ARTIFICIAL INTELLIGENCE

Versatile multicontact planning and control for legged loco-manipulation

Jean-Pierre Sleiman*, Farbod Farshidian, Marco Hutter

Loco-manipulation planning skills are pivotal for expanding the utility of robots in everyday environments. These skills can be assessed on the basis of a system's ability to coordinate complex holistic movements and multiple contact interactions when solving different tasks. However, existing approaches have been merely able to shape such behaviors with hand-crafted state machines, densely engineered rewards, or prerecorded expert demonstrations. Here, we propose a minimally guided framework that automatically discovers whole-body trajectories jointly with contact schedules for solving general loco-manipulation tasks in premodeled environments. The key insight is that multimodal problems of this nature can be formulated and treated within the context of integrated task and motion planning (TAMP). An effective bilevel search strategy was achieved by incorporating domain-specific rules and adequately combining the strengths of different planning techniques: trajectory optimization and informed graph search coupled with sampling-based planning. We showcase emergent behaviors for a quadrupedal mobile manipulator exploiting both prehensile and nonprehensile interactions to perform real-world tasks such as opening/closing heavy dishwashers and traversing spring-loaded doors. These behaviors were also deployed on the real system using a two-layer whole-body tracking controller.

INTRODUCTION

Mobile manipulators have been gaining considerable attention as we move toward integrating robotic systems into our unstructured and complex world. The reason is primarily rooted in their ability to unify the functionalities offered by fixed-base robotic manipulators and mobile platforms into a single system. A mobile base indefinitely extends a manipulator's workspace, whereas a manipulator promotes a mobile robot into an agent that can interact with its environment and actively modify it. Such a synergy enables these robots to cover a wide range of tasks akin to those tackled by a human, nature's most versatile mobile manipulator.

Numerous mobile manipulators have been developed in recent years, encompassing systems with diverse morphologies and varying degrees of autonomy. However, none has come close to matching human-level versatility in handling generic loco-manipulation problems. Loco-manipulation is a form of manipulation inherently involving a locomotion element. It is fundamentally a multicontact planning and control problem where the robot should properly exploit and coordinate contacts with its surroundings to simultaneously manipulate itself (to move and maintain balance) and other objects. One of the most impressive displays of such skills was demonstrated by Boston Dynamics' quadrupedal robot SpotMini (*1*) autonomously opening and navigating through a spring-loaded door. Although the company's work is unpublished, the complexity associated with planning and executing such a task is evident. Typically, a substantial amount of engineering effort goes into hand-crafting similar task plans in an elaborate state machine that composes a feasible sequence of connected subgoals. The success of the full scheme essentially relies on the robot's ability to perform the proper whole-body motions and apply the necessary contact forces such that all subgoals are fulfilled. In addition, various lower-level objectives must be satisfied, such as maintaining balance and stability, being robust to external disturbances, respecting the system's physical limits, avoiding self-collisions, and avoiding collisions with the manipulated object and static obstacles. Developing a framework that can automatically and holistically resolve these problems remains an active research endeavor.

A vast literature on loco-manipulation planning exists wherein different kinds of platforms and strategies have been adopted for various applications. Conducting a general yet brief overview, we came across a broad collection of interesting examples: aerial manipulators pushing doors (*2*) or movable structures (*3*); wheeled mobile manipulators wiping surfaces (*4*), plastering walls (*5*), or opening doors and drawers (*6*–*8*); humanoids moving large and heavy objects (*9*), stacking boxes (*10*), or manipulating articulated objects (*11*, *12*); quadrupedal platforms with robotic arms performing dynamic throwing (*13*) and grasping (*14*) maneuvers, opening heavy spring-loaded doors (*15*, *16*), or solving a set of tasks within a kitchen testbed (*17*); and many more (*18*–*22*).

When considering the evolution of these systems' planning architectures, one notices a shift from decentralized approaches toward more holistic solutions. The former decomposes the entire problem into a hierarchy of smaller puzzles that are easier to manage individually. This has also been a popular strategy for pure locomotion control of polyarticulated systems such as legged robots (*23*, *24*). However, designing independent yet tightly coupled submodules is a highly task-dependent and heuristics-based process that requires arduous manual tuning. In contrast, whole-body formulations for loco-manipulation implicitly account for the interactions among the different subsystems, are more intuitive to tune, and result in naturally coordinated motions. However, existing whole-body techniques have been primarily developed for the unimodal planning case, where switching of manipulation modes is either nonexistent or is predefined by a skilled engineer (*1*). Therefore, a more sophisticated framework that can produce multimodal

Robotic Systems Lab, ETH Zurich, Zurich, Switzerland.
*Corresponding author. Email: jsleiman@ethz.ch

behaviors would ultimately allow us to deal with a broader class of problems.

Recently, some of the most prominent approaches to robotic control design have been widely dominated by data-driven techniques. Relying on data eliminates the need for accurate analytical models, which can be very difficult to acquire for systems operating in complex and unpredictable settings. For instance, given the intricacy of modeling contact phenomena, model-free reinforcement learning (RL) methods have gained notable traction in contact-rich applications such as legged locomotion (*25*–*27*), dexterous manipulation (*28*–*30*), and legged mobile manipulation (*31*–*33*). However, one disadvantage of model-free RL is that it typically involves an inefficient trial-and-error process, which leads to long training times before attaining satisfactory performance. So rather than relying on real-world experience, simulators are often used to generate realistic training data efficiently (*34*). When combined with strategies that mitigate the sim-to-real gap (*35*), this can enable reliable transfer to hardware. Besides being sample inefficient, RL algorithms perform poorly in the absence of a well-defined, dense reward signal, particularly when tasked with long-horizon planning in complex environments (*36*, *37*). Therefore, experts could spend months in laborious reward-shaping and hyperparameter-tuning tailored to a specific task. One common way of overcoming this issue has been by bootstrapping RL with a form of imitation learning. For example, training can be guided by recorded human demonstrations (*28*, *29*), animal motion capture clips (*38*, *39*), or an RL-trained teacher with access to privileged information (*27*, *40*, *41*). Nevertheless, generating expert demonstrations for every newly encountered task is time consuming, and motion retargeting is often challenging, especially when the robot′s morphology differs from that of the demonstrator.

Data-driven control design has been a powerful tool for achieving robust policies. On the other hand, it does not offer a versatile framework that can systematically generalize well across diverse scenarios of a similar nature. With the aim of establishing such a framework for general loco-manipulation, we explored alternative approaches in the trajectory optimization literature. Discovering motion trajectories coupled with an optimal contact schedule is fundamentally a hybrid planning problem because it involves both continuous and discrete decision variables. In its most general form, multicontact planning can be mathematically cast as a mixed-integer nonlinear program (NLP). Such programs are computationally intractable because of their inherent exponential complexity. Moreover, if provided with a poor initial guess, they typically end up in bad local minima arising from the problem′s nonconvexity. To address these issues, some methods rely on approximations that convert the original problem into a mixed-integer convex program (*42*, *43*), offering global optimality guarantees upon convergence without requiring any form of warm start. Alternatively, contact-implicit optimization aims to resolve the combinatorial explosion of contact transitions by reformulating the problem as a continuous trajectory optimization. For instance, Mordatch *et al.* (*44*, *45*) introduced contact-dependent costs weighted by real-valued auxiliary variables acting as continuous contact flags. These were then optimized jointly with the motion trajectories [and contact forces (*45*)] within a single unconstrained NLP. The approach of Mordatch *et al.* (*44*) and other similar methods that eliminate discontinuities induced by contact dynamics (*46*–*48*) are essentially based on smooth relaxations of the contact complementarity system (*49*). These relaxations allow forces to be applied at a distance, resulting in gradients that guide the optimization routine toward contact-rich but physically inconsistent behaviors. In contrast, Posa *et al.* (*50*) maintained the hybrid structure of contact by formulating the NLP as a mathematical program with complementarity constraints. They used certain tricks that aided the solver′s convergence despite the nonconvexity and nonsmoothness of the constraint manifold. Other notable works that implicitly and accurately handle contact phenomena posed the problem as a bilevel optimization: The outer level generated motion trajectories that were constrained by the resolution of the contact dynamics in the inner-level optimization (*51*–*53*). Nevertheless, in general, optimizations involving intricate physical models are highly sensitive to initialization and take time to converge, especially when planning through multiple contact switches.

Complementary to the gradient-based perspective, we investigated the contributions of classical graph-search and sampling-based algorithms (*54*) in multicontact planning. These can be seen as crucial assets when considering the fragility of optimization-driven methods in hybrid and nonconvex scenarios. Early work by Dalibard *et al.* (*55*) addressed hybrid loco-manipulation tasks such as opening and navigating through doors by applying rapidly exploring random trees (RRT) to find a sequence of robot-object motions intertwined with discrete transitions between successive constraints. A similar problem was later solved by searching through a graph of predefined motion primitives that provide feasible transitions between base configurations and discrete manipulation modes (*56*). In both cases, the planners were tightly built around some form of task-specific knowledge. Hauser *et al.* (*57*) proposed a more generic approach for multimodal motion planning that builds a sparse tree of hybrid states by extending tree nodes toward randomly sampled adjacent modes and robot configurations. More recently, Murooka *et al.* (*58*) achieved general large-object loco-manipulation with humanoid robots by jointly planning sequences of footsteps and regrasps using a graph search with a transition model based on reachability maps. However, they relied on a zero-moment-point balance criterion, which assumes coplanar contacts by neglecting the effects of manipulation forces. Looking at additional examples in the legged locomotion (*59*, *60*) or nonprehensile manipulation (*61*, *62*) literature, it is evident that this family of multicontact planners has been predominantly applied in the quasi-static domain. One way of extending them is via prespecified dynamic motion primitives (*63*, *64*). However, such methods do not generalize well because they might require extra design effort to generate unique high-level primitives given a new task. These shortcomings have motivated synergizing search-based and sampling-based planners with trajectory optimization schemes (*65*–*67*).

In the interest of unifying and formalizing concepts, we note that most of the above approaches can be classified as an instance of task and motion planning (TAMP) problems (*68*). The work by Hauser *et al.* (*57*) can be interpreted as an extension of sampling-based motion planning to the TAMP domain. Analogously, Toussaint (*69*) proposed an optimization-based version, logic geometric programming (LGP), that incorporates first-order logic to represent mode switches within a geometrically constrained path optimization. This effectively amounts to a domain-specific extension of mixed-integer programs that can be handled more efficiently with a structure-exploiting solver. The LGP formalism was later adopted to tackle a wide range of sequential manipulation tasks (*70*–*73*) and

also inspired the loco-manipulation framework presented in this paper.

We propose a versatile approach for generating high-fidelity multimodal plans that solve long-horizon loco-manipulation tasks under the assumption of prespecified nominal object models and affordances. More specifically, our framework targets a class of problems involving general dynamic manipulation of movable or articulated objects with a mixture of prehensile and nonprehensile interaction modes. Given high-level descriptions of the robot and object, along with a task specification encoded through a sparse objective, our planner holistically discovered how the robot should move, what forces it should exert, what limbs it should use, and when and where it should establish or break contact with the object. The framework can be readily adapted to different kinds of mobile manipulators. Nonetheless, for the sake of conciseness, we limited the discussion in this work to the case of a quadrupedal platform, ANYmal (*74*, *75*), equipped with a custom-built 6-DoF robotic arm (see Fig. 1). Designing control architectures for such a high-dimensional (24-DoF) underactuated system exhibiting hybrid dynamics is a highly complex process.

The diagram in Fig. 1 provides an overview of the proposed planning and control scheme. In our previous work (*16*), we formulated a unified optimal control problem (OCP) that generates whole-body plans for combined locomotion and unimodal manipulation. Here, we built on this framework by extending it to accommodate various manipulation modes. This enabled us to automatically compose a mixture of multimodal plans without relying on task-dependent motion primitives. Manipulation modes are defined on the basis of viable pairings between user-specified object affordances and robot end effectors. As depicted in Fig. 1A, we categorize affordances into prehensile/nonprehensile contact points or surfaces and end effectors as prehensile/nonprehensile feet or arm contacts. The core insight is that such a classification enables us to encode feasible contact-switching actions via simple logic rules. This considerably reduces the branching factor in the discrete search and, accordingly, the computational complexity associated with traversing a connected graph of contact states. Therefore, inspired by the LGP formalism of TAMP problems (*69*), we devised an offline planner in the form of a bilevel optimization (*72*, *73*) that uses a rule-based informed graph search at an outer level interleaved with an inner-level trajectory optimization for switched systems. Unlike most TAMP formulations, our sparse goal was not characterized in terms of a target symbolic state but rather as a terminal set of desired base-object poses. Furthermore, we aimed to avoid the computational burden of searching through an extended graph that augments the contact state with a grid-based representation of our continuous state space (*72*). To this end, we adopted a sampling-based approach in generating the references for the inner-level problem and incrementally grew a multimodal tree of short-horizon trajectories by alternating between goal-directed and purely random extensions. This effectively covers the search space with a discrete set of reachable robot-object states coupled with their corresponding contact modes, as illustrated in Fig. 1B (i). The introduced randomness also elevates the algorithm into a strategy capable of global exploration, an essential aspect for escaping bad local minima arising from the nonconvexity of geometric constraints.

After converging to a connected sequence of optimal trajectories and contact switches leading to the goal, we applied a postprocessing step to enhance the solution′s overall quality. As shown in Fig. 1B (ii), this entails using the discovered sequence as an initial guess to warm-start a single long-horizon optimization over a fixed contact schedule. The resulting output is a smoother and lower-cost solution with increased feasibility, yielding a high-fidelity plan that can be reliably executed on the real system.

Last, to ensure a robust deployment of complex behaviors on such robotic systems, a common approach has been to decompose the problem into an offline planning phase that handles the heavy computations and a reactive control module that tracks the generated plans at high-update rates (*76*, *77*). However, when dealing with long-horizon dynamic maneuvers involving multiple discontinuous switches, the accumulated effects of unmodeled disturbances might render the plan unstabilizable with a purely reactive controller. Hence, we mitigated these issues with a two-layer tracking scheme (*78*, *79*) consisting of a short-horizon model predictive control (MPC) layer on top of a whole-body controller. The full architecture of the online module is depicted in Fig. 1C. With such a structure, we attained a task-agnostic transfer of multicontact plans with minimal tuning effort, thereby adequately bridging the gap between offline behavior generation and online execution.

We demonstrated the effectiveness of our framework in its ability to rapidly discover holistic solutions for a diverse set of tasks characterized by different objects, environments, or goal specifications. We verified the physical consistency of the offline behaviors by testing some of them on hardware and showing that, under the assumption of a reasonable object model, they can be accurately tracked by relying solely on proprioceptive feedback. One example, shown in Fig. 1D, is that of a door-traversal sequence including multiple interaction modes.

RESULTS

Movie 1 summarizes the methodology and results of the presented work. The following subsections describe the results in detail.

Overview of loco-manipulation experiments

Examining a wide variety of real-world loco-manipulation scenarios, we noticed that they can be broadly categorized on the basis of their high-level goal descriptions into two main groups: object-centric and robot-centric tasks. The former encompasses tasks centered around manipulating an object to alter its state toward a target configuration, such as opening a refrigerator, closing a dishwasher/oven, or turning a valve. On the other hand, a robot-centric perspective in the context of mobile manipulation entails a goal specification centered around the robot′s base. This often requires the agent to interact with its environment as it navigates from one state to another. Examples of this class include traversing to the other side of a closed door or navigating among movable obstacles. Here, we opted for two representative tasks per category, as illustrated in Fig. 2A.

Through a series of simulation experiments, we established the versatility of our planning framework by showcasing its ability to efficiently discover complex long-horizon behaviors with minimal manual guidance: All experiments assumed unknown task durations, sparse goal specifications, a uniform cost function, and no prior contact schedules or motion plans to guide the solver. Our results primarily illustrate how new behaviors automatically emerge from subtle variations in the task formulation, such as changes in the robot′s joint limits when opening a dishwasher,

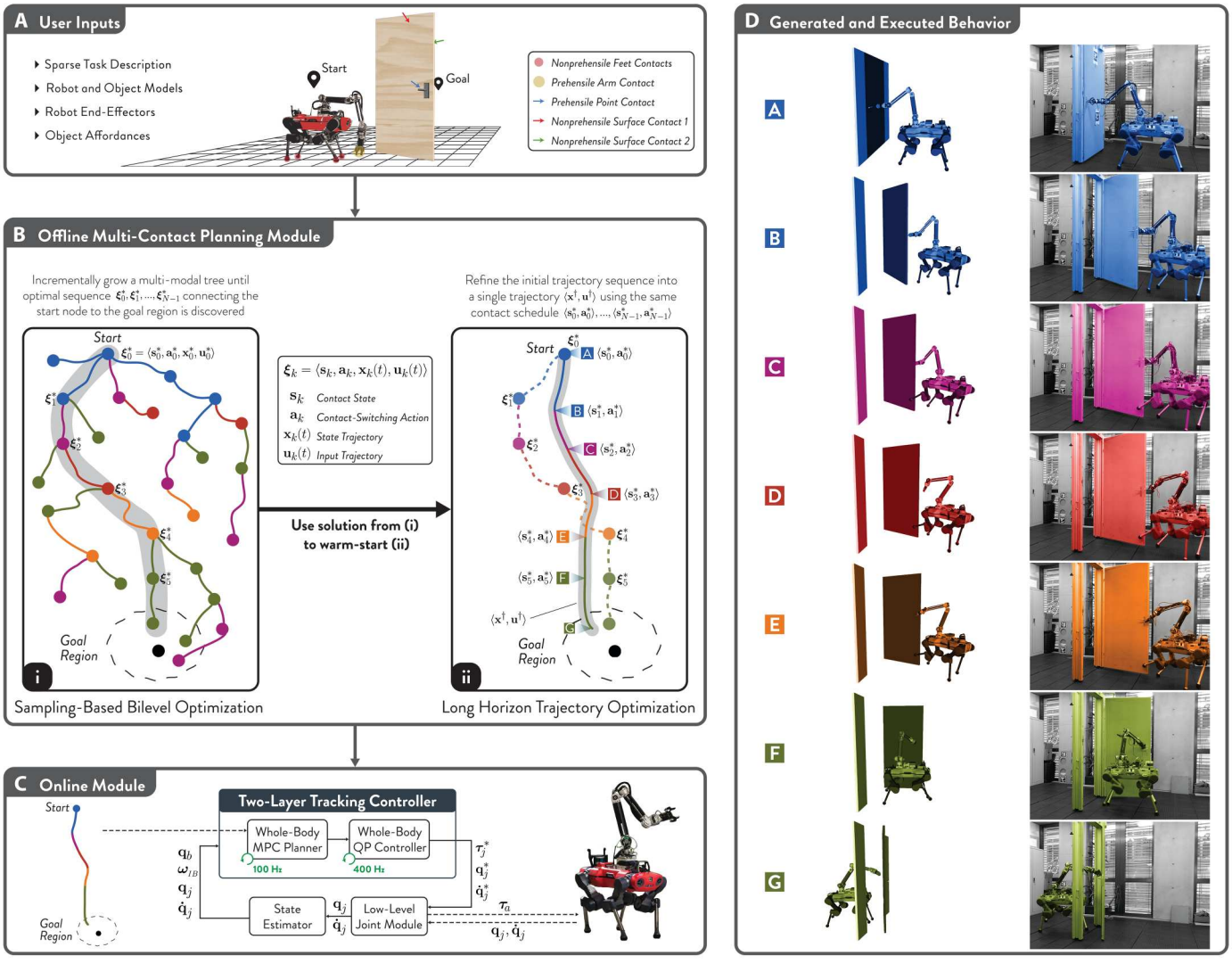


Fig. 1. Planning and control architecture for multicontact loco-manipulation. (**A**) The user provides the offline planner with a basic high-level description of the scene: A sparse task specification given by a start state and a target goal region, dynamic models of the robot and the object, specification and classification of robot end effectors, and object affordances. (**B**) The central element in the offline planner is (i) a sampling-based bilevel optimization that gradually grows a tree of mode-invariant short-horizon trajectories seeking an optimal multimodal sequence that reaches the goal region. Modes are determined by contact state and action pairs $\langle \boldsymbol{s_k}, \boldsymbol{a_k} \rangle$ that encode for each end effector whether it is a closed, open, or switching manipulation contact and the object contact it is interacting with. Each color corresponds to a specific contact state, and consecutive color variations indicate a contact switch. (ii) The second component in the planner is a long-horizon trajectory optimization that is warm-started with the bilevel optimization solution. It uses the same contact schedule and only polishes the continuous part. (**C**) To deploy multicontact behaviors, the online module relies on an MPC-based whole-body controller that tracks the offline references. (**D**) With such a framework, complex behaviors that require exploiting multiple contact interactions, such as traversing a spring-loaded pull door, can be rapidly generated (i.e., often requires less than 1 min on an ordinary laptop) and reliably executed with a real quadrupedal mobile manipulator.

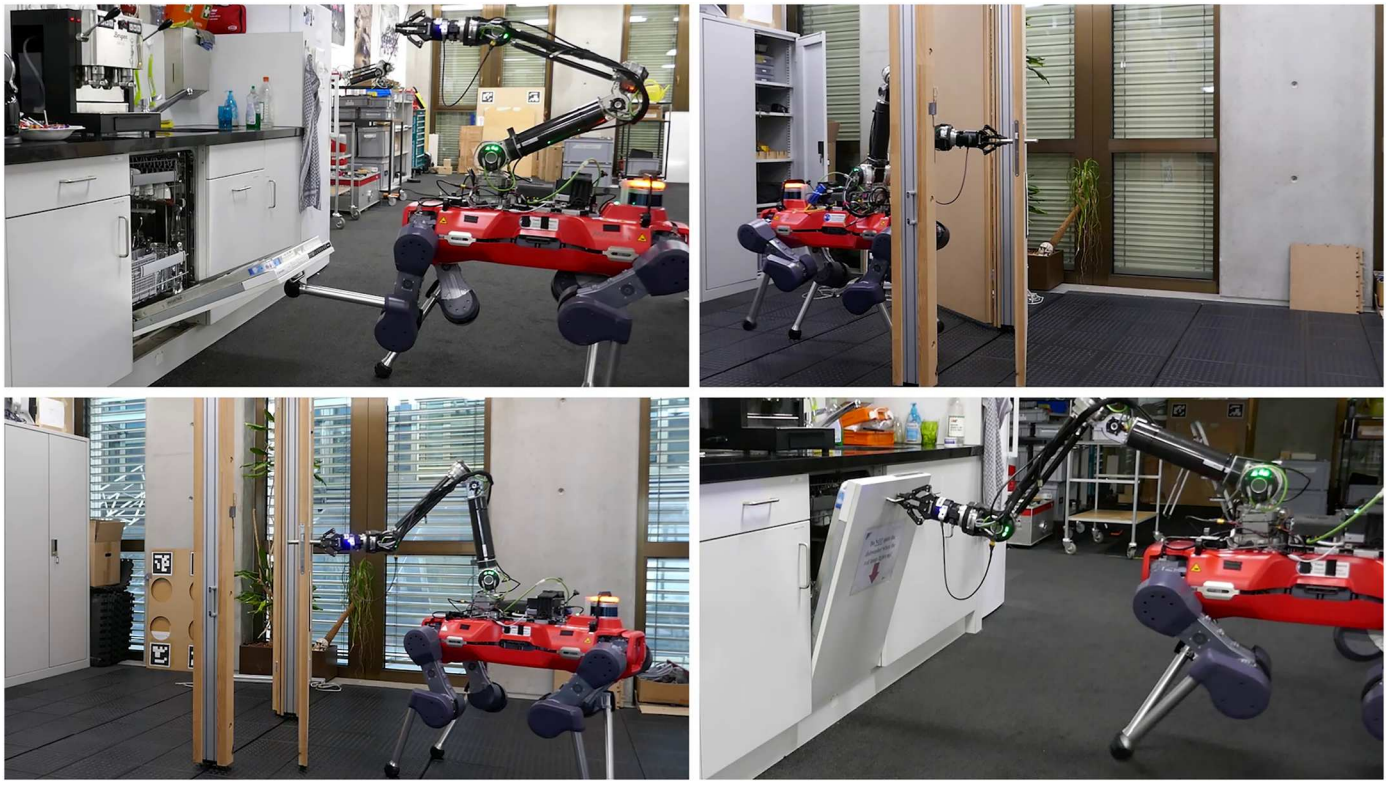
the object affordances when turning a valve, the available end effectors when navigating across a movable obstacle, and the object dynamics when traversing a door.

To confirm the dynamic feasibility of the offline-generated plans, we conducted a set of hardware experiments for tasks 1 and 4 of Fig. 2A using our MPC-based tracking controller. Furthermore, we demonstrated the reliability of these plans, in terms of task attainment, by successfully executing multiple independently computed behaviors that solve the same task, but with variations in the problem parameters (such as the robot's desired gait schedule and the object's dynamic model) or in the resulting manipulation schedule.

In the following section, we first analyze the emergent behaviors for all four tasks and then present hardware results on our quadrupedal mobile manipulator. All discussed solutions correspond to the first goal-attaining sequence found by the planner.

Behavior generation for object-centric tasks

In the first object-centric scenario, the robot manipulates a heavy dishwasher door that exhibits high stiction at the joint. As illustrated in Fig. 2A (i), three potential object contacts are assumed: the front



Movie 1. Summary of the results and methods for the loco-manipulation framework.

surface, the handle attached to it, and the back surface. All generated multimodal solutions in Fig. 2 (B and C) are presented in the form of continuous trajectories coupled with a color-based encoding of the underlying manipulation modes. We started with the dishwasher fully closed and provided a goal region centered at −90°. As shown in Fig. 2B, the resulting behavior consists of the arm initially grasping the handle, slightly opening the dishwasher, then switching to the back surface contact to complete the opening motion. The need for the intermediate contact switch can be explained by referring to Fig. 2C, where its emergence is visibly justified by the arm′s operational limits. By deactivating joint limits and self-collision constraints in our formulation, we observed that the contact switch is no longer necessary, because the dishwasher can be opened with a continuous gripper-handle interaction.

Next, we simply swapped the start and goal configurations of the dishwasher. Consequently, various closing maneuvers with diverse contact schedules were found, one of which is depicted in Fig. 2B. The solution is a fully nonprehensile interaction sequence that exploits different end-effector contacts: The robot first uses one of its feet to lift the dishwasher, making it then easier for the arm to establish a nonprehensile contact with the same surface and thereby complete the closing task.

The second object-centric task involves manipulating a large valve wheel with dynamics subject to static friction. A valve target of two full turns was assigned; however, we introduced various scenarios in which different object affordances (namely, different numbers, types, and locations of contacts) were chosen to highlight their effects on the resulting manipulation behavior. First, we considered the case of a single nonprehensile point contact located at one of the spokes of the valve. Figure 2D indicates that a feasible plan consists of a continuous valve rotation by relying on a constant interaction. In contrast, replacing this contact with a prehensile one at the wheel′s outer rim makes a regrasp necessary because of the additional constraint imposed on the gripper′s orientation with respect to the graspable point, which causes a violation in the limits of the arm′s last joint.

Next, we constructed a setting wherein the valve′s height was adequately increased such that the inner nonprehensile point was still kinematically reachable for all valve angles, but the outer graspable contact was not. Therefore, in the prehensile interaction case, one object contact was no longer sufficient to fully solve the task. We resolved this by representing the valve affordance more accurately with multiple (five) grasping locations along the outer rim. As shown in Fig. 2E, a viable turning sequence consists of several regrasps that occur whenever the arm gets close to its kinematic singularity throughout the manipulation period.

Behavior generation for robot-centric tasks

In both robot-centric tasks, the base goal location is only attainable by altering the environment, which is composed of obstructive manipulable objects and other static obstacles. We did not specify any target regions for the object; hence, it could end up in an arbitrary terminal state.

The first scenario involves a movable obstacle modeled as a 2-kg box with three degrees of freedom: one rotational and two translational, along with four contact surfaces. To adequately capture frictional effects in all directions, we approximated the box-ground patch interaction with four point contacts at the bottom vertices

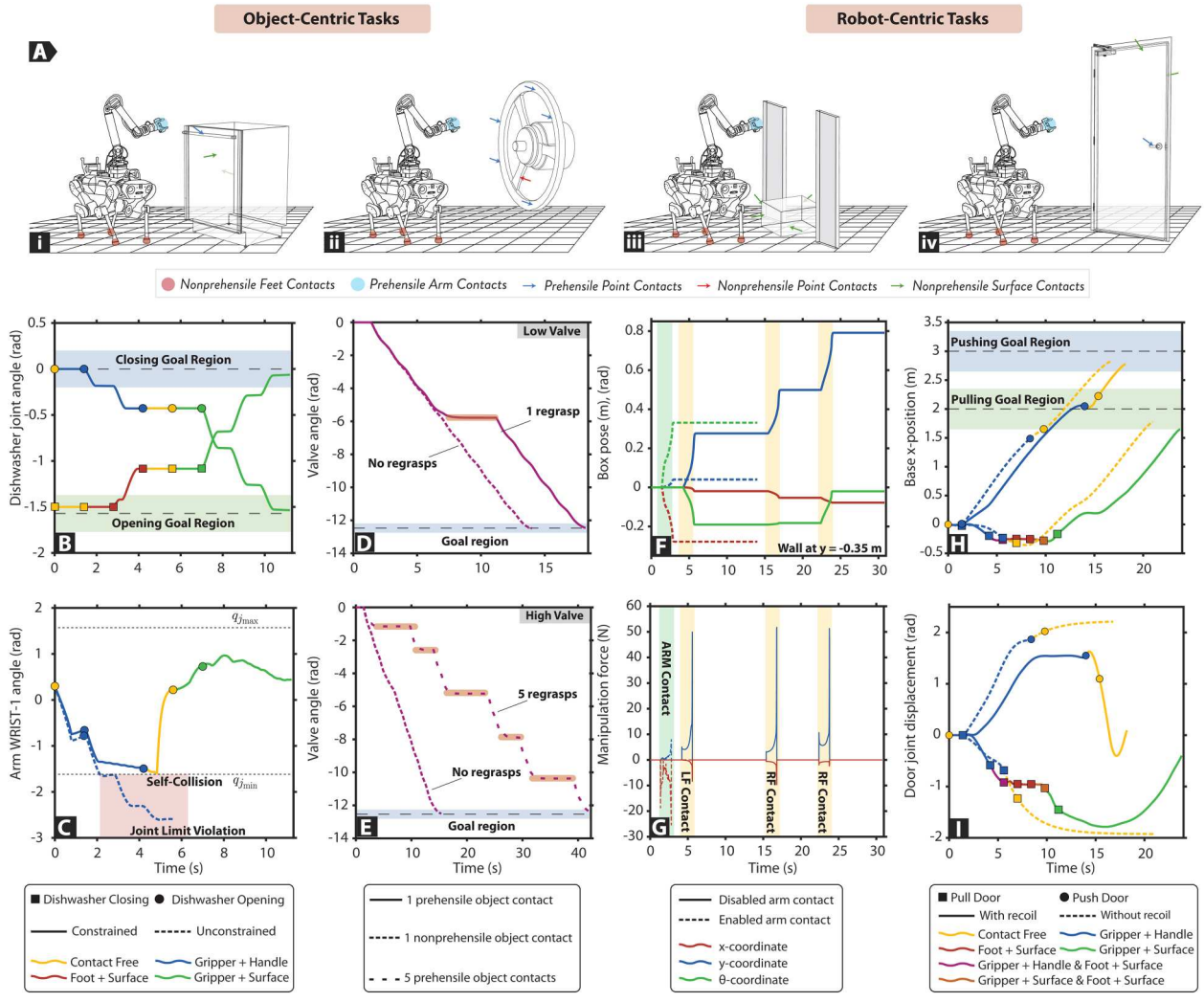


Fig. 2. Validation of loco-manipulation planner. (**A**) Illustration of two object-centric and two robot-centric tasks: (i) dishwasher manipulation (movie S1), (ii) valve manipulation (movie S2), (iii) traversal of an obstructing movable obstacle (movie S3), and (iv) traversal of a large articulated object (movie S4). Diverse behaviors emerge, highlighting the planner's adaptability to variations in each task. Task 1: (**B**) Different start states and goal regions (opening versus closing). (**C**) Opening with different joint limits and self-collision constraints for the robot. Task 2: (**D** and **E**) Different numbers, types, and locations of object contacts, for two varying valve heights (low versus high). Task 3: (**F** and **G**) Different robot end-effector contacts enabled for manipulation (enabled arm contact versus blocked arm contact). Task 4: (**H** and **I**) Different joint limits (push versus pull) and dynamics (recoil versus no recoil) for the object.

of the cuboid. If the box is not large enough to block the robot's path, the planner finds the trivial locomotion-only solution of navigating around the object without having to manipulate it. On the other hand, in the case of an obstructing box, one of the resulting feasible plans uses the arm's end effector in a single nonprehensile interaction phase that pushes the box sideways (y direction). The displacement is large enough for the robot to clear its path while ensuring that the object does not collide with the wall.

Within the same setting, we considered another example wherein the arm cannot be used for manipulation, such as when the gripper is already carrying a load. Consequently, an alternative solution that exploits the robot's feet to displace the obstacle with a series of forward pushes (x direction) emerges. These behaviors are captured in Fig. 2 (F and G), which shows the object's motion and the corresponding manipulation forces exerted by the robot. The presented plots also highlight the dynamic feasibility of the generated trajectories: unilateral pushing forces, force-motion consistency, and stick-slip motion dynamics caused by static friction.

In another scenario, the robot is required to traverse a large articulated object with nonlinear recoil dynamics, namely, a spring-loaded door. We produced variations of the same task through basic adjustments in the door's workspace limits, articulation, and dynamic parameters. Affordances were specified as one graspable point (the door handle) and all pushable surface contacts.

Starting with the case of a spring-loaded revolving door that can be pushed open, the simplest solution to this task comprises establishing contact with the handle, maintaining this contact while pushing, and breaking the contact while ensuring that the door does not collide with the robot upon recoil. By imposing an extra joint limit on the door ($q_o \leq 0$), transforming it into one that can only be pulled open, a more complex multimodal sequence arises; one of the feet is required to hold the door as the arm switches from

the handle contact to the surface contact in the middle of the opening maneuver. Otherwise, it cannot fully pass through without violating the inherent geometric constraints. A similar schedule is also observed for the sliding-door case. The discovered behaviors are presented in Fig. 2 (H and I) in the form of base-object position trajectories with a color-based encoding of the underlying modes.

Last, by removing the door's recoil, the planner converges to the same straightforward sequence that solves all three examples. As shown in Fig. 2I, the door can be easily opened and traversed using a single handle-contact interaction that is released at an early stage of the task duration.

Planner evaluation

We assessed the solver's performance considering 10 of the discussed scenarios with five independent runs each. The related results mainly consist of the timing statistics relative to the corresponding behavior duration and are reported in Table 1. The majority of the planner's computation is dedicated to the sampling-based bilevel optimization stage: For instance, in the pull-door scenario, around 82% of the computational time is spent in this stage, whereas the rest corresponds to the long-horizon trajectory optimization. Furthermore, we observed that, had the proper manipulation schedule been predefined for this task, searching over the continuous trajectory segments would have only required around 13% of the bilevel search's total time. In contrast to our sampling-based method, a deterministic approach that accurately solves the bilevel problem through a grid-based representation of the continuous state space would have resulted in less variability in the computational times and higher-quality solutions that do not require any postprocessing; however, it would have been less computationally efficient overall.

All trajectories were computed on an ordinary laptop (Intel Core i7-10750H, 2.6 GHz, Hexacore) in less than 1 min for an average computational time of 279 ms per tree extension. Considering the planner's average solve times, normalized by the behavior durations, we found that the solver performs best for the valve-turning task and takes the most time to converge for the pull-door example, with a ratio of 0.46 and 1.84, respectively.

We observed that the intricacy of the problem's dynamics does not substantially affect the planner's computational demands. This is reflected in the tree-extension solve times, which are mostly similar for all scenarios. Moreover, as a consequence of the adopted pruning rules, the planner's efficiency is not necessarily hindered by the number of total discrete modes, as demonstrated in the valve-manipulation scenario. Instead, we noticed that the computational times are strongly correlated with the level of geometric complexity in the problem (namely, the nonconvexity of collision constraints) and the degree of sparsity in the goal specification. For instance, this is evident in the planner's performance when applied to the door traversal task. In contrast to the push-door scenario, traversing a pull door requires the robot to initially move away from the goal and to also switch the arm contact across the object while avoiding collisions. Such behaviors underscore the need for an exploratory aspect in the planner.

Behavior execution

Here, we focus on validating the physical consistency of the multimodal behaviors by deploying them on hardware. In these experiments, the offline-generated trajectories were tracked by the real system using module C, depicted in Fig. 1. The two-layer tracking control and state estimation (*80*) both run on ANYmal's main onboard PC (Intel i7-8850H, 2.6 GHz, Hexacore). With a 1-s prediction horizon and a discretization time step of 0.015 s, the MPC loop is executed at 100 Hz asynchronously to the whole-body controller and the state estimator, which are updated at a rate of 400 Hz. The same hyperparameters were used in the MPC-WBC tracking module during all scenarios.

The experimental procedure simply consisted of precomputing the plans given a reasonable object model and storing them in a motion library on the robot's onboard PC. These trajectories were then mapped to the online setting by establishing a correspondence between a key reference frame in the real environment and its counterpart in the offline case. An example of such a frame is the end effector as it initially grasps the door handle.

Table 1. Planner evaluation over 10 scenarios. Planner timing statistics were based on five test runs per scenario. Tree extensions correspond to solving a mode-invariant OCP with a short horizon of 1.4 s. The minimum and maximum average solve times, as well as the longest behavior duration, are highlighted with asterisks. STD, standard deviation.

		Planner timing statistics (s)				
Loco-manipulation scenario	**Behavior minimum duration (s)**	**Mean**	**Min**	**Max**	**STD**	**Tree-extension mean solve time (ms)**
Heavy dishwasher opening	9.8	9.6	4.6	25.9	9.2	226
Heavy dishwasher closing	9.8	13.6	7.5	25.8	7.2	277
Valve + 1 nonprehensile contact	14	6.5*	5.7	7.2	0.7	296
Valve + 5 prehensile contacts	42*	19.4	16.5	21.6	2.2	233
Movable obstacle + arm contact	14	15.2	13.1	19.9	2.7	349
Movable obstacle + feet contacts	30.8	43.2	38.0	49.9	4.7	337
Push door with recoil	15.4	13.9	5.7	22.0	7.1	256
Pull door with recoil	23.8	43.7*	33.8	53.9	7.5	275
Sliding door with recoil	26.6	40.3	29.3	51.5	8.0	284
Push door without recoil	15.4	8.1	4.8	11.9	3.4	252

We selected the behaviors of highest complexity from one robot-centric and one object-centric task commonly encountered in real-world settings: traversing a spring-loaded door and manipulating a heavy dishwasher exhibiting high stiction. First, we tested the push-door case, which can be solved with a straightforward sequence described in the previous section. However, because of the sampling-based nature of our planner, we obtained various pushing maneuvers and manipulation schedules that could solve the task given the same goal specification. Three of the discovered mode sequences are depicted in Fig. 3, two of which involve the use of one of the feet to hold the door as the arm either switches to a new object contact or retracts to its nominal configuration.

Next, we showcased the more complex behavior of opening and passing through a pull door. So far, this has only been presented in (*1*), where laborious handcrafting appears to have been involved in the design process. The conducted experiments readily applied this sequence in two settings containing doors with different kinematic and dynamic parameters (different hinge side, width, inertia, and recoil) by solely changing the door's kinematic parameters in the planner formulation. Moreover, we note that varying multicontact plans with the same manipulation sequence were generated by diversifying the gait schedules used when solving the task. These were all accurately tracked by the robot, where the adopted dynamic gaits included a fast trot, a normal trot, and a standing trot, as depicted in Fig. 4A. The motion-tracking accuracy of the robot's base and manipulating limbs is presented in Fig. 4B for one of the pull-door solutions. The results demonstrate that the online execution highly matches the offline references and that the transfer of precomputed trajectories to the physical system is agnostic to variations in the task formulation parameters.

We further verified the framework's physical fidelity by manipulating a real dishwasher while exploiting multiple contact interactions, as shown in Fig. 5 (A and B). The tracking accuracy of the manipulating limbs for the opening and closing case is shown in the joint positions plots of Fig. 5C and Fig. 5 (D and E), respectively. In such a scenario, the robot was operating close to its joint limits and experienced large manipulation forces of up to 125 N because

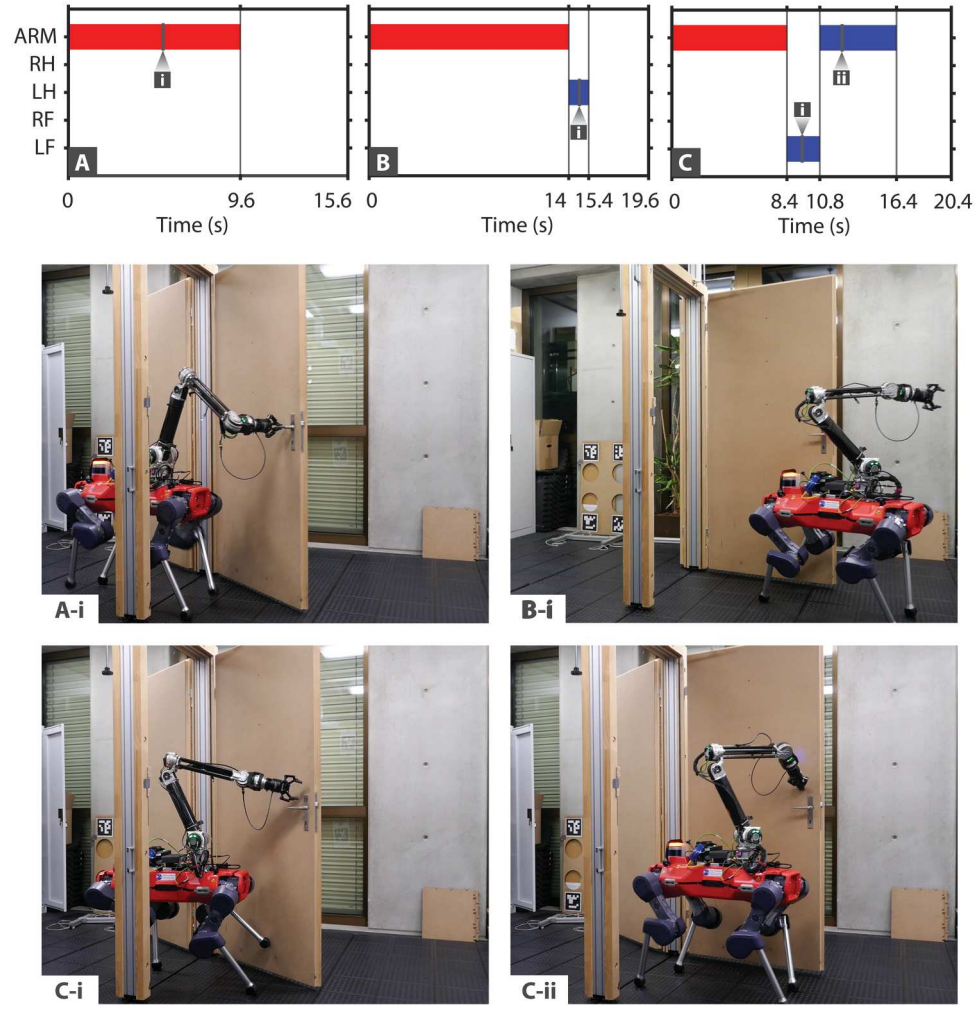


Fig. 3. Hardware experiments – Feasibility verification for push door with recoil behaviors. (**A** to **C**) Different manipulation schedules discovered and executed to solve the same task. Red, prehensile interaction with handle; blue, nonprehensile interaction with surface. ARM, 6-DoF robotic arm; LF, left front leg; RF, right front leg; LH, left hind leg; RH, right hind leg. The numbers on the schedules indicate certain instants captured in the corresponding snapshots.

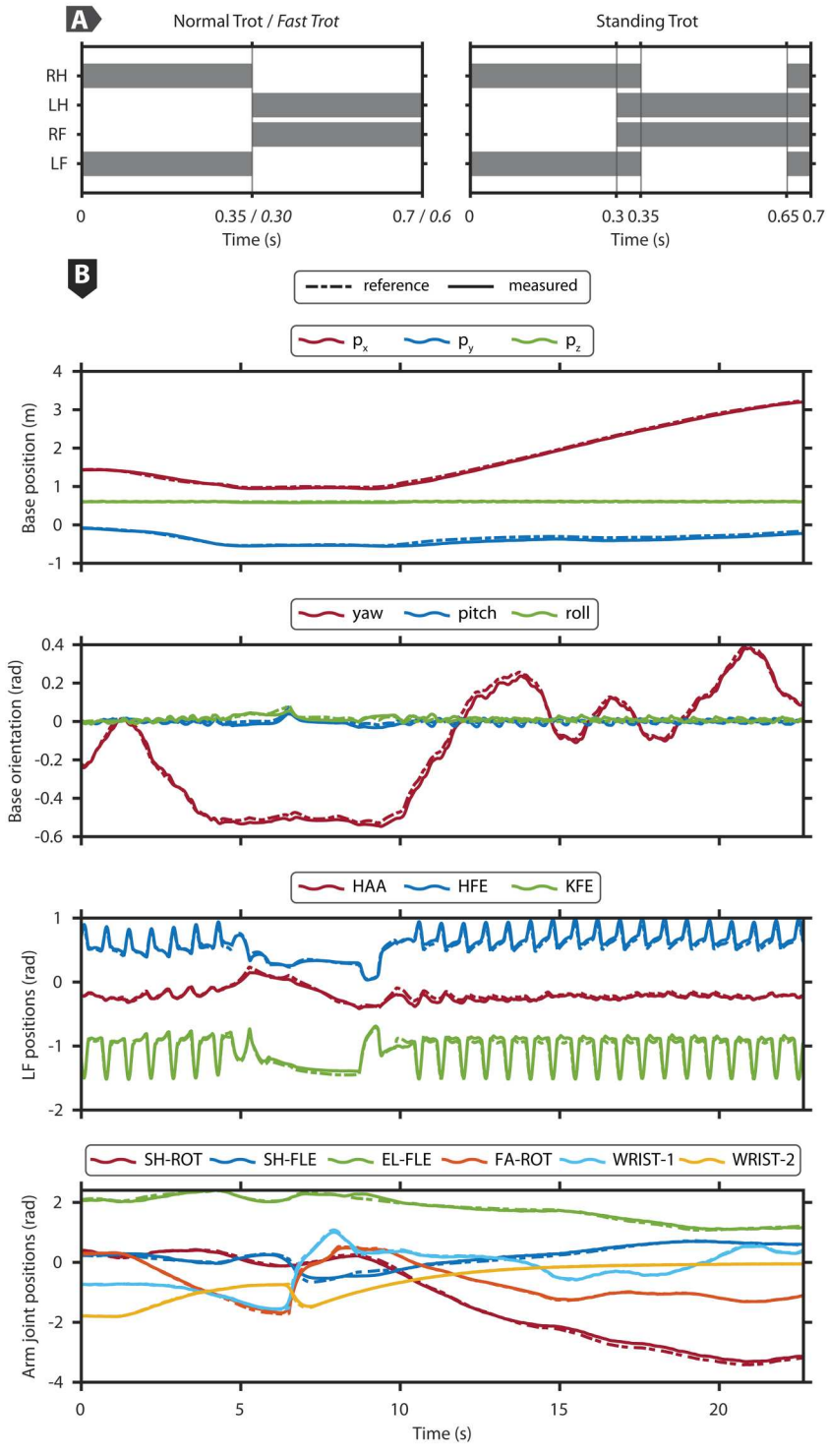


Fig. 4. Hardware experiments – Feasibility verification for pull door with recoil behaviors. (**A**) The different gait schedules adopted in this scenario. Left, normal trot and fast trot; right, standing trot. (**B**) Motion-tracking accuracy of the base pose and manipulating limbs.

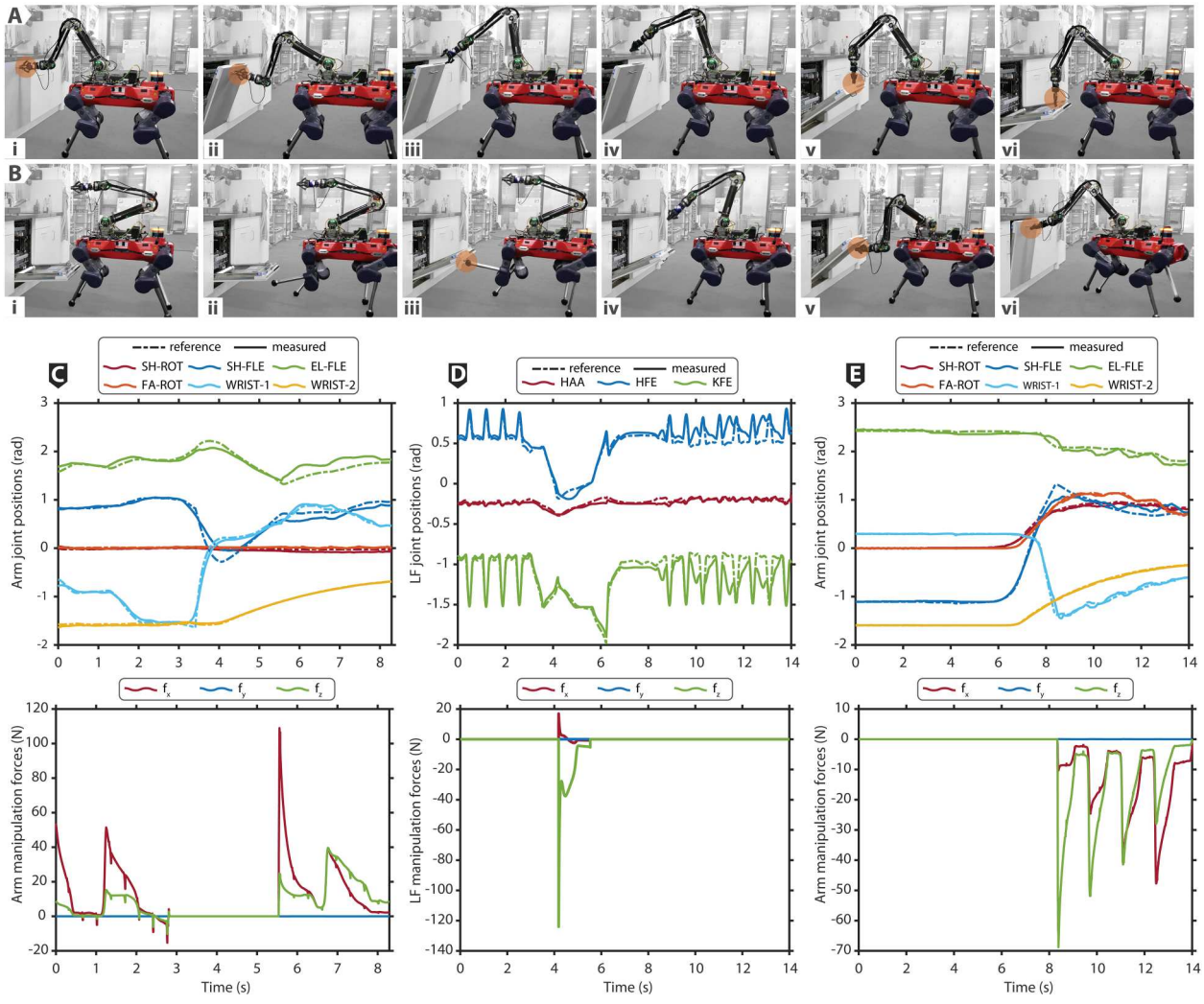


Fig. 5. Hardware experiments – Feasibility verification for dishwasher manipulation behaviors. (**A** and **B**) Image sequences of executed multicontact behaviors for manipulating a heavy dishwasher with high stiction highlighting different interaction phases: opening and closing maneuvers, respectively. (**C**) Motion-tracking accuracy and exerted manipulation forces corresponding to the arm during the opening scenario. (**D** and **E**) Motion-tracking accuracy and exerted manipulation forces corresponding to the arm and left front leg, respectively, during the closing scenario.

of the high friction in the dishwasher's joint (see force plots of Fig. 5, C to E).

The trajectories discussed in this section were successfully tested on the robot from the first trial without requiring any specific hand-tuning. Moreover, we note that each behavior was reliably executed in several test runs without any failures.

DISCUSSION

The framework presented in this paper achieved a rapid generation of holistic multimodal behaviors along with their reliable execution in challenging scenarios on a high-dimensional legged system. Robot-centric and object-centric tasks consisting of basic high-level specifications, sparse objectives, and unknown time horizons were solved within a minute for task durations that reach up to 50 s. In the general context of contact-rich planning, several approaches exist in the locomotion/manipulation literature that, in contrast to ours, mostly lead to physically inconsistent maneuvers, require long computational times, or are highly task specific and hence involve a cumbersome design process. Apart from the introduction of domain-specific guiding rules, the success of the presented approach can be attributed to its exploitation of different planning techniques whose core strengths complement one another: optimization-based planning for computing locally optimal solutions that satisfy dynamics and path constraints, informed graph search for a fast and optimal combinatorial search, and sampling-based planning for an efficient global exploration.

On the other hand, we see some limitations that we aim to address in future work. These limitations are primarily connected to the task-execution phase. First, the present capabilities of online replanning offered by the MPC-based controller are limited to solely adapting the continuous elements of the offline multimodal references. Extending our framework to allow for an online adjustment of manipulation schedules necessitates making

the bilevel optimization routine real-time capable. This could be achieved by speeding up the search through a proper parallelization of multiple tree extensions.

Second, tracking behaviors generated on the basis of a nominal world model is only viable under the assumption of a reasonably accurate description; however, certain types of modeling mismatches can be tolerated more than others. For instance, during our hardware tests, we had to ensure that object articulations and geometries were captured with relatively high certainty, whereas we only provided a simple approximation of the dynamics that roughly represents the overall system response. Therefore, none of the real-world objects actually required an accurate system identification process when defining their corresponding equations of motion (EoM) in the offline phase. The trajectories computed for the two distinct doors shown in movie S6 rely on the same nominal model in the planner except for changes in the kinematic parameters (the door width and hinge side), despite them also having different masses and recoils. The reason is that, if quasi-static effects are roughly compensated for, the unmodeled residual dynamics can be mitigated with high-gain feedback control. This inherent robustness is further demonstrated in movie S6 through the controller's disturbance-rejection capabilities. In contrast, geometric and kinematic mismatches cannot be handled with our current approach and would potentially lead to dangerous collisions or large internal forces caused by wrongly controlling position in motion-constrained directions.

Nonetheless, we see the proposed framework as a stepping stone toward developing a fully autonomous loco-manipulation pipeline. For instance, robustness to large unforeseen disturbances and unmodeled effects can be greatly improved by complementing our planner with data-driven techniques. Specifically, RL has been shown to produce policies with strong robustness properties when tailored to a specific objective (*26*, *30*, *40*). However, without a tedious reward-engineering process dedicated to the task at hand, it generally struggles to discover the right behaviors. This can be addressed by leveraging expert demonstrations: Multimodal trajectories can be encoded into motion priors that are then used to guide the training of a robust RL agent (*39*, *81*, *82*) in complex loco-manipulation settings. However, instead of relying on expert trajectories generated by human examples (*83*, *84*), one could potentially use our framework as an automatic provider of physically consistent demonstrations for an RL pipeline.

Last, full autonomy would naturally require incorporating exteroceptive information into our framework. This is necessary to accurately detect and localize the object of interest at test time, as well as to adequately react against unanticipated effects that cannot be inferred with pure proprioception (such as unknown obstacles or loss of contact with the object). In addition, perception can be used to infer important properties about the object, such as its dimensions, articulation, and affordances, thereby fully replacing user-specified inputs. In the Supplementary Discussion (Toward real-world deployment section), we discuss the practical usability of our current setup in real-world settings, despite our limiting assumption regarding a-priori-modeled environments. We also propose straightforward extensions that mildly integrate perception with our framework to facilitate deployment.

MATERIALS AND METHODS

This section provides a detailed description of the architecture depicted in Fig. 1 with a specific focus on the offline planning module. We justify the importance of different components in the Supplementary Methods (Ablation study over framework components section).

General optimization formulation

The discovery of an optimal plan $\boldsymbol{\zeta}^*(t)$ composed of a continuous state trajectory $\boldsymbol{x}^*(t) \in \mathbb{R}^{n_x}$ and input trajectory $\boldsymbol{u}^*(t) \in \mathbb{R}^{n_u}$ jointly with a sequence of discrete modes $\boldsymbol{z}^*(t) \in \mathscr{Z} \subset \mathbb{R}^{n_z}$ can be generically formulated as a mixed-integer OCP (*85*):

$$\boldsymbol{\zeta}^* = \arg\min_{\boldsymbol{\zeta}} \int_0^{t_f} L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{z})\, dt \quad \text{subject to} \quad \begin{matrix} \dot{\boldsymbol{x}} = \boldsymbol{f}_z(\boldsymbol{x}, \boldsymbol{u}), & \boldsymbol{x}(\boldsymbol{0}) = \boldsymbol{x}_{\text{init}} \\ \langle \boldsymbol{x}, \boldsymbol{u} \rangle \in S_z, & \boldsymbol{x}(t_f) \in X_{\text{goal}} \end{matrix} \tag{1}$$

where $S_z := \{\langle \boldsymbol{x}, \boldsymbol{u} \rangle \mid \boldsymbol{g}_z(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{0}, \boldsymbol{h}_z(\boldsymbol{x}, \boldsymbol{u}) \geq \boldsymbol{0}\}$ is a set of feasible continuous states and inputs satisfying mode-dependent constraints and $\boldsymbol{f}_z(\boldsymbol{x}, \boldsymbol{u})$ represents the mode-dependent system dynamics. For the tasks of interest here, the main objective is always specified in terms of a terminal set constraint that is based on a goal region X_{goal} and an unknown final time t_f, whereas the cost function $L(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{z})$ consists of task-independent terms that simply penalize large inputs, unnecessary mode-switching, and deviations from desired nominal configurations. This effectively amounts to solving hybrid constraint satisfaction problems analogous to those arising in the TAMP domain (*68*). A primary advantage of such a formalism is that it rids us of a cumbersome cost engineering process that would otherwise be required for every newly introduced scenario.

On the other hand, handling a free end-time mixed-integer nonlinear optimization with a sparse objective using off-the-shelf numerical solvers is computationally intractable. The branching factor involved in standard combinatorial techniques such as branch-and-bound would scale poorly with respect to the number of discrete modes, the time horizon, and the number of time steps used when discretizing the dynamics (*86*). Alternatively, we propose a formulation that loosens the original problem in Eq. 1 by identifying and exploiting the special structure emerging in multicontact loco-manipulation planning.

First, considering that manipulation modes mostly remain constant over multiple consecutive time steps, we divided the full trajectory into coarse intervals of mode-invariant phases. This allows mode switching to occur less frequently along the horizon, thereby reducing the number of discrete decision variables. Second, we introduced an alternative representation by substituting the set of integer variables in Eq. 1 with an equivalent set of contact states $\boldsymbol{s} \in \mathscr{S}$ and contact switching actions $\boldsymbol{a} \in \mathscr{A}$. Such a representation enables us to define a transition map $\boldsymbol{\Gamma} : \mathscr{S} \times \mathscr{A} \to \mathscr{S}$, along with a set of feasible discrete actions $\overline{\mathscr{A}} \subseteq \mathscr{A}$ constructed using domain-specific logic rules. Applying these modifications yields a more manageable "mixed-logic program" whose structure can be leveraged for a fast and effective pruning of infeasible branches.

We casted this program in the form of a bilevel optimization consisting of a graph search over discrete state-action sequences interleaved with a contact-driven trajectory optimization that is parameterized by the manipulation modes. The minimization is

framed as follows:

$$\min_{\{\boldsymbol{s}_k,\boldsymbol{a}_k,\boldsymbol{x}_{T_k}\}} \sum_{k=0}^{N-1} E(\boldsymbol{s}_k, \boldsymbol{a}_k, \boldsymbol{x}_{T_k})$$
$$\text{such that } \langle \boldsymbol{s}_0, \boldsymbol{x}_{T_0} \rangle = \langle \boldsymbol{s}_{\text{init}}, \boldsymbol{x}_{\text{init}} \rangle, \boldsymbol{x}_{T_N} \in X_{\text{goal}}$$
$$\boldsymbol{s}_{k+1} = \boldsymbol{\Gamma}(\boldsymbol{s}_k, \boldsymbol{a}_k), \boldsymbol{a}_k \in \overline{\mathscr{A}}(\boldsymbol{s}_k, \boldsymbol{s}_{k-1}, \boldsymbol{x}_{T_k})$$
$$\boldsymbol{x}_{T_{k+1}} \in \begin{cases} \underset{\boldsymbol{x}(\cdot),\boldsymbol{u}(\cdot)}{\arg\min}\, \Phi(\boldsymbol{x}_T, \tilde{\boldsymbol{r}}) + \int_0^T \mathscr{L}(\boldsymbol{x}, \boldsymbol{u}, \tilde{\boldsymbol{r}}) dt \\ \quad \text{such that } \boldsymbol{x}(0) = \boldsymbol{x}_{T_k}, \\ \quad \dot{\boldsymbol{x}} = f(x, u \mid \boldsymbol{s}_k, \boldsymbol{a}_k), \\ \quad \boldsymbol{g}(x, u, t \mid \boldsymbol{s}_k, \boldsymbol{a}_k) = \boldsymbol{0}, \\ \quad \boldsymbol{h}(x, u, t \mid \boldsymbol{s}_k, \boldsymbol{a}_k) \geq \boldsymbol{0} \end{cases} \quad (2)$$

where N is the unspecified number of directed edges connecting the starting node $\langle \boldsymbol{s}_0, \boldsymbol{x}_{T_0} \rangle$ to the goal region X_{goal}. Each edge connecting a node $\langle \boldsymbol{s}_k, \boldsymbol{x}_{T_k} \rangle$ to its neighboring node $\langle \boldsymbol{s}_{k+1}, \boldsymbol{x}_{T_{k+1}} \rangle$ is assigned a transition cost E including both discrete and continuous terms. The continuous component is deduced from the solution of a mode-invariant OCP with a fixed time horizon T and an optimal terminal state $\boldsymbol{x}_{T_{k+1}}$. The inner-level OCP minimizes an objective consisting of an intermediate cost $\mathscr{L}$ and a terminal cost Φ driven by a reference $\tilde{\boldsymbol{r}} \in \mathscr{R} \subseteq \mathbb{R}^{n_x}$ and is subject to the dynamics and path constraints that are conditioned on a fixed contact mode $\langle \boldsymbol{s}_k, \boldsymbol{a}_k \rangle$. The different elements in the above formulation will be discussed in detail in the remaining subsections.

Robot-object interaction rules

From a user-defined collection of n_e robot end effectors and n_c potential object contacts, we constructed a set of contact states $\mathscr{S} := \{0, 1, \ldots, n_c\}^{n_e}$ that encodes all possible interaction combinations. For instance, a state $\boldsymbol{s} = (0\ 0\ 2)$ indicates that the first two end effectors are open contacts and that the third one is interacting with the second object contact. Regarding our action space $\mathscr{A}$, we aimed to define a minimal set of admissible contact switches because the number of expanded edges in the graph is primarily dictated by the size of that set. To this end, we restricted every action to affect one limb at a time such that it can either keep the same contact state (open/closed), break a closed contact, or establish a new one. Similar strategies have been previously adopted for acyclic locomotion planning (*59*) and dexterous manipulation (*67*). This rule already enables us to adopt a compact representation for contact switching actions $\boldsymbol{a} \in \{0, 1, ..., n_e\} \times \{0, 1, ..., n_c\}$ where the first and second elements correspond to the limb and object contact indices, respectively. In other words, an action $\boldsymbol{a} = (e\ c)$ indicates that limb e should establish a new object contact corresponding to index c. Moreover, the end effector e should break its current contact if $c = 0$, whereas the same contact state is maintained if $e = 0$. Accordingly, the definition of the transition map Γ readily follows as

$$s_{k+1}[\mathrm{i}] = \begin{cases} s_k[1] & \text{if } i \neq a_k[1] \quad \forall i = 1, \ldots, n_e \\ a_k[2] & \text{if } i = a_k[1] \quad \forall i = 1, \ldots, n_e \end{cases} \quad (3)$$

The branching factor can be further reduced with additional logic rules that make use of loco-manipulation domain knowledge. A central component behind this is the manual classification of robot-object contacts based on certain key attributes: End effectors are categorized as prehensile/nonprehensile arm contacts or prehensile/nonprehensile feet that participate in both manipulation and locomotion, whereas object contacts are categorized as prehensile/nonprehensile points or nonprehensile surfaces. Consequently, we can restrict the space of admissible actions $\overline{\mathscr{A}}$ by introducing a set of basic rules (both generic and domain-specific): For instance, an end effector is allowed to establish a new object contact only from a previously open contact state; a point object contact can only be occupied by a single limb; and nonprehensile robot contacts cannot be paired with prehensile object contacts. The full list of such pruning rules is provided in the Supplementary Methods (Loco-manipulation logic rules section).

Contact-driven trajectory optimization

When enumerating all possible contact switches, the above conditions only provide us with a fast feasibility checker that marks which edges cannot be expanded. However, an additional step is required to ensure that the admissible modes are also kinematically and dynamically consistent. This can be achieved by adopting sufficiently complex dynamical models in the inner-level optimization of Eq. 2. We based the OCP formulation here on our previous work (*16*), wherein any arbitrary mobile manipulator is treated as a multi-limbed and polyarticulated floating-base system. Unlike most multicontact planners that rely on highly simplified kinematic or quasi-static models (*55*, *58*, *59*, *62*, *87*), the core idea is to choose a minimal yet high-fidelity model description that captures the dominant coupling effects between the robot's base, its limbs, and the manipulated object. Such a representation can be attained using the robot's full centroidal dynamics (*88*) and a first-order kinematic model augmented with the object's full dynamics. Consequently, we defined the continuous state and input as follows: $\boldsymbol{x}$: = $(\boldsymbol{x}_r\ \boldsymbol{x}_o)$ = $(\boldsymbol{h}_{com}\ \boldsymbol{q}_b\ \boldsymbol{q}_j\ \boldsymbol{q}_o\ \boldsymbol{v}_o)$ and $\boldsymbol{u} = (\boldsymbol{w}_e\ \boldsymbol{v}_j)$. The robot state $\boldsymbol{x}_r$ consists of the centroidal momentum $\boldsymbol{h}_{com}$, base pose $\boldsymbol{q}_b$, and joint positions $\boldsymbol{q}_j$, whereas the object state $\boldsymbol{x}_o$ comprises its generalized coordinates $\boldsymbol{q}_o$ and velocities $\boldsymbol{v}_o$. The input $\boldsymbol{u}$ stacks the contact wrenches $\boldsymbol{w}_e$ acting at the end effectors and the joint velocities $\boldsymbol{v}_j$. The inclusion of the manipulandum's dynamics in the EoM presumes a reasonable knowledge of the object's nominal model, in addition to its kinematic and dynamic parameters. Moreover, in contrast to our previous formulation in (*16*), the set of end-effector contact forces appearing in the object dynamics is now conditioned on the manipulation mode. A complete description of the EoM is provided in the Supplementary Methods (Contact-driven OCP formulation: Dynamics section).

The central design principle underlying our framework is one that aims to capture various loco-manipulation behaviors in a unified manner by predominantly relying on a constraint-based formulation while keeping the cost engineering process as lightweight as possible. Because of the generic contact classification discussed earlier, the same constraint set would be applicable to diverse scenarios involving arbitrary mobile manipulators or articulated objects. Besides standard continuously active constraints enforcing system operational limits, workspace limits, and collision avoidance (*89*), we introduced a collection of contact-driven constraints that are parameterized by the manipulation mode. The mode can be readily extracted from the current contact state $\boldsymbol{s}_k$ and the contact switching action $\boldsymbol{a}_k$. For instance, a closed and prehensile contact state would entail a zero relative twist condition between the gripper and the graspable object contact. In contrast, a nonprehensile interaction (wherein only sticking contacts are assumed) would

only constrain their relative linear velocity while ensuring the end-effector force lies within the friction cone. Furthermore, we preserved the same time-based switched system structure adopted in (*16*, *90*) when defining our constraints, hence the time dependence in the path constraints of Eq. 2. By this, we are able to encode certain hybrid behaviors that can be described with basic mode schedules: a fixed mode sequence coupled with a set of switching times. These could include cyclic gaits that are primarily relevant for legged mobile manipulators or simple manipulation schedules, such as breaking a closed contact or establishing a new contact. An elaborate description of the OCP constraints is presented in the Supplementary Methods (Contact-driven OCP formulation: Constraints section).

The definition of the OCP cost function consisting of $\mathscr{L}$ and Φ is straightforward, because it only encompasses simple quadratic terms that mostly use the same set of weighting matrices for different tasks. As described in the Supplementary Methods (Contact-driven OCP formulation: Cost function section), they include elements that penalize large inputs and regularize the robot's state around a nominal configuration, as well as a task-dependent term that takes care of tracking target references $\tilde{\boldsymbol{r}}$ corresponding to both the object and the robot's base.

Offline planner: Sampling-based bilevel optimization

To address the bilevel optimization in Eq. 2, we start by noting that we did not encode our loco-manipulation tasks in the form of a terminal symbolic state $\boldsymbol{s}_N$ but rather as a goal region to which a reachable continuous state $\boldsymbol{x}_{T_N}$ must belong. This would typically require augmenting the original graph state (namely, the contact state) with a discretization of the continuous space (*72*). Alternatively, instead of inefficiently searching through a dense graph of high-dimensional augmented states, we resorted to an approximate sampling-based solution to the bilevel search. Such a scheme enables a rapid computation of feasible solutions and offers an effective globalization property that helps avoid bad local minima. Although this happens at the expense of optimality, our primary target is nonetheless fulfilled: discovering a physically consistent suboptimal plan that solves the task.

To elaborate, the planning algorithm operates by incrementally building a multimodal tree in which each tree node is characterized by its contact mode and a reachable terminal state extracted from the optimal trajectory of the corresponding edge. The tree expansion is guided by two key components: a node selection mechanism and a strategy that determines the cost targets $\tilde{\boldsymbol{r}}$ during node extensions. Similar in essence to (*91*), the latter relies on a sampling-based approach that drives our inner-level OCP with targets defined in a smaller subset of the full state space, namely, the reference space $\mathscr{R}$. Specifically, in our case, this corresponds to the robot's two-dimensional base pose and the object's generalized coordinates. The goal region itself is constructed in terms of our reference space (e.g., proximity to a target base position or object configuration), rendering the task-specification step notably simpler for the user. As described in the Supplementary Methods (Reference generation section), the cost reference $\tilde{\boldsymbol{r}} := \tilde{\boldsymbol{r}}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{x}_T, \tilde{\boldsymbol{y}}) \in \mathscr{R}$ basically determines the direction of extension and is generated as a function of the contact mode, the node's continuous state, and a random variable $\tilde{\boldsymbol{y}} \in SE(2) \times \mathbb{R}^{n_o}$. We sampled $\tilde{\boldsymbol{y}}$ either from a uniform distribution $\mathscr{U}$ or a goal-directed normal distribution $\mathscr{N}$ whose mean is used in constructing the goal set X_{goal}.

As for the outer-level selection step, each tree node is first assigned a cumulative cost $C_n = \sum_{i=0}^{n-1} E_i + \alpha H$ that is composed of the accumulated edge costs from the starting node and a heuristic function H. The one with the lowest cost C_n is then chosen for expansion using an anytime nonparametric A* (ANA*) strategy (*92*). This method aims for a quick discovery of feasible plans by initially applying a best-first selection criterion and then keeps enhancing the solution as it converges toward an A*-type search by gradually decreasing the weight α to one. Each edge cost combines a discrete element penalizing contact switches with a merit function composed of continuous regularization terms and penalties on constraint violations in the corresponding OCP. Moreover, we evaluated our heuristic cost as $H = N_{\text{segments}} \cdot E_{\text{average}}$, where N_{segments} roughly estimates the number of unconstrained trajectory segments required by the robot/object to reach the goal assuming that it can move freely at maximum velocity, whereas E_{average} denotes the average edge cost of all branches in the existing tree. The aim with such a choice is to ultimately converge toward a near-optimal solution by approaching an admissible heuristic function that underestimates the true cost-to-go.

The pseudo-code for an elementary version of the algorithm used to solve the bilevel optimization in Eq. 2 is presented below:

Algorithm 1: Multicontact loco-manipulation planner.
Initialize $\boldsymbol{s}_{\text{init}}$, $\boldsymbol{x}_{\text{init}}$, $\mathscr{X}_{\text{goal}}$, $\mathscr{N}$, $\mathscr{U}$, T, α
Define tree $\mathscr{T}$, set of open nodes $\mathscr{O}$, and solution set $\mathscr{S}^*$
Initialize start node n_s using initial states and append to $\mathscr{T}$
  while not termination condition reached **do**
    /* Goal directed extension */
    Draw sample $\tilde{\boldsymbol{y}}$ from $\mathscr{N}$
    while not $\mathscr{O}$ is empty and not max iterations reached **do**
      Set node last appended to $\mathscr{T}$ as initial node n_i
      if n_i state belongs to $\mathscr{X}_{\text{goal}}$ **then**
        Append n_i to $\mathscr{S}^*$; update weight α; break
      end if
      Extend n_i and update $\mathscr{O}$, $\mathscr{T}$: GenerateSuccessors(n_i, $\tilde{\boldsymbol{y}}$)
    end while
    /* Uniformly random extension */
    while not new node added to tree **do**
      Draw sample $\tilde{\boldsymbol{y}}$ from $\mathscr{U}$
      Set node in $\mathscr{T}$ nearest to $\tilde{\boldsymbol{y}}$ as n_i using a proper metric
      Extend n_i and update $\mathscr{O}$, $\mathscr{T}$: GenerateSuccessors(n_i, $\tilde{\boldsymbol{y}}$)
    end while
  end while
Algorithm 2: Function to generate successors in Algorithm 1.
function GenerateSuccessors(n_i, $\tilde{\boldsymbol{y}}$)
  for each $\boldsymbol{a} \in \overline{\mathscr{A}}(n_i)$ **do**
    Compute OCP references $\tilde{\boldsymbol{r}}(n_i, \boldsymbol{a}, \tilde{\boldsymbol{y}})$
    Solve OCP given n_i, $\boldsymbol{a}$, $\tilde{\boldsymbol{r}}$, and the time horizon T
    if OCP solution found **then**
      Set state and cost info of new node n_f from solution
      if cost from n_s to n_f < smallest cost in $\mathscr{S}^*$ **then**
        Append n_f to $\mathscr{O}$
      end if
    end if
  end for
  Select node in $\mathscr{O}$ with least cumulative cost and append to $\mathscr{T}$
end function

In summary, the above algorithm boils down to an alternation between goal-directed tree expansions and an RRT-like exploration of the reference space. The core component driving our planner is a targeted balance between exploration and exploitation. This aspect was fundamental in choosing an appropriate nearest-neighbor metric and in our introduction of supplementary pruning strategies that help maintain a sparse yet sufficiently rich tree. Implementation details regarding the underlying OCP solver, the exploration metric, and the pruning process can be found in the Supplementary Methods (Implementation details section), whereas the common hyperparameters used for all tasks are given in table S4.

Offline planner: Long-horizon trajectory optimization

A complete plan connecting a sequence of smooth trajectories may not necessarily be smooth itself, especially when randomness is involved in generating the plan. Therefore, we introduced a postprocessing step that enhances the quality of the solutions obtained from the bilevel search. To this end, we used the trajectory sequence as a feasible initial guess to warm start a long-horizon trajectory optimization routine. Although not required, the sequence can also be applied as a cost reference $\tilde{\boldsymbol{r}}$ to guide the optimization. The resulting OCP structure resembles that in Eq. 2, but at this stage, the short time horizon T is replaced with the total duration $N \cdot T$ needed to solve the loco-manipulation task. Furthermore, the state-action pairs governing the path constraints over the N time segments are now predetermined from the discovered contact schedule.

Apart from smoothening the full trajectory and getting rid of superfluous motions, the long-horizon optimization offers extra benefits with regard to how mode switches are handled. Unlike the contact-dependent OCPs in Eq. 2, the added look-ahead capabilities result in refined solutions that suitably alter contact locations during mode switching while accounting for future events and maneuvers (e.g., adapted foothold locations or adapted end-effector positions when interacting with a surface contact). Ultimately, this leads to smaller contact forces and more robust behaviors associated with increased stability and higher chances for successful task attainment.

Two-layer tracking controller

When it comes to executing the offline multicontact plans on the real robot, a pointwise tracking scheme would normally be sufficient for quasi-static motions and statically stable systems. Because this does not hold in our case of a quadrupedal mobile manipulator performing dynamic loco-manipulation tasks, we resorted to a two-layer tracking architecture providing stronger robustness properties. A schematic diagram detailing the interconnections between the two layers is provided in fig. S4.

The first layer is a whole-body model predictive controller (*16*) that acts as an effective filter with disturbance-rejection properties. Incorporating this component is instrumental in achieving behaviors involving multiple contact switches. Otherwise, such phenomena would induce unmodeled effects that could render the reference trajectory unstabilizable with a simple one-step look-ahead strategy. In contrast to (*16*), the MPC in this work is perceived as a tracker of precomputed physically consistent plans rather than a planner that generates new solutions for object-centric or robot-centric objectives. With such a perspective, we managed to keep the MPC formulation as elementary as possible by removing the object state $\boldsymbol{x}_o$, reducing the number of constraints, and adopting simple state-input quadratic cost functions driven by the robot's offline references $\langle \boldsymbol{x}_r^*, \boldsymbol{u}^* \rangle$. As a result, we obtained a task-independent MPC-based tracker with a considerable gain in its computational rate. The adopted MPC cost weights are shown in table S6.

The second layer is a whole-body reactive controller that outputs joint position-velocity-torque commands to track the higher-level MPC references. The references are first mapped into desired generalized coordinates and velocities using conversions between the full centroidal dynamic model and the full rigid-body dynamics (*16*). A series of cascaded quadratic programs are then solved in a hierarchical fashion based on a list of prioritized tasks (*93*, *94*). This results in optimal generalized accelerations and contact forces that are used to compute reference joint torques from the inverse dynamics. A list of the tasks included in the hierarchical QP along with their corresponding priorities is provided in table S7.

Supplementary Materials

This PDF file includes:
Methods and Discussion
Figs. S1 to S4
Tables S1 to S7
References (*95*–*109*)

Other Supplementary Material for this manuscript includes the following:
Movies S1 to S7

REFERENCES AND NOTES

1. "New dog-like robot from Boston Dynamics can open doors (video)," *Guardian*, 12 February 2018; www.theguardian.com/technology/video/2018/feb/13/new-dog-like-robot-from-boston-dynamics-can-open-doors-video
2. D. Lee, H. Seo, D. Kim, H. J. Kim, in *2020 IEEE International Conference on Robotics and Automation, ICRA 2020* (IEEE, 2020), pp. 1237–1242.
3. D. Lee, H. Seo, I. Jang, S. J. Lee, H. J. Kim, Aerial manipulator pushing a movable structure using a DOB-based robust controller. *IEEE Robot. Autom. Lett.* **6**, 723–730 (2021).
4. J. Pankert, M. Hutter, Perceptive model predictive control for continuous mobile manipulation. *IEEE Robot. Autom. Lett.* **5**, 6177–6184 (2020).
5. J. Pankert, G. Valsecchi, D. Baret, J. Zehnder, L. L. Pietrasik, M. Bjelonic, M. Hutter, Design and motion planning for a reconfigurable robotic base. arXiv:2206.15298 (2022). https://doi.org/10.48550/arXiv.2206.15298
6. S. Chitta, B. J. Cohen, M. Likhachev, in *IEEE International Conference on Robotics and Automation, ICRA 2010* (IEEE, 2010), pp. 1799–1806.
7. Y. Karayiannidis, C. Smith, F. E. V. Barrientos, P. Ögren, D. Kragic, An adaptive control approach for opening doors and drawers under uncertainties. *IEEE Trans. Robot.* **32**, 161–175 (2016).
8. M. Arduengo, C. Torras, L. Sentis, Robust and adaptive door operation with a mobile robot. *Intell. Serv. Robot.* **14**, 409–425 (2021).
9. M. Murooka, S. Nozawa, Y. Kakiuchi, K. Okada, M. Inaba, Whole-body pushing manipulation with contact posture planning of large and heavy object for humanoid robot, in *IEEE International Conference on Robotics and Automation, ICRA 2015* (IEEE, 2015), pp. 5682–5689.
10. D. Berenson, S. S. Srinivasa, J. J. Kuffner, Task space regions. *Int. J. Robot. Res.* **30**, 1435–1460 (2011).
11. F. Burget, A. Hornung, M. Bennewitz, Whole-body motion planning for manipulation of articulated objects, in *2013 IEEE International Conference on Robotics and Automation* (IEEE, 2013), pp. 1656–1662.
12. S. J. Jorgensen, M. Vedantam, R. Gupta, H. Cappel, L. Sentis, in *2020 IEEE International Conference on Robotics and Automation, ICRA 2020* (IEEE, Paris, France, May 31 to 31 August 2020), pp. 6611–6617.
13. M. P. Murphy, B. Stephens, Y. Abe, A. A. Rizzi, *Proc. SPIE* **8387**, Unmanned Systems Technology XIV, 83870V (2012).
14. S. Zimmermann, R. Poranne, S. Coros, in *IEEE International Conference on Robotics and Automation, ICRA 2021* (IEEE, 2021), pp. 4488–4494.

15. C. D. Bellicoso, K. Krämer, M. Stäuble, D. Sako, F. Jenelten, M. Bjelonic, M. Hutter, ALMA - Articulated Locomotion and Manipulation for a Torque-Controllable Robot, in *International Conference on Robotics and Automation, ICRA, 2019* (IEEE, 2019), pp. 8477–8483.
16. J. Sleiman, F. Farshidian, M. V. Minniti, M. Hutter, A unified MPC framework for whole-body dynamic locomotion and manipulation. *IEEE Robot. Autom. Lett.* **6**, 4688–4695 (2021).
17. M. Mittal, D. Hoeller, F. Farshidian, M. Hutter, A. Garg, Articulated Object Interaction in Unknown Scenes with Whole-Body Mobile Manipulation. arXiv:2103.10534 (2021). https://doi.org/10.48550/arXiv.2103.10534
18. L. Sentis, O. Khatib, in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006* (IEEE, Orlando, FL, USA, 15 to 19 May 2006), pp. 2641–2648.
19. M. V. Minniti, F. Farshidian, R. Grandia, M. Hutter, Whole-body MPC for a dynamically stable mobile manipulator. *IEEE Robot. Autom. Lett.* **4**, 3687–3694 (2019).
20. F. Ruggiero, V. Lippiello, A. Ollero, Aerial manipulation: A literature review. *IEEE Robot. Autom. Lett.* **3**, 1957–1964 (2018).
21. V. Morlando, M. Selvaggio, F. Ruggiero, in *2022 International Conference on Robotics and Automation, ICRA 2022* (IEEE, 2022), pp. 6628–6634.
22. M. P. Polverini, A. Laurenzi, E. M. Hoffman, F. Ruscelli, N. G. Tsagarakis, Multi-contact heavy object pushing with a centaur-type humanoid robot: Planning and control for a real demonstrator. *IEEE Robot. Autom. Lett.* **5**, 859–866 (2020).
23. J. E. Pratt, J. Carff, S. V. Drakunov, A. Goswami, in *2006 6th IEEE-RAS International Conference on Humanoid Robots* (IEEE, 2006), pp. 200–207.
24. C. D. Bellicoso, F. Jenelten, C. Gehring, M. Hutter, Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots. *IEEE Robot. Autom. Lett.* **3**, 2261–2268 (2018).
25. G. Ji, J. Mun, H. Kim, J. Hwangbo, Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion. *IEEE Robot. Autom. Lett.* **7**, 4630–4637 (2022).
26. J. Siekmann, K. Green, J. Warila, A. Fern, J. W. Hurst, in *Robotics: Science and Systems XVII*, D. A. Shell, M. Toussaint, M. A. Hsieh, Eds. (2021).
27. T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci. Robot.* **7**, eabk2822 (2022).
28. H. Zhu, A. Gupta, A. Rajeswaran, S. Levine, V. Kumar, in *International Conference on Robotics and Automation, ICRA 2019* (IEEE, 2019), pp. 3651–3657.
29. A. Rajeswaran et al., in *Robotics: Science and Systems XIV*, H. Kress-Gazit, S. S. Srinivasa, T. Howard, N. Atanasov, Eds. (2018).
30. OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. M. Grew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, W. Zaremba, Learning Dexterous In-Hand Manipulation. arXiv:1808.00177 [cs.LG] (2018).
31. Z. Fu, X. Cheng, D. Pathak, arXiv abs/2210.10044 (2022).
32. Y. Ma, F. Farshidian, T. Miki, J. Lee, M. Hutter, Combining learning-based locomotion policy with model-based manipulation for legged mobile manipulators. *IEEE Robot. Autom. Lett.* **7**, 2377–2384 (2022).
33. Y. Ma, F. Farshidian, M. Hutter, Learning arm-assisted fall damage reduction and recovery for legged mobile manipulators, in *2023 IEEE International Conference on Robotics and Automation* (IEEE, 2023); 10.1109/ICRA48891.2023.10160582.
34. N. Rudin, D. Hoeller, P. Reist, M. Hutter, in *Conference on Robot Learning*, A. Faust, D. Hsu, G. Neumann, Eds., vol. 164 of *Proceedings of Machine Learning Research* (PMLR, 2021), pp. 91–100.
35. W. Zhao, J. P. Queralta, T. Westerlund, in *2020 IEEE Symposium Series on Computational Intelligence, SSCI 2020* (IEEE, 2020), pp. 737–744.
36. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
37. A. Faust et al., in *2018 IEEE International Conference on Robotics and Automation, ICRA 2018* (IEEE, 2018), pp. 5113–5120.
38. X. B. Peng et al., in Robotics: Science and Systems XVI, M. Toussaint, A. Bicchi, T. Hermans, Eds. (2020).
39. S. Bohez, S. Tunyasuvunakool, P. Brakel, F. Sadeghi, L. Hasenclever, Y. Tassa, E. Parisotto, J. Humplik, T. Haarnoja, R. Hafner, M. Wulfmeier, M. Neunert, B. Moran, N. Siegel, A. Huber, F. Romano, N. Batchelor, F. Casarini, J. Merel, R. Hadsell, N. Heess, Imitate and Repurpose: Learning Reusable Robot Movement Skills From Human and Animal Behaviors. arXiv:2203.17138 [cs.RO] (2022).
40. J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.* **5**, 5986 (2020).
41. A. Kumar, Z. Fu, D. Pathak, J. Malik, in *Robotics: Science and Systems XVII*, D. A. Shell, M. Toussaint, M. A. Hsieh, Eds. (2021).
42. B. Aceituno-Cabezas, C. Mastalli, H. Dai, M. Focchi, A. Radulescu, D. G. Caldwell, J. Cappelletto, J. C. Grieco, G. Fernandez-Lopez, C. Semini, Simultaneous Contact, Gait and Motion Planning for Robust Multi-Legged Locomotion via Mixed-Integer Convex Optimization. arXiv:1904.04595 [cs.RO] (2019).
43. B. Aceituno-Cabezas, A. Rodriguez, in *Robotics: Science and Systems XVI*, M. Toussaint, A. Bicchi, T. Hermans, Eds. (2020).
44. I. Mordatch, E. Todorov, Z. Popovic, Discovery of complex behaviors through contact-invariant optimization. *ACM Trans. Graph.* **31**, 1–8 (2012).
45. I. Mordatch, Z. Popovic, E. Todorov, in *Proceedings of the 2012 Eurographics/ACM SIGGRAPH Symposium on Computer Animation, SCA 2012*, J. Lee, P. G. Kry, eds. (Eurographics Association, 2012), pp. 137–144.
46. M. Neunert, M. Stauble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, J. Buchli, Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robot. Autom. Lett.* **3**, 1458–1465 (2018).
47. Y. Tassa, T. Erez, E. Todorov, in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012* (IEEE, 2012), pp. 4906–4913.
48. Y. Tassa, E. Todorov, in *Robotics: Science and Systems VI*, Y. Matsuoka, H. F. Durrant-Whyte, J. Neira, Eds. (MIT Press, 2010).
49. M. Anitescu, F. A. Potra, Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *Nonlinear Dyn.* **14**, 231–247 (1997).
50. M. Posa, C. Cantu, R. Tedrake, A direct method for trajectory optimization of rigid bodies through contact. *Int. J. Robot. Res.* **33**, 69–81 (2014).
51. J. Carius, R. Ranftl, V. Koltun, M. Hutter, Trajectory optimization with implicit hard contacts. *IEEE Robot. Autom. Lett.* **3**, 3316–3323 (2018).
52. B. Landry, J. Lorenzetti, Z. Manchester, M. Pavone, *Robotics Research - The 19th International Symposium ISRR 2019*, T. Asfour, E. Yoshida, J. Park, H. Christensen, O. Khatib, Eds., vol. 20 of *Springer Proceedings in Advanced Robotics* (Springer, 2019), pp. 789–804.
53. Y. Zhu, Z. Pan, K. Hauser, in *IEEE International Conference on Robotics and Automation, ICRA 2021* (IEEE, 2021), pp. 9921–9927.
54. S. M. LaValle, *Planning Algorithms* (Cambridge University Press, 2009).
55. S. Dalibard, A. Nakhaei, F. Lamiraux, J. Laumond, in *10th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2010* (IEEE, 2010), pp. 518–523.
56. S. Gray, S. Chitta, V. Kumar, M. Likhachev, in *2013 IEEE International Conference on Robotics and Automation* (IEEE, 2013), pp. 3839–3846.
57. K. K. Hauser, V. Ng-Thow-Hing, Randomized multi-modal motion planning for a humanoid robot manipulation task. *Int. J. Robot. Res.* **30**, 678–698 (2011).
58. M. Murooka, I. Kumagai, M. Morisawa, F. Kanehiro, A. Kheddar, Humanoid loco-manipulation planning based on graph search and reachability maps. *IEEE Robot. Autom. Lett.* **6**, 1840–1847 (2021).
59. S. Tonneau, A. Del Prete, J. Pettre, C. Park, D. Manocha, N. Mansard, An efficient acyclic contact planner for multiped robots. *IEEE Trans. Robot.* **34**, 586–601 (2018).
60. A. Short, T. Bandyopadhyay, Legged motion planning in complex three-dimensional environments. *IEEE Robot. Autom. Lett.* **3**, 29–36 (2018).
61. J. E. King, J. A. Haustein, S. S. Srinivasa, T. Asfour, in *IEEE International Conference on Robotics and Automation, ICRA 2015* (IEEE, 2015), pp. 2508–2515.
62. X. Cheng, E. Huang, Y. Hou, M. T. Mason, in *IEEE International Conference on Robotics and Automation, ICRA 2021* (IEEE, 2021), pp. 6520–6526.
63. A. Pasricha, Y. Tung, B. Hayes, A. Roncone, PokeRRT: Poking as a skill and failure recovery tactic for planar non-prehensile manipulation. *IEEE Robot. Autom. Lett.* **7**, 4480–4487 (2022).
64. J. Z. Woodruff, K. M. Lynch, in *2017 IEEE International Conference on Robotics and Automation, ICRA 2017* (IEEE, 2017), pp. 4066–4073.
65. Y. Lin, B. Ponton, L. Righetti, D. Berenson, in *International Conference on Robotics and Automation, ICRA 2019* (IEEE, 2019), pp. 5280–5286.
66. H. J. T. Suh, X. Xiong, A. Singletary, A. D. Ames, J. W. Burdick, in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2020* (IEEE, 2020), pp. 7027–7033.
67. C. Chen, P. Culbertson, M. Lepert, M. Schwager, J. Bohg, in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2021* (IEEE, 2021), pp. 8262–8268.
68. C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, T. Lozano-Pérez, Integrated Task and Motion Planning. arXiv:2010.01083 [cs.RO] (2020).
69. M. Toussaint, in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015*, Q. Yang, M. J. Wooldridge, eds. (AAAI Press, 2015), pp. 1930–1936.
70. M. Toussaint, K. R. Allen, K. A. Smith, J. B. Tenenbaum, in *Robotics: Science and Systems XIV*, H. Kress-Gazit, S. S. Srinivasa, T. Howard, N. Atanasov, Eds. (2018).
71. T. Migimatsu, J. Bohg, Object-centric task and motion planning in dynamic environments. *IEEE Robot. Autom. Lett.* **5**, 844–851 (2020).

72. T. Stouraitis, I. Chatzinikolaidis, M. Gienger, S. Vijayakumar, Online hybrid motion planning for dyadic collaborative manipulation via bilevel optimization. *IEEE Trans. Robot.* **36**, 1452–1471 (2020).
73. Z. Zhao, Z. Zhou, M. Park, Y. Zhao, SyDeBO: Symbolic-decision-embedded bilevel optimization for long-horizon manipulation in dynamic environments. *IEEE Access* **9**, 128817–128826 (2021).
74. M. Hutter et al., in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2016* (IEEE, 2016), pp. 38–44.
75. ANYbotics, ANYmal D – Autonomous Legged Robot. URL: https://www.anybotics.com/anymal-legged-robot.
76. H. Ferrolho, V. Ivan, W. Merkt, I. Havoutis, S. Vijayakumar, Roloma: Robust loco-manipulation for quadruped robots with arms. arXiv:2203.01446 [cs.RO] (2022).
77. A. Winkler, D. Bellicoso, M. Hutter, J. Buchli, *IEEE Robotics and Automation Letters* (2018).
78. M. Bjelonic, R. Grandia, M. Geilinger, O. Harley, V. S. Medeiros, V. Pajovic, E. Jelavic, S. Coros, M. Hutter, Offline motion libraries and online MPC for advanced mobility skills. *Intl. J. Robot. Res.* **41**, 903–924 (2022).
79. S. Kuindersma, "Recent progress on Atlas, the world's most dynamic humanoid robot (video)," *Robotics Today*; www.youtube.com/watch?v=EGABAx52GKI.
80. M. Blösch et al., in *Robotics: Science and Systems VIII*, N. Roy, P. Newman, S. S. Srinivasa, Eds. (MIT Press, 2013).
81. X. B. Peng, Z. Ma, P. Abbeel, S. Levine, A. Kanazawa, AMP: Adversarial Motion Priors for Stylized Physics-Based Character Control. arXiv:2104.02180 [cs.GR] (2021).
82. E. Vollenweider et al., arXiv abs/2203.14912 (2022).
83. T. Zhang et al., in *2018 IEEE International Conference on Robotics and Automation, ICRA 2018* (IEEE, 2018), pp. 1–8.
84. H. Ito, K. Yamamoto, H. Mori, T. Ogata, Efficient multitask learning with an embodied predictive model for door opening and entry with whole-body control. *Sci. Robot.* **7**, eaax8177 (2022).
85. P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, A. Mahajan, Mixed-integer nonlinear optimization. *Acta Numer.* **22**, 1–131 (2013).
86. J. T. Betts, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming* (Cambridge University Press, 2010), second edn.
87. E. Jelavic, F. Farshidian, M. Hutter, Combined Sampling and Optimization Based Planning for Legged-Wheeled Robots, in *IEEE International Conference on Robotics and Automation, ICRA 2021* (IEEE, 2021), pp. 8366–8372.
88. D. E. Orin, A. Goswami, S. Lee, Centroidal dynamics of a humanoid robot. *Auton. Robot.* **35**, 161–176 (2013).
89. J. Chiu, J. Sleiman, M. Mittal, F. Farshidian, M. Hutter, in *2022 International Conference on Robotics and Automation, ICRA 2022* (IEEE, 2022), pp. 4686–4693.
90. F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, J. Buchli, in *IEEE International Conference on Robotics and Automation* (IEEE, 2017), pp. 93–100.
91. Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, G. Fiore, Real-time motion planning with applications to autonomous urban driving. *IEEE Trans. Control. Syst. Technol.* **17**, 1105–1118 (2009).
92. J. van den Berg, R. Shah, A. Huang, K. Y. Goldberg, in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2011*, W. Burgard, D. Roth, eds. (AAAI Press, 2011).
93. C. D Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, M. Hutter, Dynamic locomotion and whole-body control for quadrupedal robots, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, Vancouver, Canada, 24 to 28 September 2017), pp. 3359–3365.
94. C. D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, M. Hutter, in *16th IEEE-RAS International Conference on Humanoid Robots* (IEEE, Cancun, Mexico, 15 to 17 November 2016), pp. 558–564.
95. J. D. Carlo, P. M. Wensing, B. Katz, G. Bledt, S. Kim, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2018), pp. 1–9.
96. R. Grandia, F. Farshidian, R. Ranftl, M. Hutter, in *IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2019), pp. 4730–4737.
97. E. Gilbert, D. Johnson, S. Keerthi, A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE J. Robot. Autom.* **4**, 193–203 (1988).
98. J. Pan, S. Chitta, D. Manocha, in *2012 IEEE International Conference on Robotics and Automation* (IEEE, 2012), pp. 3859–3866.
99. R. Grandia, F. Jenelten, S. Yang, F. Farshidian, M. Hutter, Perceptive locomotion through nonlinear model predictive control. arXiv:2208.08373 (2022). https://doi.org/10.48550/arXiv.2208.08373
100. J. Nocedal, S. J. Wright, *Numerical Optimization* (Springer, 2006).
101. G. Frison, M. Diehl, HPIPM: A high-performance quadratic programming framework for model predictive control. arXiv:2003.02547 (2020). https://doi.org/10.48550/arXiv.2003.02547
102. J. Sleiman, F. Farshidian, M. Hutter, in *IEEE International Conference on Robotics and Automation, ICRA 2021* (IEEE, 2021), pp. 8209–8215.
103. J. Carpentier, F. Valenza, N. Mansard, Pinocchio: Fast forward and inverse dynamics for poly-articulated systems, https://stack-of-tasks.github.io/pinocchio (2015–2019).
104. J. Carpentier, in *IEEE International Symposium on System Integrations (SII)* (2019).
105. CppAD, Computational Infrastructure for Operations Research, https://cppad.readthedocs.io/en/latest/user_guide.html.
106. OCS2: An open source library for optimal control of switched systems. [Online].; https://zenodo.org/badge/latestdoi/384033964.
107. I. A. Sucan, M. Moll, L. E. Kavraki, The open motion planning library. *IEEE Robot. Autom. Mag.* **19**, 72–82 (2012).
108. L. Kocsis, C. Szepesvári, in *Machine Learning: ECML 2006*, J. Fürnkranz, T. Scheffer, M. Spiliopoulou, eds. (Springer, 2006), pp. 282–293.
109. G. Jocher, A. Chaurasia, A. Stoken, J. Borovec; NanoCode012, Y. Kwon, K. Michael, F. TaoXie, I. Jiacong, Z. Y. Lorna, C. Wong, V. Abhiram, D. Montes, Z. Wang, C. Fati, J. Nadar, U. Laughing, V. Sonck, Y. Tkianai, S. Piotr, A. Hogan, D. Nair, M. Strobel, M. Jain, ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation (2022).

Acknowledgments: We thank J. Preisig, S. Arreghini, and J.-R. Chiu for their valuable help in developing the autonomous door traversal pipeline and their help with hardware experiments. **Funding:** The project was funded, in part, by the Intel Network on Intelligent Systems, the Swiss National Science Foundation (SNF) through the National Centre of Competence in Research Robotics (NCCR Robotics), the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program grant agreement no. 852044, the Swiss National Science Foundation through the National Centre of Competence in Digital Fabrication (NCCR dfab), and TenneT TSO. This work has been conducted as part of ANYmal Research, a community to advance legged robotics. **Author contributions:** J.-P.S. formulated and implemented the planning and control architecture, designed and performed all simulations and real-world experiments, and wrote the manuscript. F.F. contributed to the development of the planning algorithm and the design of the experiments. F.F. and M.H. contributed to the analysis of the data, revised the manuscript, and refined ideas. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials. Other materials can be found at SupplementaryCode and SupplementaryData (DOI: 10.5061/dryad.vq83bk3zp).

Submitted 31 December 2022
Accepted 18 July 2023
Published 16 August 2023
10.1126/scirobotics.adg5014

Science Robotics

Versatile multicontact planning and control for legged loco-manipulation

Jean-Pierre Sleiman, Farbod Farshidian, and Marco Hutter

Sci. Robot. **8** (81), eadg5014. DOI: 10.1126/scirobotics.adg5014

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.