

Semester Thesis

Contact-Implicit MPC for Mobile Manipulation

Spring Term 2021

Supervised by:

Jean-pierre Sleiman
Dr. Farbod Farshidian
Prof. Dr. Marco Hutter

Authors:

Yifei Dong
Zeren Luo

Declaration of Originality

I hereby declare that the written work I have submitted entitled

Contact-Implicit MPC for Mobile Manipulation

is original work which I alone have authored and which is written in my own words.¹

Author(s)

Yifei
Zeren

Dong
Luo

Student supervisor(s)

Jean-pierre
Farbod

Sleiman
Farshidian

Supervising lecturer

Marco

Hutter

With the signature I declare that I have been informed regarding normal academic citation rules and that I have read and understood the information on ‘Citation etiquette’ (<https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesseliste/leistungskontrollen/plagiarism-citationetiquette.pdf>). The citation conventions usual to the discipline in question here have been respected.

The above written work may be tested electronically for plagiarism.

Zürich Nov.13, 2021
Place and date

Yifei Dong
Signature

¹Co-authored work: The signatures of all authors are required. Each signature attests to the originality of the entire piece of written work in its final form.

Intellectual Property Agreement

The student acted under the supervision of Prof. Hutter and contributed to research of his group. Research results of students outside the scope of an employment contract with ETH Zurich belong to the students themselves. The results of the student within the present thesis shall be exploited by ETH Zurich, possibly together with results of other contributors in the same field. To facilitate and to enable a common exploitation of all combined research results, the student hereby assigns his rights to the research results to ETH Zurich. In exchange, the student shall be treated like an employee of ETH Zurich with respect to any income generated due to the research results.

This agreement regulates the rights to the created research results.

1. Intellectual Property Rights

1. The student assigns his/her rights to the research results, including inventions and works protected by copyright, but not including his moral rights (“Urheberpersönlichkeitsrechte”), to ETH Zurich. Herewith, he cedes, in particular, all rights for commercial exploitations of research results to ETH Zurich. He is doing this voluntarily and with full awareness, in order to facilitate the commercial exploitation of the created Research Results. The student’s moral rights (“Urheberpersönlichkeitsrechte”) shall not be affected by this assignment.
2. In exchange, the student will be compensated by ETH Zurich in the case of income through the commercial exploitation of research results. Compensation will be made as if the student was an employee of ETH Zurich and according to the guidelines “Richtlinien für die wirtschaftliche Verwertung von Forschungsergebnissen der ETH Zürich”.
3. The student agrees to keep all research results confidential. This obligation to confidentiality shall persist until he or she is informed by ETH Zurich that the intellectual property rights to the research results have been protected through patent applications or other adequate measures or that no protection is sought, but not longer than 12 months after the collaborator has signed this agreement.
4. If a patent application is filed for an invention based on the research results, the student will duly provide all necessary signatures. He/she also agrees to be available whenever his aid is necessary in the course of the patent application process, e.g. to respond to questions of patent examiners or the like.

2. Settlement of Disagreements

Should disagreements arise out between the parties, the parties will make an effort to settle them between them in good faith. In case of failure of these agreements, Swiss Law shall be applied and the Courts of Zurich shall have exclusive jurisdiction.

Zürich Mar. 13, 2021

Place and date

Yifei Dong

Signature

Contents

Preface	v
Abstract	vii
Symbols	ix
1 Introduction	1
2 Problem Formulation	3
2.1 Model Formulation	3
2.1.1 Generic Cost Function	4
2.1.2 System Dynamics	4
2.1.3 System Constraints	5
2.2 Contact-implicit Approach	6
2.2.1 A Variant of Complementarity Model	6
2.2.2 Complementary Comments	6
2.3 Model Augmentation	7
2.3.1 Single-object Multi-contact Manipulation	7
2.3.2 Multi-object Single-contact Manipulation	8
2.3.3 Multi-object Multi-contact Manipulation	9
2.4 Full Control Architecture	9
3 Results	11
3.1 Single Contact Manipulation	11
3.2 Multiple Contacts Manipulation	13
3.3 Multiple Objects Manipulation	15
3.4 Contact Model	15
3.4.1 Complementarity Constraints	15
3.4.2 Advantages of CIO	16
3.5 Object-State Estimation	17
4 Conclusion and Future Work	19
Bibliography	22

Preface

The basis for this research originally stemmed from my interest in developing better methods of quadruped robot's mobile manipulation. As the world moves further into the automation age, robots are massively applied in our daily life and expected to be highly interactive with its surroundings. Robust and efficient mobile manipulation is currently in an immense demand. It is my passion to find out ways of effectively controlling articulated floating base system in mobile manipulation.

In truth, I could not have achieved my current level of fulfillment without a strong support group. First of all, Prof. Marco Hutter, Jean-pierre Sleiman and Dr. Farbod Farshidian, who supported me insightfully with great ideas and full heart. And secondly, my teammate Zeren Luo, who has provided patient advice throughout the research process. Most importantly, I am grateful to my parents and friends who always love me. Thank you all for your unwavering support.

Abstract

In this work, we present a legged robot optimal control framework with objects' dynamics augmented, capable of dynamic multi-object manipulation and mobile locomotion. We encode multiple manipulation contact points to tackle elaborate tasks and implement the mobile locomotion through pre-definitions of gait sequences in the switched system. The object manipulation and the mobile locomotion are performed in contact-implicit and multi-phase fashions, respectively. The dynamics of each manipulated-object are separately considered by introducing virtual contact points. The virtual contact points also decouple the gait-related forces and manipulation-related forces in the tasks of foot end-effector manipulation. Thus contact-implicit and multi-phase schemes could be applied in one contact point simultaneously.

On the other side, we assume a hard contact model and propose a reformulation of Contact-Implicit Optimization (CIO) by relaxing the complementarity contact model. The contact strategy of CIO is automatically generated by the solver in real-time, without the need of prior knowledge of mode sequences. The superiority of CIO aids us in resolving multi-object contact-rich manipulation problems involving shift of contact points and recapture of recoiling objects.

We devise an open-loop object-state estimation strategy in the whole body controller. By estimating the object state based on the system modeling without measurement, the open-loop state estimation could efficiently track the object's true values in Gazebo. We further test the feasibility of real-time SLQ-MPC solver in resolving various manipulation tasks with different manipulated objects, initial configurations and desired final states.

Keywords— Contact-Implicit Optimization, Multi-Object Manipulation, Trajectory Optimization, Legged Robots

Symbols

Symbols

\mathbf{x}	State Space
\mathbf{u}	Input Space
$\Phi(\cdot)$	Stage Cost
$\Phi_f(\cdot)$	Final Cost
\mathbf{h}_{com}	Robot's Center of Mass Momentum
\mathbf{q}_b	Robot's Base Pose
\mathbf{q}_j	Generalized Coordinate
\mathbf{f}_k	General Contact Forces
\mathbf{f}_{g_i}	Gait-related Contact Forces
\mathbf{f}_{m_i}	Manipulation-related Contact Forces
\mathbf{v}_j	Generalized Velocity
\mathbf{p}	Penalty Terms
$\boldsymbol{\tau}_i$	Contact Torques
n_a	Number of Actuation Joint
n_o	Number of Manipulated Object's Total DoFs
n_c	Number of Total Contact Points
n_g	Numbers of Gait-related Contact Points
n_{cio}	Numbers of Manipulation-related Contact Points
n_p	Number of Penalty Terms
$\{\mathcal{I}\}$	Inertia Frame
$\{\mathcal{B}\}$	Base Frame
$\{\mathcal{G}\}$	CoM Frame
$\{\mathcal{E}_i\}$	Limb's End-effector Frame
$\{\mathcal{O}_i\}$	Manipulated-objects Frame
\mathbf{r}_{IE}	Arm's End-effector Position
ζ_{IE}	Arm's End-effector Orientation Error
$\mathbf{Q}_{ee_p}, \mathbf{Q}_{ee_o}$	State Weight Matrices for EE Pose Control
\mathbf{Q}_r	Robot's State Weight Matrices
\mathbf{Q}_o	Object's State Weight Matrix
\mathbf{R}	Input Weight Matrix
\mathbf{g}	Gravity Constant
$\mathbf{r}_{com,i}$	Position of Contact Point i w.r.t CoM
\mathbf{A}_b	CMM of Base
\mathbf{A}_j	CMM of Joint
\mathbf{M}_o	Manipulated-object's Generalized Mass Matrix
\mathbf{b}_o	Nonlinear Term of the Object

\mathbf{J}_{c_o}	Manipulated-object's Matrix of Stacked Contact Jacobian
\mathcal{C}	Closed Contacts Set
μ_s	Friction Coefficient
$\phi_k(\mathbf{x})$	Generalized Gap Distance
$\lambda_k(\mathbf{u})$	Normal Contact Force
\mathbf{K}_p	Proportional Term in Joint Level Torque Control
\mathbf{K}_d	Derivative Term in Joint Level Torque Control

Acronyms and Abbreviations

CIO	Contact-Implicit Optimization
TO	Trajectory Optimization
NMPC	Nonlinear Model Predictive Control
ODE	Ordinary Differential Equation
OC	Optimal Control
NLP	Nonlinear Program
iLQR	iterative Linear Quadratic Regulator
DDP	Differential Dynamic Programming
SLQ	Sequential Linear Quadratic
CoM	Center of Mass
DoF	Degree of Freedom
EE	End-Effecter
CMM	Centroidal Momentum Matrices
MPC	Model Predictive Control
HQP	Hierarchical Quadratic Programming
WBC	Whole Body Controller
IMU	Inertia Measurement Unit
KF	Kalman Filter
ALMA C	A Poly-articulated Quadruped Robot

Chapter 1

Introduction

Nowadays research in the robotics community is putting more emphasis on the maneuverability and dexterity of mobile robots. Robots are expected to be highly interactive with its surroundings and able to deal with various mobile manipulation tasks, assisted by its flexible legs and high degrees of mobility. Quadrupedal robots with an on-board manipulator arm provide advantages for the sole-object mobile manipulation in an object-augmented dynamical system. In terms of complicated manipulation tasks involving several contact points and manipulated objects, work in the literature rarely engages owing to the elaborate planning framework.

Related work on manipulation investigates in sole-object manipulation tasks like door opening, payload throwing, etc. In an earlier framework, Murphy et al. [1] develop coordinated manipulation motion offline and the planning of object motion is separated from the locomotion planner. In the door-opening scenario of [2], a desired contact force is computed based on desired door angle and gripper velocity, and commanded at the gripper. The gripper passively follows the kinematically constrained path prescribed by the door motion. Sleiman et al. [3] solve the optimal policy within a formulated CIO program of pushing a load to a desired one-dimension coordinate.

The previous works above mostly resort to the multi-phase approach, which pre-specifies the mode sequences and hand-crafts the mode switching time. It requires prior knowledge of mode types and their order of occurrence. The multi-phase approach produces accurate and dynamically feasible trajectories. However, in practice, the pre-definition strategy is inapplicable in tasks of dynamically manipulating multiple objects using several contact points. The reasons are as follows: First, the contact modes and mode sequences increase exponentially as we introduce more contact points. Second, the feasible region shrinks with more manipulated objects, which makes the pre-determination of contact schemes computationally arduous.

To this end, contact-implicit approach, also called mode-invariant approach, serves as an appealing approach during the contact-rich and contact-unspecified interaction with the environment. Contact locations, timings and mode sequences are optimized in CIO rather than pre-specified. In the contact-implicit scheme, the contact constraints are enforced at every single grid-point. It does not require inter-mode constraints or prior knowledge of mode sequences, which makes the TO problem more computationally tractable.

Remarkable work has been done under the framework of CIO. Inspired by Stewart and Trinkle's time-stepping scheme [4], Pose et al. [5] first propose an algorithm posing the optimization problem as a Mathematical Program with Complementarity Constraints (MPCC). Mordatch et al. [6, 7, 8] add contact forces into the vector of decision variables and solve it indirectly through the optimization problem. Neunert et al. [9] implement a NMPC algorithm that is based on CIO and soft-contact model. Dai et al. [10] exploits centroidal dynamics in whole-body planning, applies complementarity contact model to motion planner and implements tasks like climbing a monkey bar on a humanoid robot. Sleiman et al. [3] reformulates the CIO approach by assuming a hard-contact model and

unilateral constraints. The two TO approaches are actually derived from two mainstream numerical simulation schemes: event-driven approach and time-stepping approach, respectively. The two schemes aim to deal with non-smooth discontinuous dynamical system. Event-driven approach keeps detecting the key events (discontinuity points) in the ODE. Once an event is detected, the integration is re-initialized. Integrability is thus guaranteed in each part segmented by event points. In the case of multiple discrete events, it is difficult to accurately capture them. Time-stepping approach, on the other hand, directly discretizes the non-smooth system dynamics at the cost of integration accuracy. Computationally heavy tasks are more tractable with this scheme, since event detection is skipped and the discretization is unrelated to discontinuous points.

Our nonlinear MPC framework based on CIO is generally transcribed into an optimization problem with finite dimensions and subsequently solved by offline or online approaches. Popular offline approaches are on the basis of direct methods like single shooting, multiple shooting and direct collocation. The multiple shooting method segments the original problem into several shorter simulations by discretizing both the inputs and the states. The direct collocation method shares the same discretization strategy while an additional collocation constraint is included. The direct methods above transcribe the infinite-dimension problem into a finite-dimension nonlinear program and aid the solver in localizing the global optimum efficiently. Time and memory consumption is huge in receiving measurements, discretizing the dynamics and consequently resolving the finite-dimensional problem. On the other hand, its computational complexity explodes with state-space expands. Therefore it normally cannot solve the problem in real-time with some exceptions like [11].

Unified locomotion MPC-scheme has been nonetheless proven successful in real-time that takes advantages of the fast optimization methods, like Gauss-Newton Hessian approximation and Riccati backward sweep. The MPC-based real-time planning schemes, such as iLQR/G (a DDP-based method[12, 13, 14]) and SLQ [15], are also adopted to formulate the multiple-contact problems [16, 17].

In this work, we present a CIO scheme for mobile robots to implement dynamic multi-contact multi-object manipulation tasks in an SLQ-MPC fashion. Our main contributions include:

- We propose an object-augmented legged robot system capable of both dynamic multi-object manipulation and mobile locomotion. With a couple of manipulation contact points encoded besides the on-board arm's end-effector, the robot could deal with intricate tasks involving several manipulated-objects. We separately consider the dynamics of each manipulated-object by introducing virtual contact points and virtual forces. The virtual forces aids a leg in dually functioning in gait generation and object manipulation. The mobile locomotion is fulfilled by the pre-definition of multi-phase gait sequences in the switched system.
- We present a reformulation of CIO - a relaxed version of non-smooth complementarity contact model, and prove its advantages in autonomous contact scheme decisions over its counterpart. The formulation facilitates the solver's convergence and mostly preserves the physical fidelity by penalizing the relaxation terms. The complementarity constraints are applied to each (virtual) contact point during the generalisation of multi-contact manipulation.
- We manage to resolve multi-object contact-rich manipulation problems in simulation. A wide variety of involved tasks could be tackled concerning shift of contact points and recapture of recoiling objects. We estimate object states in open-loop based on accurate system modeling of manipulated-objects in the whole body controller. MPC could be solved in real-time on the legged robot's on-board computer in a receding-horizon fashion.

Chapter 2

Problem Formulation

In this chapter, we formulate the OC model starting from a simple single-contact single-object problem in 2.1. Afterwards in 2.2, the contact model of CIO is discussed at length. With the purpose of adapting the model in 2.1 for multi-contact multi-object manipulation, we furthermore augment it in 2.3 by introducing virtual contact points and virtual forces. The full control architecture is briefly presented in 2.4.

2.1 Model Formulation

Let us first consider the single-contact single-object tasks, where the arm's end-effector is the only contact point regarding the manipulated object.

After transcribing from the initial infinite-dimensional OC problem, the discrete nonlinear OC problem we concern about can be formulated as

$$\begin{aligned} \min_{\boldsymbol{u}[\cdot]} \Phi_f(\boldsymbol{x}[n_f]) + \sum_{n=0}^{n_f-1} \Phi(\boldsymbol{x}[n], \boldsymbol{u}[n], n) \\ \text{s.t. } \left\{ \begin{array}{l} \dot{\boldsymbol{x}}[n] = \boldsymbol{f}(\boldsymbol{x}[n], \boldsymbol{u}[n], n) \\ \boldsymbol{x}[0] = \boldsymbol{x}_0 \\ \boldsymbol{g}_1(\boldsymbol{x}[n], \boldsymbol{u}[n], n) = 0 \\ \boldsymbol{g}_2(\boldsymbol{x}[n], n) = 0 \\ \boldsymbol{h}(\boldsymbol{x}[n], \boldsymbol{u}[n], n) \geq 0 \end{array} \right. \end{aligned} \quad (2.1)$$

For the sake of simplification, the time dependencies would be omitted in the following expressions. The OC problem is an NLP subject to system dynamics, the initial state, and multiple constraints.

The state vector \boldsymbol{x} comprises robot's state $\boldsymbol{x}_r = (\boldsymbol{h}_{\text{com}}, \boldsymbol{q}_b, \boldsymbol{q}_j) \in \mathbb{R}^{12+n_a}$ and manipulated-object state \boldsymbol{x}_o .

$$\boldsymbol{x} = (\boldsymbol{h}_{\text{com}}, \boldsymbol{q}_b, \boldsymbol{q}_j, \boldsymbol{x}_o) \in \mathbb{R}^{12+n_a+2n_o} \quad (2.2)$$

For single-contact manipulation problems, the input vector \boldsymbol{u} comprises gait-related contact forces \boldsymbol{f}_{g_i} , manipulation-related contact forces \boldsymbol{f}_{m_i} , generalized velocity \boldsymbol{v}_j and penalty terms of equality constraints \boldsymbol{p} . Contact torques $\boldsymbol{\tau}_i$ are not included in the input space by ideally considering points of contact only instead of patches.

$$\boldsymbol{u} = (\boldsymbol{f}_{g_1}, \dots, \boldsymbol{f}_{g_{n_c}}, \boldsymbol{f}_{m_1}, \dots, \boldsymbol{f}_{m_{n_c}}, \boldsymbol{v}_j, \boldsymbol{p}) \in \mathbb{R}^{3n_c+n_a+n_p} \quad (2.3)$$

where $n_c = n_g + n_{cio}$ is the number of total contact points; n_g, n_{cio} are the numbers of gait-related and manipulation-related contact points, respectively; $n_p = 2n_{cio}$ means the

number of penalty terms. For the simple arm's end-effector manipulation task, $n_g = 4$, $n_{cio} = 1$ and $n_c = 5$.

2.1.1 Generic Cost Function

The generic cost function is expressed in a quadratic reference tracking fashion.

$$\begin{aligned}\Phi(\mathbf{x}, \mathbf{u}, t) = & \alpha_1 \cdot \left(\|\mathbf{r}_{IE} - \mathbf{r}_{IE}^*\|_{\mathbf{Q}_{ee_p}}^2 + \|\zeta_{IE}\|_{\mathbf{Q}_{ee_o}}^2 \right) + \\ & \alpha_2 \cdot \|\mathbf{x}_r - \mathbf{x}_r^*\|_{\mathbf{Q}_r}^2 + \alpha_3 \cdot \|\mathbf{x}_o - \mathbf{x}_o^*\|_{\mathbf{Q}_o}^2 + \|\mathbf{u} - \mathbf{u}^*\|_{\mathbf{R}}^2\end{aligned}\quad (2.4)$$

All the weight matrices above are positive definite. Symbols with a superscript * indicate the tracking references of the corresponding values.

The upcoming task descriptions were encoded in this single cost function, in which we could switch modes by specifying the values of $\alpha_1, \alpha_2, \alpha_3$ from $\{0, 1\}$. We could customize the mode in terms of specific tasks, like end-effector tracking ($\alpha_1 = 1, \alpha_2 = 1, \alpha_3 = 0$), base tracking ($\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 0$) and object manipulation ($\alpha_1 = 0, \alpha_2 = 1, \alpha_3 = 1$).

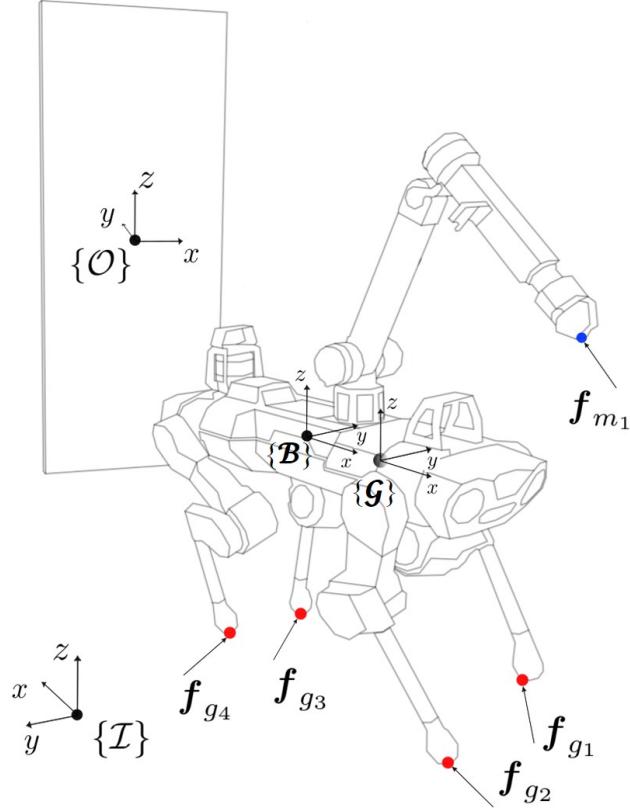


Figure 2.1: Illustration of the object-augmented multi-limbed floating-base system, describing the case of single-arm contact manipulating a door. The gait-related forces are applied on points represented by red dots, while the manipulation-related forces are with blue dots.

2.1.2 System Dynamics

In our framework, the system dynamics capture two main aspects: robot equation of motion and manipulated-object dynamics. Firstly, we express the dynamics in terms of the robot's centroidal momentum

$$\dot{\mathbf{h}}_{com} = \begin{bmatrix} mg + \sum_{i=1}^{n_g} \mathbf{f}_{g_i} + \sum_{j=1}^{n_{cio}} \mathbf{f}_{m_j} \\ \boldsymbol{\tau}_i + \sum_{i=1}^{n_g} \mathbf{r}_{com,i} \times \mathbf{f}_{g_i} + \sum_{j=1}^{n_{cio}} \mathbf{r}_{com,j} \times \mathbf{f}_{m_j} \end{bmatrix} \quad (2.5)$$

where $\mathbf{r}_{com,i}$ ($\mathbf{r}_{com,j}$) represents the position of contact point i (j) with respect to the center of mass. The Newton-Euler equation indicates that the linear and angular momentum rates are affected by external forces and torques.

Secondly, the robot's full kinematics is comprised to describe the influence of joint velocities on the base pose

$$\dot{\mathbf{q}}_b = \mathbf{A}_b^{-1} (\mathbf{h}_{com} - \mathbf{A}_j \dot{\mathbf{q}}_j) \quad (2.6)$$

$$\dot{\mathbf{q}}_j = \mathbf{v}_j \quad (2.7)$$

where \mathbf{A}_b , \mathbf{A}_j denotes the CMM of base and joint, respectively, mappings from the generalized coordinates' rate of change to the centroidal momentum.

Thirdly, the planner is aware of the manipulated object dynamics

$$\dot{\mathbf{x}}_o = \begin{bmatrix} \mathbf{v}_o \\ \mathbf{M}_o^{-1} (-\mathbf{J}_{c_o}^T \mathbf{f}_{m_1} - \mathbf{b}_o) \end{bmatrix} \quad (2.8)$$

where \mathbf{f}_{m_1} represents the arm's end-effector contact force. In manipulation tasks, the dynamics-coupling of robot and object is essential in that it makes the object desired state more attainable. In order to fully capture the manipulated object dynamics, we assume the properties of the object are recorded in advance, like weight, size, dynamical coefficient, etc. It is also required to precisely estimate the object's state, which will be fed back to the MPC solver.

2.1.3 System Constraints

All of the constraints are defined at the level of the potential contact points. At the arm's end-effector, the complementarity constraints concerning task manipulation are deferred to section 2.2. The other constraints could be divided into two categories: gait-related constraints and manipulation-related constraints.

The gait-related constraints are coupled with gait sequences, based on the multi-phase approach. The gait sequences (i.e. stance, trot) are predefined through mode schedules, consisting of mode sequences and corresponding switching times. For each contact point, a mode sequence is a series of contact statuses (open or closed) aligned in order within a period of time. Given a fixed mode instance \mathcal{C}_j that starts at time t_j^0 and ends at t_j^f , the gait-related constraints corresponding to this mode can be established as the following:
 $\forall t \in [t_j^0, t_j^f] \text{ and } \forall i \in \{1, \dots, n_g\}$

$$\left\{ \begin{array}{ll} \mathbf{v}_i = \mathbf{0} & \text{if } c_i \in \mathcal{C}_j \\ \mathbf{v}_i \cdot \hat{\mathbf{n}} = v^*(t) & \text{if } c_i \in \bar{\mathcal{C}}_j \\ \mathbf{f}_{g_i} = \mathbf{0} & \text{if } c_i \in \bar{\mathcal{C}}_j \\ \mu_s f_{g_i,z} - \sqrt{f_{g_i,x}^2 + f_{g_i,y}^2 + \epsilon^2} \geq 0 & \text{if } c_i \in \mathcal{C}_j \end{array} \right. \quad (2.9)$$

where c_i denotes the i 'th contact point, \mathbf{v}_i is the absolute linear velocity of contact point c_i in the inertia frame, $\hat{\mathbf{n}}$ denotes the surface normal vector, $v^*(t)$ is a reference trajectory normal to the surface. At open contacts, the contact forces disappear and the foot end-effectors follow a trajectory normal to the ground. At closed contacts, the feet do not slip or detach with respect to the ground, and the feet' contact forces are confined within the friction cone. ϵ is a non-zero constant introduced to smoothen the constraint and avoid the saddle point at the origin[18].

The threefold manipulation-related constraints are irrelevant to mode sequences and formulated in a neater form

$$\begin{cases} -\mathbf{v}_{j_{max}} \leq \mathbf{v}_j^{arm} \leq \mathbf{v}_{j_{max}} \\ -\boldsymbol{\tau}_{\max} \leq \mathbf{g} + \mathbf{J}_{m_1}^T \mathbf{f}_{m_1} \leq \boldsymbol{\tau}_{\max} \\ \lambda_1 \phi_1 \geq 0 \end{cases} \quad (2.10)$$

where \mathbf{J}_{m_1} is the matrix of contact Jacobian for the arm's end-effector, λ_1 and ϕ_1 are the end-effector normal force and gap distance (detailed definition in Fig. 2.2), respectively, expressed in $\{\mathcal{O}\}$. The constraints guarantee that the arm's joint velocity and torque limits are not violated and that in the door pushing task, the arm end-effector remains inside the door frame when making contact.

2.2 Contact-implicit Approach

2.2.1 A Variant of Complementarity Model

Contact model is crucial in manipulation tasks. The model we use is the contact-implicit approach, which could be further categorized into the soft contact model and hard contact model. In the soft contact model, hard-contact surfaces are approximated using a damping spring model. Specifically speaking, the contact forces are explicitly expressed as a function of the distance between the contact point and the object surface. We adopt the non-smooth hard contact model instead, also known as the linear complementarity model. The model contains more physical fidelity compared with the soft contact model, as it does not penetrate or generate forces at a distance. We could devise the original form and perturbed form of the complementarity contact model as follows: $\forall k \in \{1, \dots, n_{cio}\}$

$$\begin{cases} \phi_k \geq 0 \\ \lambda_k \geq 0 \\ \phi_k \cdot \lambda_k = 0 \\ \dot{\phi}_k \cdot \lambda_k = 0 \end{cases} \longrightarrow \begin{cases} \phi_k \geq 0 \\ \lambda_k \geq 0 \\ \phi_k \cdot \lambda_k = p_{k_1} \\ \dot{\phi}_k \cdot \lambda_k = p_{k_2} \end{cases} \quad (2.11)$$

where $n_{cio} = n_{mp}n_{obj}$, implying the number of complementarity constraint sets is related to both the numbers of manipulation contact points and manipulated objects. The constraints have the following indications: the contact point only applies pushing forces; the contact point does not penetrate object surfaces; the contact modes can be switched between open and closed contacts and there is no impulsive acceleration in contact. Instead of the original form on the left of (2.11), we introduce slack variables p_{k_1}, p_{k_2} and augment them as penalty terms (\mathbf{p}) in the input space. The slack variables are dynamic offsets that make the constraints less strict and more tractable in the solver. They aid the solver in efficiently converging to the local minimum.

2.2.2 Complementary Comments

The contact force and gap distance utilized in (2.11) are expressed in $\{\mathcal{O}_i\}$. The normal contact force $\lambda_k(\mathbf{u})$ could be readily resolved by

$$\lambda_k(\mathbf{u}) = [\ 0 \ 1 \ 0 \] \cdot (R_{\mathcal{I}\mathcal{O}_i})^T \cdot \mathbf{f}_k \quad (2.12)$$

where $R_{\mathcal{I}\mathcal{O}_i}$ represents the rotation matrix from $\{\mathcal{I}\}$ to $\{\mathcal{O}_i\}$. Nonetheless the formulation of generalized gap distance $\phi_k(\mathbf{x})$ varies in accordance with the properties of specific manipulated objects. In terms of the shape of a manipulated object, we categorize it into two types, geometric objects and point-mass objects. The geometric objects feature planar contact surfaces, like doors, payloads, etc. The gap distance with respect to a geometric object is defined as the distance between the contact point and the object's surface

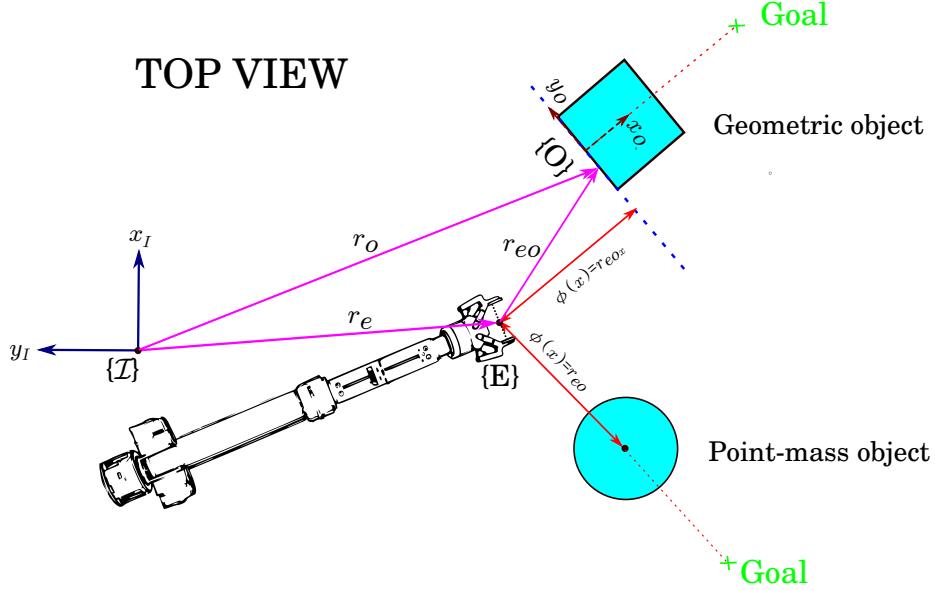


Figure 2.2: Illustration of generalized gap distance function with respect to a geometric object and a point-mass object.

$$\phi_k(\mathbf{x}) = [\begin{array}{ccc} 0 & 1 & 0 \end{array}] \cdot (R_{\mathcal{IO}_i})^T \cdot \tau r_{eo} \quad (2.13)$$

where τr_{eo} is the vector connecting end-effector and any point on the manipulated-object surface expressed in $\{\mathcal{I}\}$.

In the case of manipulation of a point-mass object, like the ball throwing task, the formulation is simplified as Euclidean distance between the contact point and CoM of the object

$$\phi_k(\mathbf{x}) = \|_O r_{eo} \|_2 \quad (2.14)$$

It is worth noting that no-penetration and no-pulling force constraints are no more required when manipulating a point-mass object, since $\phi(\mathbf{x})$ remains positive and pulling forces are permitted.

2.3 Model Augmentation

Our ultimate target is to perform multi-object manipulation using multiple contact points. In this section, we reformulate and augment the original model step by step.

2.3.1 Single-object Multi-contact Manipulation

In terms of multi-contact manipulation, elbow contact is an intuitive one we could think of besides the arm end-effector. To achieve an elbow and end-effector dual-contact task, the input space will be augmented with the elbow contact force, with $n_{mp} = 2$ and $n_c = 6$.

Then we expect one of the feet to act as a second manipulator, like the left-front foot. To this end, there will be two categories of forces applied on the foot end-effector, gait-related force and manipulation-related force, with $n_{mp} = 3$ and $n_c = 7$. The gait-related force represents the interaction with the ground in the switched system, while the manipulation-related force captures the contact with the manipulated object. We find it necessary to

express the two forces separately in the input space, even though they are both applied on the same contact point. Hence we regard there is a virtual contact point, with the manipulation-related force applied on it, which coincides exactly with the original foot end-effector contact point.

Generally speaking, n_{mp} implies the number of manipulation-related contact points, no matter a virtual point or an utterly new one. The manipulation-related forces affect the base centroidal momentum rates as described in (2.5). The object dynamics is reformulated as follows

$$\dot{\mathbf{x}}_o = \left[\begin{array}{c} \mathbf{v}_o \\ M_o^{-1} \left(- \sum_{i=1}^{n_{mp}} J_{o_i}^T(\mathcal{O}\mathbf{r}_{c_i}) \mathbf{f}_{m_i} - \mathbf{b}_o \right) \end{array} \right] \quad (2.15)$$

where \mathbf{f}_{m_i} indicates the manipulation-related force; $J_{o_i}(\mathcal{O}\mathbf{r}_{c_i})$ is the i 'th contact Jacobian of the object as a function of the position of the i 'th contact point in the object frame $\mathcal{O}\mathbf{r}_{c_i}$. For instance, in the door manipulation task the Jacobians are related to the distance between the contact point and the door hinge.

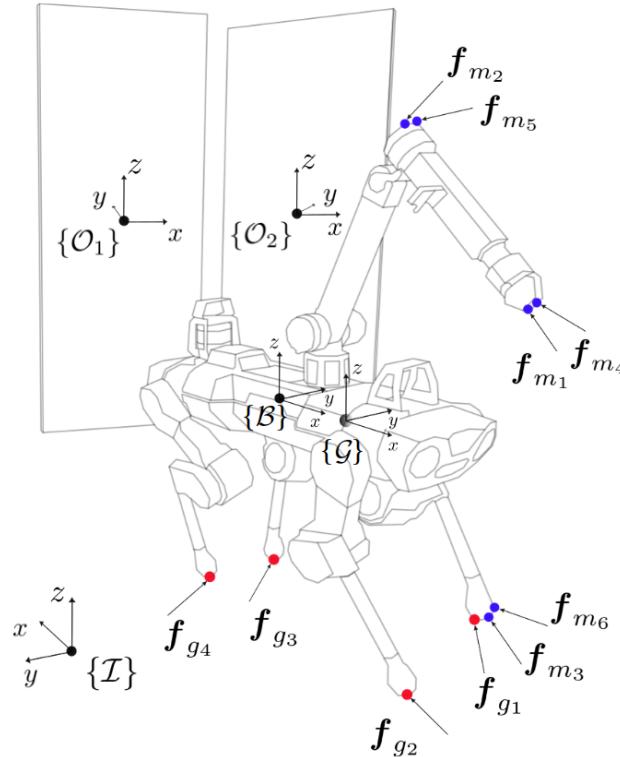


Figure 2.3: Illustration of the object-augmented multi-limbed floating-base system in paw-up status (gait mode of left-front leg lifting), depicting the arm end-effector, elbow and foot end-effector multiple contacts case with two manipulated doors. The gait-related forces are applied on points represented by red dots, while the manipulation-related forces are with blue dots. Virtual contact points might exist if several forces applying on the same contact point.

2.3.2 Multi-object Single-contact Manipulation

We naturally think of introducing several manipulated objects instead of a single one from the perspective of inductive reasoning, with $n_{obj} > 1$ and $n_{mp} = 1$. A typical instance is the task of double-door pushing towards desired angles utilizing arm's end-effector only, where $n_{obj} = 2$ and $n_{mp} = 1$.

The state and input spaces could be readily augmented in accordance with (2.2) and (2.3). It is worth being noted that

$$n_o = 2 \sum_{j=1}^{n_{obj}} n_{dof,j} \quad (2.16)$$

where $n_{dof,j}$ indicates the number of DoFs of the j 'th object. In the case of double-door manipulation, the objects' state-space comprises the angle and angular velocity of each door, with $n_o = 4$.

$\forall j \in \{1, \dots, n_{obj}\}$, the j 'th object's flow-map could be set up as follows

$$\dot{\mathbf{x}}_{o,j} = \begin{bmatrix} \mathbf{v}_{o,j} \\ \mathbf{M}_{o,j}^{-1} \left(-\mathbf{J}_{o,j}^T (\mathcal{O}_j \mathbf{r}_{c_j}) \mathbf{f}_{m_j} - \mathbf{b}_{o,j} \right) \end{bmatrix} \quad (2.17)$$

The merits of virtual forces and contact points primarily lie in the convenient composition of the dynamics of the robot's centroidal momentum and each manipulated-object. The dynamics of the j 'th object is solely affected by the corresponding manipulation-related force \mathbf{f}_{m_j} , when the arm's end-effector touches the j 'th object. All manipulation-related forces \mathbf{f}_m impact the CoM momentum rates consistent with (2.5).

2.3.3 Multi-object Multi-contact Manipulation

With multiple contact points, the robot is capable of manipulating several objects simultaneously, with $n_{obj} > 1$ and $n_{mp} > 1$.

$\forall i \in \{1, \dots, n_{mp}\}$ and $\forall j \in \{1, \dots, n_{obj}\}$, the j 'th object's flow-map could be set up as follows

$$\dot{\mathbf{x}}_{o,j} = \begin{bmatrix} \mathbf{v}_{o,j} \\ \mathbf{M}_{o,j}^{-1} \left(-\sum_{i=1}^{n_{cio}} \mathbf{J}_{o_i,j}^T (\mathcal{O}_j \mathbf{r}_{c_{i,j}}) \mathbf{f}_{m_{l(i,j)}} - \mathbf{b}_{o,j} \right) \end{bmatrix} \quad (2.18)$$

where $l(i, j) = (j-1)n_{mp} + i$. Assuming the i 'th manipulation contact point makes contact with the j 'th object, the corresponding manipulation-related force $\mathbf{f}_{m_{l(i,j)}}$ turns non-zero. $\mathbf{f}_{m_{l(i,j)}}$ acts on the CoM momentum rates in line with (2.5) as well as the dynamics of the j 'th object.

$\forall k \in \{1, \dots, n_{cio}\}$, there is a unique set of complementarity constraints as is formulated in (2.11). Manifold complementarity constraints facilitate the decoupling of contact forces in the switched system and manipulation, despite the fact that computational complexity of the OC problem increases owing to expanded input and state spaces.

We illustrate the case in Figure 2.3 with an instance of dual-door pushing using multiple contacts, where $n_{obj} = 2$, $n_{mp} = 3$ and consequently $n_{cio} = 6$. Note that manipulation-related contact forces and a gait-related contact force are simultaneously enforced on the left-front foot EE. Thus we decouple the CIO and multi-phase patterns existing in the foot contact. In other words, the contact forces on the foot's EE involve mixed strategies of self-optimization (CIO) and pre-definition (Multi-phase Approach).

Consequently, we generalize the application of CIO approach and establish a multi-object multi-contact augmented model for the articulated floating base system.

2.4 Full Control Architecture

The full control architecture of the quadruped robot is composed of a high-level controller, state estimator, low-level controller and joint controller, as depicted in Fig. 2.4. Specifically, the high-level control module consists of a whole-body planner and a whole-body controller.

The model formulation and augmentation of the whole-body planner are discussed in depth in sections 2.1 and 2.3. The optimal policy $\mathbf{u}[\cdot]$ of the NLP is solved utilizing a variant of continuous-time SLQ algorithm[16, 17, 12], which is widely used for OC problems of the switched system (hybrid system). The variant absorbs a previously implemented projection technique and augmented-Lagrangian approach[19]. The whole-body planner outputs the optimal state and input reference trajectories, which are subsequently forwarded to WBC with motion and force tracking controller units.

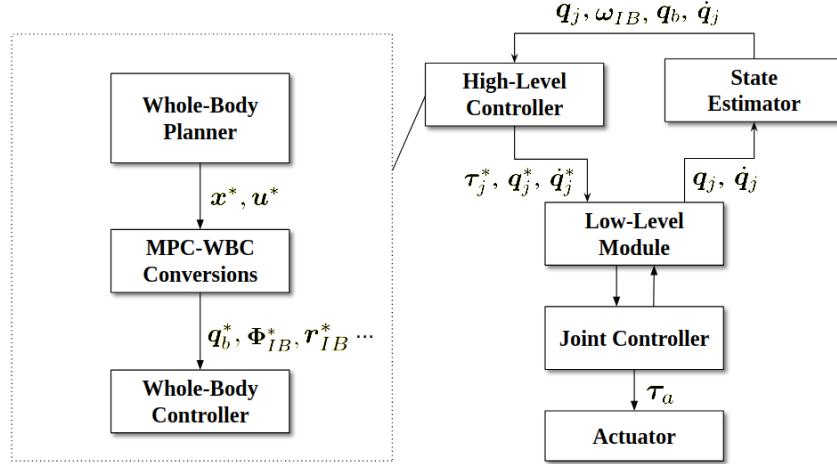


Figure 2.4: A schematic diagram illustrating the whole control framework. The High-Level Controller is described in detail on the left.

We implement the optimal state reference x^* and input reference u^* trajectory tracking by WBC that is on the basis of HQP [20, 21]. HQP fulfills a series of weighted tasks in a prioritized order. There is a conversion procedure within WBC module, where x^* and u^* are transformed into the WBC tracking references (with detailed discussion in [22]).

In WBC, we adopt an open-loop strategy to estimate the manipulated objects' state. To be specific, we estimate the object state based on the system modeling without measurement units. It's required to have knowledge of the mass, size, dynamical coefficients of all manipulated objects in advance. With accurate system modeling, the open-loop state estimation could be adequately efficient.

Optimal generalized accelerations q_j^* and optimal joint torques τ_j^* , outputted from the high-level controller, are directly obtained through inverse dynamics. It is important to note that the optimal arm contact force from the planner is directly tracked as a reference trajectory since the object dynamics are not integrated into WBC.

Both the whole body planner and controller obtain their feedback from a state estimator that fuses encoder readings and IMU measurements. In the low-level module, the control law on the joint level is derived as follows:

$$\tau_j = \tau_j^* + \mathbf{K}_p(q_j^* - q_j) + \mathbf{K}_d(\dot{q}_j^* - \dot{q}_j) \quad (2.19)$$

Chapter 3

Results

In this chapter, we implement various simulation experiments in whole-body planner visualization as well as in Gazebo. Firstly we implement various manipulation tasks from single-contact (3.1), multi-contact (3.2) to multi-contact multi-object (3.3). Secondly, we verify the physical fidelity of the complementarity contact model as well as the advantages of it over the multi-phase method in 3.4. Lastly, we substantiate the feasibility of an open-loop object-state estimation strategy in 3.5.

For all simulations, with time horizon $T = 1s$, the MPC loop computes trajectories at an update rate of $60Hz$.

3.1 Single Contact Manipulation

For single contact manipulation, we perform experiments in the Gazebo physics engine on the task of pushing a resistive door. The door is modeled as a rotational mass-stiffness-damper system, with constant stiffness and damping coefficients by assuming linear dynamics. The manipulated-object state, which is continuously estimated in an open-loop fashion, is transmitted to the whole-body planner as a state observation. Besides the arm's EE contact (Fig. 3.1 (a)), we add manifold contact points by imposing complementarity constraints on the left forefoot (Fig. 3.1 (b)) and elbow (Fig. 3.1 (c)).

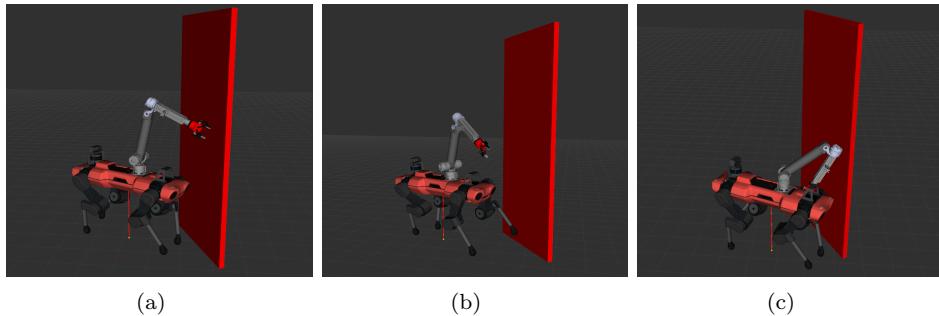


Figure 3.1: The robot dynamically pushes a door with different contact points in Gazebo simulation: (a) arm end-effector; (b) left-front foot end-effector; (c) elbow.

Two door-pushing tasks are performed using arm's EE contact(Fig. 3.1(a)), with stiffness coefficients of 0 and $5N/rad$, respectively. The results of the estimated door angles and the optimal force trajectories are presented in Fig. 3.2 for a simple arm end-effector pushing task in which the impact-events are directly perceived. In the zero stiffness case, a single impulsive impact ($15N$) with a duration less than $1sec$ is imposed by the arm. In the

non-zero stiffness case, multiple impulsive impacts nevertheless are generated with larger magnitudes for the sake of overcoming the stiffness of the door. The behavior also indicates the richness of the CIO contact scheme, since it leads to TO strategies that are difficult to be pre-determined in the multi-phase framework.

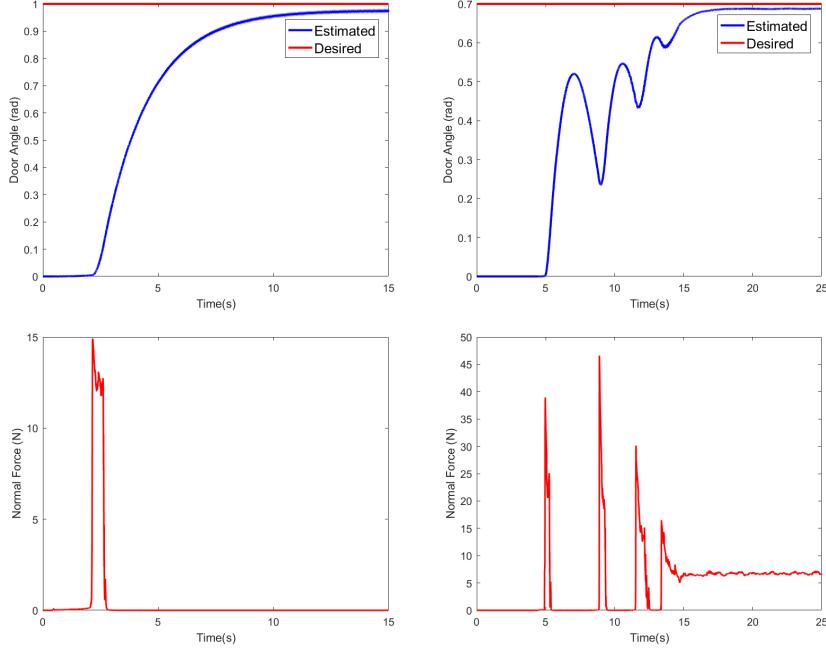


Figure 3.2: Normal contact force applied to the door and estimated door angle plots. (left) stiffness = 0; (right) stiffness = 5N/rad

In both two scenarios, the actual final door angle closely approximates the desired one. Moreover, we could diminish steady-state error simply by increasing the weight of the tracking object's state and reducing the penalty on the arm contact forces in the cost function (2.4).

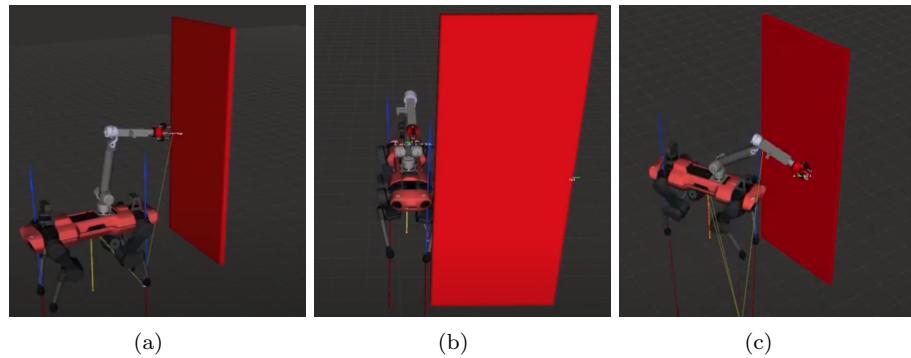


Figure 3.3: Snapshots of door-pushing tasks in Gazebo with different initial poses: (a) robot's base pose in $x - axis$ of $\{\mathcal{I}\}$; (b) base pose in $y - axis$ of $\{\mathcal{I}\}$; (c) base orientation.

To further test the versatility and extremes of the planner, we design door-pushing tasks with different initial poses: (1) the base position farther away from the door; (2) the arm' EE outside the door frame; (3) a higher yaw angle w.r.t. the manipulated-object. In all scenarios, the robot manages to trot forward, move towards the inside of the door frame

or tune its base orientation followed by pushing the door.

On the other side, we testify that the robot could dynamically interact with a manipulated object utilizing foot in the paw-up status, or elbow in the arm-folded circumstance. Impulsive impacts are observed similar to the arm's EE case and thus we do not repeatedly analyze the results.

3.2 Multiple Contacts Manipulation

After introducing virtual forces in the input space and additional contact points, we are able to implement a door-pushing task involving multiple contacts and contact switches. The results of two sets of simulation experiments are illustrated in Fig.3.4, where we activate two combinations of contact points.

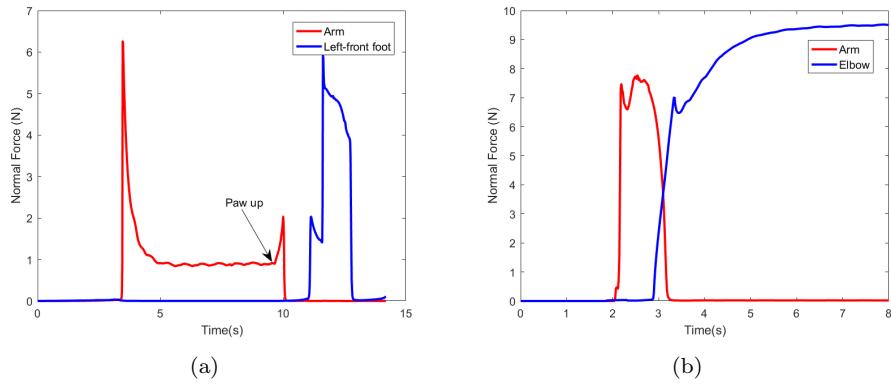


Figure 3.4: Plots of the normal forces to the manipulated door in the door-pushing cases with multiple contact points: arm EE and left-front foot EE (a); arm's EE and elbow (b);

In the instance concerning EEs of the arm and left forefoot, shown in Fig.3.4 (a) and Fig.3.5, the arm's EE impulsively pushes the door open initially, and continuously interacts with it to prevent it from recoiling (Fig.3.5(a)). The shift of contacts occurs after we switch the gait mode from trot to paw-up, as is recorded in Fig.3.5(b). Since the planner believes foot manipulation is a cost-efficient way compared with arm manipulation, the arm makes no more contact with the door, and instead the foot kicks it open with an impulsive force (Fig.3.5(c)).

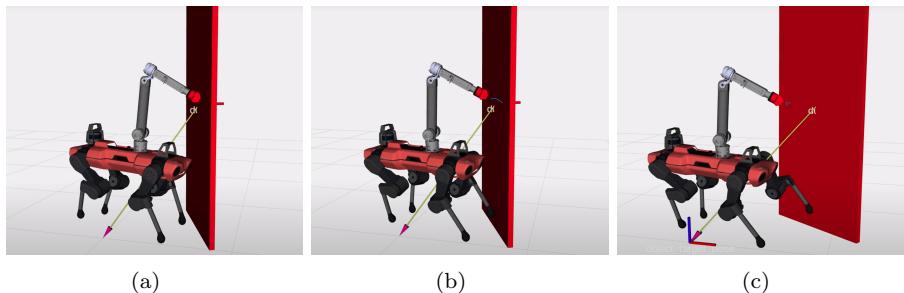


Figure 3.5: Snapshots of multi-contact door-pushing tasks in Gazebo: (a) Arm EE contact in trot gait; (b) Gait switch to paw-up accompanied by contact switching to left-fore foot; (c) Foot contact in paw-up gait.

In the arm's EE and elbow contacts case, the initial configuration of the arm's joints is set

such that the forearm is roughly parallel to the object's surface. As Fig.3.3 (b) depicts, the arm's EE opens up the door and subsequently, at 2.91sec , the contact point switcher to the elbow. Afterward, the elbow constantly supports the door at the desired final object-state.

The two cases above device contact-point switches either motivated by the gait change (a) or spontaneously (b). One of the reasons behind of contact switch primarily lies in the CIO frameworks determines contact schemes merely premised on the minimization of the total cost. If a secondary contact point is a lower-cost way in manipulation, the switch naturally happens. The optimized contact schedules in CIO are yielded without pre-defining contact sequences. The complexity of contact schemes rises dramatically as more contact points introduced. The multi-phase approach, therefore, is not an appropriate solution anymore in this multi-contact circumstance.

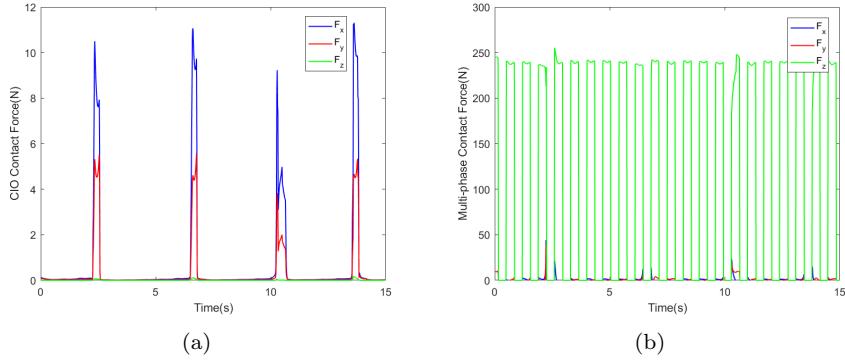


Figure 3.6: Plots showing CIO manipulation-related contact forces (a) and multi-phase gait-related contact forces (b) of the left-front foot in the foot's EE manipulation while trotting.

Moreover, with additional virtual end-effector contact forces introduced, the planner is able to decouple the forces generated through multi-phase and CIO contact schemes at the foot EE. We visualize this decoupling effect in a task of pushing the door with left-fore foot EE while constantly trotting, as illustrated in Fig.3.6 and visualized in Fig.3.7 (a)(b)(c).

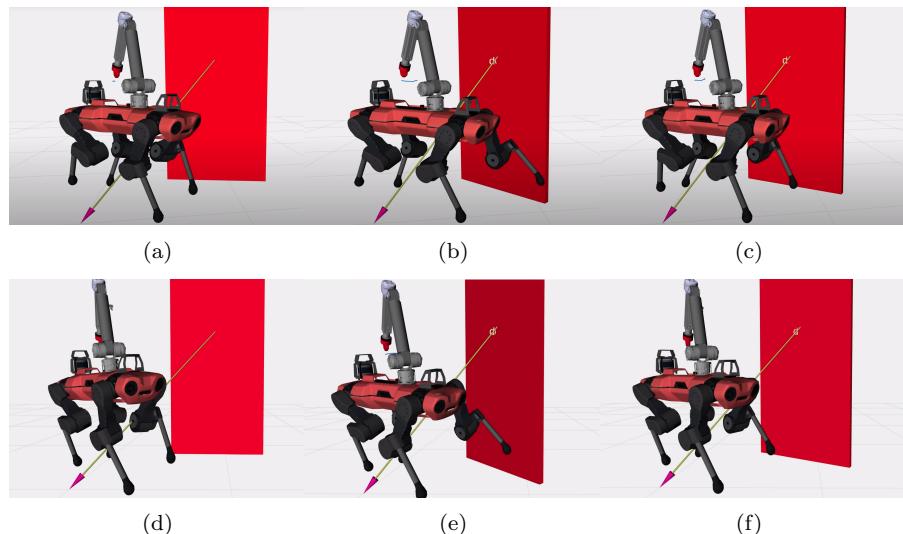


Figure 3.7: Snapshots of door-pushing utilizing left fore foot EE in the gaits of trot (upper) as well as paw-up (lower).

The initially-trotting (a) leg impulsively shoves the door open (b) and then recovers its normal status of trot (c). As the door recoils, the leg seeks for a break while in the air, gradually reaches the door and nudges it once more. The manipulation-related and gait-related contact forces are resolved utilizing the CIO and multi-phase schemes, respectively.

On the other side, we command the robot to manipulate a door with stiffness in paw-up gait, as illustrated in Fig.3.7 (d)(e)(f). The robot retrieves its left foreleg to maintain stability after manipulating the door with it.

3.3 Multiple Objects Manipulation

After simple tasks involving a single manipulated-object, we further investigate the results of generalization to multiple contact points on multiple objects, with the formulation given in (2.18).

Fig. 3.8 showcases the multiple object manipulation task of an ordinary daily occasion: double-door pushing. With the purpose of efficiently manipulating two doors, multi-contacts using both arm and foot would be a good choice. We augment the left-fore foot EE and arm EE with virtual forces in terms of the second door.

As the robot is placed in stance initially on the side of Door 2 in Fig.3.8(a), it nudges Door 1 utilizing the arm manipulator with a higher level of extension and flexibility. Door 1 is shoved open (b) subsequently by the foot manipulator, which is instinctive since the leg is in the vicinity of it compared with the arm. Consequently, the two doors reach their corresponding final states as illustrated in (c).

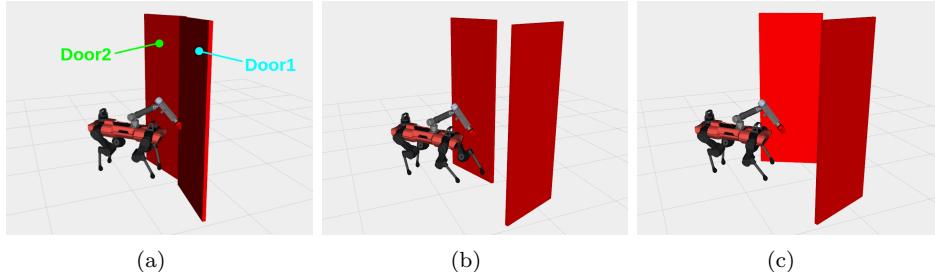


Figure 3.8: Snapshots of the robot dynamically pushing two doors towards respective target positions with arm and foot end-effectors

3.4 Contact Model

3.4.1 Complementarity Constraints

To present the principles of complementarity constraints, we perform a door-pushing task with an impulsive impact depicted in (b) of Fig.3.9. The figure keeps a record of the normal contact force λ_k , the EE-to-door gap distance ϕ_k as well as the slack value p_{k_1} in the mode-transition constraint. The plot indicates that λ_k and ϕ_k are always positive along the shown time horizon, confined by no pulling force constraint and no penetration constraint. On the other side, there is no contact force while the gap is non-zero and the contact mode is open (approaching or detached), and vice versa. Regarding the non-zero slack value, it mainly appears in the mode transition from open-contact (approaching mode) to closed-contact (contact mode), as is highlighted by an arrow in (a).

In conclusion, the perturbed form of complementarity model formulated in 2.2 preserves the physical fidelity of hard-contact and eases the solver in finding optimum by introducing slack in the constraints.

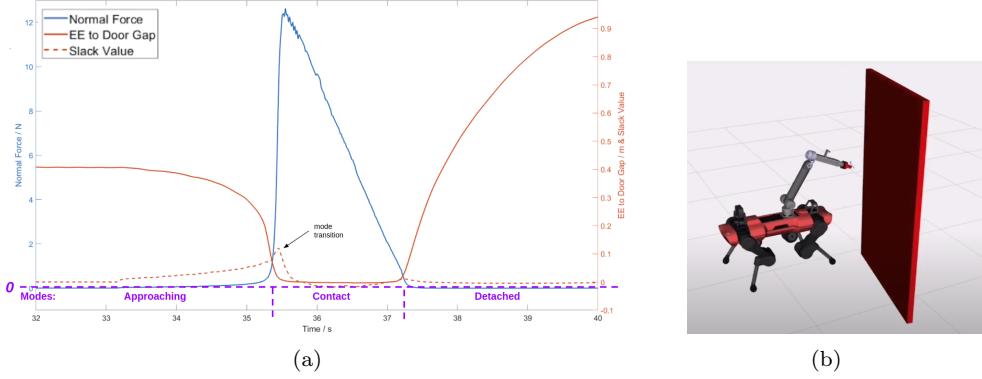


Figure 3.9: (a) Plot of normal contact force, EE-to-door gap distance and slack value in terms of time. (b) Snapshot of door-pushing task with an impulsive impact.

3.4.2 Advantages of CIO

Besides the benefits of CIO in 3.4.1, we choose it as a contact model owing to its advantage of autonomous decisions on contact time, location, and contact points.

To start with, we design a comparative experiment on load-throwing towards a desired location, so as to verify CIO outperforms its counterpart in contact time decisions. As is shown in Fig. 3.10, we are obliged to explicitly instruct the robot's gripper switch from closed to open contact at a certain time in the multi-phase scheme. If an insufficient switching time is defined, the final position of the load ends up far away from the target position (origin of $\{\mathcal{I}\}$ at the bottom-right corner of (a) (b) (c)). We have no choice but to manually adjust the value of switching time till we discover a satisfactory result. However, the CIO scheme implicitly calculates the optimal switching time (0.829s) as well as the optimal contact forces.

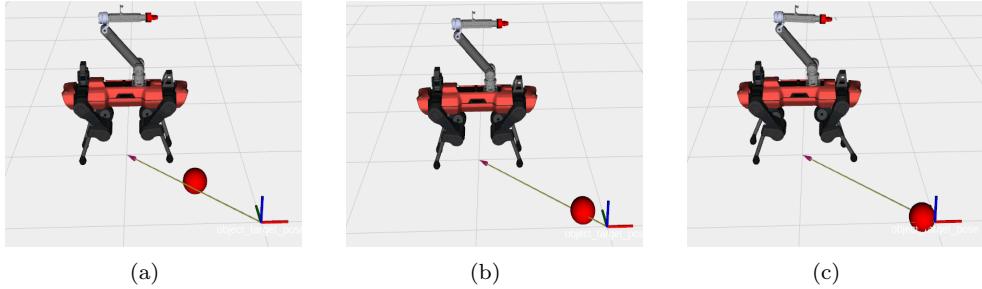


Figure 3.10: Snapshots of the final state of ball throwing task using multi-phase and CIO schemes. (a) Multi-phase scheme with switching time specified as 0.1s, (b) Multi-phase scheme with switching time specified as 0.7s, (c) CIO scheme with optimal switching time 0.829s

Secondly, autonomous contact location discovery is demonstrated by a test case of door-pushing with door-frame constraints. We supplement the constraint set with the door-frame constraint in (2.10). The constraint requires the robot to only make contact with the door within the door frames. If the projection of the arm's end-effector in the door plane lies initially outside the door frame, the planner generates trajectories forcing the EE to translate towards the door before the contact (Fig.3.11).

Thirdly, we present autonomous contact points decisions of the CIO in a door-pushing task with arm's EE and elbow contacts. In the multi-contact door-pushing task, it is intriguing to observe how the planner determines which contact point to adopt and lets it push the door. By adjusting the joint angle of the elbow, there is a combination of initial



Figure 3.11: Snapshots of arm’s EE approaching the door confined by the door-frame constraint, and eventually pushes it to a target angle.

poses of the manipulator. Based on the experimental results, we deduce that the closer contact point with respect to the door would manipulate the object (Fig.3.12 (a)(b)). If the two robot’s contact points are of the same distance from the door, they would act simultaneously (Fig.3.12 (c)).

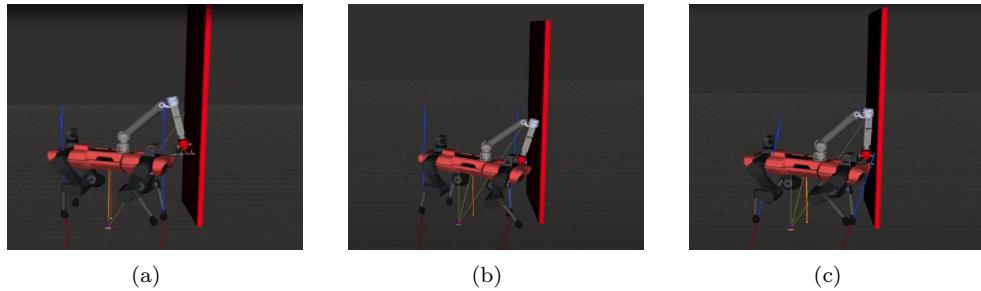


Figure 3.12: Snapshots of the door-pushing task in Gazebo utilizing (a) arm’s EE (b) elbow and (c) arm’s EE together with elbow.

3.5 Object-State Estimation

In this final section, we demonstrate the experiments of commanding the manipulated object to reach different desired final states in the door-pushing task. As briefed in 2.4, we intuitively implement an open-loop object-state estimation strategy on the basis of an accurate object-dynamical system modeling. We exhibit some of the results in Fig.3.13, with the red(blue) curve representing the planner estimated-state (the Gazebo real-state).

As shown in (a)(c), the desired final angles ($0.5rad$ and $1.5rad$) could be basically reached in both planner and Gazebo, with the estimation tracking the true-value within an offset within 15%. The reasons why there are offsets between the two curves lie in (1) inaccuracy of system modeling and (2) external disturbances. Since the door dynamical coefficients, mass, position and size in the planner do not perfectly describe the real ones, there would be discrepancies. On the other side, disturbances exist in most dynamic systems in reality. For instance, a slight nonlinear effect in the door dynamics could possibly affect the estimation in the long run. As we can imagine, the discrepancies would accumulate with time going on, like in the case of pushing the door multiple times. That is to say, the open-loop object-state estimation model is based on assumptions of accurate object-system modeling and zero external disturbances.

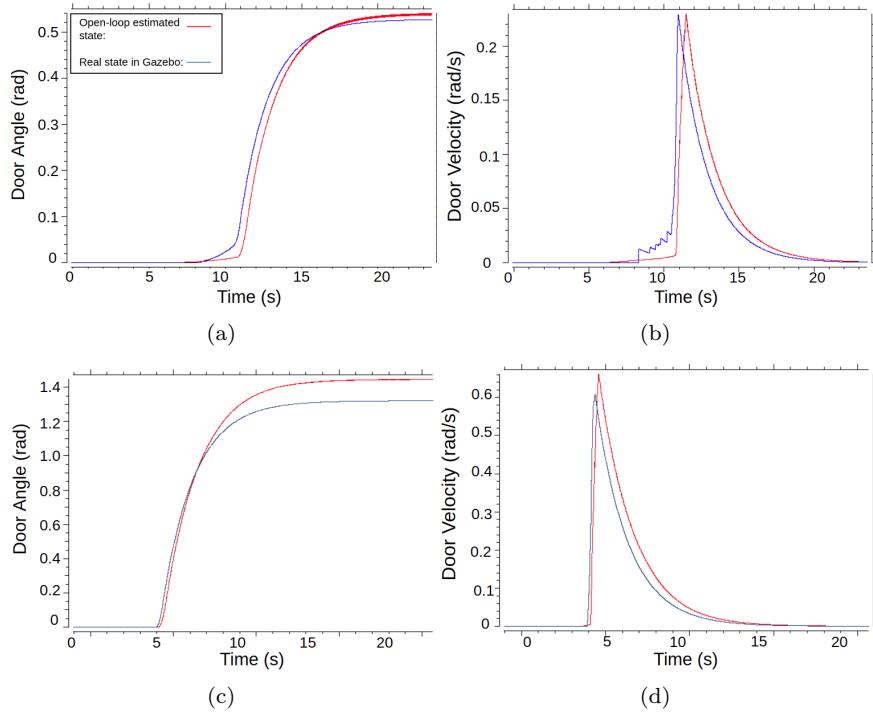


Figure 3.13: Plots of simulation results of manipulated-door states in the door-pushing task. (a)(b) illustrate the door angle and velocity in terms of time with a desired final angle of 0.5rad , with the open-loop estimation and real value in Gazebo plotted. (c)(d) indicate the case with a desired final angle of 1.5rad .

Chapter 4

Conclusion and Future Work

In this work, we adopt the contact-implicit optimization approach to tackle mobile manipulation problems, performed by a quadrupedal mobile robot-ALMA C-equipped with a 6 DoF robot-manipulator. We propose, as far as we know, one of the first optimization-based frameworks that encodes multi-object multi-contact manipulation utilizing the CIO scheme. Motivated by the foot EE manipulation task, we introduce virtual forces to describe the interaction with manipulated-objects besides the original contact forces in the switched system. The virtual forces are applied on their corresponding virtual contact points, which coincide with initially existing contact points. The planner is thereby capable of decoupling manipulation-related forces from gait-related ones on the same contact point. The one-by-one correlation between the contact point and the manipulated-object facilitates us in explicitly expressing the objects' dynamics.

On the other side, we assume a hard-contact model and present a reformulation of complementarity constraints. The introduction of slack variables penalized in the input space softens the equality constraints and aids the solver in efficiently converging to the local minimum. In the whole-body controller, we propose an intuitive open-loop estimation model to approximately track the manipulated-objects state.

The verification of the effectiveness of our approach is based on a wide range of object manipulation test cases. We concentrate on tasks varying from single-contact single-object manipulation, multi-contact single-object manipulation to multi-contact multi-object manipulation. The diversity of resulting optimal state and input trajectories indicates the feasibility of this approach in terms of discovering contact-schedules. Moreover, we experimentally substantiate the physical fidelity of the variant of complementarity model as well as its merits over its counterpart, the multi-phase approach. On top of that, the open-loop object's state estimation is proved to be effective by comparing the estimation and true values.

When it comes to future work, we expect to verify the MPC-based mobile manipulation approach in hardware tests. By comparing the simulation and hardware results, we could provide a solid substantiation for the implementation. One of the other natural extensions of this work would be improving the object's state estimation strategy utilizing sensor-based information from LiDAR or camera. For instance, an on-board camera could detect an April-tag labeled on the manipulated-object and identify the pose and velocity of it. With measurement units and feedback information, the estimator is more robust and involved manipulation tasks could be possibly resolved.

Another perspective is to identify the manipulated objects' dynamical properties in real-time, since every so it is arduous to acquire the prior knowledge of the manipulated-objects. Possible ways of achieving this would be combining online parameter learning with parameter tracking techniques such as KF, so as to make the model adaptive to disturbances and model mismatches of objects.

Furthermore, it is an appealing topic to implement CIO-based gait sequences. The switched

system is currently encoded by the multi-phase scheme, with switching times pre-specified. Assuming four leg end-effectors could automatically detect contact points with respect to the ground, the gait would be constantly under dynamic adjustment.

Bibliography

- [1] M. P. Murphy, B. Stephens, Y. Abe, and A. A. Rizzi, “High degree-of-freedom dynamic manipulation,” in *Unmanned Systems Technology XIV*, vol. 8387. International Society for Optics and Photonics, 2012, p. 83870V.
- [2] C. D. Bellicoso, K. Krämer, M. Stäuble, D. Sako, F. Jenelten, M. Bjelonic, and M. Hutter, “Alma-articulated locomotion and manipulation for a torque-controllable robot,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8477–8483.
- [3] J.-P. Sleiman, J. Carius, R. Grandia, M. Wermelinger, and M. Hutter, “Contact-implicit trajectory optimization for dynamic object manipulation,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6814–6821.
- [4] D. E. Stewart and J. C. Trinkle, “An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction,” *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.
- [5] M. Posa, C. Cantu, and R. Tedrake, “A direct method for trajectory optimization of rigid bodies through contact,” *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 69–81, 2014.
- [6] I. Mordatch, Z. Popović, and E. Todorov, “Contact-invariant optimization for hand manipulation,” in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, 2012, pp. 137–144.
- [7] I. Mordatch, E. Todorov, and Z. Popović, “Discovery of complex behaviors through contact-invariant optimization,” *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–8, 2012.
- [8] I. Mordatch, J. M. Wang, E. Todorov, and V. Koltun, “Animating human lower limbs using contact-invariant optimization,” *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, pp. 1–8, 2013.
- [9] M. Neunert, M. Stäuble, M. Gifthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, “Whole-body nonlinear model predictive control through contacts for quadrupeds,” *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.
- [10] H. Dai, A. Valenzuela, and R. Tedrake, “Whole-body motion planning with centroidal dynamics and full kinematics,” in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 295–302.
- [11] Y. Wang and S. Boyd, “Fast model predictive control using online optimization,” *IEEE Transactions on control systems technology*, vol. 18, no. 2, pp. 267–278, 2009.
- [12] E. Todorov and W. Li, “A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems,” in *Proceedings of the 2005, American Control Conference, 2005*. IEEE, 2005, pp. 300–306.

- [13] D. H. Jacobson and D. Q. Mayne, “Differential dynamic programming,” 1970.
- [14] W. Li and E. Todorov, “Iterative linear quadratic regulator design for nonlinear biological movement systems.” in *ICINCO (1)*. Citeseer, 2004, pp. 222–229.
- [15] A. Sideris and J. E. Bobrow, “An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems,” in *Proceedings of the 2005, American Control Conference, 2005*. IEEE, 2005, pp. 2275–2280.
- [16] F. Farshidian, M. Neunert, A. W. Winkler, G. Rey, and J. Buchli, “An efficient optimal planning and control framework for quadrupedal locomotion,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 93–100.
- [17] F. Farshidian, E. Jelavic, A. Satapathy, M. Giftthaler, and J. Buchli, “Real-time motion planning of legged robots: A model predictive control approach,” in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 577–584.
- [18] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, “Feedback mpc for torque-controlled legged robots,” *arXiv preprint arXiv:1905.06144*, 2019.
- [19] J.-P. Sleiman, F. Farshidian, and M. Hutter, “Constraint handling in continuous-time ddp-based model predictive control,” *ArXiv*, vol. abs/2101.06067, 2021.
- [20] C. D. Bellicoso, C. Gehring, J. Hwangbo, P. Fankhauser, and M. Hutter, “Perception-less terrain adaptation through whole body control and hierarchical optimization,” in *2016 IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 558–564.
- [21] C. D. Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, “Dynamic locomotion and whole-body control for quadrupedal robots,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3359–3365.
- [22] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, “A unified mpc framework for whole-body dynamic locomotion and manipulation,” *arXiv preprint arXiv:2103.00946*, 2021.