

Sequential Motion Planning for Bipedal Somersault via Flywheel SLIP and Momentum Transmission with Task Space Control

Xiaobin Xiong and Aaron D. Ames

Abstract—In this paper, we present a sequential motion planning and control method for generating somersaults on bipedal robots. The somersault (backflip or frontflip) is considered as a coupling between an axile hopping motion and a rotational motion about the center of mass of the robot; these are encoded by a hopping Spring-loaded Inverted Pendulum (SLIP) model and the rotation of a Flywheel, respectively. We thus present the Flywheel SLIP model for generating the desired motion on the ground phase. In the flight phase, we present a momentum transmission method to adjust the orientation of the lower body based on the conservation of the centroidal momentum. The generated motion plans are realized on the full-dimensional robot via momentum-included task space control. Finally, the proposed method is implemented on a modified version of the bipedal robot Cassie in simulation wherein multiple somersault motions are generated.

I. INTRODUCTION

Human athletes perform somersaults as an ultimate demonstration of mobility and dynamic maneuverability. The anthropomorphic partner in the robotics world, the hydraulic-actuated robot Atlas from Boston dynamics, was able to backflip [1], showing the decades of dedicated effort towards advancing robotic technologies; this dates back to the frontflips of the Raibert’s hopper [2], [3] in the 1980s. Regardless of the practical ramifications of doing somersaults on bipedal robots, achieving this kind of highly dynamic behavior benchmarks the stage of maturity of engineering robotic systems.

The back-flip of Atlas inspired the quadrupedal locomotion community which recently has built and controlled electric-powered quadrupedal robots to perform backflips [4], [5]. However, these results have not been widely seen on full-scale bipedal robots, despite the existence of studies of bipedal somersaulting in [6], [7], [8].

This is in part due to the fact that bipedal robots typically have more complex designs and, roughly speaking, less region of stability comparing to quadrupeds. These somersault behaviors require robot design to be able to make the behaviors feasible so that feedback control can play a role. The first observation is that the actuators need to be powerful to generate large vertical acceleration so that the robot can jump up to a certain height. The angular acceleration should be large enough so that enough momentum is generated on the ground to enable whole-body rotation in the flight phase.

Another observation is that backflipping requires the robot to have an upper body with large inertia and large ranges of

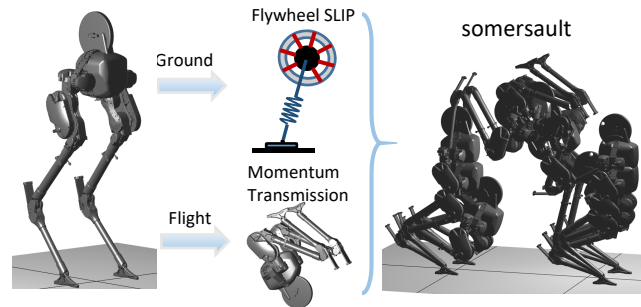


Fig. 1. Illustration of the proposed methodology for generating somersaults.

motion at certain joints. To achieve the flip motion, the robot should have enough momentum at lift-off, which should be mainly generated by the upper body. This is because the lower body typically extends for jumping, the momentum of which cannot be freely generated. Additionally, the robot should have large ranges of motion at joints that connect to the linkages with large inertia; for instance, if the hip joints have small ranges of motion, then it requires impractical torques to accelerate the joints to the required rotational velocity within the ranges of motion.

As illustrated thus far, achieving somersaults is a complex system-level problem of design and control. In this paper, we mainly investigate the motion planning and feedback control component in achieving these motions. The robot design to make somersaults feasible is out of the scope of this paper, even though it is an important and challenging problem.

In [9], [4], the motion plans for backflips are generated via trajectory optimization on the full-dimensional model of the quadrupeds, which has computational challenges for more complex bipedal robots. The formulated optimization is also highly nonconvex in a large dimensional space [10], [11], [12]. The optimization typically requires good initial guesses to solve, which can have feasibility issues. If the robot design renders the motion infeasible to realize, the trajectory optimization runs into infeasibility and provides little information towards the possible improvement in the robot design. Moreover, the optimized plans based on the full model typically are not robust to modeling errors and external disturbances. Thus a principled simple representation, e.g. the Linear Inverted Pendulum (LIP) [13] for walking and the Spring-loaded Inverted Pendulum (SLIP) [14] for running, is preferred for studying the essential components which realize the somersault.

To address this, we present a Flywheel SLIP (FSLIP)

*This work is supported by NSF 1924526, NSF 1932091, NSF 1923239. The authors are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125 xxiong@caltech.edu, ames@caltech.edu

model to capture somersault dynamics on the ground phase, including jumping and landing phases. The FSLIP has a flywheel [15], [16], [17] which is attached on a SLIP [18] model. The point mass dynamics of the FSLIP represents the hopping part of the somersault, while the flywheel captures the centroidal angular dynamics [19]. Trajectory optimization is formulated on the FSLIP to generate jumping and landing behaviors subject to the task constraints and additional physical constraints of the somersault. With the low dimensionality, it is easy and fast to solve the optimization.

When the robot is in the flight phase, the centroidal angular momentum [20], [21], [22], [23], [24] is conserved. We present a moment transmission control for a specific linkage to modulate the angle of the rest of the body. The specific linkage can be a flywheel, a tail, or arms of the robot. Moreover, additional assistive strategies are presented towards the landing.

The motion planning from the reduced-order representation is generated in sequence to the domains of the somersault. The motion plan is realized on the full-dimensional robot via an optimization-based controller. The optimization-based controller also takes into consideration the ground reaction force constraints and the input constraints.

The proposed method is realized on a modified version of the bipedal robot Cassie in simulation. We added a flywheel on the top of its pelvis to increase the inertia of its upper body and to amplify its capability to control the momentum. The flywheel can also be viewed as a representation of arms on the robot that have a free range of motion at the shoulders. Various somersault motions are generated, including front/back-flips and frontflips with over 1m forward jumping distance. Robustness and practical feasibility are also analyzed and discussed. With the formulated FSLIP model and the correspondence to the realization on a full dimensional robot, we hope that this study can shed some light on bipedal robot design and motion generation towards realizing versatile aerial motion on bipedal robots.

II. MODELLING

We start by presenting the model of the robot with the modification. Then we model the hybrid dynamics of the somersault, which is used in the simulation and analysis of the proposed planning and control methods.

A. Robot Model

The robot Cassie is a bipedal robot that and its predecessor ATRIAS [25] are built to best mimic the canonical Spring-loaded Inverted Pendulum (SLIP) model. As a result, it has a concentrated upper body and mechanical springs in the leg. The robot has shown good mechanical properties for walking [26], [27]. However, the trivial torso on Cassie makes it extremely difficult to produce any movement with aerial phases.

For instance, in [28], vertical hopping is realized on the robot with a goal of producing zero momentum before the flight phase. The leg extension generates a non-zero angular momentum, and thus the torso has to cancel out the angular

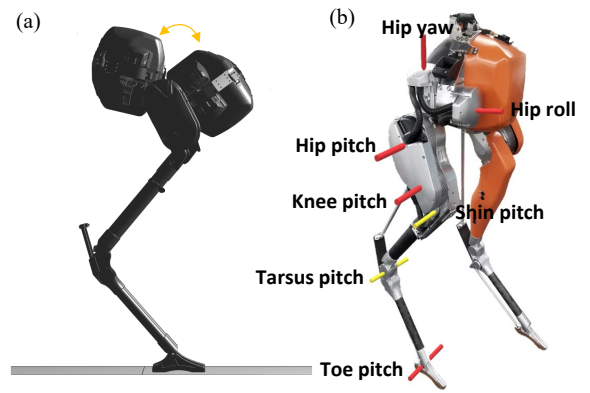


Fig. 2. (a) The range of motion of the pelvis. (b) The joints on the lower body of the bipedal robot Cassie by Agility Robotics.

momentum by producing a large forward rotation of the pelvis, which is limited by its range of motion (see Fig. 2 (a)), and maximum speed at the hip pitch joint. In other words, with a small torso to generate a large momentum, the required torso rotational speed is impractical to be realized within the range of motion. Thus using the trivial torso to generate somersaults also becomes impractical. To enable the capability, we present the following modification.

Hardware Modification: A flywheel is added on the top of the pelvis of the robot Cassie. It can be viewed as a robotic tail in some sense, e.g. the tail on Salto [29]. The design can be optimized for the location and size of the flywheel. Since the focus of this paper is on the planning and control, we manually select the installation of the flywheel based on intuition. The flywheel is installed 15cm above the pelvis, assuming the flywheel is made of aluminum. The exact specifications of the flywheel are discussed in section VI. Except the flywheel, all physical parameters of the robot remain unchanged.

Dynamics Model: We use a model of 22 degrees of freedom (DOFs) to represent the robot Cassie [28]. With the flywheel modification, the robot has 23 DOFs. Let $q \in SE(3) \times \mathbb{R}^{17}$ represent the configuration of the robot. The Euler-Lagrange equation for its dynamics is:

$$M(q)\ddot{q} + H(q, \dot{q}) = Bu + J_s^T \tau_s + J_{h,v}^T F_{h,v}, \quad (1)$$

$$J_{h,v}(q)\ddot{q} + \dot{J}_{h,v}(q)\dot{q} = 0, \quad (2)$$

where $M(q)$ is the mass matrix, $H(q, \dot{q})$ is the Coriolis, centrifugal and gravitational term, B and $u \in \mathbb{R}^{11}$ are the actuation matrix and the motor torque vector, τ_s and J_s are the spring joint moment vector and the corresponding Jacobian, and $F_{h,v} \in \mathbb{R}^{n_{h,v}}$ and $J_{h,v}$ are the holonomic force vector and the corresponding Jacobian respectively. The subscript v is used to indicate different domains which have different numbers of holonomic constraints.

B. Hybrid Model of Somersaulting

The somersault is a hybrid dynamical phenomenon with ground and flight phases. We assume that the robot hops in the sagittal plane and its left and right feet have the same

contact mode. For simplification, we also assume the feet always flatly contact the ground, which will be encoded as one of the control objectives. Based on the sequence of the contact modes, we divide the motion into three domains, i.e., $v = \text{jumping, flight, and landing}$. In jumping, the feet are in contact with the ground. It transits from jumping to flight when the ground reaction normal forces reach to zero. In the flight phase, the feet are above the ground. The robot goes through a large whole-body rotation in its sagittal plane. It transits into landing when the feet strike the ground. The foot-ground impact is modeled as plastic impact [30], where the position trajectories are smooth while the velocities are discontinuous.

III. THE FLYWHEEL SLIP MODEL

In this section, we explain the Flywheel SLIP (FSLIP) model for generating the jumping and landing trajectories for the robot. The point mass dynamics of the FSLIP captures the hopping dynamics of the center of mass (COM) dynamics of the robot; the rotational dynamics of the flywheel corresponds to the rotational part of the centroidal dynamics of the robot. Using parametric regressions on the stiffness of the leg spring and the centroidal inertia of the robot, we identify the Flywheel SLIP model to best capture the features of the robot.

A. The Dynamics

The Flywheel SLIP is the actuated SLIP model in [18], [31] with a flywheel attached on the mass and an actuated foot (see Fig. 3 (a)). The actuated SLIP is the canonical SLIP model with leg length actuation and a prismatic spring. The flywheel is actuated to rotate about the mass location. The foot actuation is added to work with the flywheel to generate appropriate acceleration. The dynamics are given by:

$$m\ddot{x} = -\frac{\tau + u}{r} \cos(\beta) + F^s \sin(\beta) \quad (3)$$

$$m\ddot{z} = -mg + \frac{\tau + u}{r} \sin(\beta) + F^s \cos(\beta) \quad (4)$$

$$I\ddot{\theta} = \tau \quad (5)$$

$$\ddot{s} = \ddot{L} - \ddot{r} \quad (6)$$

where:

- m is the mass of the mass and the flywheel,
- I is the rotational inertia of the flywheel,
- τ is the torque on the flywheel,
- u is the foot actuation,
- θ is the angle of the flywheel,
- β is the angle of the leg,
- r is the real leg length,
- L is the actuated leg length,
- s is the spring deformation,
- F^s is the spring force,
- x, z are the horizontal and vertical mass positions.

The kinematic constraints are $s + r = L$ and $\dot{s} + \dot{r} = \dot{L}$. The spring force is calculated by: $F^s = Ks + D\dot{s}$, where K, D are the stiffness and damping of the spring.

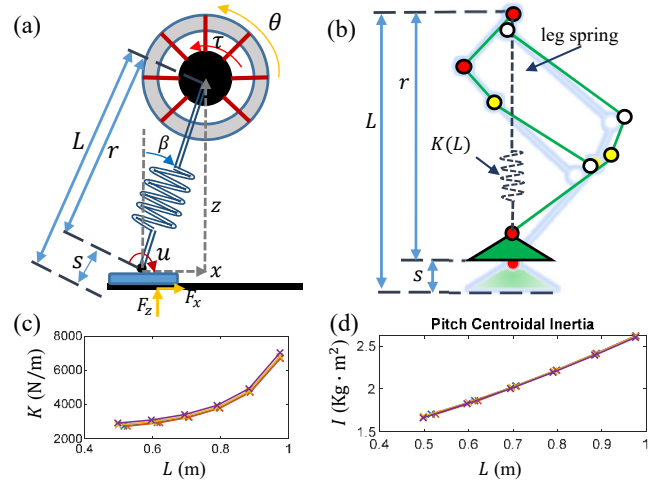


Fig. 3. (a) The Flywheel SLIP. (b) Illustration of the leg spring from the leg of Cassie. (c, d) The stiffness of the leg spring and the pitch centroidal inertia of the robot v.s. the leg length for different standing configurations.

Correspondence to the robot: The Flywheel SLIP model described above is general. Here we use parametric regressions to best match the model to the robot. Similar to [28], [18], we identify the stiffness and damping of the leg spring from the joint springs of the robot. Then parametric regressions are used to project the stiffness K and damping D of the leg spring over the leg length L . Thus the leg spring of the FSLIP mimics the leg of the robot, as illustrated by Fig. 3 (b). Additionally, we regress the centroidal pitch inertia [32] $I(L)$ of the robot, which becomes the inertia of the Flywheel. It is important to note that it is the inertia of the whole robot instead of the flywheel on the pelvis that corresponds to the flywheel on the FSLIP. The regressed FSLIP model from the robot is expected to produce behaviors which can be translated to Cassie. Fig. 3 (c) (d) show the leg stiffness of the leg spring and the pitch inertia over different leg length for different configurations, indicating that they are mainly changed by L .

B. Jumping Optimization

Now we use the Flywheel SLIP model to generate a jumping motion from standing for enabling the somersault. The planned motion is required to satisfy the actuation constraints, the ground force constraints, the required flipping momentum, and the apex constraints. The planned motion contains the trajectories of the point mass and desired centroidal momentum, which will be tracked by the robot.

Similar to [28], we use a direct transcription method [33], [10] to describe and solve the optimization problem. The trajectories are discretized and the dynamics are integrated via trapezoidal integration. Here we mainly describe the constraints and the cost function of the optimization.

Task Constraints: The goal of the jumping optimization is to create enough vertical velocity and angular momentum to enable somersaults. The task is encoded via the velocities

at the lift-off event. We assume that the robot lands with a similar self-configuration to that at lift-off. Thus the duration of the flight phase is given by:

$$T_{\text{Flight}} = \frac{2z^{\text{lift-off}}}{g}, \quad (7)$$

where $\dot{z}^{\text{lift-off}}$ is the vertical velocity of the point mass at lift-off. If the robot fixes its configuration during the flight phase, the angular velocity of the whole body at lift-off should be:

$$\dot{\theta}^{\text{lift-off}} = \frac{\pm 2\pi - 2\beta^{\text{lift-off}}}{T_{\text{Flight}}} = \frac{(\pm\pi - \beta^{\text{lift-off}})g}{\dot{z}^{\text{lift-off}}}. \quad (8)$$

Note that we will relax the condition of the fixed configuration on the robot later. Eq. (8) establishes the constraint on the jumping phase to produce a flip motion. The sign \pm depends on if the somersault is a backflip or frontflip.

Physical Constraints: During jumping, we enforce non-slipping condition on the foot via,

$$-F_z\mu \leq F_x \leq F_z\mu, \quad (\text{Friction Cone})$$

where F_z is the vertical ground reaction force (GRF), and μ is the friction coefficient. It is also desirable to keep the feet flat on the ground, thus the zero moment point (ZMP) constraint is included:

$$-F_z \frac{l}{2} \leq u \leq F_z \frac{l}{2}, \quad (\text{ZMP})$$

where l is the foot length. Additional constraints include the actuation limits, leg length limits and non-negative normal ground reaction forces. The final ground normal force should reach to 0 to trigger the lift-off event, i.e., $F_z^{\text{lift-off}} = 0$.

Cost: The cost for the jumping optimization is on the virtually consumed energy:

$$J_{\text{Jumping}} = \sum_i (c_1 \dot{L}_i^2 + c_2 u_i^2 + c_3 \tau_i^2) \Delta T, \quad (9)$$

where $c_{1,2,3}$ are the coefficients, ΔT is time discretization, and i is the index for the discretized variables. Fig. 4 illustrates an example of the optimized trajectories of a jumping behavior.

C. Landing Optimization

The landing optimization is performed after the touchdown event at the end of the flight phase. Since it uses the same Flywheel SLIP model, we briefly explain the landing planning here.

During landing, the system rests to a static configuration from the flight phase. The task can be directly encoded via the equality constraints on the final states:

$$\begin{aligned} z_N &= z^{\text{des}}, \\ \dot{z}_N &= x_N = \dot{x}_N = \dot{\theta} = 0, \\ F_N^s &= mg, \end{aligned}$$

where the subscript N indicates the final state of landing. The physical constraints are the same to that of the jumping

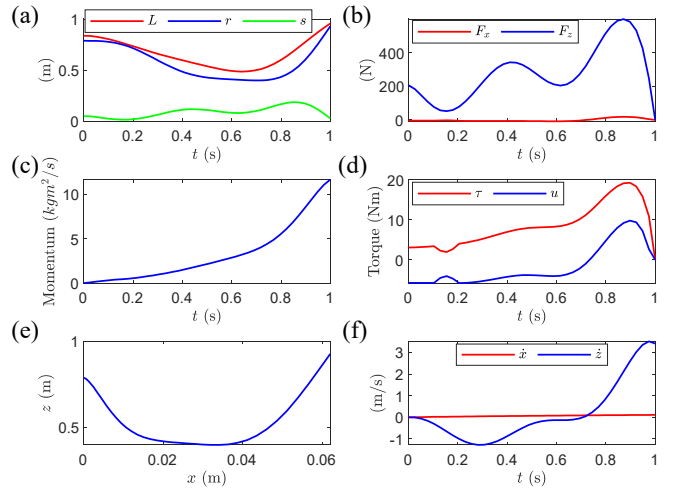


Fig. 4. An example of the optimized jumping trajectories of the leg length and spring deformation (a), ground reaction forces (b), the momentum of the flywheel (c), the torques (d), the position of the point mass (e) and the velocities of the mass (f).

planning. The initial states are constrained based on the post-impact state of the robot. The cost is the cost of jumping plus a penalty on the spring oscillation:

$$J_{\text{Landing}} = J_{\text{Jumping}} + \sum_i \alpha \dot{s}_i^2 \Delta T, \quad (10)$$

where α is a weighting coefficient.

IV. FLIGHT PHASE PLANNING

In this section, we present the planning methodology for the flight phase. The core idea here is to change the momentum of the flywheel to control the configuration of the lower body of the robot. This is due to the conservation of the angular centroidal momentum of the robot in the flight phase. Additionally, we add two components to assist the flipping motion with a view toward landing the jump.

A. Momentum Transmission on the Flywheel

As the flight phase is underactuated due to the conservation of momentum, direct controlling the rotation of the floating robot is not possible. However, it is possible to change the momentum of the lower body of the robot (linkages except the flywheel) by controlling the momentum of the flywheel. The total centroidal pitch momentum $\mathcal{H}_{\text{pitch}}$ of the robot in the flight phase is,

$$\mathcal{H}_{\text{pitch}} = I_{\text{flywheel}} \omega_{\text{flywheel}} + \sum_{i=2}^N I_i \omega_i, \quad (11)$$

where $I_{\text{flywheel}}, \omega_{\text{flywheel}}$ are the pitch inertia and velocity of the flywheel about the COM of the robot, and I_i, ω_i are the pitch inertia and velocity of the other linkages about the COM. The conservation of momentum indicates $\mathcal{H}_{\text{pitch}}$ is constant. The averaged angular velocity of lower body can be defined as:

$$\bar{\omega} = \frac{\sum_{i=2}^N I_i \omega_i}{\sum_{i=2}^N I_i}, \quad (12)$$

which is controllable by changing the momentum of the flywheel. If we assume the robot is rigid, the flipping condition becomes,

$$\int_0^{T_{\text{Flight}}} \bar{\omega}(t) dt = \pm 2\pi - 2\beta^{\text{lift-off}}. \quad (13)$$

The initial pitch angle of the lower body is $\theta(t=0) = \beta^{\text{lift-off}}$, and the final angle at touch-down is $\theta(t=T_{\text{Flight}}) = \pm 2\pi - \beta^{\text{lift-off}}$, where \pm depends on the direction of the somersault. The initial velocity $\bar{\omega}^{\text{lift-off}}$ is calculated from the robot, and we let the final velocity $\bar{\omega}^{\text{touch-down}} = 0$. Thus, trajectories of $[\theta^{\text{des}}(t), \bar{\omega}^{\text{des}}(t)]$ can be constructed based on the boundary conditions. Since it is desirable to adjust the pitch angle of the lower body to prepare for landing at good configurations, we add a feedback law on the desired average velocity:

$$\dot{\theta}_{cl}^{\text{des}}(t) = \bar{\omega}^{\text{des}}(t) - k_p(\theta_{\text{pelvis}}(t) - \theta^{\text{des}}(t)), \quad (14)$$

where we will use the pelvis pitch angle θ_{pelvis} as a reference for the orientation of the lower body. Thus we control the pelvis pitch angle to be the same of the leg angle β on the ground phase. Finally, the desired centroidal momentum of the flywheel is:

$$\mathcal{H}_{\text{flywheel}}^{\text{des}}(t) = \mathcal{H}_{\text{pitch}} - \dot{\theta}_{cl}^{\text{des}}(t)(\sum_{i=2}^N I_i). \quad (15)$$

The rotational momentum required to achieve the whole body rotation is mainly accumulated on the flywheel during the jumping phase. In the flight phase, the momentum is transmitted from the flywheel to the lower body to achieve the desired rotation. Thus we refer Eq. (15) as the *momentum transmission* of the flywheel.

B. Assistive Components

In addition to the control of the flywheel, the lower body is also controlled to assist the somersault. We consider two assistive components as follows:

Leg Retraction: The legs of the robot can retract in the beginning of the flight phase and then extend towards the landing. This is referred to as *tucking* in [2]. The purpose of the tucking is to decrease the centroidal inertia and thus increase the angular rotational rate. We implement this strategy by controlling the leg length. The desired trajectory is constructed smoothly with appropriate timing.

Foot Placement: The robot is expected to achieve the desired angular rotation and then land on the ground. To have the appropriate landing configuration, the robot should control the horizontal positions of the feet, which is controlled by adjusting the leg angle to the opposite of the leg angle at lift-off. By symmetry, the opposite leg angle can ensure the feasibility of landing in the optimization. Additionally, we control the feet to be flat to prepare for landing.

V. OUTPUT CONSTRUCTION AND STABILIZATION

In this section, we explain the control of the aforementioned plans as they are realized on the 3D robot Cassie. We first explain the output selection and construction in each domain, then derive the optimization-based control method for stabilization.

In our previous work [28], [34], we applied the control Lyapunov function based quadratic programs (CLF-QPs) [35] for output stabilization with momentum outputs. A different optimization-based control, i.e., task space control (TSC) [36], [37], [38], [39], can serve the same purpose of stabilization. Here we derive TSC with minimum variable formulation for illustration.

A. Output Construction

The output construction is used to specify the desired behavior from the Flywheel SLIP model and the momentum transmission for the robot via featured functions (outputs) of the states. There are different outputs to be specified in the ground and flight phases.

1) *Jumping/Landing:* The Flywheel SLIP provides the desired point mass dynamics and rotational dynamics for the robot. The point mass dynamics in the leg axial direction is enforced by the leg length $L_{\text{robot}}(q)$ on the robot; the mass dynamics in the translational/forward direction is enforced by the COM forward position $x_{\text{com}}(q)$ of the robot. The rotational dynamics is enforced via the centroidal pitch momentum $\mathcal{H}_{\text{Pitch}}(q, \dot{q})$ of the robot. Additionally, we enforce zero motion in the coronal and transversal planes by constraining the lateral position of the COM $y_{\text{com}}(q) \rightarrow 0$ and the yaw angle of the pelvis $\phi_{\text{yaw}}(q) \rightarrow 0$. The outputs are the same for the jumping and landing phase, which are:

$$\mathcal{Y}_1^{J/L}(q, \dot{q}, t) = \mathcal{H}_{\text{Pitch}}(q, \dot{q}) - \mathcal{H}_{\text{FSLIP}}(t), \quad (16)$$

$$\mathcal{Y}_2^{J/L}(q, t) = \begin{bmatrix} L_{\text{robot}}(q) \\ x_{\text{com}}(q) \\ y_{\text{com}}(q) \\ \phi_{\text{pitch}}(q) \\ \phi_{\text{yaw}}(q) \end{bmatrix} - \begin{bmatrix} L_{\text{FSLIP}}(t) \\ x_{\text{FSLIP}}(t) \\ 0 \\ \phi_{\text{pitch}}^{\text{des}}(t) \\ 0 \end{bmatrix}. \quad (17)$$

where $\mathcal{H}_{\text{FSLIP}}(t) = I_{\text{FSLIP}}(t)\theta(t)$ is the momentum of the Flywheel SLIP, $x_{\text{FSLIP}}(t)$ is the forward position of the Flywheel SLIP, and $\phi_{\text{pitch}}^{\text{des}}(t)$ is the desired pitch angle of the pelvis by construction. The subscript $1/2$ indicates the number of times one must differentiate to have the torque u appear.

2) *Flight:* As mentioned in the previous section, the flight phase is underactuated. The pitch momentum is transmitted between the flywheel and the lower body by Eq. (15). Thus the momentum of the flywheel $\mathcal{H}_{\text{flywheel}}(q, \dot{q})$ is controlled:

$$\mathcal{Y}_1^F(q, \dot{q}) = \mathcal{H}_{\text{flywheel}}(q, \dot{q}) - \mathcal{H}_{\text{flywheel}}^{\text{des}}(t). \quad (18)$$

To enable body tucking, the leg length $L_{\text{robot}}(q)$ are controlled to retract first and then extend. The rest joints of the body $q_{/\text{knee}}$, i.e. motor joints except the knee joints, are controlled as fixed. Thus the outputs are defined as:

$$\mathcal{Y}_2^F(q, t) = \begin{bmatrix} L_{\text{robot}}(q) \\ q_{/\text{knee}} \end{bmatrix} - \begin{bmatrix} L^{\text{des}}(t) \\ q_{/\text{knee}}^+ \end{bmatrix}, \text{ if } t < T_1 \quad (19)$$

where T_1 is manually decided, e.g., $T_1 = \frac{3}{4}T_{\text{Flight}}$. As approaching landing, it is desirable to prepare for the touch-down event by controlling the leg angle $\beta_{\text{leg}}(q)$ and toe pitch

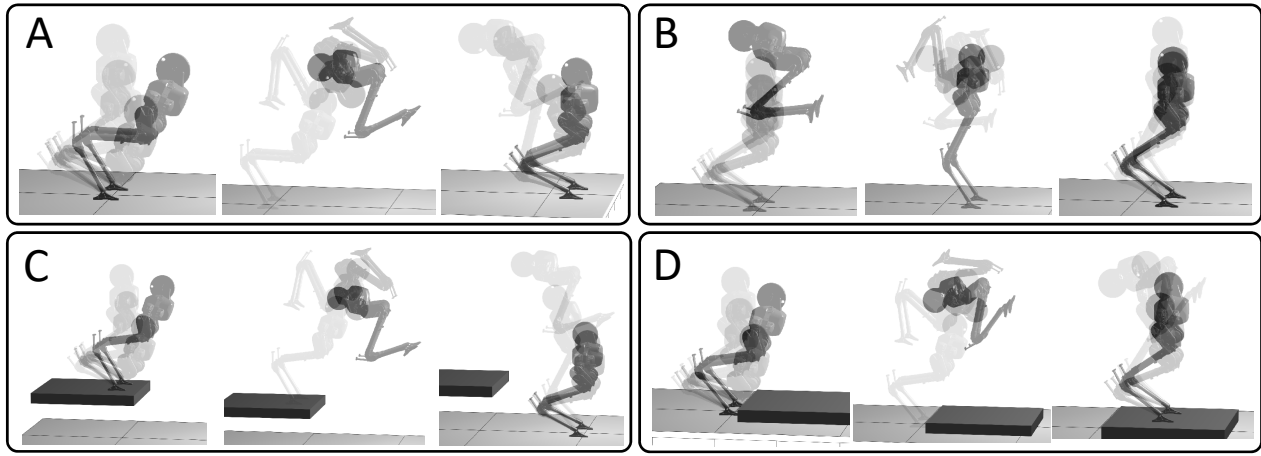


Fig. 5. Snapshots of the generated somersaults: (A) a frontflip with a 1m forward target distance, (B) a backflip, (C) a frontflip from a platform of 0.5m height, and (D) a frontflip without knowing the landing surface, which is with a 0.1m height from the ground.

angle $q_{\text{toe}}^{\text{pitch}}(q)$. The outputs are:

$$\mathcal{Y}_2^F(q, t) = \begin{bmatrix} L_{\text{robot}}(q) \\ \beta_{\text{leg}}^{\text{pitch}}(q) \\ q_{\text{toe}}^{\text{pitch}}(q) \\ q_{\text{rest}} \end{bmatrix} - \begin{bmatrix} L^{\text{des}}(t) \\ -\beta_{\text{leg}}^+ \\ 0 \\ q_{\text{rest}}^+ \end{bmatrix}, \text{ if } t \geq T_1, \quad (20)$$

where β_{leg}^+ is the leg angle at the lift-off event. q_{rest} denotes the rest motor joints except for the constrained one, which in this case are the hip roll and yaw joints.

B. Momentum-Included Task Space Control

Finally we present the output stabilization via the Task Space Control (TSC) to drive $\mathcal{Y}_2 \rightarrow 0, \mathcal{Y}_1 \rightarrow 0$. Typically,

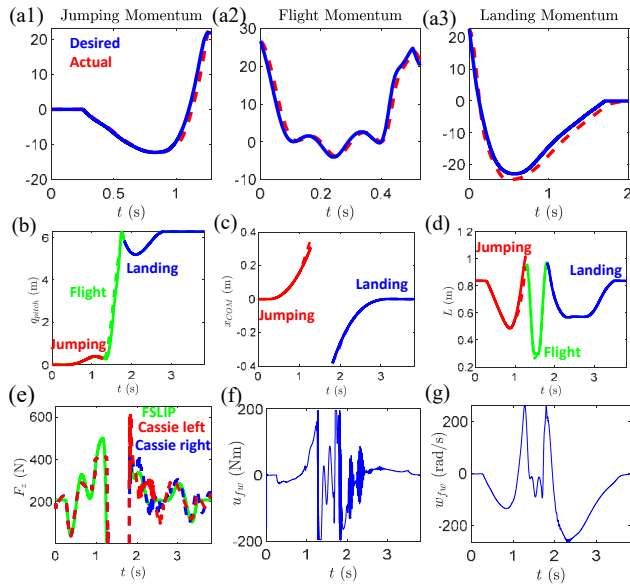


Fig. 6. Simulation plots of the forward frontflip A (dashed lines are the actual; solid lines are the desired): (a1-a3) The momentum output trajectories in each domain (The flywheel momentum is the output in (a2)). (b) The leg length trajectories. (c) The ground norm forces of Cassie v.s. that of the FSLIP. (d) x_{COM} . (e) The pitch angle of the pelvis. (f) The applied torque on the flywheel. (g) The rotational speed of the flywheel.

the joint accelerations, motor torques, and ground reaction forces are all included as optimization variables in the TSC formulation. Here we present the TSC formulation with only the motor torques as the optimization variables, which lets the TSC based quadratic program be solved efficiently.

First, we notice that the ground reaction forces are affine functions of the input torques u as:

$$F_v = A_v u + b_v, \quad (21)$$

$$A_v = -(J_v M^{-1} J_v^T)^{-1} J_v M^{-1} B, \quad (22)$$

$$b_v = (J_v M^{-1} J_v^T)^{-1} (J_v M^{-1} H - \dot{J}_v \dot{q}). \quad (23)$$

Plugging this into the equation of motion yields,

$$M(q)\ddot{q} + H(q, \dot{q}) - \underbrace{J_s^T \tau_s(q)}_{\bar{H}} - \underbrace{J_{h,v}^T b_v}_{\bar{B}_v} = \underbrace{(B + J_{h,v}^T A_v)}_{\bar{B}} u,$$

which yields the affine relation $\ddot{q} = \bar{A}u + \bar{b}$ with

$$\bar{A} = M^{-1} \bar{B}, \bar{b} = -M^{-1} \bar{H}. \quad (24)$$

As the derivatives of outputs are affine w.r.t. \ddot{q} , they are also affine w.r.t. u :

$$\dot{\mathcal{Y}}_1 = A_1 \ddot{q} + b_1 = A_1 u + \mathcal{B}_1, \quad (25)$$

$$\dot{\mathcal{Y}}_2 = A_2 \ddot{q} + b_2 = A_2 u + \mathcal{B}_2, \quad (26)$$

where the explicit expression of each term is omitted. Compactly,

$$\dot{\mathbf{Y}} = \mathbf{A}u + \mathbf{B} \quad (27)$$

with $\mathbf{Y} = [\mathcal{Y}_1; \mathcal{Y}_2]$. The desired derivatives (accelerations) for stabilization are,

$$\dot{\mathcal{Y}}_1^{\text{des}} = -K_1^p \mathcal{Y}_1, \quad (28)$$

$$\dot{\mathcal{Y}}_2^{\text{des}} = -K_2^p \mathcal{Y}_2 - K_2^d \dot{\mathcal{Y}}_2. \quad (29)$$

This motivates a quadratic cost to minimize the difference between the actual and desired accelerations:

$$J_{\text{TSC}}(u) = (\mathbf{A}u + \mathbf{B} - \dot{\mathcal{Y}}^{\text{des}})^T (\mathbf{A}u + \mathbf{B} - \dot{\mathcal{Y}}^{\text{des}}), \quad (30)$$

with $\dot{y}^{des} = [\dot{y}_1^{desT}, \dot{y}_2^{desT}]^T$.

Additionally, two physical constraints are included for the feasibility of the torques on the robots. In the ground phase, the ground reaction forces have to satisfy the physics constraints such as non-negative normal forces and non-slipping condition, formulated by $C_v F_v \leq 0$, where C_v is a constant matrix. The constraint is also on u from Eq. (21):

$$C_v A_v u \leq -C_v b_v. \quad (31)$$

Moreover, the motor torque must be within the feasible limits of the robot hardware. For electrical motors, the available motor torque depends on the rotational speed. Then we encode that motor torque limit constraint as:

$$u_{lb}(\dot{q}) \leq u \leq u_{ub}(\dot{q}). \quad (32)$$

Thus the quadratic program based controller for each domain is formulated as:

$$\begin{aligned} u^* = \operatorname{argmin}_{u \in \mathbb{R}^{11}} \quad & J_{TSC}(u), \\ \text{s.t.} \quad & C_v A_v u \leq -C_v b_v, \quad (\text{GRF}) \\ & u_{lb}(\dot{q}) \leq u \leq u_{ub}(\dot{q}). \quad (\text{Torque Limit}) \end{aligned}$$

VI. RESULTS AND DISCUSSION

The proposed planning and optimization-based controller are implemented in simulation on the modified robot model of Cassie. The dynamics are integrated using Matlab ODE45 with event triggering. The trajectory optimization is solved via IPOPT [40]. Due to the simple dynamics, the optimizations are typically solved within seconds without code optimization for speed. The TSC is solved at 2kHz using qpOASES [41]. The results can be seen in [42].

Using the proposed method, we generated several somersault motions on the robot. We specify the desired jumping height and forward distance, and then the optimization on Flywheel SLIP is performed. Jumping is thus generated on the robot. When the robot lifts off, the momentum transmission control is applied. After touch-down, the landing optimization is performed. Then the robot lands to a static configuration by tracking the optimized landing trajectory.

Even though the model difference between the Flywheel SLIP and the robot can cause about 0.1m difference on the jumping height, the sequential planning and control still can achieve successful landing and accurate forward distance. Due to the numerical precision of integration, the robot can behave asymmetrically across the frontal plane; since lateral COM position is controlled at the ground phase, the robot is still stabilized. Fig. 5 demonstrates four somersaults. Fig. 6 show the trajectories of the produced somersault A.

Robust to ground height change: Suppose the ground height of the landing is not exactly known, the robot can still land successfully. The reason is that landing planning is performed after the post-impact event. The somersault D in Fig. 5 is an example where the ground height increases by 0.1m, which shows the robustness of the planning and control with respect to the ground height change.

Practical Feasibility: The result in Fig. 6 is on the robot with flywheel I selected in Table I, which has relatively small inertia comparing to the inertia of the rest of the robot. Therefore, the required rotational speed is high, and we replace Eq. (32) on the flywheel by a constant torque constraint to explore the required actuation. As a comparison, the knee motor module of Cassie has a maximum torque of 195.2(Nm); while the maximum speed after gear reduction is only 81.25(rpm). This implies that we need a significant improvement on the actuation design on the flywheel to realize such somersaulting motion.

We also tried different flywheel specifications in Table I to realize somersault C in Fig. 5, with an apex height of 1.2m from the platform. By limiting the maximum joint torque, the maximum required flywheel speed decreases with the increases of the flywheel inertia. However, if the mass increases, the lower body requires an increase of the maximum torques to generate the desired apex height. If the desired apex increases, which leads to a lower required flywheel momentum due to a longer flight duration, the knee actuation has to be increased. Another way to decrease the required momentum generation on the flywheel is to increase the hip pitch rotational speed at lift-off. We found that the contribution to decreasing the flywheel momentum is small unless the maximum hip joint speed is significantly increased. With the above discussion, we conclude that the practical realization of somersaults relies on powerful actuation design and good inertia distribution. This is a challenging topic to study in the field of legged locomotion.

VII. CONCLUSION AND FUTURE WORK

To conclude, we proposed a sequential planning and control method for generating somersaults on bipedal robots. The planning on the ground phase is based on the Flywheel SLIP model, and that in the flight phase follows the principle of conservation of momentum using momentum transmission. The planned trajectories are extracted and tracked by momentum-included task space control on the robot Cassie.

Future work will be devoted to understanding the necessary components in achieving other types of aerial motions, including periodic hopping, running, and aerial flipping in the transversal and coronal planes. Additionally, it is interesting to study how to generate somersaults on robots without free-rotational joints. From our perspective, the bottleneck for achieving this kind of highly dynamic motion on the physical robot is on the necessity to unify mechanical design with control implementation at the system level. Future work will explore this trade-off with a view toward control-informed robot design for realizing dynamic aerial motion.

TABLE I
DIFFERENT FLYWHEEL SPECIFICATIONS

Flywheel	I_{pitch} (kg · m ²)	m (kg)	τ_{max} (Nm)	w_{max} (rpm)
I	0.1074	9.54	195	1645
II	0.2148	19.09	195	812
III	0.3222	28.63	195	573

REFERENCES

- [1] “What’s new, atlas?, boston dynamics,” in <https://www.youtube.com/watch?v=fRj34o4hN4I>, 2018.
- [2] J. Hodgins and M. H. Raibert, “Biped gymnastics,” *Dynamically Stable Legged Locomotion*, p. 79, 1988.
- [3] R. R. Playter and M. H. Raibert, “Control of a biped somersault in 3d,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1. IEEE, 1992, pp. 582–589.
- [4] B. Katz, J. Carlo, and S. Kim, “Mini cheetah: A platform for pushing the limits of dynamic quadruped control,” in *Robotics and Automation (ICRA), 2019 IEEE International Conference on*. IEEE, 2019.
- [5] “Aliengo robot new progress: backflip etc., unitree robotics,” in <https://www.youtube.com/watch?v=Ayxoji2xg3k>, 2019.
- [6] T. Tamada, W. Ikarashi, D. Yoneyama, K. Tanaka, Y. Yamakawa, T. Senoo, and M. Ishikawa, “High-speed bipedal robot running using high-speed visual feedback,” in *2014 IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2014, pp. 140–145.
- [7] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, “Deepmimic: Example-guided deep reinforcement learning of physics-based character skills,” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.
- [8] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, “Deep reinforcement learning from human preferences,” in *Advances in Neural Information Processing Systems*, 2017, pp. 4299–4307.
- [9] B. Stephens, S. Kuindersma, and A. Saunders, “Atlas’s day out,” *Dynamic Walking Conference*, 2019.
- [10] A. Hereid, E. A. Cousineau, C. M. Hubicki, and A. D. Ames, “3d dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1447–1454.
- [11] 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.
- [12] Y. Ding, C. Li, and H.-W. Park, “Single leg dynamic motion planning with mixed-integer convex optimization,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–6.
- [13] S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, “The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation,” in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 239–246.
- [14] R. Altendorfer, U. Saranlı, H. Komsuoglu, D. Koditschek, H. B. Brown, M. Buehler, N. Moore, D. McMordie, and R. Full, “Evidence for spring loaded inverted pendulum running in a hexapod robot,” in *Experimental Robotics VII*. Springer, 2001, pp. 291–302.
- [15] J. Pratt, J. Carff, S. Drakunov, and A. Goswami, “Capture point: A step toward humanoid push recovery,” in *2006 6th IEEE-RAS international conference on humanoid robots*. IEEE, 2006, pp. 200–207.
- [16] B. Stephens, “Push recovery control for force-controlled humanoid robots,” Ph.D. dissertation, Carnegie Mellon University, The Robotics Institute, 2011.
- [17] T. Takenaka, “Real time motion generation and control for biped robot-3rd report: Gait pattern modification to compensate approximated dynamics error,” in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, 2009.
- [18] X. Xiong and A. D. Ames, “Coupling reduced order models via feedback control for 3d underactuated bipedal robotic walking,” in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.
- [19] G. Nava, D. Pucci, and F. Nori, “Momentum control of humanoid robots with series elastic actuators,” *arXiv preprint arXiv:1703.01882*, 2017.
- [20] D. E. Orin, A. Goswami, and S.-H. Lee, “Centroidal dynamics of a humanoid robot,” *Autonomous Robots*, vol. 35, no. 2-3, pp. 161–176, 2013.
- [21] A. Saccon, S. Traversaro, F. Nori, and H. Nijmeijer, “On centroidal dynamics and integrability of average angular velocity,” *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 943–950, 2017.
- [22] S.-H. Lee and A. Goswami, “Reaction mass pendulum (rmp): An explicit model for centroidal angular momentum of humanoid robots,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 4667–4672.
- [23] T. Koolen, S. Bertrand, G. Thomas, T. De Boer, T. Wu, J. Smith, J. Engelsberger, and J. Pratt, “Design of a momentum-based control framework and application to the humanoid robot atlas,” *International Journal of Humanoid Robotics*, vol. 13, no. 01, p. 1650007, 2016.
- [24] A. Hofmann, M. Popovic, and H. Herr, “Exploiting angular momentum to enhance bipedal center-of-mass control,” in *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*. IEEE, 2009, pp. 4423–4429.
- [25] J. A. Grimes and J. W. Hurst, “The design of atrias 1.0 a unique monopod, hopping robot,” in *Adaptive Mobile Robotics*. World Scientific, 2012, pp. 548–554.
- [26] Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. van de Panne, “Feedback control for cassie with deep reinforcement learning,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1241–1246.
- [27] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, “Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway,” *arXiv preprint arXiv:1809.07279*, 2018.
- [28] X. Xiong and A. D. Ames, “Bipedal hopping: Reduced-order model embedding via optimization-based control,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Spain*, <https://arxiv.org/pdf/1807.08037.pdf>.
- [29] D. W. Haldane, J. K. Yim, and R. S. Fearing, “Repetitive extreme-acceleration (14-g) spatial jumping with salto-1p,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3345–3351.
- [30] J. W. Grizzle, C. Chevallereau, R. W. Sinnet, and A. D. Ames, “Models, feedback control, and open problems of 3d bipedal robotic walking,” *Automatica*, vol. 50, no. 8, pp. 1955–1988, 2014.
- [31] X. Xiong and A. D. Ames, “Orbit characterization, stabilization and composition of 3d underactuated bipedal walking via hybrid passive linear inverted pendulum model,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- [32] D. E. Orin and A. Goswami, “Centroidal momentum matrix of a humanoid robot: Structure and properties,” in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 653–659.
- [33] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *Journal of guidance, control, and dynamics*, vol. 21, no. 2, pp. 193–207, 1998.
- [34] T. Westenbroek, X. Xiong, A. D. Ames, and S. S. Sastry, “Optimal control of piecewise-smooth control systems via singular perturbations,” in *IEEE Conference on Decision and Control (CDC)*, 2019.
- [35] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, “Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics,” *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.
- [36] B. Xian, M. S. de Queiroz, D. Dawson, and I. Walker, “Task-space tracking control of robot manipulators via quaternion feedback,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 160–167, 2004.
- [37] P. M. Wensing and D. E. Orin, “Generation of dynamic humanoid behaviors through task-space control with conic optimization,” in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3103–3109.
- [38] A. Escande, N. Mansard, and P.-B. Wieber, “Hierarchical quadratic programming: Fast online humanoid-robot motion generation,” *The International Journal of Robotics Research*, vol. 33, no. 7, pp. 1006–1028, 2014.
- [39] X. Xiong and A. D. Ames, “Dynamic and versatile humanoid walking via embedding 3d actuated slip model with hybrid lip based stepping,” *IEEE Robotics and Automation Letters*, 2020.
- [40] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.
- [41] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, “qpOASES: A parametric active-set algorithm for quadratic programming,” *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.
- [42] Simulation Results: <https://youtu.be/HchdmWrBd2o>.