Inverse Dynamics with Rigid Contact and Friction

Samuel Zapolsky · Evan Drumwright

Abstract Inverse dynamics is used extensively in robotics and biomechanics applications. In manipulator and legged robots, it can form the basis of an effective nonlinear control strategy by providing a robot with both accurate positional tracking and active compliance. In biomechanics applications, inverse dynamics control can approximately determine the net torques applied at anatomical joints that correspond to an observed motion.

In the context of robot control, using inverse dynamics requires knowledge of all contact forces acting on the robot; accurately perceiving external forces applied to the robot requires filtering and thus significant time delay. An alternative approach has been suggested in recent literature: predicting contact and actuator forces simultaneously under the assumptions of rigid body dynamics, rigid contact, and friction. Existing such inverse dynamics approaches have used approximations to the contact models, which permits use of fast numerical linear algebra algorithms. In contrast, we describe inverse dynamics methods that are derived only from first principles.

We assess these inverse dynamics approaches in a control context on a locomoting quadrupedal robot using perfectly accurate sensor data from simulation. The data collected from these experiments gives an upper bound on the performance of such controllers *in situ*. For points of comparison, we assess performance on the same tasks with both error feedback control and inverse dynamics control with virtual contact force sensing.

Samuel Zapolsky
George Washington University
Dept. of Computer Science
samzapo@gwu.edu

Evan Drumwright
George Washington University
Dept. of Computer Science
drum@gwu.edu

1 Introduction

Inverse dynamics can be a highly effective method in multiple contexts including control of robots and physically simulated characters and estimating muscle forces in biomechanics. The inverse dynamics problem, which entails computing actuator (or muscular) forces that yield desired accelerations, requires knowledge of all "external" forces acting on the robot, character, or human. Measuring such forces is limited by the ability to instrument surfaces and filter force readings, and such filtering effectively delays force sensing to a degree unacceptable for real-time operation. An alternative is to use contact force predictions, for which reasonable agreement between models and reality have been observed (see, *e.g.*, Aukes & Cutkosky, 2013). Formulating such approaches is technically challenging, however, because the actuator forces are generally coupled to the contact forces, requiring simultaneous solution.

Inverse dynamics approaches that simultaneously compute contact forces exist in literature, but these approaches use approximations to the rigid contact model. These approximations generally can provide reasonable performance, particularly on robots where multiple unmodeled phenomena may be present and limited system identification (for kinematic properties, inertial values, *etc.*) may have been conducted. However, this article focuses only on models that are derived from first principles, irrespective of implications on physical accuracy:

it is conceivable that a model *not* derived from first principles performs better on a particular robot.

It has also been claimed that the rigid contact model is not invertible (Todorov, 2014), implying that inverse dynamics is unsolveable for multi-rigid bodies subject to rigid contact. If forces on the multi-body *other than contact forces* at state { $\boldsymbol{q}, \dot{\boldsymbol{q}}$ } are designated $\boldsymbol{x}$ and contact forces are designated $\boldsymbol{y}$, then the rigid contact model (to be described in detail in Section 2.2.3) yields the relationship $\boldsymbol{y} = f_{\boldsymbol{q}, \dot{\boldsymbol{q}}}(\boldsymbol{x})$. It is then true that there exists no left inverse $g(.)$ of f that provides the mapping $\boldsymbol{x} = g_{\boldsymbol{q}, \dot{\boldsymbol{q}}}(\boldsymbol{y})$ for $\boldsymbol{y} = f_{\boldsymbol{q}, \dot{\boldsymbol{q}}}(\boldsymbol{x})$. However, this article will show that there does exist a right inverse $h(.)$ of f such that, for $\boldsymbol{h}_{\boldsymbol{q}, \dot{\boldsymbol{q}}}(\boldsymbol{y}) = \boldsymbol{x}$, $f_{\boldsymbol{q}, \dot{\boldsymbol{q}}}(\boldsymbol{x}) = \boldsymbol{y}$, and in Section 4 we will show that this mapping is computable in expected polynomial time. This article will use this mapping to formulate *ab initio* (derived from first principles) inverse dynamics approaches for rigid contact with both no-slip constraints and frictional surface properties.

The rigid contact model is known to be susceptible to the problem of contact indeterminacy, the presence of multiple equally valid solutions to the contact force-acceleration mapping. This indeterminacy is the factor that prevents strict invertibility and which, when used for contact force predictions in the context of inverse dynamics, can result in torque chatter that is potentially destructive for physically situated robots. We show that computing a mapping from accelerations to contact forces that evolves without harmful torque chatter is no worse than NP-hard in the number of contacts modeled for Coulomb friction and can be calculated in polynomial time for the case of infinite (no-slip) friction.

This article also describes a computationally tractable approach for mitigating torque chatter that is based upon a rigid contact model without complementarity conditions (see Sections 2.2.3 and 2.2.4). The model appears to produce reasonable predictions: Anitescu (2006); Drumwright & Shell (2010); Todorov (2014) have all used the model within simulation and physical artifacts have yet to become apparent.

We will assess these inverse dynamics methods in the context of controlling a virtual locomoting robot. We will examine performance of error feedback and inverse dynamics controllers with virtual contact force sensors for points of comparison. Performance will consider smoothness of torque commands, trajectory tracking accuracy, locomotion performance, and computation time.

This paper provides the following contributions:

1. Proof that the coupled problem of computing inverse dynamics-derived torques and contact forces under the rigid body dynamics, non-impacting rigid contact, and Coulomb friction models (with linearized friction cone) is solvable in expected polynomial time.
2. An algorithm that computes inverse dynamics-derived torques without torque chatter under the rigid body dynamics model and the rigid contact model *assuming no slip along the surfaces of contact*, in expected polynomial time.
3. An algorithm that yields inverse dynamics-derived torques without torque chatter under the rigid body dynamics model and the rigid, non-impacting contact model with Coulomb friction in exponential time in the number of points of contact, and hence a proof that this problem is no harder than NP-hard.
4. An algorithm that computes inverse dynamics-derived torques without torque chatter under the rigid body dynamics model and a rigid contact model with Coulomb friction but does not enforce complementarity conditions (again, see Sections 2.2.3 and 2.2.4), in expected polynomial time.

This work validates controllers based upon these approaches in simulation to determine their performance under a variety of measurable phenomena that would not be accessible on present day physically situated robot hardware. Experiments indicating performance differentials on present day (prototype quality) hardware might occur due to control forces exciting unmodeled effects, for example; legged robot systems are sufficiently sophisticated, yet brittle (from a maintenance standpoint), that isolating variables responsible for performance changes is presently infeasible. Results derived from simulation using ideal sensing and perfect torque control indicate the limit of performance for control software described in this work.

1.1 Contents

Section 2 describes background in rigid body dynamics and rigid contact, as well as related work in inverse dynamics with contact and friction. Section 3 presents an *ab initio* inverse dynamics formulation with contact force prediction that utilizes the non-impacting rigid contact model (to be described in Section 2.2) with no-slip frictional constraints. Section 4 presents an *ab initio* inverse dynamics formulation with contact force prediction that utilizes the rigid contact model with Coulomb friction constraints. We show that the problem of mitigating torque chatter from indeterminate contact configurations is no harder than NP-hard. Section 5 presents an *ab initio* inverse dynamics formulation that uses a rigid impact model and permits the contact force prediction problem to be convex. This convexity will allow us to mitigate torque chatter from

indeterminacy. Section 6 describes experimental setups for assessing the inverse dynamics formulations in the context of simulated robot control along multiple dimensions: accuracy of trajectory tracking; contact force prediction accuracy; general locomotion stability; and computational speed. Tests are performed on terrains with varied friction and compliance. Presented controllers are compared against both PID control and inverse dynamics control with sensed contact and perfectly accurate virtual sensors. Assessment under both rigid and compliant contact models permits both exact and in-the-limit verification that controllers implementing these inverse dynamics approaches for control work as expected. These experiments also examine behavior when modeling assumptions break down. Section 7 analyzes the findings from these experiments.

1.2 The multi-body

This paper centers around *a multi-body*, which is the system of rigid bodies to which inverse dynamics is applied. The multi-body may come into contact with "fixed" parts of the environment (*e.g.*, solid ground) which are sufficiently modeled as non-moving bodies. Alternatively, the multi-body may contact other bodies, in which case effective inverse dynamics will require knowledge of those bodies' kinematic and dynamic properties.

The articulated body approach can be extended to a multi-body to account for physically interacting with movable rigid bodies by appending the six degree-of-freedom velocities ($\boldsymbol{v}_{\mathrm{cb}}$) and external wrenches ($\boldsymbol{f}_{\mathrm{cb}}$) of each contacted rigid body to the velocity and external force vectors and by augmenting the generalized inertia matrix ($\mathbf{M}$) similarly:

$$\boldsymbol{v} = \left[\boldsymbol{v}_{\mathrm{robot}}{}^{\mathsf{T}} \; {}^{+}\boldsymbol{v}_{\mathrm{cb}}{}^{\mathsf{T}} \right]^{\mathsf{T}} \tag{1}$$

$$\boldsymbol{f}_{\mathrm{ext}} = \left[\boldsymbol{f}_{\mathrm{robot}}{}^{\mathsf{T}} \; \boldsymbol{f}_{\mathrm{cb}}{}^{\mathsf{T}} \right]^{\mathsf{T}} \tag{2}$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{\mathrm{robot}} & 0 \\ 0 & \mathbf{M}_{\mathrm{cb}} \end{bmatrix} \tag{3}$$

Without loss of generality, we will hereafter consider only a single multi-body in contact with a static environment.

2 Background and related work

This section surveys the independent parts that are combined to formulate algorithms for calculating inverse dynamics torques with simultaneous contact force computation. Section 2.1 introduces the rigid body dynamics model for Newtonian mechanics under generalized coordinates. Section 2.2 covers the rigid contact model, and unilaterally constrained contact. Sections 2.2.1–2.2.3 show how to formulate constraints on the rigid contact model to account for Coulomb friction and no-slip constraints. Section 2.2.4 describe an algebraic impact model that will form the basis of one of the inverse dynamics methods. Section 2.3 describes the phenomenon of "indeterminacy" in the rigid contact model. Lastly, Sections 2.4 and 2.5 discusses other work relevant to inverse dynamics with simultaneous contact force computation.

2.1 Rigid body dynamics

The multi rigid body dynamics (generalized Newton) equation governing the dynamics of a robot undergoing contact can be written in its generalized form as:

$$\mathbf{M}\ddot{\boldsymbol{q}} = \boldsymbol{f}_C + \mathbf{P}^{\mathsf{T}}\boldsymbol{\tau} + \boldsymbol{f}_{\mathrm{ext}} \tag{4}$$

Equation 4 introduces the variables $\boldsymbol{f}_{\mathrm{ext}} \in \mathbb{R}^m$ ("external", non-actuated based, forces on the m degree-of-freedom multibody, like gravity and Coriolis forces); $\boldsymbol{f}_C \in \mathbb{R}^m$ (contact forces); unknown actuator torques $\boldsymbol{\tau} \in \mathbb{R}^{nq}$ (nq is the number of actuated degrees of freedom); and a binary selection matrix $\mathbf{P} \in \mathbb{R}^{nq \times m}$. If all of the degrees-of-freedom of the system are actuated $\mathbf{P}$ will be an identity matrix. For, *e.g.*, legged robots, some variables in the system will correspond to unactuated, "floating base" degrees-of-freedom (DoF); the corresponding columns of the binary matrix $\mathbf{P} \in \mathbb{R}^{(m-6) \times m}$ will be zero vectors, while every other column will possess a single "1".

2.2 Rigid contact model

This section will summarize existing theory of modeling non-impacting rigid contact and draws from Stewart & Trinkle (1996); Trinkle et al. (1997); Anitescu & Potra (1997). Let us define a set of gap functions $\phi_i(\boldsymbol{x})$ (for $i = 1, \ldots, n$), where gap function i gives the signed distance between a feature on a link of the robot and a feature on another rigid body (another link of the robot, part of the environment, an object to be manipulated, *etc.*)

Our notation assumes independent coordinates $\boldsymbol{x}$ (and velocities $\boldsymbol{v}$ and accelerations $\dot{\boldsymbol{v}}$), and that generalized forces and inertias are also given in minimal coordinates.
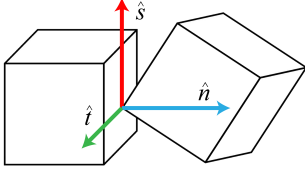
Fig. 1: *The contact frame consisting of $\hat{n}$, $\hat{s}$, and , $\hat{t}$ vectors corresponding to the normal, first tangential, and second tangential directions (for 3D) to the contact surface.*

The gap functions return a positive real value if the features are separated, a negative real value if one or both features lie inside the other rigid body, and zero if features are in a "kissing" configuration. The rigid contact model specifies that the bodies never overlap, *i.e.*:

$$\phi_i(\boldsymbol{x}) \geq 0 \quad \text{for } i = 1, \ldots, n \tag{5}$$

One or more points of contact between bodies is determined for two features in a kissing configuration ($\phi_i(\boldsymbol{x}) = 0$), yielding $n \leq q$ total points of contact. In the absence of friction, the constraints on the gap functions are enforced by forces that act along the contact normal. Projecting these forces along the contact normal yields scalars $f_{N_1}, \ldots, f_{N_n}$. The forces should be compressive (*i.e.*, forces that can not pull bodies together), which is denoted by restricting the sign of these scalars to be non-negative:

$$f_{N_i} \geq 0 \quad \text{for } i = 1, \ldots, n \tag{6}$$

A *complementarity* constraint keeps frictional contacts from doing work: when the constraint is inactive ($\phi_i > 0$) no force is applied and when force is applied, the constraint must be active ($\phi_i = 0$). This constraint is expressed mathematically as $\phi_i \cdot f_{N_i} = 0$. All three constraints can be combined into one equation using the following notation:

$$0 \leq f_{N_i} \perp \phi_i(\boldsymbol{x}) \geq 0 \quad \text{for } i = 1, \ldots, n \tag{7}$$

These constraints can be differentiated with respect to time to yield velocity-level or acceleration-level constraints suitable for expressing the differential algebraic equations (DAEs), as an index 1 DAE:

$$0 \leq f_{N_i} \perp \dot{\phi}_i(\boldsymbol{x}) \geq 0 \quad \text{for } i = 1, \ldots, n \tag{8}$$

$$0 \leq f_{N_i} \perp \ddot{\phi}_i(\boldsymbol{x}) \geq 0 \quad \text{for } i = 1, \ldots, n \tag{9}$$

2.2.1 Modeling Coulomb friction

Dry friction is often simulated using the Coulomb model, a relatively simple, empirically derived model that yields the approximate outcome of sophisticated physical interactions. Coulomb friction covers two regimes: sticking/rolling friction (where the tangent velocity at a point of contact is zero) and sliding friction (nonzero tangent velocity at a point of contact). Rolling friction is distinguished from sticking friction by whether the bodies are moving relative to one another other than at the point of contact.

Coulomb friction uses a unitless friction coefficient, commonly denoted μ. If we define the tangent velocities and accelerations in 3D frames (located at the ith point of contact) as v_{S_i}/v_{T_i} and a_{S_i}/a_{T_i}, respectively, and the tangent forces as f_{S_i} and f_{T_i}, then the sticking/rolling constraints which are applicable exactly when $0 = v_{S_i} = v_{T_i}$, can be expressed via the Fritz-John optimality conditions (Mangasarian & Fromovitz, 1967):

$$0 \leq \mu_i^2 f_{N_i}^2 - f_{S_i}^2 - f_{T_i}^2 \perp a_{S_i}^2 + a_{T_i}^2 \geq 0 \tag{10}$$

$$\mu_i f_{N_i} a_{S_i} + f_{S_i} \sqrt{a_{S_i}^2 + a_{T_i}^2} = 0 \tag{11}$$

$$\mu_i f_{N_i} a_{T_i} + f_{T_i} \sqrt{a_{S_i}^2 + a_{T_i}^2} = 0 \tag{12}$$

These conditions ensure that the friction force lies within the friction cone (Equation 10) and that the friction forces push against the tangential acceleration (Equations 11–12).

In the case of sliding at the ith contact ($v_{S_i} \neq 0$ or $v_{T_i} \neq 0$), the constraints become:

$$\mu_i^2 f_{N_i} - f_{S_i}^2 - f_{T_i}^2 \geq 0 \tag{13}$$

$$\mu_i f_{N_i} v_{S_i} + f_{S_i} \sqrt{v_{S_i}^2 + v_{T_i}^2} = 0 \tag{14}$$

$$\mu_i f_{N_i} v_{T_i} + f_{T_i} \sqrt{v_{S_i}^2 + v_{T_i}^2} = 0 \tag{15}$$

Note that this case is only applicable if $v_{S_i}^2 + v_{T_i}^2 > 0$, so there is no need to include such a constraint in Equation 13 (as was necessary in Equation 10).

The rigid contact model with Coulomb friction is subject to *inconsistent configurations* (Stewart, 2000a), exemplified by Painlevé's Paradox (Painlevé, 1895), in which impulsive forces may be necessary to satisfy all constraints of the model *even when the bodies are not impacting*. The acceleration-level dynamics can be approximated using finite differences; a first order approximation is often used (see, *e.g.*, Posa & Tedrake, 2012), which moves the problem to the velocity/impulsive force domain. Such an approach generally uses an *algebraic collision law* (see Chatterjee & Ruina, 1998)

to model all contacts, both impacting, as inelastic impacts; typical "time stepping methods" (Moreau, 1983) for simulating dynamics often treat the generalized coordinates as constant over the small timespan of contact/impact (*i.e.*, a first order approximation); see, *e.g.*, Stewart & Trinkle (2000).

Upon moving to the velocity/impulsive force domain, Equations 10–15 require a slight transformation to the equations:

$$0 \leq \mu_i^2 f_{N_i}^2 - f_{S_i}^2 - f_{T_i}^2 \perp v_{S_i}^2 + v_{T_i}^2 \geq 0 \tag{16}$$

$$\mu f_{N_i} v_{S_i} + f_{S_i} \sqrt{v_{S_i}^2 + v_{T_i}^2} = 0 \tag{17}$$

$$\mu f_{N_i} v_{T_i} + f_{T_i} \sqrt{v_{S_i}^2 + v_{T_i}^2} = 0 \tag{18}$$

and there is no longer separate consideration of sticking/rolling and slipping contacts.

2.2.2 No-slip constraints

If the Coulomb friction constraints are replaced by no-slip constraints, which is a popular assumption in legged locomotion research, one must also use the discretization approach; without permitting impulsive forces, slip can occur even with infinite friction (Lynch & Mason, 1995). The no-slip constraints are then simply $v_{S_i} = v_{T_i} = 0$ (replacing Equations 16–18).

2.2.3 Model for rigid, non-impacting contact with Coulomb friction

The model of rigid contact with Coulomb friction for two bodies in non-impacting rigid contact at $\boldsymbol{p}$ can be summarized by the following equations:

$$0 \leq f_n \perp a_n \geq 0 \tag{19}$$

$$0 \leq \mu^2 f_n^2 - f_s^2 - f_t^2 \perp \sqrt{v_s^2 + v_t^2} \geq 0 \tag{20}$$

$$0 = \mu f_n v_s + f_s \sqrt{v_s^2 + v_t^2} \tag{21}$$

$$0 = \mu f_n v_t + f_t \sqrt{v_s^2 + v_t^2} \tag{22}$$

$$0 = \mu f_n a_s + f_s \sqrt{a_s^2 + a_t^2} \tag{23}$$

$$0 = \mu f_n a_t + f_t \sqrt{a_s^2 + a_t^2} \tag{24}$$

where f_n, f_s, and f_t are the signed magnitudes of the contact force applied along the normal and two tangent directions, respectively; a_n is the relative acceleration of the bodies normal to the contact surface; and v_s and v_t are the relative velocities of the bodies projected along the two tangent directions. The operator $\perp$ indicates that $a \cdot b = 0$, for scalars a and b. Detailed interpretation of these equations can be found

in Trinkle et al. (1997); we present a summary below. Equation 19 ensures that (1) only compressive forces are applied ($f_n \geq 0$); (2) bodies cannot interpenetrate ($a_n \geq 0$); and (3) no work is done for frictionless contacts ($f_n \cdot a_n = 0$). Equation 20 models Coulomb friction: either the velocity in the contact tangent plane is zero—which allows frictional forces to lie within the friction cone—or the contact is slipping and the frictional forces must lie strictly on the edge of the friction cone. Equations 21 and 22—applicable to sliding contacts (*i.e.*, those for which $v_s \neq 0$ or $v_t \neq 0$)—constrain friction forces to act against the direction of slip, while Equations 23 and 24 constrain frictional forces for rolling or sticking contacts (*i.e.*, those for which $v_s = v_t = 0$) to act against the direction of tangential acceleration.

These equations form a nonlinear complementarity problem (Cottle et al., 1992), and this problem may not possess a solution with nonimpulsive forces due to the existence of inconsistent configurations like Painlevé's Paradox (Stewart, 2000b). This issue led to the movement to the impulsive force/velocity domain for modeling rigid contact, which can provably model the dynamics of such inconsistent configurations.

A separate issue with the rigid contact model is that of indeterminacy, where multiple sets of contact forces and possibly multiple sets of accelerations—or velocities, if an impulse-velocity model is used—can satisfy the contact model equations. Our inverse dynamics approaches, which use rigid contact models, address inconsistency and, given some additional computation, can address indeterminacy (useful for controlled systems).

2.2.4 Polynomial time computable simultaneous rigid impact model

Anitescu (2006) described a contact model without complementarity (Equation 7) used for multi-rigid body simulation. Drumwright & Shell (2010) and Todorov (2014) rediscovered this model (albeit with slight modifications, addition of viscous friction, and guarantees of solution existence and non-negativity energy dissipation in the former); Drumwright & Shell (2010) also motivated acceptability of removing the complementarity condition based on work by Chatterjee (1999). If treated as a simultaneous impact model[1], the model is

---

[1] In simulation, at least, impact models are often used for non-impacting contact and *vice versa*. This use can be at least somewhat justified by considering that distinguishing between impacting and non-impacting contact using floating-point arithmetic requires selecting an arbitrary tolerance value.

consistent with first principles. Additionally, using arguments in Smith et al. (2012), it can be shown that solutions of this model exhibit symmetry. This impact model, under the assumption of inelastic impact—it is possible to model partially or fully elastic impact as well, but one must then consider numerous models of restitution, see, *e.g.*, Chatterjee & Ruina (1998)—will form the basis of the inverse dynamics approach described in Section 5.

The model is formulated as the convex quadratic program below. For consistency of presentation with the non-impacting rigid model described in the previous section, only a single impact point is considered.

Complementarity-free impact model (single point of contact)

dissipate kinetic energy maximally:

$$\underset{^{+}\boldsymbol{v},\boldsymbol{f}_n,\boldsymbol{f}_s,\boldsymbol{f}_t}{\text{minimize}} \quad \frac{1}{2}{}^{+}\boldsymbol{v}^{\mathsf{T}}\mathbf{M}\,{}^{+}\boldsymbol{v} \tag{25}$$

non−interpenetration:

$$\text{subject to: } \mathbf{n}\,{}^{+}\boldsymbol{v} \geq \mathbf{0} \tag{26}$$

compressive normal forces

$$\boldsymbol{f}_n \geq \mathbf{0} \tag{27}$$

Coulomb friction:

$$\mu^2 \boldsymbol{f}_n \geq \boldsymbol{f}_s + \boldsymbol{f}_t \tag{28}$$

first−order dynamics:

$$^{+}\boldsymbol{v} = {}^{-}\boldsymbol{v} + \mathbf{M}^{-1}(\mathbf{n}^{\mathsf{T}}\boldsymbol{f}_n + \mathbf{s}^{\mathsf{T}}\boldsymbol{f}_s + \mathbf{t}^{\mathsf{T}}\boldsymbol{f}_t) \tag{29}$$

---

where f_n, f_s, and f_t are the signed magnitudes of the impulsive forces applied along the normal and two tangent directions, respectively; $^{-}\boldsymbol{v} \in \mathbb{R}^m$ and $^{+}\boldsymbol{v} \in \mathbb{R}^m$ are the generalized velocities of the multi-body immediately before and after impact, respectively; $\mathbf{M} \in \mathbb{R}^{m \times m}$ is the generalized inertia matrix of the m degree of freedom multi-body; and $\boldsymbol{n} \in \mathbb{R}^m$, $\boldsymbol{s} \in \mathbb{R}^m$, and $\boldsymbol{t} \in \mathbb{R}^m$ are the Jacobian matrices that transform the multi-body's generalized velocity to the velocity at the point of contact, projected along the normal and two tangential directions at the contact surface, respectively (see Appendix A for further details on these matrices).

The physical interpretation of the impact model is straightforward: it selects impact forces that maximize the rate of kinetic energy dissipation. Finally, we note that rigid impact models do not enjoy the same degree of community consensus as the non-impacting rigid contact models because three types of impact models (algebraic, incremental, and full deformation) currently exist (Chatterjee & Ruina, 1998), because simultaneous impacts and impacts between multi-bodies can

be highly sensitive to initial conditions (Ivanov, 1995), and because intuitive physical parameters for capturing all points of the feasible impulse space do not yet exist (Chatterjee & Ruina, 1998), among other issues. These difficulties lead this article to consider consider only inelastic impacts, a case for which the feasible impulse space is constrained.

2.3 Contact force indeterminacy

In previous work (Zapolsky et al., 2013), we found that indeterminacy in the rigid contact model can be a significant problem for controlling quadrupedal robots (and, presumably, hexapods, *etc.*) by yielding torques that switch rapidly between various actuators. The problem can also occur for real bipedal walkers; for example, Collins et al. (2001) observed instability from rigid contact indeterminacy in passive walkers.

Rigid contact configurations can be indeterminate in terms of forces; for the example of a table with all four legs perfectly touching a ground plane, infinite enumerations of force configurations satisfy the contact model (as discussed in Mirtich, 1996), although the accelerations predicted by the model are unique. Other rigid contact configurations can be indeterminate in terms of predicting different accelerations/velocities through multiple sets of valid force configurations.

2.4 Contact models for inverse dynamics in the context of robot control

This section focuses on "hard", by which we mean perfectly rigid, models of contact incorporated into inverse dynamics and whole body control for robotics. We are unaware of research that has attempted to combine inverse dynamics with compliant contact (one possible reason for absence of such work is that such compliant models can require significant parameter tuning for accuracy and to prevent prediction of large contact forces).

Mistry et al. (2010) developed a fast inverse dynamics control framework for legged robots in contact with rigid environments under the assumptions that feet do not slip or accelerate upward. Righetti et al. (2013) extended this work with a framework that permits quickly optimizing a mixed linear/quadratic function of motor torques and contact forces using fast linear algebra techniques. Hutter & Siegwart (2012) also uses this formulation in an operational space control scheme, simplifying the contact mathematics by assuming contacts are sticking. These works do not incorporate unilateral contact constraints, and thereby permit prediction

of tensile (adhesive) forces that pull along the contact normals and "dead" collisions (the latter shown to diverge from expected impact modeling by Chatterjee & Ruina). Despite divergence from the rigid contact model, Mistry et al.; Righetti et al.; Hutter & Siegwart demonstrate effective trajectory tracking performance on quadrupedal robots.

The inverse dynamics approach of Ames (2013) assumes sticking impact upon contact with the ground and immediate switching of support to the new contact, while enforcing a unilateral constraint of the normal forces and predicting no-slip frictional forces.

Stephens & Atkeson (2010) incorporate a contact model into an inverse dynamics formulation for dynamic balance force control. Their approach uses a quadratic program (QP) to estimate contact forces quickly on a simplified model of a bipedal robot's dynamics. Newer work by Feng et al. (2013) builds on this by approximating the friction cone with a circumscribed friction pyramid. In similar fashion to Mistry et al. (2010), forces along the contact normals are not constrained to be compressive.

Ott et al. (2011) also use an optimization approach for balance, modeling contact to distribute forces among a set of pre-defined contacts to enact a wrench on a simplified model of a biped; their controller seeks to minimize the Euclidian norm of the predicted contact forces to mitigate slip. In underconstrained cases (where multiple solutions to the inverse dynamics with contact system exist), Saab et al. (2013) and Zapolsky et al. (2013) use a multi-phase QP formulation for bipeds and quadrupeds, respectively. Zapolsky et al. mitigates the indeterminacy in the rigid contact model by selecting a solution that minimizes total actuator torques, while Saab et al. use the rigid contact model in the context of cascades of QPs to perform several tasks in parallel (*i.e.*, whole body control). The latter work primarily considers contacts without slip, but does describe modifications that would incorporate Coulomb friction (inconsistent and indeterminate rigid contact configurations are not addressed). Todorov (2014) uses the same contact model (to be described below) but without using a two-stage solution; that approach uses regularization to make the optimization problem strictly convex (yielding a single, globally optimal solution). None of Saab et al.; Zapolsky et al.; Todorov utilize the *complementarity constraint* (*i.e.*, $f_N \perp \phi$ in Equation 7) in their formulation. Zapolsky *et al.* and Todorov motivate dropping this constraint in favor of maximizing energy dissipation through contact, an assumption that they show performs reasonably in practice (Drumwright & Shell, 2010; Todorov, 2011).

Kuindersma et al. (2014) use a no-slip constraint but allow for bounded violation of that constraint in order to avoid optimizing over an infeasible or inconsistent trajectory.

2.5 Contact models for inverse dynamics in the context of biomechanics

Inverse dynamics is commonly applied in biomechanics to determine approximate net torques at anatomical joints for observed motion capture and force plate data. Standard Newton-Euler inverse dynamics algorithms (as described in Featherstone, 2008) are applied; least squares is required because the problem is overconstrained. Numerous such approaches are found in biomechanics literature, including (Kuo, 1998; Hatze, 2002; Blajer et al., 2007; Bisseling & Hof, 2006; Yang et al., 2007; Van Den Bogert & Su, 2008; Sawers & Hahn, 2010). These force plate based approaches necessarily limit the environments in which the inverse dynamics computation can be conducted.

3 Inverse dynamics with no-slip constraints

Some contexts where inverse dynamics may be used (biomechanics, legged locomotion) may assume absence of slip (see, *e.g.*, Righetti, et al., 2011; Zhao, et al., 2014). This section describes an inverse dynamics approach that computes reaction forces from contact using the non-impacting rigid contact model with no-slip constraints. Using no-slip constraints results in a symmetric, positive semidefinite LCP. Such problems are equivalent to convex QPs by duality theory in optimization, which implies polynomial time solvability. Convexity also permits mitigating indeterminate contact configurations, as will be seen in Section 3.6. This formulation inverts the rigid contact problem in a practical sense and is derived from first principles.

We present two algorithms in this section: Algorithm 1 ensures the no-slip constraints on the contact model are non-singular and thus guarantees that the inverse dynamics problem with contact is invertible; Algorithm 2 presents a method of mitigating torque chatter from indeterminate contact (for contexts of inverse dynamics based control) by *warm-starting* (Nocedal & Wright, 2006) the solution for the LCP solver with the last solution.

As broached in Section 2.2.1, dynamics are discretized using a first order approximation to acceleration to avoid inconsistent contact configurations. Thus, the solution to the equation of motion $\dot{\boldsymbol{v}} = \mathbf{M}^{-1}\boldsymbol{f}$ over $[t_0, t_f]$ is approximated by $^+\boldsymbol{v} = {}^-\boldsymbol{v} + \Delta t\, ^-\mathbf{M}^{-1}\, ^-\boldsymbol{f}$, where $\Delta t =$

$(t_f - t_0)$. We use the superscript "+" to denote that a term is evaluated at t_f and the superscript "−" is applied to denote that a term is computed at t_0. For example, the generalized inertia matrix $\mathbf{M}$ has the superscript "−" and the generalized post-contact velocity ($^{+}\boldsymbol{v}$) has the superscript "+". We will hereafter adopt the convention that application of a superscript once will indicate implicit evaluation of that quantity at that time thereafter (unless another superscript is applied). For example, we will continue to treat $\mathbf{M}$ as evaluated at t_0 in the remainder of this paper.

3.1 Normal contact constraints

The equation below extends Equation 8 to multiple points of contact (via the relationship $\dot{\boldsymbol{\phi}} = \mathbf{N}\boldsymbol{v}$), where $\mathbf{N} \in \mathbb{R}^n$ is the matrix of generalized wrenches along the contact normals (see Appendix A):

$$0 \le {}^{-}\mathbf{N}\,{}^{+}\boldsymbol{v} \perp f_N \ge \mathbf{0} \tag{30}$$

Because $\boldsymbol{\phi}$ is clearly time-dependent and the control loop executes at discrete points in time, $\mathbf{N}$ must be treated as constant over a time interval. ${}^{-}\mathbf{N}$ indicates that points of contact are drawn from the current configuration of the environment and multi-body. Analogous to time-stepping approaches for rigid body simulations with rigid contact, *all possible points of contact between rigid bodies over the interval t_0 and t_f can be incorporated into $\mathbf{N}$ as well*: as in time stepping approaches for simulation, it may not be necessary to apply forces at all of these points (the approaches implicitly can treat unnecessary points of contact as inactive, though additional computation will be necessary). Stewart (1998) showed that such an approach will converge to the solution of the continuous time dynamics as $\Delta t = (t_f - t_0) \to 0$. Given a sufficiently high control rate, Δt will be small and errors from assuming constant $\mathbf{N}$ over this interval should become negligible.

3.2 Discretized rigid body dynamics equation

The discretized version of Equation 4, now separating contact forces into normal ($^{-}\mathbf{N}$) and tangential wrenches ($^{-}\mathbf{S}$ and $^{-}\mathbf{T}$ are matrices of generalized wrenches along the first and second contact tangent directions for all contacts) is:

$$\mathbf{M}(^{+}\boldsymbol{v} - {}^{-}\boldsymbol{v}) = \mathbf{N}^\mathsf{T}\boldsymbol{f}_N + \mathbf{S}^\mathsf{T}\boldsymbol{f}_S + \mathbf{T}^\mathsf{T}\boldsymbol{f}_T - \ldots \tag{31}$$
$$\mathbf{P}^\mathsf{T}\boldsymbol{\tau} + \Delta t\,{}^{-}\boldsymbol{f}_{\text{ext}}$$

Treating inverse dynamics at the velocity level is necessary to avoid the inconsistent configurations that can arise in the rigid contact model when forces are required to be non-impulsive (Stewart, 2000a, as also noted in Section 2.2.1). As noted above, Stewart has shown that for sufficiently small Δt, $^{+}\boldsymbol{v}$ converges to the solution of the continuous time dynamics and contact equations (1998).

3.3 Inverse dynamics constraint

The inverse dynamics constraint is used to specify the desired velocities only at actuated joints:

$$\mathbf{P}\,{}^{+}\boldsymbol{v} = \dot{\boldsymbol{q}}_{\text{des}} \tag{32}$$

Desired velocities $\dot{\boldsymbol{q}}_{\text{des}}$ are calculated as:

$$\dot{\boldsymbol{q}}_{\text{des}} \equiv \dot{\boldsymbol{q}} + \Delta t \ddot{\boldsymbol{q}}_{\text{des}} \tag{33}$$

3.4 No-slip (infinite friction) constraints

Utilizing the first-order discretization (revisit Section 2.2.2 to see why this is necessary), preventing tangential slip at a contact is accomplished by using the constraints:

$$\mathbf{S}\,{}^{+}\boldsymbol{v} = \mathbf{0} \tag{34}$$
$$\mathbf{T}\,{}^{+}\boldsymbol{v} = \mathbf{0} \tag{35}$$

These constraints indicate that the velocity in the tangent plane is zero at time t_f. The equations are compiled below:

Complementarity-based inverse dynamics without slip

non−interpenetration, compressive force, and

normal complementarity constraints:

$$0 \le \boldsymbol{f}_N \perp \mathbf{N}\,{}^{+}\boldsymbol{v} \ge \mathbf{0}$$

no−slip constraints:

$$\mathbf{S}\,{}^{+}\boldsymbol{v} = \mathbf{0}$$

$$\mathbf{T}\,{}^{+}\boldsymbol{v} = \mathbf{0}$$

inverse dynamics:

$$\mathbf{P}\,{}^{+}\boldsymbol{v} = \dot{\boldsymbol{q}}_{\text{des}}$$

first−order dynamics:

$$^{+}\boldsymbol{v} = {}^{-}\boldsymbol{v} + \mathbf{M}^{-1}(\mathbf{N}^\mathsf{T}\boldsymbol{f}_N + \mathbf{S}^\mathsf{T}\boldsymbol{f}_S + \ldots$$
$$\mathbf{T}^\mathsf{T}\boldsymbol{f}_T - \mathbf{P}^\mathsf{T}\boldsymbol{\tau} + \Delta t\boldsymbol{f}_{ext})$$

Combining Equations 30–32, 34, and 35 into a *mixed linear complementarity problem* (MLCP, see Appendix B) yields:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{P}^\mathsf{T} & -\mathbf{S}^\mathsf{T} & -\mathbf{T}^\mathsf{T} & -\mathbf{N}^\mathsf{T} \\ \mathbf{P} & 0 & 0 & 0 & 0 \\ \mathbf{S} & 0 & 0 & 0 & 0 \\ \mathbf{T} & 0 & 0 & 0 & 0 \\ \mathbf{N} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} {}^+\!\boldsymbol{v} \\ \boldsymbol{\tau} \\ \boldsymbol{f}_S \\ \boldsymbol{f}_T \\ \boldsymbol{f}_N \end{bmatrix} + \begin{bmatrix} \boldsymbol{\kappa} \\ -\dot{\boldsymbol{q}}_{\text{des}} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \boldsymbol{w}_N \end{bmatrix} \quad (36)$$

$$\boldsymbol{f}_N \geq \mathbf{0}, \boldsymbol{w}_N \geq \mathbf{0}, \boldsymbol{f}_N^\mathsf{T} \boldsymbol{w}_N = 0 \tag{37}$$

where $\boldsymbol{\kappa} \triangleq -\Delta t \boldsymbol{f}_{\text{ext}} - \mathbf{M}^-\boldsymbol{v}$. We define the MLCP block matrices—in the form of Equations 110–113 from Appendix B—and draw from Equations 36 and 37 to yield:

$$\mathbf{A} \equiv \begin{bmatrix} \mathbf{M} & -\mathbf{P}^\mathsf{T} & -\mathbf{S}^\mathsf{T} & -\mathbf{T}^\mathsf{T} \\ \mathbf{P} & 0 & 0 & 0 \\ \mathbf{S} & 0 & 0 & 0 \\ \mathbf{T} & 0 & 0 & 0 \end{bmatrix} \quad \mathbf{C} \equiv \begin{bmatrix} -\mathbf{N}^\mathsf{T} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{D} \equiv -\mathbf{C}^\mathsf{T} \qquad\qquad \mathbf{B} \equiv \mathbf{0}$$

$$\boldsymbol{x} \equiv \begin{bmatrix} {}^+\!\boldsymbol{v} \\ \boldsymbol{\tau} \\ \boldsymbol{f}_S \\ \boldsymbol{f}_T \end{bmatrix} \qquad \boldsymbol{g} \equiv \begin{bmatrix} -\boldsymbol{\kappa} \\ \dot{\boldsymbol{q}}_{\text{des}} \\ 0 \\ 0 \end{bmatrix}$$

$$\boldsymbol{y} \equiv \boldsymbol{f}_N \qquad\qquad \boldsymbol{h} \equiv \mathbf{0}$$

Applying Equations 115 and 116 (again see Appendix B), we transform the MLCP to LCP $(\boldsymbol{r}, \mathbf{Q})$. Substituting in variables from the no-slip inverse dynamics model and then simplifying yields:

$$\mathbf{Q} \equiv \mathbf{C}^\mathsf{T} \mathbf{A}^{-1} \mathbf{C} \tag{38}$$

$$\boldsymbol{r} \equiv \mathbf{C}^\mathsf{T} \mathbf{A}^{-1} \boldsymbol{g} \tag{39}$$

The definition of matrix $\mathbf{A}$ from above may be singular, which would prevent inversion, and thereby, conversion from the MLCP to an LCP. We defined $\mathbf{P}$ as a selection matrix with full row rank, and the generalized inertia ($\mathbf{M}$) is symmetric, positive definite. If $\mathbf{S}$ and $\mathbf{T}$ have full row rank as well, or we identify the largest subset of row blocks of $\mathbf{S}$ and $\mathbf{T}$ such that full row rank is attained, $\mathbf{A}$ will be invertible as well (this can be seen by applying blockwise matrix inversion identities). Algorithm 1 performs the task of ensuring that matrix $\mathbf{A}$ is invertible.

From Bhatia (2007), a matrix of $\mathbf{Q}$'s form must be non-negative definite, *i.e.*, either positive-semidefinite (PSD) or positive definite (PD). $\mathbf{Q}$ is the right product of $\mathbf{C}$ with its transpose about a symmetric PD matrix, $\mathbf{A}$. Therefore, $\mathbf{Q}$ is symmetric and either PSD or PD.

The singularity check on Lines 6 and 10 of Algorithm 1 is most quickly performed using Cholesky factorization; if the factorization is successful, the matrix is non-singular. Given that $\mathbf{M}$ is non-singular (it is symmetric, PD), the maximum size of $\mathbf{X}$ in Algorithm 1 is $m \times m$; if $\mathbf{X}$ were larger, it would be singular.

The result is that the time complexity of Algorithm 1 is dominated by Lines 6 and 10. As $\mathbf{X}$ changes by at most one row and one column per Cholesky factorization, singularity can be checked by $O(m^2)$ updates to an initial $O(m^3)$ Cholesky factorization. The overall time complexity is $O(m^3 + nm^2)$.

Algorithm 1 FIND-INDICES($\mathbf{M}, \mathbf{P}, \mathbf{S}, \mathbf{T}$), determines the row indices ($\mathcal{S}$, and $\mathcal{T}$) of $\mathbf{S}$ and $\mathbf{T}$ such that the matrix $\mathbf{A}$ (Equation 110 in Appendix B) is non-singular.

1: $\mathcal{S} \leftarrow \emptyset$
2: $\mathcal{T} \leftarrow \emptyset$
3: **for** $i = 1, \dots, n$ **do**   ▷ n is the number of contacts
4:     $\mathcal{S}^* \leftarrow \mathcal{S} \cup \{i\}$
5:     Set $\mathbf{X} \leftarrow \begin{bmatrix} \mathbf{P}^\mathsf{T} & \mathbf{S}_{\mathcal{S}^*}^\mathsf{T} & \mathbf{T}_{\mathcal{T}}^\mathsf{T} \end{bmatrix}$
6:     **if** $\mathbf{X}^\mathsf{T} \mathbf{M}^{-1} \mathbf{X}$ not singular **then**
7:         $\mathcal{S} \leftarrow \mathcal{S}^*$
8:     $\mathcal{T}^* \leftarrow \mathcal{T} \cup \{i\}$
9:     Set $\mathbf{X} \leftarrow \begin{bmatrix} \mathbf{P}^\mathsf{T} & \mathbf{S}_{\mathcal{S}}^\mathsf{T} & \mathbf{T}_{\mathcal{T}^*}^\mathsf{T} \end{bmatrix}$
10:     **if** $\mathbf{X}^\mathsf{T} \mathbf{M}^{-1} \mathbf{X}$ not singular **then**
11:         $\mathcal{T} \leftarrow \mathcal{T}^*$
12: **return** $\{\mathcal{S}, \mathcal{T}\}$

3.5 Retrieving the inverse dynamics forces

Once the contact forces have been determined, one solves Equations 31 and 32 for $\{{}^+\!\boldsymbol{v}, \boldsymbol{\tau}\}$, thereby obtaining the inverse dynamics forces. While the inverse dynamics problem is solvable, it is possible that the desired accelerations are inconsistent. As an example, consider a legged robot standing on a ground plane without slip (such a case is similar to, but not identical to infinite friction, as noted in Section 2.2.2), and attempting to splay its legs outward while remaining in contact with the ground. Such cases can be readily identified by verifying that $\mathbf{N}^+\!\boldsymbol{v} \geq \mathbf{0}$. If this constraint is not met, consistent desired accelerations can be determined *without re-solving the LCP*. For example, one could determine accelerations that deviate minimally from the desired accelerations by solving a quadratic program:

$$\underset{{}^+\!\boldsymbol{v}, \boldsymbol{\tau}}{\text{minimize}} \; ||\mathbf{P}^+\!\boldsymbol{v} - \dot{\boldsymbol{q}}_{\text{des}}|| \tag{40}$$

$$\text{subject to: } \mathbf{N}^+\!\boldsymbol{v} \geq \mathbf{0} \tag{41}$$

$$\mathbf{S}^+\!\boldsymbol{v} = \mathbf{0} \tag{42}$$

$$\mathbf{T}^+\!\boldsymbol{v} = \mathbf{0} \tag{43}$$

$$\mathbf{M}^+\!\boldsymbol{v} = \mathbf{M}^-\boldsymbol{v} + \mathbf{N}^\mathsf{T} \boldsymbol{f}_N + \mathbf{S}^\mathsf{T} \boldsymbol{f}_S + \dots \tag{44}$$

$$\mathbf{T}^\mathsf{T} \boldsymbol{f}_T + \mathbf{P}^\mathsf{T} \boldsymbol{\tau} + \Delta t \boldsymbol{f}_{\text{ext}} \tag{45}$$

This QP is always feasible: $\boldsymbol{\tau} = \mathbf{0}$ ensures that $\mathbf{N}^+\!\boldsymbol{v} \geq \mathbf{0}$, $\mathbf{S}^+\!\boldsymbol{v} = \mathbf{0}$, and $\mathbf{T}^+\!\boldsymbol{v} = \mathbf{0}$.

3.6 Indeterminacy mitigation

We warm start a pivoting LCP solver (see Appendix C) to bias the solver toward applying forces at the same points of contact (see Figure 2)—tracking points of contact using the rigid body equations of motion—as were active on the previous inverse dynamics call (see Algorithm 2). Kuindersma et al. (2014) also use warm starting to solve a different QP resulting from contact force prediction. However, we use warm starting to address indeterminacy in the rigid contact model while Kuindersma et al. use it to generally speed the solution process.

Fig. 2: Warm-Starting Example

Iteration i Iteration $i+1$ Iteration $i+2$
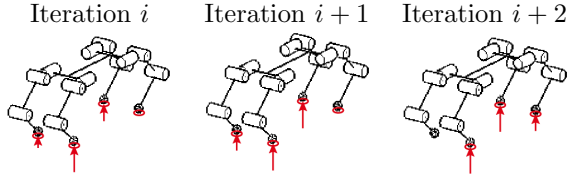
Left ("cold start"): with four active contacts, the pivoting solver chooses three arbitrary *non-basic* indices (in $\overline{\beta}$, see Appendix C) to solve the LCP and then returns the solution. The solution applies the majority of upward force to two feet and applies a smaller amount of force to the third.
Center ("warm start"): With four active contacts, the pivoting solver chooses the *same* three non-basic indices as the last solution to attempt to solve the LCP. The warm-started solution will distribute upward forces similarly to the last solution, tending to provide consecutive solves with continuity over time.
Right ("cold start"): one foot of the robot has broken contact with the ground; there are now three active contacts. The solver returns a new solution, applying the majority of normal force to two legs, and applying a small amount of force to the third.

Using warm starting, Algorithm 2 will find a solution that predicts contact forces applied at the exact same points on the last iteration *assuming that such a solution exists*. Such solutions do not exist (1) when the numbers and relative samples from the contact manifold change or (2) as contacts become active ($\dot{\phi}_i(\boldsymbol{x}, \boldsymbol{v}) \leq 0$) or inactive ($\dot{\phi}_i(\boldsymbol{x}, \boldsymbol{v}) > 0$). Case (2) implies immediate precedence or subsequence of case (1), which means that discontinuities in actuator torques will occur for at most two control loop iterations around a contact change (one discontinuity will generally occur due to the contact change itself).

3.7 Scaling inverse dynamics runtime linearly in number of contacts

The multi-body's number of generalized coordinates (m) are expected to remain constant. The number of contact points, n, depends on the multi-body's link geometries, the environment, and whether the inverse dynamics approach should anticipate *potential* contacts in $[t_0, t_f]$ (as discussed in Section 3.1). This section describes a method to solve inverse dynamics problems with simultaneous contact force computation that scales linearly with additional contacts. This method will be applicable to all inverse dynamics approaches presented in this paper except that described in Section 4: that problem results in a copositive LCP (Cottle et al., 1992) that the algorithm we describe cannot generally solve.

To this point in the article presentation, time complexity has been dominated by the $O(n^3)$ expected time solution to the LCPs. However, a system with m degrees-of-freedom requires no more than m positive force magnitudes applied along the contact normals to satisfy the constraints for the no-slip contact model. Proof is provided in Appendix D. Below, we describe how that proof can be leveraged to generally decrease the expected time complexity.

Modified PPM I Algorithm We now describe a modification to the Principle Pivoting Method I (Cottle et al., 1992) (PPM) for solving LCPs (see description of this algorithm in Appendix C) that leverages the proof in Appendix D to attain expected $O(m^3 + nm^2)$ time complexity. A brief explanation of the mechanics of pivoting algorithms is provided in Appendix C; we use the common notation of β as the set of *basic variables* and $\overline{\beta}$ as the set of *non-basic variables*.

The PPM requires few modifications toward our purpose. These modifications are presented in Algorithm 2. First, the full matrix $\mathbf{N}\mathbf{M}^{-1}\mathbf{N}^{\mathsf{T}}$ is never constructed, because the construction is unnecessary and would require $O(n^3)$ time. Instead, Line 11 of the algorithm constructs a maximum $m \times m$ system; thus, that operation requires only $O(m^3)$ operations. Similarly, Lines 12 and 13 also leverage the proof from Appendix D to compute $\boldsymbol{w}^{\dagger}$ and $\boldsymbol{a}^{\dagger}$ efficiently (though these operations do not affect the asymptotic time complexity). Expecting that the number of iterations for a pivoting algorithm is $O(n)$ in the size of the input (Cottle et al., 1992) and assuming that each iteration requires at most two pivot operations (each rank-1 update operation to a matrix factorization will exhibit $O(m^2)$ time complexity), the asymptotic complexity of the modified PPM I algorithm is $O(m^3 + m^2 n)$. The ter-

mination conditions for the algorithm are not affected by our modifications.

Finally, we note that Baraff (1994) has proven that LCPs of the form $(\mathbf{H}\boldsymbol{w}, \mathbf{H}\mathbf{Q}^{-1}\mathbf{H}^\mathsf{T})$, where $\mathbf{H} \in \mathbb{R}^{p \times q}, \mathbf{Q} \in \mathbb{R}^{q \times q}, \mathbf{w} \in \mathbb{R}^q$, and $\mathbf{Q}$ is symmetric, PD are solvable. Thus, the inverse dynamics model will always possess a solution.

Algorithm 2 $\{\boldsymbol{z}, \boldsymbol{w}, \overline{\mathcal{B}}\} = \mathrm{PPM}(\boldsymbol{N}, \boldsymbol{M}, \boldsymbol{f}^*, \boldsymbol{z}^-)$ Solves the LCP $(\mathbf{N}\mathbf{M}^{-1}\boldsymbol{f}^*, \mathbf{N}\mathbf{M}^{-1}\mathbf{N}^\mathsf{T})$ resulting from convex, rigid contact models (the no-slip model and the complementarity-free model with Coulomb friction). $\overline{\mathcal{B}}^*$ are the set of non-basic indices returned from the last call to PPM.

1: $n \leftarrow \mathrm{rows}(\mathbf{N})$
2: $\boldsymbol{r} \leftarrow \mathbf{N} \cdot \boldsymbol{f}^*$
3: $i \leftarrow \arg\min_i r_i$ ▷ Check for trivial solution
4: **if** $r_i \geq 0$ **then**
5:     **return** $\{\mathbf{0}, \boldsymbol{r}\}$
6: $\overline{\mathcal{B}} \leftarrow \overline{\mathcal{B}}^*$
7: **if** $\overline{\mathcal{B}} = \emptyset$ **then**
8:     $\overline{\mathcal{B}} \leftarrow \{i\}$ ▷ Establish initial nonbasic indices
9: $\mathcal{B} \leftarrow \{1, \dots, n\} - \overline{\mathcal{B}}$ ▷ Establish basic indices
10: **while** *true* **do**
11:     $\mathbf{A} \leftarrow \mathbf{N}_{\overline{\mathcal{B}}} \cdot \mathbf{M}^{-1} \cdot \mathbf{N}_{\overline{\mathcal{B}}}^\mathsf{T}$
12:     $\boldsymbol{b} \leftarrow \mathbf{N}_{\overline{\mathcal{B}}} \cdot \boldsymbol{f}^*$
13:     $\boldsymbol{z}^\dagger \leftarrow \mathbf{A}^{-1} \cdot -\boldsymbol{b}$ ▷ Compute $\boldsymbol{z}$ non-basic components
14:     $\boldsymbol{a}^\dagger \leftarrow \mathbf{M}^{-1} \cdot \mathbf{N}_{\overline{\mathcal{B}}}^\mathsf{T} \boldsymbol{z}^\dagger + \boldsymbol{f}^*$
15:     $\boldsymbol{w}^\dagger \leftarrow \mathbf{N} \cdot \boldsymbol{a}^\dagger$
16:     $i \leftarrow \arg\min_i w_i^\dagger$ ▷ Search for index to move into non-basic set
17:     **if** $w_i^\dagger \geq 0$ **then**
18:         $j \leftarrow \arg\min_i \boldsymbol{z}_i^\dagger$ ▷ No index to move into the non-basic set; search for index to move into the basic set
19:         **if** $z_j^\dagger < 0$ **then**
20:             $k \leftarrow \overline{\mathcal{B}}(j)$
21:             $\mathcal{B} \leftarrow \mathcal{B} \cup \{k\}$ ▷ Move index k into basic set
22:             $\overline{\mathcal{B}} \leftarrow \overline{\mathcal{B}} - \{k\}$
23:             **continue**
24:         **else**
25:             $\boldsymbol{z} \leftarrow \mathbf{0}$
26:             $\boldsymbol{z}_{\overline{\mathcal{B}}} \leftarrow \boldsymbol{z}^\dagger$
27:             $\boldsymbol{w} \leftarrow \mathbf{0}$
28:             $\boldsymbol{w}_{\mathcal{B}} \leftarrow \boldsymbol{w}^\dagger$
29:             **return** $\{\boldsymbol{z}, \boldsymbol{w}\}$
30:     **else**
31:         $\overline{\mathcal{B}} \leftarrow \overline{\mathcal{B}} \cup \{i\}$ ▷ Move index i into non-basic set
32:         $\mathcal{B} \leftarrow \mathcal{B} - \{i\}$
33:         $j \leftarrow \arg\min_i \boldsymbol{z}_i^\dagger$ ▷ Try to find an index to move into the basic set
34:         **if** $z_j^\dagger < 0$ **then**
35:             $k \leftarrow \overline{\mathcal{B}}(j)$
36:             $\mathcal{B} \leftarrow \mathcal{B} \cup \{k\}$ ▷ Move index k into basic set
37:             $\overline{\mathcal{B}} \leftarrow \overline{\mathcal{B}} - \{k\}$

---

4 Inverse dynamics with Coulomb friction

Though control without slip facilitates locomotion for walking robots, for example, this behavior is not an accurate reflection of friction in most cases of contact (see further discussion in Section 7.1). Implementations of no-slip controllers limiting contact forces to within the bounds of a friction constraint have been suggested to reduce the occurrence of unintentional slip in walking robots (Righetti et al., 2013). They also greatly limit the accessible space of accelerations for the robot as all movements yielding sliding contact are infeasible under this model.

We formulate inverse dynamics using the computational model of rigid contact with Coulomb friction developed by Stewart & Trinkle (1996) and Anitescu & Potra (1997); the equations in this section closely follow those in Anitescu & Potra (1997).

4.1 Coulomb friction constraints

Still utilizing the first-order discretization of the rigid body dynamics (Equation 31), we reproduce the linearized Coulomb friction constraints from Anitescu & Potra (1997) without explanation (identical except for slight notational differences):

$$0 \leq \mathbf{E}\boldsymbol{\lambda} + \bar{\mathbf{F}}^+\boldsymbol{v} \perp \boldsymbol{f}_F \geq \mathbf{0} \tag{46}$$

$$0 \leq \boldsymbol{\mu}f_N - \mathbf{E}^\mathsf{T}\boldsymbol{f}_F \perp \boldsymbol{\lambda} \geq \mathbf{0} \tag{47}$$

where $\mathbf{E} \in \mathbb{R}^{n \times nk}$ (k is the number of edges in the polygonal approximation to the friction cone) retains its definition from Anitescu & Potra (1997) as a sparse selection matrix containing blocks of "ones" vectors, $\boldsymbol{\mu} \in \mathbb{R}^{n \times n}$ is a diagonal matrix with elements corresponding to the coefficients of friction at the n contacts, $\boldsymbol{\lambda}$ is a variable roughly equivalent to magnitude of tangent velocities after contact forces are applied, and $\mathbf{F} \in \mathbb{R}^{nk \times m}$ (equivalent to $\mathbf{D}$ in Anitescu & Potra, 1997) is the matrix of wrenches of frictional forces at k tangents to each contact point. If the friction cone is approximated by a pyramid (an assumption we make in the remainder of the article), then:

$$\mathbf{F} \equiv \begin{bmatrix} \mathbf{S}^\mathsf{T} & -\mathbf{S}^\mathsf{T} & \mathbf{T}^\mathsf{T} & -\mathbf{T}^\mathsf{T} \end{bmatrix}^\mathsf{T}$$

$$\boldsymbol{f}_F \equiv \begin{bmatrix} \boldsymbol{f}_S^{+\mathsf{T}} & \boldsymbol{f}_S^{-\mathsf{T}} & \boldsymbol{f}_T^{+\mathsf{T}} & \boldsymbol{f}_T^{-\mathsf{T}} \end{bmatrix}^\mathsf{T}$$

where $\boldsymbol{f}_S = \boldsymbol{f}_S^+ - \boldsymbol{f}_S^-$ and $\boldsymbol{f}_T = \boldsymbol{f}_T^+ - \boldsymbol{f}_T^-$. Given these substitutions, the contact model with inverse dynamics becomes:

Complementarity-based inverse dynamics

non−interpenetration, compressive force, and

normal complementarity constraints:

$$0 \leq \boldsymbol{f}_N \perp \mathbf{N}^+\boldsymbol{v} \geq \mathbf{0} \tag{48}$$

Coulomb friction constraints:

$$0 \leq \lambda\boldsymbol{e} + \mathbf{F}^+\boldsymbol{v} \perp \boldsymbol{f}_F \geq \mathbf{0} \tag{49}$$

$$0 \leq \mu\boldsymbol{f}_N - \boldsymbol{e}^\mathsf{T}\boldsymbol{f}_F \perp \boldsymbol{\lambda} \geq \mathbf{0} \tag{50}$$

inverse dynamics:

$$\mathbf{P}^+\boldsymbol{v} = \dot{\boldsymbol{q}}_{\text{des}} \tag{51}$$

first−order dynamics:

$$^+\boldsymbol{v} = {}^-\boldsymbol{v} + \mathbf{M}^{-1}(\mathbf{N}^\mathsf{T}\boldsymbol{f}_N + \dots \tag{52}$$
$$\mathbf{F}^\mathsf{T}\boldsymbol{f}_F + \Delta t\boldsymbol{f}_{ext} - \mathbf{P}^\mathsf{T}\boldsymbol{\tau})$$

4.2 Resulting MLCP

Combining Equations 30–32 and 46–47 results in the MLCP:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{P}^\mathsf{T} & -\mathbf{N}^\mathsf{T} & -\mathbf{F}^\mathsf{T} & \mathbf{0} \\ \mathbf{P} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{N} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{E} \\ \mathbf{0} & \mathbf{0} & \mu & -\mathbf{E}^\mathsf{T} & \mathbf{0} \end{bmatrix} \begin{bmatrix} {}^+\boldsymbol{v} \\ \boldsymbol{\tau} \\ \boldsymbol{f}_N \\ \boldsymbol{f}_F \\ \boldsymbol{\lambda} \end{bmatrix} + \begin{bmatrix} -\boldsymbol{\kappa} \\ \dot{\boldsymbol{q}}_{\text{des}} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{w}_N \\ \boldsymbol{w}_F \\ \boldsymbol{w}_\lambda \end{bmatrix} \tag{53}$$

$$\boldsymbol{f}_N \geq \mathbf{0}, \boldsymbol{w}_N \geq \mathbf{0}, \boldsymbol{f}_N^\mathsf{T}\boldsymbol{w}_N = 0 \tag{54}$$

$$\boldsymbol{f}_F \geq \mathbf{0}, \boldsymbol{w}_F \geq \mathbf{0}, \boldsymbol{f}_F^\mathsf{T}\boldsymbol{w}_F = 0 \tag{55}$$

$$\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{w}_\lambda \geq \mathbf{0}, \boldsymbol{\lambda}^\mathsf{T}\boldsymbol{w}_\lambda = 0 \tag{56}$$

Vectors $\boldsymbol{w}_N$ and $\boldsymbol{w}_F$ correspond to the normal and tangential velocities after impulsive forces have been applied.

4.2.1 Transformation to LCP and proof of solution existence

The MLCP can be transformed to a LCP as described by Cottle et al. (1992) by solving for the unconstrained variables ${}^+\boldsymbol{v}$ and $\boldsymbol{\tau}$. This transformation is possible because the matrix:

$$\mathbf{X} \equiv \begin{bmatrix} \mathbf{M} & \mathbf{P}^\mathsf{T} \\ \mathbf{P} & \mathbf{0} \end{bmatrix} \tag{57}$$

is non-singular. Proof comes from blockwise invertibility of this matrix, which requires only invertibility of $\mathbf{M}$ (guaranteed because generalized inertia matrices are positive definite) and $\mathbf{PM}^{-1}\mathbf{P}^\mathsf{T}$. This latter matrix selects exactly those rows and columns corresponding to the joint space inertia matrix (Featherstone, 1987), which is also positive definite. After eliminating the unconstrained variables ${}^+\boldsymbol{v}$ and $\boldsymbol{\tau}$, the following LCP results:

$$\begin{bmatrix} \mathbf{NM}^{-1}\mathbf{N}^\mathsf{T} & \mathbf{NM}^{-1}\mathbf{F}^\mathsf{T} & \mathbf{E} \\ \mathbf{FM}^{-1}\mathbf{N}^\mathsf{T} & \mathbf{FM}^{-1}\mathbf{F}^\mathsf{T} & \mathbf{0} \\ -\mathbf{E}^\mathsf{T} & \mu & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_N \\ \boldsymbol{f}_F \\ \boldsymbol{\lambda} \end{bmatrix} - \begin{bmatrix} \mathbf{NM}^{-1}\boldsymbol{\kappa} \\ \mathbf{FM}^{-1}\boldsymbol{\kappa} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_N \\ \boldsymbol{w}_F \\ \boldsymbol{w}_\lambda \end{bmatrix} \tag{58}$$

$$\boldsymbol{f}_N \geq \mathbf{0}, \boldsymbol{w}_N \geq \mathbf{0}, \boldsymbol{f}_N^\mathsf{T}\boldsymbol{w}_N = 0 \tag{59}$$

$$\boldsymbol{f}_F \geq \mathbf{0}, \boldsymbol{w}_F \geq \mathbf{0}, \boldsymbol{f}_F^\mathsf{T}\boldsymbol{w}_F = 0 \tag{60}$$

$$\boldsymbol{\lambda} \geq \mathbf{0}, \boldsymbol{w}_\lambda \geq \mathbf{0}, \boldsymbol{\lambda}^\mathsf{T}\boldsymbol{w}_\lambda = 0 \tag{61}$$

The discussion in Stewart & Trinkle (1996) can be used to show that this LCP matrix is *copositive* (see Cottle, et al., 1992, Definition 3.8.1), since for any vector $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{f}_N^\mathsf{T} & \boldsymbol{f}_F^\mathsf{T} & \boldsymbol{\lambda}^\mathsf{T} \end{bmatrix}^\mathsf{T} \geq \mathbf{0}$,

$$\begin{bmatrix} \boldsymbol{f}_N \\ \boldsymbol{f}_F \\ \boldsymbol{\lambda} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \mathbf{NM}^{-1}\mathbf{N}^\mathsf{T} & \mathbf{NM}^{-1}\mathbf{F}^\mathsf{T} & \mathbf{E} \\ \mathbf{FM}^{-1}\mathbf{N}^\mathsf{T} & \mathbf{FM}^{-1}\mathbf{F}^\mathsf{T} & \mathbf{0} \\ \mu & -\mathbf{E}^\mathsf{T} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_N \\ \boldsymbol{f}_F \\ \boldsymbol{\lambda} \end{bmatrix} =$$
$$(\mathbf{N}^\mathsf{T}\boldsymbol{f}_N + \mathbf{F}^\mathsf{T}\boldsymbol{f}_N)^\mathsf{T}\mathbf{M}^{-1}(\mathbf{N}^\mathsf{T}\boldsymbol{f}_N + \mathbf{F}^\mathsf{T}\boldsymbol{f}_N) + \boldsymbol{f}_N^\mathsf{T}\mu\boldsymbol{\lambda} \geq 0 \tag{62}$$

because $\mathbf{M}^{-1}$ is positive definite and $\boldsymbol{\mu}$ is a diagonal matrix with non-negative elements. The transformation from the MLCP to the LCP yields $(k + 2)n$ LCP variables (the per-contact allocation is: one for the normal contact magnitude, k for the frictional force components, and one for an element of $\boldsymbol{\lambda}$) and at most $2m$ unconstrained variables.

As noted by Anitescu & Potra (1997), Lemke's Algorithm can provably solve such copositive LCPs (Cottle et al., 1992) if precautions are taken to prevent cycling through indices. After solving the LCP, joint torques can be retrieved exactly as in Section 3.5. *Thus, we have shown—using elementary extensions to the work in Anitescu & Potra (1997)—that a right inverse of the non-impacting rigid contact model exists* (as first broached in Section 1). Additionally, the expected running time of Lemke's Algorithm is cubic in the number of variables, so this inverse can be computed in expected polynomial time.

4.3 Contact indeterminacy

Though the approach presented in this section produces a solution to the inverse dynamics problem with simultaneous contact force computation, the approach can converge to a vastly different, but equally valid solution at each controller iteration. However, unlike the no-slip model, it is unclear how to bias the solution to the LCP, because a means for warm starting Lemke's Algorithm is currently unknown (our experience confirms the common wisdom that using the basis corresponding to the last solution usually leads to excessive pivoting).

Generating a torque profile that would evolve without generating torque chatter requires checking all possible solutions of the LCP if this approach is to be used for robot control. Generating all solutions requires a linear system solve for each combination of basic and nonbasic indices among all problem variables. Enumerating all possible solutions yields exponential time complexity, the same as the worst case complexity of Lemke's Algorithm (Lemke, 1965). After all solutions have been enumerated, torque chatter would be eliminated by using the solution that differs minimally from the last solution. Algorithm 3 presents this approach.

Algorithm 3 $\{x, \epsilon\}$ =MINDIFF$(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, g, h, \overline{\mathcal{B}}, x_0, n)$ Computes the solution to the LCP $(h - \mathbf{D}\mathbf{A}^{-1}g, \mathbf{B} - \mathbf{D}\mathbf{A}^{-1}\mathbf{C})$ that is closest (by Euclidean norm) to vector x_0 using a recursive approach. n is always initialized as rows($\mathbf{B}$).

1: **if** $n > 0$ **then**
2:     $\{x_1, \epsilon_1\}$ =MINDIFF$(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, g, h, \overline{\mathcal{B}}, x_0, n-1)$
3:     $\overline{\mathcal{B}} \leftarrow \{\overline{\mathcal{B}}, n\}$ ▷ Establish nonbasic indices
4:     $\{x_2, \epsilon_2\}$ =MINDIFF$(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, g, h, \overline{\mathcal{B}}, x_0, n-1)$
5:     **if** $\epsilon_1 < \epsilon_2$ **then return** $\{x_1, \epsilon_1\}$
6:     **else**   **return** $\{x_2, \epsilon_2\}$
7: **else**
8:     $\{z, w\} = \text{LCP}(h_{nb} - \mathbf{D}_{nb}\mathbf{A}^{-1}g, \mathbf{B}_{nb} - \mathbf{D}_{nb}\mathbf{A}^{-1}\mathbf{C}_{nb})$
9:     $x \leftarrow \mathbf{A}^{-1}(\mathbf{C}_{nb}z_{nb} + g)$
10:     **return** $\{x, \|x - x_0\|\}$

The fact that we can enumerate all solutions to the problem in exponential time proves that solving the problem is at worst NP-hard, though following an enumerative approach is not practical.

5 Convex inverse dynamics without normal complementarity

This section describes an approach for inverse dynamics that mitigates indeterminacy in rigid contact using the impact model described in Section 2.2.4. The approach is almost identical to the "standard" rigid contact model described in Section 2.2, but for the absence of the normal complementarity constraint (as motivated in Section 2.2.4). The approach works by determining contact and actual forces in a first step and then solving within the nullspace of the objective function (Equation 25) such that joint forces are minimized. The resulting problem is strictly convex, and thus torques are continuous in time (and more likely safe for a robot to apply) if the underlying dynamics are smooth. This latter assumption is violated only when a discontinuity occurs from one control loop iteration to the next, as it would when contact is broken, bodies newly come

into contact, the contact surface changes, or contact between two bodies switches between slipping and sticking.

Torque chatter due to contact indeterminacy can be avoided by ensuring that contact forces are not cycled rapidly between points of contact under potentially indeterminate contact configurations across successive controller iterations. Zapolsky & Drumwright (2014) eliminate torque chatter using a QP stage that searches over the optimal set of contact forces (using a convex relaxation to the rigid contact model) for forces that minimize the ℓ_2-norm of joint torques.

An alternative, one-stage approach is described by Todorov (2014), who regularizes the quadratic objective matrix to attain the requisite strict convexity. Another one-stage approach (which we test in Section 6) uses the warm starting-based solution technique described in Section 3 to mitigate contact indeterminacy. The two-stage approach described below confers the following advantages over one-stage approaches: (1) no regularization factor need be chosen as in Todorov (2014)—Todorov does not motivate the physical intuition behind the effect of larger regularization factors, and computing an optimal regularization factor through singular value decomposition would impact running time considerably; and (2) the two-stage approach allows the particular solution to be selected using an arbitrary objective criterion (minimizing actuator torques is particularly relevant for robotics applications). The presented approach is somewhat slower than a one-stage approach, though we have demonstrated performance suitably fast for real-time control loops on quadrupedal robots in Zapolsky & Drumwright (2014).

5.1 Computing inverse dynamics and contact forces simultaneously (Stage I)

For simplicity of presentation, we will assume that the number of edges in the approximation of the friction cone for each contact is four; in other words, we will use a friction pyramid in place of a cone. The inverse dynamics problem is formulated as follows:

<div style="border:1px solid black; padding:10px">

Complementarity-free inverse dynamics: Stage I

dissipate kinetic energy maximally:

$$\underset{\boldsymbol{f}_N, \boldsymbol{f}_F, {}^{+}\boldsymbol{v}, \boldsymbol{\tau}}{\text{minimize}} \quad \frac{1}{2} {}^{+}\boldsymbol{v}^{\mathsf{T}} \mathbf{M} \, {}^{+}\boldsymbol{v} \tag{63}$$

$non-interpenetration constraint:$

$$\text{subject to: } \mathbf{N} \, {}^{+}\boldsymbol{v} \geq \mathbf{0} \tag{64}$$

$variable \; non-negativity$

(for formulation convenience):

$$\boldsymbol{f}_N \geq \mathbf{0}, \quad \boldsymbol{f}_F \geq \mathbf{0} \tag{65}$$

Coulomb friction:

$$\mu f_{N_i} \geq \mathbf{1}^{\mathsf{T}} \boldsymbol{f}_{F_i} \tag{66}$$

$first-order \; velocity \; relationship:$

$$
\begin{aligned}
{}^{+}\boldsymbol{v} = {}^{-}\boldsymbol{v} + \mathbf{M}^{-1}(\mathbf{N}^{\mathsf{T}} \boldsymbol{f}_N + \dots \\
\mathbf{F}^{\mathsf{T}} \boldsymbol{f}_F + \Delta t \boldsymbol{f}_{ext} - \mathbf{P}^{\mathsf{T}} \boldsymbol{\tau})
\end{aligned} \tag{67}
$$

inverse dynamics:

$$\mathbf{P} \, {}^{+}\boldsymbol{v} = \dot{\boldsymbol{q}}_{\text{des}} \tag{68}$$

</div>

As discussed in Section 2.2.4, we have shown that the contact model always has a solution (*i.e.*, the QP is always feasible) and that the contact forces will not do positive work (Drumwright & Shell, 2010). The addition of the inverse dynamics constraint (Equation 68) will not change this result—the frictionless version of this QP is identical to an LCP of the form that Baraff has proven solvable (see Section 3.7), which means that the QP is feasible. As in the inverse dynamics approach in Section 4, the first order approximation to acceleration avoids inconsistent configurations that can occur in rigid contact with Coulomb friction. The worst-case time complexity of solving this convex model is polynomial in the number of contact features (Boyd & Vandenberghe, 2004). High frequency control loops limit n to approximately four contacts given present hardware and using fast active-set methods.

5.1.1 Removing equality constraints

The optimization in this section is a convex quadratic program with inequality and equality constraints. We remove the equality constraints through substitution. This reduces the size of the optimization problem; removing linear equality constraints also eliminates significant variables if transforming the QP to a LCP via optimization duality theory (Cottle et al., 1992).[2]

---

[2] We use such a transformation in our work, which allows us to apply LEMKE (Fackler & Miranda), which is freely available, numerically robust (using Tikhonov regularization), and relatively fast.

The resulting QP takes the form:

$$\underset{\boldsymbol{f}_N, \boldsymbol{f}_F}{\text{minimize}} \begin{bmatrix} \boldsymbol{f}_N \\ \boldsymbol{f}_F \end{bmatrix}^{\mathsf{T}} \left(\begin{bmatrix} \mathbf{N}\mathbf{X}^{-1}\mathbf{N}^{\mathsf{T}} & \mathbf{N}\mathbf{X}^{-1}\mathbf{F}^{\mathsf{T}} \\ \mathbf{F}\mathbf{X}^{-1}\mathbf{N}^{\mathsf{T}} & \mathbf{F}\mathbf{X}^{-1}\mathbf{F}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_N \\ \boldsymbol{f}_F \end{bmatrix} + \dots \right.$$
$$\left. \begin{bmatrix} -\mathbf{N}\boldsymbol{\kappa} \\ -\mathbf{F}\boldsymbol{\kappa} \end{bmatrix} \right) \tag{69}$$

$$\text{subject to: } \begin{bmatrix} \mathbf{N}\mathbf{X}^{-1}\mathbf{N}^{\mathsf{T}} & \mathbf{N}\mathbf{X}^{-1}\mathbf{F}^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}_N \\ \boldsymbol{f}_F \end{bmatrix} - \mathbf{N}\boldsymbol{\kappa} \geq \mathbf{0} \tag{70}$$

$$\boldsymbol{f}_N \geq \mathbf{0}, \boldsymbol{f}_F \geq \mathbf{0} \tag{71}$$

$$\mu f_{N_i} \geq \mathbf{1}^{\mathsf{T}} \boldsymbol{f}_{F_i} \tag{72}$$

Once $\boldsymbol{f}_N$ and $\boldsymbol{f}_F$ have been determined, the inverse dynamics forces are computed using:

$$\begin{bmatrix} {}^{+}\boldsymbol{v} \\ \boldsymbol{\tau} \end{bmatrix} = \mathbf{X}^{-1} \begin{bmatrix} -\boldsymbol{\kappa} + \mathbf{N}^{\mathsf{T}} \boldsymbol{f}_N + \mathbf{F}^{\mathsf{T}} \boldsymbol{f}_F \\ \dot{\boldsymbol{q}}_{\text{des}} \end{bmatrix} \tag{73}$$

As in Section 3.5, consistency in the desired accelerations can be verified and modified without re-solving the QP if found to be inconsistent.

5.1.2 Minimizing floating point computations

Because inverse dynamics may be used within real-time control loops, this section describes an approach that can minimize floating point computations over the formulation described above.

Assume that we first solve for the joint forces $\boldsymbol{f}_{ID}$ necessary to solve the inverse dynamics problem under no contact constraints. The new velocity ${}^{+}\boldsymbol{v}$ is now defined as:

$$
{}^{+}\boldsymbol{v} = {}^{-}\boldsymbol{v} + \mathbf{M}^{-1}(\mathbf{N}^{\mathsf{T}} \boldsymbol{f}_N + \mathbf{F}^{\mathsf{T}} \boldsymbol{f}_F + \Delta t \boldsymbol{f}_{ext} + \begin{bmatrix} \mathbf{0} \\ \Delta t (\boldsymbol{f}_{ID} + \boldsymbol{x}) \end{bmatrix}) \tag{74}
$$

where we define $\boldsymbol{x}$ to be the actuator forces that are added to $\boldsymbol{f}_{ID}$ to counteract contact forces. To simplify our derivations, we will define the following vectors and

matrices:

$$\mathbf{R} \equiv \begin{bmatrix} \mathbf{N}^\mathsf{T} & \mathbf{F}^\mathsf{T} \end{bmatrix} \tag{75}$$

$$\boldsymbol{z} \equiv \begin{bmatrix} \boldsymbol{f}_N & \boldsymbol{f}_F \end{bmatrix}^\mathsf{T} \tag{76}$$

$$\mathbf{M} \equiv \begin{matrix} nb \\ nq \end{matrix} \begin{bmatrix} \overset{nb}{\mathbf{A}} & \overset{nq}{\mathbf{B}} \\ \mathbf{B}^\mathsf{T} & \mathbf{C} \end{bmatrix} \tag{77}$$

$$\mathbf{M}^{-1} \equiv \begin{matrix} nb \\ nq \end{matrix} \begin{bmatrix} \overset{nb}{\mathbf{D}} & \overset{nq}{\mathbf{E}} \\ \mathbf{E}^\mathsf{T} & \mathbf{G} \end{bmatrix} \tag{78}$$

$$\boldsymbol{j} \equiv \boldsymbol{v}_b + \begin{bmatrix} \mathbf{D} & \mathbf{E} \end{bmatrix} (\Delta t \boldsymbol{f}_{ext} + \begin{bmatrix} \mathbf{0} \\ \Delta t \boldsymbol{f}_{ID} \end{bmatrix}) \tag{79}$$

$$\boldsymbol{k} \equiv \boldsymbol{v}_q + \begin{bmatrix} \mathbf{E}^\mathsf{T} & \mathbf{G} \end{bmatrix} (\Delta t \boldsymbol{f}_{ext} + \begin{bmatrix} \mathbf{0} \\ \Delta t \boldsymbol{f}_{ID} \end{bmatrix}) \tag{80}$$

$$\boldsymbol{f}_{ID} \equiv \mathbf{C} \ddot{\boldsymbol{q}} - \boldsymbol{f}_{ext,nq} \tag{81}$$

Where nq is the total joint degrees of freedom of the robot, and nb is the total base degrees of freedom for the robot ($nb = 6$ for floating bases).

The components of $^+\boldsymbol{v}$ are then defined as follows:

$$^+\boldsymbol{v}_b \equiv \boldsymbol{j} + \begin{bmatrix} \mathbf{D} & \mathbf{E} \end{bmatrix} (\mathbf{R}\boldsymbol{z} + \begin{bmatrix} \mathbf{0} \\ \Delta t \boldsymbol{x} \end{bmatrix}) \tag{82}$$

$$^+\boldsymbol{v}_q \equiv \boldsymbol{k} + \begin{bmatrix} \mathbf{E}^\mathsf{T} & \mathbf{G} \end{bmatrix} (\mathbf{R}\boldsymbol{z} + \begin{bmatrix} \mathbf{0} \\ \Delta t \boldsymbol{x} \end{bmatrix}) = \dot{\boldsymbol{q}}_{\text{des}} \tag{83}$$

From the latter equation we can solve for $\boldsymbol{x}$:

$$\boldsymbol{x} = \frac{\mathbf{G}^{-1}(^+\boldsymbol{v}_q - \boldsymbol{k} - \begin{bmatrix} \mathbf{E}^\mathsf{T} & \mathbf{G} \end{bmatrix} \mathbf{R}\boldsymbol{z})}{\Delta t} \tag{84}$$

Equation 84 indicates that once contact forces are computed, determining the actuator forces for inverse dynamics requires solving only a linear equation. Substituting the solution for $\boldsymbol{x}$ into Eqn. 82 yields:

$$^+\boldsymbol{v}_b = \boldsymbol{j} + \begin{bmatrix} \mathbf{D} & \mathbf{E} \end{bmatrix} \mathbf{R}\boldsymbol{z} + \mathbf{E}\mathbf{G}^{-1}(^+\boldsymbol{v}_q - \boldsymbol{k} - \begin{bmatrix} \mathbf{E}^\mathsf{T} & \mathbf{G} \end{bmatrix} \mathbf{R}\boldsymbol{z}) \tag{85}$$

To simplify further derivation, we will define a new matrix and a new vector:

$$\mathbf{Z} \equiv (\begin{bmatrix} \mathbf{D} & \mathbf{E} \end{bmatrix} - \mathbf{E}\mathbf{G}^{-1} \begin{bmatrix} \mathbf{E}^\mathsf{T} & \mathbf{G} \end{bmatrix}) \mathbf{R} \tag{86}$$

$$\boldsymbol{p} \equiv \boldsymbol{j} + \mathbf{E}\mathbf{G}^{-1}(^+\boldsymbol{v}_q - \boldsymbol{k}) \tag{87}$$

Now, $^+\boldsymbol{v}_b$ can be defined simply, and solely in terms of $\boldsymbol{z}$, as:

$$^+\boldsymbol{v}_b \equiv \mathbf{Z}\boldsymbol{z} + \boldsymbol{p} \tag{88}$$

We now represent the objective function (Eqn. 63) in block form as:

$$f(.) \equiv \frac{1}{2} \begin{bmatrix} ^+\boldsymbol{v}_b \\ ^+\boldsymbol{v}_q \end{bmatrix}^\mathsf{T} \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\mathsf{T} & \mathbf{C} \end{bmatrix} \begin{bmatrix} ^+\boldsymbol{v}_b \\ ^+\boldsymbol{v}_q \end{bmatrix} \tag{89}$$

which is identical to:

$$f(.) \equiv \frac{1}{2} {}^+\boldsymbol{v}_b{}^\mathsf{T} \mathbf{A} \, {}^+\boldsymbol{v}_b + {}^+\boldsymbol{v}_b \mathbf{B}^\mathsf{T} \, {}^+\boldsymbol{v}_q + \frac{1}{2} {}^+\boldsymbol{v}_q{}^\mathsf{T} \mathbf{C} \, {}^+\boldsymbol{v}_q \tag{90}$$

As we will be attempting to minimize $f(.)$ with regard to $\boldsymbol{z}$, which the last term of the above equation does not depend on, that term is ignored hereafter. Expanding remaining terms using Equation 82, the new objective function is:

$$f(.) \equiv \frac{1}{2} \boldsymbol{z}^\mathsf{T} \mathbf{Z}^\mathsf{T} \mathbf{A} \mathbf{Z} \boldsymbol{z} + \boldsymbol{z}^\mathsf{T} \mathbf{Z}^\mathsf{T} \mathbf{A} \boldsymbol{p} + \boldsymbol{z}^\mathsf{T} \mathbf{Z}^\mathsf{T} \mathbf{B} \, {}^+\boldsymbol{v}_q \tag{91}$$

$$\equiv \frac{1}{2} \boldsymbol{z}^\mathsf{T} \mathbf{Z}^\mathsf{T} \mathbf{A} \mathbf{Z} \boldsymbol{z} + \boldsymbol{z}^\mathsf{T} (\mathbf{Z}^\mathsf{T} \mathbf{A} \boldsymbol{p} + \mathbf{Z}^\mathsf{T} \mathbf{B} \, {}^+\boldsymbol{v}_q) \tag{92}$$

subject to the following constraints:

$$\mathbf{N}^\mathsf{T} \begin{bmatrix} \mathbf{Z}\boldsymbol{z} + \boldsymbol{p} \\ \boldsymbol{v}_q^* \end{bmatrix} \geq 0 \tag{93}$$

$$\boldsymbol{f}_{N,i} \geq 0 \qquad (\text{for } i = 1 \dots n) \tag{94}$$

$$\mu \boldsymbol{f}_{N,i} \geq \boldsymbol{c}_{S,i} + \boldsymbol{c}_{T,i} \tag{95}$$

Symmetry and positive semi-definiteness of the QP follows from symmetry and positive definiteness of $\mathbf{A}$. Once the solution to this QP is determined, the actuator forces $\boldsymbol{x} + \boldsymbol{f}_{ID}$ determined via inverse dynamics can be recovered.

5.1.3 Floating point operation counts

Operation counts for matrix-vector arithmetic and numerical linear algebra are taken from Hunger (2007).

Before simplifications: Floating point operations (*flops*) necessary to setup the *Stage I* model as presented initially sum to 77,729 flops, substituting: $m = 18, nq = 16, n = 4, k = 4$.

operation	flops
$LDL^\mathsf{T}(\mathbf{X})$	$\frac{m^3}{3} + m^2(\text{nq}) + m^2 + m(\text{nq})^2 + 2m(\text{nq})$
	$- \frac{7m}{3} + \frac{(\text{nq})^3}{3} + (\text{nq})^2 - \frac{7(\text{nq})}{3} + 1$
$\mathbf{X}^{-1}\mathbf{N}^\mathsf{T}$	$m + 2m^2n + (nq) + 4mn(nq) + 2n(nq)^2$
$\mathbf{X}^{-1}\mathbf{F}^\mathsf{T}$	$m + 2km^2n + (nq) + 4kmn(nq) + 2kn(nq)^2$
$\mathbf{N}\mathbf{X}^{-1}\mathbf{N}^\mathsf{T}$	$2mn^2 - mn$
$\mathbf{F}\mathbf{X}^{-1}\mathbf{N}^\mathsf{T}$	$2mn^2k - mn$
$\mathbf{F}\mathbf{X}^{-1}\mathbf{F}^\mathsf{T}$	$2mnk^2 - mnk$
κ	$2m^2 - m$
$\mathbf{X}^{-1}\kappa$	$2(m + (nq))^2$
$\mathbf{N}\mathbf{X}^{-1}\kappa$	$mn - m$
$\mathbf{F}\mathbf{X}^{-1}\kappa$	$mnk - m$
$\boldsymbol{\tau}$	$2(m + (nq))^2$

Table 1: *Floating point operations (*flops*) per task without floating point optimizations.*

After simplifications: Floating point operations necessary to setup the *Stage I* model after modified to reduce computational costs sums to 73,163 flops when substituting: $m = 18, nq = 16, n = 4, k = 4, nb = m - nq$, a total of 6.24% reduction in computation. When substituting: $m = 18, nq = 12, n = 4, k = 4, nb = m - nq$, we observed 102,457 flops for this new method and 62,109 flops before simplification. Thus, a calculation of the total number of floating point operations should be performed to determine whether the floating point simplification is actually more efficient for a particular robot.

operation	flops
$(LL^\mathsf{T}(\mathbf{M}))^{-1}$	$\frac{2}{3}m^3 + \frac{1}{2}m^2 + \frac{5}{6}m$
$LL^\mathsf{T}(\mathbf{G})$	$\frac{1}{3}(nq)^3 + \frac{1}{2}(nq)^2 + \frac{1}{6}nq$
$\mathbf{Z}$	$nb\ m + nq\ n(nk + 1)(2m - 1) +$
	$nb\ m(2nq - 1) + 2nq^2 m$
$\boldsymbol{p}$	$2nb\ nq + 2nq^2 + 3nq + 2nb + 2m + 2m\ nq$
$\mathbf{Z}^\mathsf{T}\mathbf{A}\mathbf{Z}$	$2nb^2(n(nk + 1)) - nb\ (n(nk + 1))$
	$+2nb\ (n(nk + 1))^2 - (n(nk + 1))^2$
$\mathbf{Z}^\mathsf{T}\mathbf{A}\boldsymbol{p}$	$(n(nk + 1) + nb)(2nb - 1)$
$\mathbf{Z}^\mathsf{T}\mathbf{B}\boldsymbol{v}_q^*$	$(n(nk + 1) + nq)(2nq - 1)$
$\mathbf{N}^\mathsf{T}\mathbf{Z}$	$n^2(nk + 1)(2nb - 1)$
$\mathbf{N}^\mathsf{T} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v}_q^* \end{bmatrix}$	$n(2m - 1)$

Table 2: *Floating point operations (*flops*) with floating point optimizations.*

5.1.4 Recomputing Inverse Dynamics to Stabilize Actuator Torques (Stage II)

In the case that the matrix $\mathbf{Z}^\mathsf{T}\mathbf{A}\mathbf{Z}$ is singular, the contact model is only convex rather than strictly convex (Boyd & Vandenberghe, 2004). Where a strictly convex system has just one optimal solution, the convex problem may have multiple equally optimal solutions. Conceptually, contact forces that predict that two legs, three legs, or four legs support a walking robot are all equally valid. A method is thus needed to optimize within the contact model's solution space while favoring solutions that predict contact forces at all contacting links (and thus preventing the rapid torque cycling between arbitrary optimal solutions). One such method would select—from the space of optimal solutions—the one that minimizes the ℓ_2-norm of joint torques. If we denote the solution that was computed in the last section as $\boldsymbol{z}_0$, the following optimization problem will yield the desired result:

> **Complementarity-free inverse dynamics: Stage II**
>
> find minimum norm motor torques:
>
> $$\underset{\boldsymbol{f}_N, \boldsymbol{f}_F}{\text{minimize}}\ \frac{1}{2}\boldsymbol{\tau}^\mathsf{T}\boldsymbol{\tau} \tag{96}$$
>
> subject to: Equations 64–68
>
> maintain Stage I objective:
>
> $$\frac{1}{2}\overset{+}{\boldsymbol{v}}{}^\mathsf{T}\mathbf{M}\overset{+}{\boldsymbol{v}} = f(\boldsymbol{z}_0) \tag{97}$$

We call the method described in Section 5.1.2 *Stage I* and the optimization problem above *Stage II*. Constraining for a quadratic objective function (Equation 63) with a quadratic inequality constraint yields a QCQP that may not be solvable sufficiently quickly for high frequency control loops. We now show how to use the nullspace of $\mathbf{Z}^\mathsf{T}\mathbf{A}\mathbf{Z}$ to perform this second optimization *without explicitly considering the quadratic inequality constraint*; thus, a QP problem formulation is retained. Assume that the matrix $\mathbf{W}$ gives the nullspace of $\mathbf{Z}^\mathsf{T}\mathbf{A}\mathbf{Z}$. The vector of contact forces will now be given as $\boldsymbol{z} + \mathbf{W}\boldsymbol{w}$, where $\boldsymbol{w}$ will be the optimization vector.

The kinetic energy from applying the contact impulses is:

$$\begin{aligned}
\epsilon_2 &= \frac{1}{2}(\boldsymbol{z} + \mathbf{W}\boldsymbol{w})^\mathsf{T}\mathbf{Z}^\mathsf{T}\mathbf{A}\mathbf{Z}(\boldsymbol{z} + \mathbf{W}\boldsymbol{w}) \\
&\quad + (\boldsymbol{z} + \mathbf{W}\boldsymbol{w})^\mathsf{T}(\mathbf{Z}^\mathsf{T}\mathbf{A}\boldsymbol{p} + \mathbf{Z}^\mathsf{T}\mathbf{B}\overset{+}{\boldsymbol{v}}_q) \\
&= \frac{1}{2}\boldsymbol{z}^\mathsf{T}\mathbf{Z}^\mathsf{T}\mathbf{A}\mathbf{Z}\boldsymbol{z} + \boldsymbol{z}^\mathsf{T}(\mathbf{Z}^\mathsf{T}\mathbf{A}\boldsymbol{p} + \mathbf{Z}^\mathsf{T}\mathbf{B}\overset{+}{\boldsymbol{v}}_q) \\
&\quad + \boldsymbol{w}^\mathsf{T}\mathbf{W}^\mathsf{T}(\mathbf{Z}^\mathsf{T}\mathbf{A}\boldsymbol{p} + \mathbf{Z}^\mathsf{T}\mathbf{B}\overset{+}{\boldsymbol{v}}_q)
\end{aligned}$$

The terms $\frac{1}{2}\boldsymbol{w}^\mathsf{T}\mathbf{W}^\mathsf{T}\mathbf{Z}^\mathsf{T}\mathbf{A}\mathbf{Z}\mathbf{W}\boldsymbol{w}$ and $\boldsymbol{z}\mathbf{Z}^\mathsf{T}\mathbf{A}\mathbf{Z}\mathbf{W}\boldsymbol{w}$ are not included above because both are zero: $\mathbf{W}$ is in the nullspace of $\mathbf{Z}^\mathsf{T}\mathbf{A}\mathbf{Z}$. The energy dissipated in the second stage. ϵ_2, should be equal to the energy dissipated in the first stage, ϵ_1. Thus, we want $\epsilon_2 - \epsilon_1 = 0$. Algebra yields:

$$\boldsymbol{w}^\mathsf{T}\mathbf{W}^\mathsf{T}(\mathbf{Z}^\mathsf{T}\mathbf{A}\boldsymbol{p} + \mathbf{Z}^\mathsf{T}\mathbf{B}\overset{+}{\boldsymbol{v}}_q) = 0 \tag{98}$$

First defining $\boldsymbol{y}$ as:

$$\boldsymbol{y} \equiv \frac{\mathbf{G}^{-1}\left(\overset{+}{\boldsymbol{v}}_q - \boldsymbol{k} - \begin{bmatrix}\mathbf{E}^\mathsf{T} & \mathbf{G}\end{bmatrix}\mathbf{R}(\boldsymbol{z} + \mathbf{W}\boldsymbol{w})\right)}{\Delta t} \tag{99}$$

The resulting objective is:

$$g(\boldsymbol{w}) \equiv \frac{1}{2}\boldsymbol{y}^\mathsf{T}\boldsymbol{y} \tag{100}$$
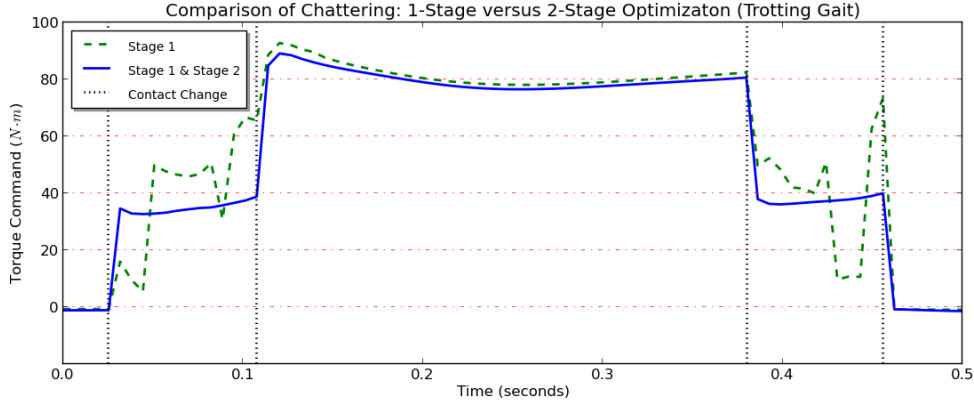
Fig. 3: *Plot of torque chatter while controlling with inverse dynamics using an indeterminate contact model (Stage 1) versus the smooth torque profile produced by a determinate contact model (Stage 1 & Stage 2).*

From this, the following optimization problem arises:

$$\min_{\boldsymbol{w}} \frac{1}{2} \boldsymbol{y}^{\mathsf{T}} \boldsymbol{y} \tag{101}$$

subject to: $(\boldsymbol{p}^{\mathsf{T}} \mathbf{A}^{\mathsf{T}} + {}^{+}\boldsymbol{v}_q{}^{\mathsf{T}} \mathbf{B}^{\mathsf{T}}) \mathbf{Z} \mathbf{W} \boldsymbol{w} = \mathbf{0} \tag{102}$

$$\mathbf{N}^{\mathsf{T}} \begin{bmatrix} \mathbf{Z}(\boldsymbol{z} + \mathbf{W}\boldsymbol{w}) + \boldsymbol{p} \\ {}^{+}\boldsymbol{v}_q \end{bmatrix} \geq \mathbf{0} \tag{103}$$

$$(\boldsymbol{z} + \mathbf{W}\boldsymbol{w})_i \geq \mathbf{0}, \quad (\text{for } i = 1 \ldots 5n) \tag{104}$$

$$\mu_i(\boldsymbol{z} + \mathbf{W}\boldsymbol{w})_i \geq \mathbf{X}_{S_i}(\boldsymbol{z} + \mathbf{W}\boldsymbol{w})_{S_i} + \ldots$$
$$\mathbf{X}_{T_i}(\boldsymbol{z} + \mathbf{W}\boldsymbol{w})_{T_i}$$
$$(\text{for } i = 1 \ldots n) \tag{105}$$

Equation 104 constrains the contact force vector to only positive values, accounting for 5 positive directions to which force can be applied at the contact manifold $(\hat{n}, \hat{s}, -\hat{s}, \hat{t}, -\hat{t})$.[3] We use a proof that $\mathbf{Z} \cdot \ker(\mathbf{Z}^{\mathsf{T}} \mathbf{A} \mathbf{Z}) = \mathbf{0}$ (Zapolsky & Drumwright, 2014) to render $n+1$ (Equations 102 and 103) of $7n + 1$ linear constraints (Equations 102–105) unnecessary.

Finally, expanding and simplifying Equation 100 (removing terms that do not contain $\boldsymbol{w}$, scaling by Δt^2), and using the identity $\mathbf{U} \equiv \begin{bmatrix} \mathbf{E}^{\mathsf{T}} & \mathbf{G} \end{bmatrix} \mathbf{R}$ yields:

$$g(\boldsymbol{w}) \equiv \frac{1}{2} \boldsymbol{w}^{\mathsf{T}} \mathbf{W}^{\mathsf{T}} \mathbf{U}^{\mathsf{T}} \mathbf{G}^{-1\mathsf{T}} \mathbf{G}^{-1} \mathbf{U} \mathbf{W} \boldsymbol{w} \tag{106}$$
$$+ \boldsymbol{z}^{\mathsf{T}} \mathbf{U}^{\mathsf{T}} \mathbf{G}^{-1\mathsf{T}} \mathbf{G}^{-1} \mathbf{U} \mathbf{W} \boldsymbol{w} - {}^{+}\boldsymbol{v}_q{}^{\mathsf{T}} \mathbf{G}^{-1\mathsf{T}} \mathbf{G}^{-1} \mathbf{U} \mathbf{W} \boldsymbol{w}$$
$$+ \boldsymbol{k}^{\mathsf{T}} \mathbf{G}^{-1\mathsf{T}} \mathbf{G}^{-1} \mathbf{U} \mathbf{W} \boldsymbol{w}$$

Finally, actuator torques for the robot can be retrieved by calculating $\boldsymbol{y} + \boldsymbol{f}_{ID}$.

5.1.5 Feasibility and time complexity

It should be clear that a feasible point ($\boldsymbol{w} = \mathbf{0}$) always exists for the optimization problem. The dimensionality

$(n \times n$ in the number of contact points) of $\mathbf{Z}^{\mathsf{T}} \mathbf{A} \mathbf{Z}$ yields a nullspace computation of $O(n^3)$ and represents one third of the Stage II running times in our experiments. For quadrupedal robots with single point contacts, for example, the dimensionality of $\boldsymbol{w}$) is typically at most two, yielding fewer than $6n + 2$ total optimization variables (each linear constraint introduces six KKT dual variables for the simplest friction cone approximation). Timing results given n contacts for this virtual robot are available in Section 7.5.1.
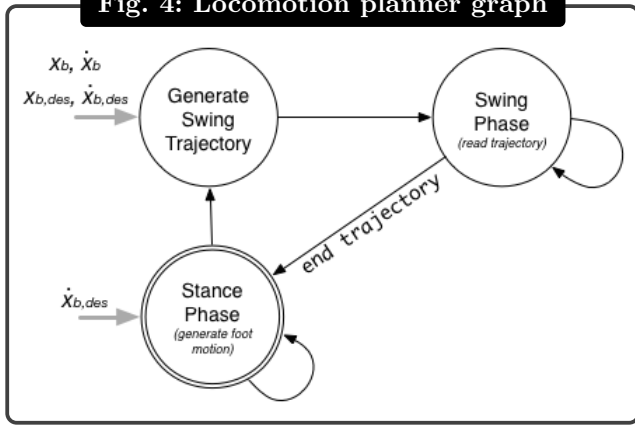
6 Experiments

This section assesses the inverse dynamics controllers under a range of conditions on a virtual, locomoting quadrupedal robot (depicted in Figures 8). For points of comparison, we also provide performance data for three reference controllers (depicted in Figures 5 and 7). These experiments also serve to illustrate that the inverse dynamics approaches function as expected.

We explore the effects of possible modeling infidelities and sensing inaccuracies by testing locomotion performance on rigid planar terrain, rigid non-planar terrain, and on compliant planar terrain. The last of these is an example of modeling infidelity, as the compliant terrain violates the assumption of rigid contact. Sensing inaccuracies may be introduced from physical sensor noise or perception error producing, *e.g.*, erroneous friction or contact normal estimates.

All code and data for experiments conducted in simulation, as well as videos of the virtual robots walking, are located (and can thus be reproduced) at:
http://github.com/PositronicsLab/idyn-experiments.

---

[3] We consider negative and positive $\hat{s}$ and $\hat{t}$ because LEMKE requires all variables to be positive.

Fig. 4: Locomotion planner graph

iteration. The phase of the gait for each foot is determined by a gait timing pattern and gait duty-factor assigned to each foot.

The planner (illustrated in Figure 4) takes as input desired planar base velocity ($\dot{\boldsymbol{x}}_{b,des} = [\dot{x}, \dot{y}, \dot{\theta}]$) and plans touchdown and liftoff locations connected with splined trajectories for the feet to drive the robot across the environment at the desired velocity. The planner then outputs a trajectory for each foot in the local frame of the robot. After end effector trajectories have been planned, joint trajectories are determined at each controller cycle using inverse kinematics.

6.2 Evaluated controllers

We use the same error-feedback and balance stabilization controller in all cases for purposes of reducing joint tracking error from drift and balancing the robot as it performs its task (see baseline controller in Figure 5). The gains used for PID control are identical between all controllers. The PID error feedback controller is defined in configuration space. The stabilization implementation uses an inverted pendulum model for balance, applying only virtual compressive forces along the contact normal to stabilize the robot (Sugihara & Nakamura, 2003). The error-feedback and stabilization modules accumulate error-feedback from stability and joint-space errors into the vector of desired acceleration ($\ddot{q}_{des}$).

6.1 Source of planned trajectories

We assess the performance of the reference controllers (described below) against the inverse dynamics controllers with contact force prediction presented in this work. The robot follows trajectories generated by our open source legged locomotion software PACER[4]. Our software implements balance stabilization, footstep planning, spline-based trajectory planning (described below), and inverse dynamics-based controllers with simultaneous contact force computation that are the focus of this article. The following terms will be used in our description of the planning system:

gait Cyclic pattern of timings determining when to make and break contact with the ground during locomotion.

stance phase The interval of time where a foot is planned to be in contact with the ground (foot applies force to move robot)

swing phase The planned interval of time for a foot to be swinging over the ground (to position the foot for the next stance phase)

duty-factor Planned portion of the gait for a foot to be in the stance phase

touchdown Time when a foot makes contact with the ground and transitions from swing to stance phase

liftoff Time when a foot breaks contact with the ground and transitions from stance to swing phase
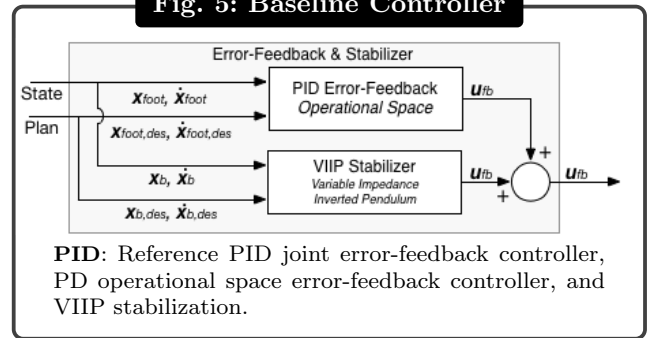
The trajectories generated by the planner are defined in operational space. Swing phase behavior is calculated as a velocity-clamped cubic spline at the start of each step and replanned during that swing phase as necessary. Stance foot velocities are determined by the desired base velocity and calculated at each controller

Fig. 5: Baseline Controller

PID: Reference PID joint error-feedback controller, PD operational space error-feedback controller, and VIIP stabilization.
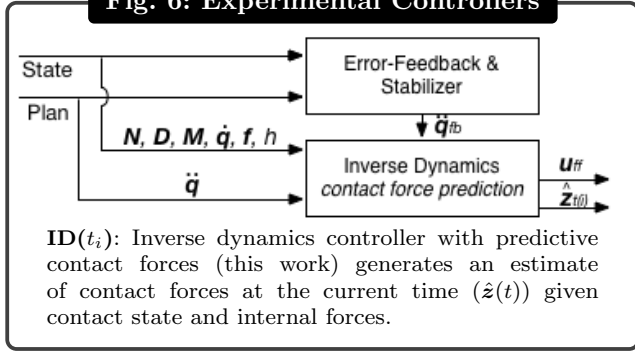
We compare the controllers described in this article, which will hereafter be referred to as $\mathbf{ID}(t_i)_{\texttt{solver,friction}}$, where the possible solvers are: $\texttt{solver} = \{\text{QP, LCP}\}$ for QP and LCP-based optimization models, respectively; and the possible friction models are: $\texttt{friction} = \{\mu, \infty\}$ for finite Coulomb friction and no-slip models, respectively.

We compare *ab initio* controllers implemented using the methods described in Sections 3, 4 and 5, see Figure 6) against the reference controllers (Figure 7), using finite and infinite friction coefficients to permit comparison against no-slip and Coulomb friction models, respectively. Time "t_i" in $\mathbf{ID}(t_i)$ refers to the use
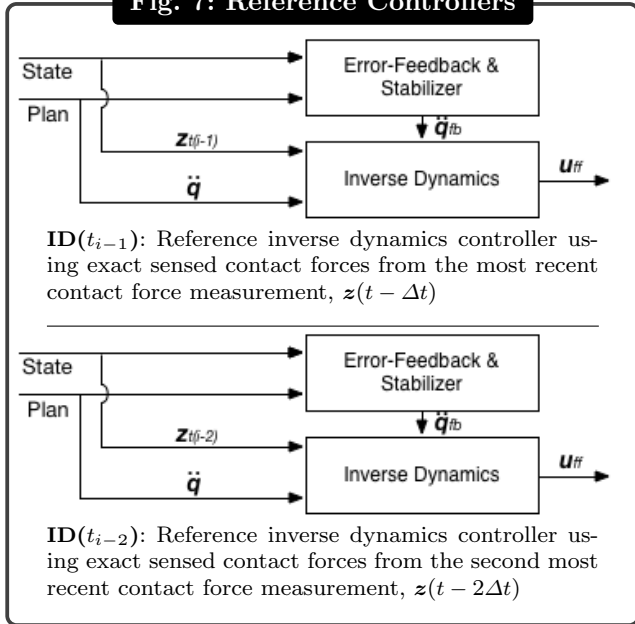
---

[4] PACER is available from:
http://github.com/PositronicsLab/Pacer

of contact forces predicted at the current controller time. The experimental (presented) controllers include: $\mathbf{ID}(t_i)_{LCP,\infty}$ is the *ab initio* controller from Section 3 that uses an LCP model, to predict contact reaction forces with no-slip constraints; $\mathbf{ID}(t_i)_{LCP,\mu}$ is the *ab initio* controller from Section 4 that uses an LCP model, to predict contact reaction forces with Coulomb friction; $\mathbf{ID}(t_i)_{QP,\mu}$ is the controller from Section 5 that uses a QP-based optimization approach for contact force prediction; $\mathbf{ID}(t_i)_{QP,\infty}$ is the same controller as $\mathbf{ID}(t_i)_{QP,\mu}$ from Section 5.1, but set to allow infinite frictional forces.

Fig. 6: Experimental Controllers

$\mathbf{ID}(t_i)$: Inverse dynamics controller with predictive contact forces (this work) generates an estimate of contact forces at the current time ($\hat{\boldsymbol{z}}(t)$) given contact state and internal forces.

Fig. 7: Reference Controllers

$\mathbf{ID}(t_{i-1})$: Reference inverse dynamics controller using exact sensed contact forces from the most recent contact force measurement, $\boldsymbol{z}(t - \Delta t)$

$\mathbf{ID}(t_{i-2})$: Reference inverse dynamics controller using exact sensed contact forces from the second most recent contact force measurement, $\boldsymbol{z}(t - 2\Delta t)$

The reference inverse dynamics controllers use sensed contact forces; the sensed forces are the exact forces applied by the simulator to the robot on the previous simulation step (*i.e.*, there is a sensing lag of Δt on these measurements, the simulation step size). We denote the controller using these exact sensed contact forces as $\mathbf{ID}(t_{i-1})$. Controller "$\mathbf{ID}(t_{i-1})$" uses the exact value of sensed forces from the immediately previous time step in simulation and represents an upper limit on the performance of using sensors to incorporate sensed

contact forces into an inverse dynamics model. *In situ* implementation of contact force sensing should result in worse performance than the controller described here, as it would be subject to inaccuracies in sensing and delays of multiple controller cycles as sensor data is filtered to smooth noise; we examine the effect of a second controller cycle delay with $\mathbf{ID}(t_{i-2})$ (see Figure 7).

6.3 Platforms

We evaluate performance of all controllers on a simulated quadruped (see Figure 8). Our test platform measures 16 cm tall and has three degree of freedom legs. Its feet are modeled as spherical collision geometries, often creating a a single point contact manifold with the terrain. We use this platform to assess the effectiveness of our inverse dynamics implementation given one to four contacts. Results we present from this platform are applicable to biped locomotion, the only differentiating factor being the presence of a more involved planning and balance system driving the robot.
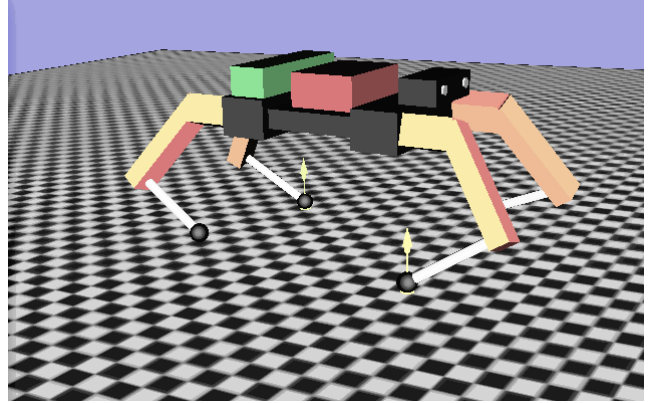
Fig. 8: *Snapshot of a quadruped robot in the* MOBY *simulator on planar terrain.*

6.4 Software and simulation setup

PACER runs alongside the open source simulator MOBY[5], which was used to simulate the legged locomotion scenarios used in the experiments. MOBY was arbitrarily set to simulate contact with the Stewart-Trinkle / Anitescu-Potra rigid contact models (Stewart & Trinkle, 1996; Anitescu & Potra, 1997); therefore, the contact models utilized by the simulator match those used in our reference controllers, *permitting us to compare*

---

[5] Obtained from `https://github.com/PositronicsLab/Moby`

our contact force predictions directly against those determined by the simulator. Both simulations and controllers had access to identical data: kinematics (joint locations, link lengths, *etc.*), dynamics (generalized inertia matrices, external forces), and friction coefficients at points of contact. MOBY provides accurate of the time of contact calculation for impacting bodies, yielding more accurate contact point and normal information. Other simulators step past the moment of contact, and approximate contact information based on the intersecting geometries. The accurate contact information provided by MOBY allows us to test the inverse dynamics controllers under more realistic conditions: contact may break or be formed between control loops.

We explore three aspects of the presented controllers in the following experiments: we show (in Section 7.1) that the controllers we present in Sections 4 and 3 are feasible for controlling a quadruped robot over a trot; we compare contact force predictions made by all presented controllers to the reaction forces generated by the simulation (Section 7.3); and we measure running times of all presented controllers given numerous additional contacts in Section 7.5.1.

6.5 Terrain types

We evaluate the performance of the baseline, reference, and presented controllers on a planar terrain. We use four cases to encompass expected surface properties experienced in locomotion. We model sticky and slippery terrain with frictional properties corresponding to a Coulomb friction of 0.1 for low (slippery) friction and infinity for high (sticky) friction. We also model rigid and compliant terrain, using rigid and penalty (spring-damper) based contact model, respectively. The compliant terrain is modeled to be relatively hard, yielding only 1 mm interpenetration of the foot into the plane while the robot is at rest. Our contact prediction models all assume rigid contact; a compliant terrain will assess whether the inverse dynamics model for predictive contact is viable when the contact model of the environment does not match its internal model. Successful locomotion using such a controller on a compliant surface would indicate (to some confidence) that the controller is robust to modeling infidelities and thus more robust in an uncertain environment.

We also test the controllers on a rigid height map terrain with non-vertical surface normals and varied surface friction to assess robustness on a more natural terrain. We limited extremes in the variability of the terrain, limiting bumps to 3 cm in height (about one fifth the height of the quadruped see Figure 9). so that the performance of the foothold planner (not presented

in this work) would not bias performance results. Friction values were selected between the upper and lower limits of Coulomb friction (between $\mu \sim \mathcal{U}(0.1, 1.5)$) found in various literature on legged locomotion.
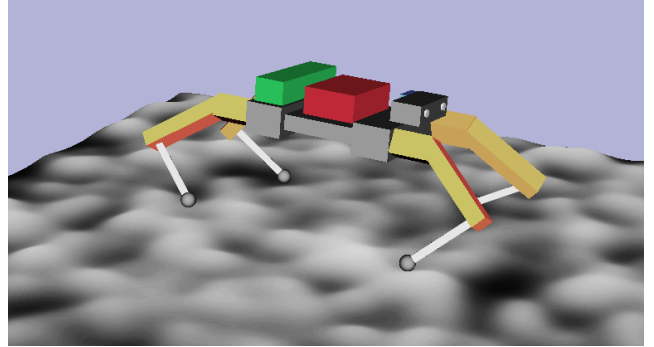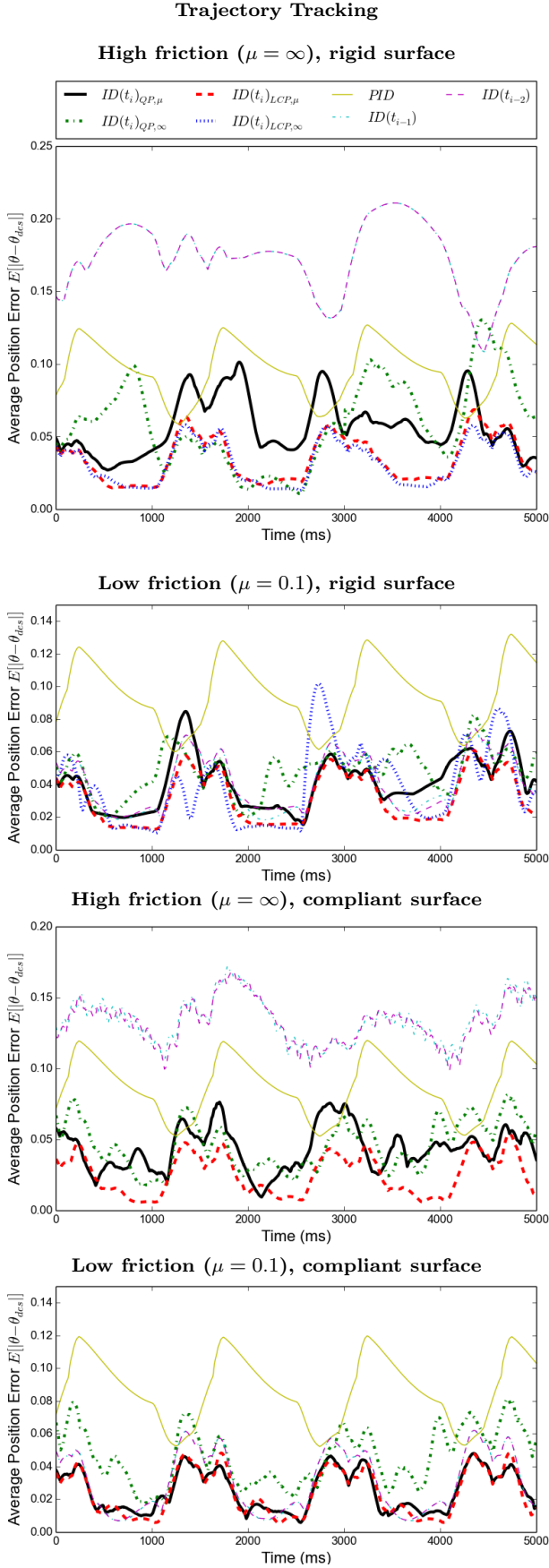
Fig. 9: *Snapshot of a quadruped robot in the* MOBY *simulator on rough terrain.*

6.6 Tasks

We simulate the robot trotting between a set of waypoints on a planar surface for 30 (virtual) seconds. The process of trotting to a waypoint and turning toward a new goal stresses some basic abilities needed to locomote successfully: (1) acceleration to and from rest; (2) turning in place, and (3) turning while moving forward. For the trotting gait we assigned gait duration of 0.3 seconds per cycle, duty-factor 75% of the total gait duration, a step height of 1.5 cm, and touchdown times $\{0.25, 0.75, 0.75, 0.25\}$ for {left front, right front, left hind, right hind} feet, respectively. These values were chosen as a generally functional set of parameters for a trotting gait on a 16 cm tall quadruped. The desired forward velocity of the robot over the test was 20 cm/s. Over the interval of the experiment, the robots are in both determinate and *possibly* indeterminate contact configurations and, as noted above, undergo numerous contact transitions.

7 Results

This section quantifies and plots results from the experiments in the previous section in five ways: (1) joint trajectory tracking; (2) accuracy of contact force prediction; (3) torque command smoothness; (4) center-of-mass behavior over a gait; (5) computation speed.

Trajectory Tracking

High friction ($\mu = \infty$), rigid surface

Low friction ($\mu = 0.1$), rigid surface

High friction ($\mu = \infty$), compliant surface

Low friction ($\mu = 0.1$), compliant surface

Fig. 10: *Average Position Error for all joints ($E[\|\theta - \theta_{des}\|]$) over time while performing a trotting gait.*

7.1 Trajectory tracking on planar surfaces:

We analyze tracking performance on both rigid planar and compliant planar surfaces (see Figure 10). Joint tracking data was collected from a simulated robot using the baseline, reference and experimental controllers to locomote on a planar terrain with varying surface properties. Numerical results for these experiments are presented in Table 3. The experimental controllers implementing contact force prediction, $\mathbf{ID}(t_i)$, either outperformed or matched the performance of the inverse dynamics formulations using sensed contact, $\mathbf{ID}(t_{i-1})$ and $\mathbf{ID}(t_{i-2})$.

As expected, the baseline controller (PID) performed substantially worse than all inverse dynamics systems for positional tracking on a low friction surface. Also, only the inverse dynamics controllers that use predictive contact managed to gracefully locomote with no-slip contact.

The reference inverse dynamics controllers with sensed contact performed the worst on high friction surfaces, only serving to degrade locomotion performance from the baseline controller over the course of the experiment. We assume that the performance of a well tuned PID error-feedback controller may be due to the control system absorbing some of the error introduced by poor planning, where more accurate tracking of planned trajectories may lead to worse overall locomotion stability.

7.2 Smoothness of torque commands
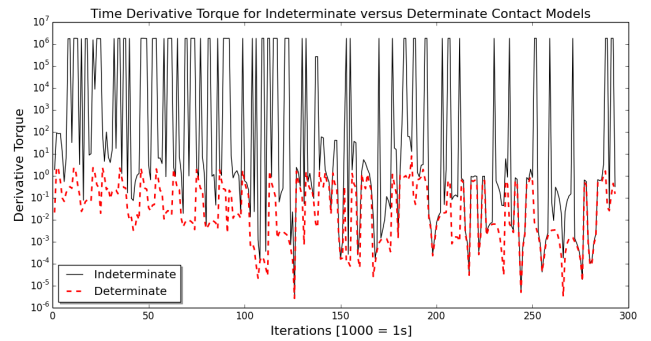
Fig. 11: *Time derivative torque when using our inverse dynamics method ($\mathbf{ID}(t_i)_{QP,\mu}$) with means for mitigating torque chatter from indeterminate contact (red/dotted) vs. no such approach (black/solid).*

Trajectory Tracking Error

Rigid, Low Friction				Rigid, High Friction		
Controller	positional error	velocity error		Controller	positional error	velocity error
$\mathbf{ID}(t_i)_{QP,\mu}$	0.0310	1.9425		$\mathbf{ID}(t_i)_{QP,\mu}$	0.0486	2.2596
$\mathbf{ID}(t_i)_{QP,\infty}$	0.0483	2.6295		$\mathbf{ID}(t_i)_{QP,\infty}$	0.0654	2.5737
$\mathbf{ID}(t_i)_{LCP,\mu}$	**0.0239**	1.8386		$\mathbf{ID}(t_i)_{LCP,\mu}$	**0.0259**	1.8784
$\mathbf{ID}(t_i)_{LCP,\infty}$	-	-		$\mathbf{ID}(t_i)_{LCP,\infty}$	0.0260	1.8950
PID	0.0895	**1.5569**		**PID**	0.0916	**1.4653**
$\mathbf{ID}(t_{i-1})$	0.0325	1.7952		$\mathbf{ID}(t_{i-1})$	0.1317	2.5585
$\mathbf{ID}(t_{i-2})$	0.0328	1.7830		$\mathbf{ID}(t_{i-2})$	0.1316	2.5608
Compliant, Low Friction				Compliant, High Friction		
Controller	positional error	velocity error		Controller	positional error	velocity error
$\mathbf{ID}(t_i)_{QP,\mu}$	**0.0217**	2.0365		$\mathbf{ID}(t_i)_{QP,\mu}$	0.0342	2.9360
$\mathbf{ID}(t_i)_{QP,\infty}$	-	-		$\mathbf{ID}(t_i)_{QP,\infty}$	0.0446	3.9779
$\mathbf{ID}(t_i)_{LCP,\mu}$	0.0219	2.0786		$\mathbf{ID}(t_i)_{LCP,\mu}$	**0.0226**	2.1243
$\mathbf{ID}(t_i)_{LCP,\infty}$	-	-		$\mathbf{ID}(t_i)_{LCP,\infty}$	-	-
PID	0.0850	**1.5845**		**PID**	0.0850	**1.5845**
$\mathbf{ID}(t_{i-1})$	0.0265	1.8858		$\mathbf{ID}(t_{i-1})$	0.1270	4.4377
$\mathbf{ID}(t_{i-2})$	0.0267	1.8742		$\mathbf{ID}(t_{i-2})$	0.1270	4.2061

Table 3: *Expected trajectory tracking error (positional: mean magnitude of radian error for all joints over trajectory duration ($E[E[|\theta - \theta_{des}|]]$), velocity: mean magnitude of radians/second error for all joints over trajectory duration ($E[E[|\dot{\theta} - \dot{\theta}_{des}|]]$)) of inverse dynamics controllers (**ID(..)**) and baseline (**PID**) controller.*

Torque Smoothness

Controller	$E[|\Delta\tau|]$	$E[|\tau|]$
$\mathbf{ID}(t_i)_{QP,\mu}$	**52.5718**	**0.2016**
$\mathbf{ID}(t_i)_{QP,\infty}$	245.5117	0.4239
$\mathbf{ID}(t_i)_{LCP,\mu}$	677.9693	0.8351
$\mathbf{ID}(t_i)_{LCP,\infty}$	351.1803	0.4055
$\mathbf{ID}(t_{i-1})$	974.3312	0.9918
$\mathbf{ID}(t_{i-2})$	17528.0	2.7416
PID	100.1571	0.3413

Table 4: *Average derivative torque magnitude (denoted $E[|\Delta\tau|]$) and average torque magnitude (denoted $E[|\tau|]$) for all controllers.*

Figure 11 shows the effects of an indeterminate contact model on torque smoothness. Derivative of torque commands are substantially smaller when incorporating a method of mitigating chatter in the inverse dynamics-derived joint torques. We observe a five order of magnitude reduction in the maximum of the time derivative torque when using a torque smoothing stage with our contact model. Controller $\mathbf{ID}(t_i)_{LCP,\mu}$ is the only presented controller unable to mitigate torque chatter (seen in the "indeterminate" case in Figure 11) and therefore produces the worst performance from our presented inverse dynamics methods. Though it demonstrates successful behavior in simulation, this method would likely not be suitable for use on a physical platform. The reference inverse dynamics controllers ($\mathbf{ID}(t_{i-1})$ and $\mathbf{ID}(t_{i-2})$) exhibit significant torque chatter also.

We measure the "smoothness" of torque commands as the mean magnitude of derivative torque over time. We gather from the data presented in Table 4 that the two phase QP-based inverse dynamics controller ($\mathbf{ID}(t_i)_{QP,\mu}$) followed by the baseline controller (**PID**)

are the most suitable for use on a physical platform. Controller $\mathbf{ID}(t_i)_{QP,\mu}$ uses the lowest torque to locomote while also mitigating sudden changes in torque that may damage robotic hardware.

7.3 Verification of correctness of inverse dynamics

We verify correctness of the inverse dynamics approaches by comparing the contact predictions against the reaction forces generated by the simulation. The comparison considers only the ℓ_1-norm of normal forces, though frictional forces are coupled to the normal forces (so ignoring the frictional forces is not likely to skew the results). We evaluate each experimental controller's contact force prediction accuracy given sticky and slippery frictional properties on rigid and compliant surfaces.

The QP-based controllers are able to predict the contact normal force in simulation to a relative error between 12–30%.[6] The $\mathbf{ID}(t_i)_{LCP,\mu}$, $\mathbf{ID}(t_i)_{LCP,\mu}$ controllers demonstrated contact force prediction between 1.16-1.94% relative error while predicting normal forces on a rigid surface (see Table 5). The QP based controllers performed as well on a compliant surface as they did on the rigid surfaces, while the performance of the $\mathbf{ID}(t_i)_{LCP,\mu}$, $\mathbf{ID}(t_i)_{LCP,\mu}$ controllers was substantially degraded on compliant surfaces.

The LCP-based inverse dynamics models ($\mathbf{ID}(t_i)_{LCP,\mu}$ and $\mathbf{ID}(t_i)_{LCP,\infty}$) use a contact model that matches

---

[6] The QP-based inverse dynamics models use a contact model that differs from the model used within the simulator. When the simulation uses the identical QP-based contact model, prediction exhibits approximately 1% relative error.
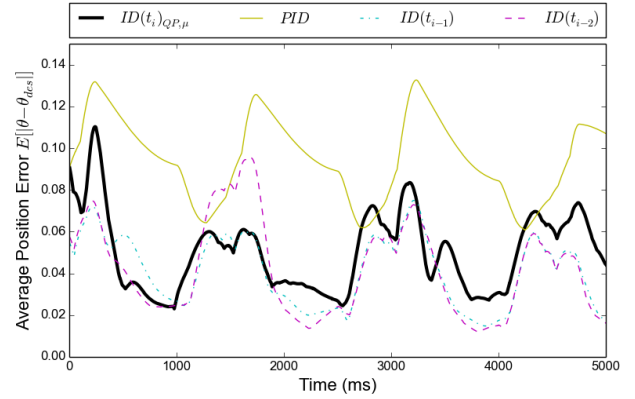
that used by the simulator. Nevertheless no inverse dynamics predictions always match the measurements provided by the simulator. Investigation determined that the slight differences are due to (1) occasionally inconsistency in the desired accelerations (we do not use the check described in Section 3.5); (2) the approximation of the friction cone by a friction pyramid in our experiments (the axes of the pyramid do not necessarily align between the simulation and the inverse dynamics model); and (3) the regularization occasionally necessary to solve the LCP (inverse dynamics might require regularization while the simulation might not, or *vice versa*).

Fig. 12: *Joint trajectory tracking for a robot on a rigid heightmap with uniform random friction* $\mu \sim \mathcal{U}(0.1, 1.5)$.

7.4 Controller behavior

The presented data support the utilization of the QP-based inverse dynamics model incorporating Coulomb friction, at least for purposes of control of real robots. We also observed that utilizing Coulomb friction in inverse dynamics leads to much more stable locomotion control on various friction surfaces. The no-slip contact models proved to be more prone to predicting excessive tangential forces and destabilizing the robot while not offering much additional performance for trajectory tracking. Accordingly, subsequent results are reported only for $\mathbf{ID}(t_i)_{QP,\mu}$ which will henceforth be referred to more generally as "the" inverse dynamics controller with contact force prediction or $\mathbf{ID}(t_i)$

7.5 Center-of-mass tracking performance

The ability of the controller to track the quadruped's center-of-mass over a path is a meta metric, as one expects this metric to be dependent upon joint tracking accuracy. Figure 13 shows that both the PID and the $\mathbf{ID}(t_i)$ methods are able to track straight line paths fairly well. $\mathbf{ID}(t_{i-1})$, which yielded better joint position tracking, does not track the center-of-mass as well. $\mathbf{ID}(t_{i-2})$ results in worse tracking with respect to both joint position and center-of-mass position. We hypothesize that this discrepancy is due to our observation that $\mathbf{ID}(t_{i-1})$ and $\mathbf{ID}(t_{i-2})$ yield significantly larger joint velocity tracking errors than the PID and $\mathbf{ID}(t_i)$ controllers.

7.5.1 Running time experiments

Rigid non-planar surface: Figure 9 plots performance with respect to tracking trajectories on the three reference controllers *vs.* the experimental control on rigid terrain with variable friction (ranging between low and high values of Coulomb friction for contacting materials as reported in literature). During this experiment only the ideal sensor controller $\mathbf{ID}(t_{i-1})$ consistently produced better positional tracking, $\mathbf{ID}(t_i)$ reduced tracking error over PID control by 19%.
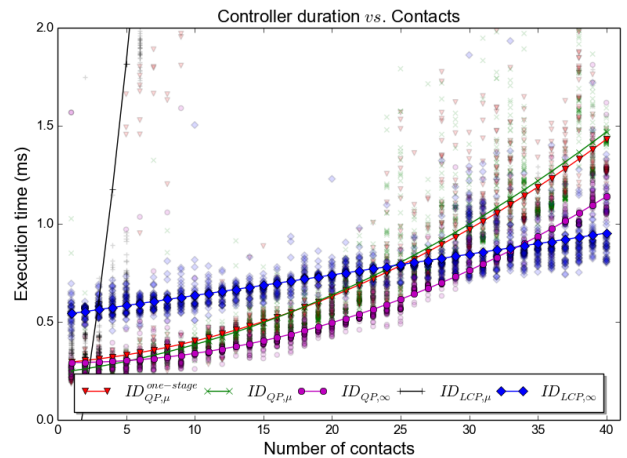

Fig. 14: *Inverse dynamics controller runtimes for increasing numbers of contacts (Quadruped with spherical feet).*

Contact Force Prediction Error

Rigid, Low Friction			Rigid, High Friction		
Controller	absolute error	relative error	Controller	absolute error	relative error
$\mathbf{ID}(t_i)_{QP,\mu}$	3.8009 N	12.53%	$\mathbf{ID}(t_i)_{QP,\mu}$	13.8457 N	27.48%
$\mathbf{ID}(t_i)_{QP,\infty}$	8.4567 N	22.26%	$\mathbf{ID}(t_i)_{QP,\infty}$	12.4153 N	25.26%
$\mathbf{ID}(t_i)_{LCP,\mu}$	0.9371 N	1.94%	$\mathbf{ID}(t_i)_{LCP,\mu}$	1.2768 N	1.55%
$\mathbf{ID}(t_i)_{LCP,\infty}$	-	-	$\mathbf{ID}(t_i)_{LCP,\infty}$	0.3572 N	1.16 %
Compliant, Low Friction			Compliant, High Friction		
Controller	absolute error	relative error	Controller	absolute error	relative error
$\mathbf{ID}(t_i)_{QP,\mu}$	7.8260 N	17.12%	$\mathbf{ID}(t_i)_{QP,\mu}$	14.9225 N	30.86%
$\mathbf{ID}(t_i)_{QP,\infty}$	-	-	$\mathbf{ID}(t_i)_{QP,\infty}$	15.1897 N	30.66%
$\mathbf{ID}(t_i)_{LCP,\mu}$	4.6385 N	6.37%	$\mathbf{ID}(t_i)_{LCP,\mu}$	12.4896 N	24.00%
$\mathbf{ID}(t_i)_{LCP,\infty}$	-	-	$\mathbf{ID}(t_i)_{LCP,\infty}$	-	-

Table 5: *Average contact force prediction error (summed normal forces) of inverse dynamics controllers* vs. *measured reaction forces from simulation. The robot weighs 47.0882 N under standard gravity. Results marked with a "-" indicate that the robot was unable to complete the locomotion task before falling.*
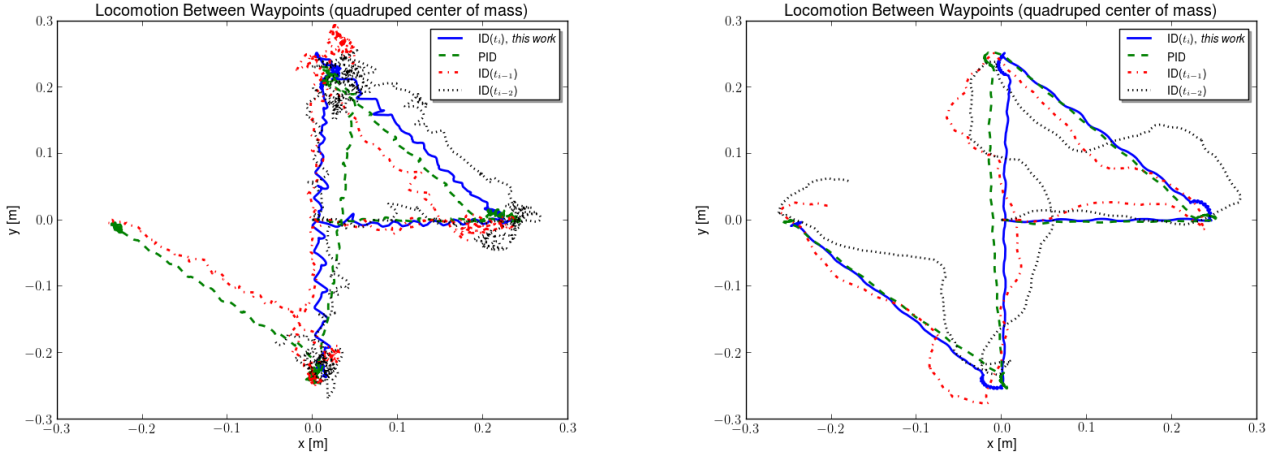
Fig. 13: *Center-of-mass path in the horizontal plane between waypoints over 30 seconds. (top) high friction; (bottom) low friction. The robot is commanded to follow straight line paths between points* $\{(0,0),(0.25,0),(0,0.25),(0,-0.25),(-0.25,0)\}$.

We simulated a biped standing on a plane with superfluous contacts added at the foot-ground contact manifold.[7] Figure 14 shows that inverse dynamics methods scale linearly with additional contacts. The fast pivoting algorithm ($\mathbf{ID}(t_i)_{LCP,\infty}$) can process in excess of 40 contacts while maintaining below a 1 ms runtime—capable of 1000 Hz control rate. The experimental QP-based controllers: $\mathbf{ID}(t_i)_{QP,\infty}$, $\mathbf{ID}(t_i)_{QP,\mu}^{\text{one-stage}}$, $\mathbf{ID}(t_i)_{QP,\mu}$ supported a control rate of 1000 Hz to about 30 total contacts when warm-starting the LCP solver with the previous solution. Controller $\mathbf{ID}(t_i)_{LCP,\mu}$ did not support warm-starting or the fast pivoting algorithm and was only able to maintain around a 1 ms expected runtime for fewer than 4 contacts. We observe that the runtime for $\mathbf{ID}(t_i)_{LCP,\infty}$ was substantially higher than the QP model, despite a significantly reduced problem size, for fewer than approximately 30 points of contact.

This disparity is due to the high computational cost of Lines 6 and 10 in Algorithm 1.

7.6 Discussion of inverse dynamics based control for legged locomotion

The inverse dynamics controller that predicts contact forces ($\mathbf{ID}(t_i)_{QP,\mu}$) performs well, at least in simulation, while mitigating destructive torque chatter. Over all tests, we observe the use of inverse dynamics control with predicted contact forces, *i.e.*, $\mathbf{ID}(t_i)$, can more closely track a joint trajectory than the alternatives—including inverse dynamics methods using sensed contact forces (even with perfect sensing) and PID control. The inverse dynamics controller described in this work $\mathbf{ID}(t_i)$ and the baseline PID controller, were able to track center-of-mass position of quadrupeds with little deviation from the desired direction of motion, while

the inverse dynamics controllers using sensed contact forces performed substantially worse at this task (as seen in Section 7.5). With these results in mind we find that contact force prediction has more potential for producing successful robot locomotion on terrain with varied surface properties. We expect that as sensor technology and computational speeds improve, contact force prediction for inverse dynamics control will even more greatly outperform methods dependent on force sensors.

8 Conclusion

We presented multiple, fast inverse dynamics methods—a method that assumes no slip (extremely fast), a QP-based method without complementarity (very fast), that same method with torque chattering mitigation (fast enough for real-time control loops at 1000Hz using current computational hardware on typical quadrupedal robots), and an LCP-based method that enforces complementarity (fast). We showed that a right inverse exists for the rigid contact model with complementarity, and we conducted asymptotic time complexity analysis for all inverse dynamics methods. Each method is likely well suited to a particular application. For example, Section 7 found that the last of these methods yields accurate contact force predictions (and therefore better joint trajectory tracking) for virtual robots simulated with the Stewart-Trinkle/Anitescu-Potra contact models. Finally, we assessed performance—running times, joint space trajectory tracking accuracy, and an indication of task execution capability (for legged locomotion)—under various contact modeling assumption errors.

9 Acknowledgements

Appendices

A Generalized contact wrenches

A contact wrench applied to a rigid body will take the form:

$$q \equiv \begin{bmatrix} \hat{q} \\ r \times \hat{q} \end{bmatrix} \tag{107}$$

where $\hat{q}$ is a vector in $\mathbb{R}^3$ and r is the vector from the center of mass of the rigid body to the point of contact

(which we denote p). For a multi-rigid body defined in m minimal coordinates, a *generalized contact wrench* $\mathbf{Q} \in \mathbb{R}^m$ for single point of contact p would take the form:

$$\mathbf{Q} = \mathbf{J}^\mathsf{T} q \tag{108}$$

where $\mathbf{J} \in \mathbb{R}^{6 \times m}$ is the manipulator Jacobian (see, *e.g.*, Sciavicco & Siciliano, 2000) computed with respect to p.

B LCPs and MLCPs

Unilateral contact problems are posed as complementarity problems, which are mathematical programming problems. This section describes the *linear complimentarily problem* (LCP), which can model rigid contact with a polygonal approximation to the friction cone.

The LCP, $(r, \mathbf{Q})$, where $r \in \mathbb{R}^a$ and $\mathbf{Q} \in \mathbb{R}^{a \times a}$ signifies:

$$w = \mathbf{Q}z + r \quad w \geq 0 \quad z \geq 0 \quad z^\mathsf{T}w = 0 \tag{109}$$

for unknown vectors $z, w \in \mathbb{R}^a$. A mixed linear complementarity problem (MLCP) is defined by the matrices $\mathbf{A} \in \mathbb{R}^{p \times s}$, $\mathbf{C} \in \mathbb{R}^{p \times t}$, $\mathbf{D} \in \mathbb{R}^{r \times s}$, $\mathbf{B} \in \mathbb{R}^{r \times t}$, $x \in \mathbb{R}^s$, $y \in \mathbb{R}^t$, $g \in \mathbb{R}^p$, and $h \in \mathbb{R}^r$ (where $p = s$ and $r = t$) and is subject to the following constraints:

$$\mathbf{A}x + \mathbf{C}z + g = 0 \tag{110}$$
$$\mathbf{D}x + \mathbf{B}z + h \geq 0 \tag{111}$$
$$z \geq 0 \tag{112}$$
$$z^\mathsf{T}(\mathbf{D}x + \mathbf{B}z + h) = 0 \tag{113}$$

The x variables are unconstrained, while the z variables must be non-negative. If $\mathbf{A}$ is non-singular, the unconstrained variables can be computed as:

$$x = -\mathbf{A}^{-1}(\mathbf{C}z + g) \tag{114}$$

Substituting x into Equations 110–113 yields the "pure" LCP $(r, \mathbf{Q})$:

$$\mathbf{Q} \equiv \mathbf{B} - \mathbf{D}\mathbf{A}^{-1}\mathbf{C} \tag{115}$$
$$r \equiv h - \mathbf{D}\mathbf{A}^{-1}g \tag{116}$$

A solution (z, w) to this LCP obeys the relationship $\mathbf{Q}z + r = w$; given z, x is determined via Equation 114, solving the MCLP.

C The Principle Pivoting Method for solving LCPs

The Principal Pivot Method I (Cottle, 1968; Murty, 1988) (PPM), which solves LCPs with P-matrices (complex square matrices with fully non-negative principal minors (Murty, 1988) that includes positive semi-definite matrices as a proper subset). The resulting algorithm limits the size of matrix solves and multiplications.

The PPM uses sets α, $\overline{\alpha}$, β, and $\overline{\beta}$ for LCP variables $\boldsymbol{z}$ and $\boldsymbol{w}$. The first two sets correspond to the $\boldsymbol{z}$ variables while the latter two correspond to the $\boldsymbol{w}$ variables. The sets have the following properties for an LCP of order n:

1. $\alpha \cup \overline{\alpha} = \{1, \ldots, n\}$
2. $\alpha \cap \overline{\alpha} = \emptyset$
3. $\beta \cup \overline{\beta} = \{1, \ldots, n\}$
4. $\beta \cap \overline{\beta} = \emptyset$

Of a pair of LCP variables, (z_i, w_i), index i will either be in α and $\overline{\beta}$ or β and $\overline{\alpha}$. If an index belongs to α or β, the variable is a *basic variable*; otherwise, it is a *non-basic variable*. Using this set, partition the LCP matrices and vectors as shown below:

$$\begin{bmatrix} \boldsymbol{w}_\beta \\ \boldsymbol{w}_{\overline{\beta}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\beta\alpha} & \mathbf{A}_{\beta\overline{\alpha}} \\ \mathbf{A}_{\overline{\beta}\alpha} & \mathbf{A}_{\overline{\beta}\overline{\alpha}} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_\alpha \\ \boldsymbol{z}_{\overline{\alpha}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{q}_\beta \\ \boldsymbol{q}_{\overline{\beta}} \end{bmatrix}$$

Isolating the basic and non-basic variables on different sides yields:

$$\begin{bmatrix} \boldsymbol{z}_{\overline{\alpha}} \\ \boldsymbol{w}_{\overline{\beta}} \end{bmatrix} = \begin{bmatrix} -\mathbf{A}_{\beta\overline{\alpha}}\mathbf{A}_{\beta\alpha} & \mathbf{A}_{\beta\overline{\alpha}}^{-1} \\ \mathbf{A}_{\overline{\beta}\alpha} - \mathbf{A}_{\overline{\beta}\overline{\alpha}}\mathbf{A}_{\beta\overline{\alpha}}^{-1}\mathbf{A}_{\beta\alpha} & \mathbf{A}_{\overline{\beta}\overline{\alpha}}\mathbf{A}_{\beta\overline{\alpha}}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_\alpha \\ \boldsymbol{w}_\beta \end{bmatrix} + \ldots$$
$$\begin{bmatrix} -\mathbf{A}_{\beta\overline{\alpha}}^{-1}\boldsymbol{q}_\beta \\ -\mathbf{A}_{\overline{\beta}\overline{\alpha}}\mathbf{A}_{\beta\overline{\alpha}}^{-1}\boldsymbol{q}_\beta + \boldsymbol{q}_{\overline{\beta}} \end{bmatrix}$$

If we set the values of the basic variables to zero, then solving for the values of the non-basic variables $\boldsymbol{z}_{\overline{\alpha}}$ and $\boldsymbol{w}_{\overline{\beta}}$ entails only computing the vector (repeated from above):

$$\begin{bmatrix} -\mathbf{A}_{\beta\overline{\alpha}}^{-1}\boldsymbol{q}_\beta \\ -\mathbf{A}_{\overline{\beta}\overline{\alpha}}\mathbf{A}_{\beta\overline{\alpha}}^{-1}\boldsymbol{q}_\beta + \boldsymbol{q}_{\overline{\beta}} \end{bmatrix} \tag{117}$$

PPM I operates in the following manner: (1) Find an index i of a basic variable x_i (where x_i is either w_i or z_i, depending which of the two is basic) such that $x_i < 0$; (2) swap the variables between basic and non-basic sets for index i (*e.g.*, if w_i is basic and z_i is non-basic, make w_i non-basic and z_i basic); (3) determine new values of $\boldsymbol{z}$ and $\boldsymbol{w}$; (4) repeat (1)–(3) until no basic variable has a negative value.

D Proof that no more than m positive force magnitudes need be applied along contact normals to a m degree of freedom multibody to solve contact model constraints

This proof will use the matrix of generalized contact wrenches, $\mathbf{N} \in \mathbb{R}^{n \times m}$ (introduced in Section 3.1), and $\mathbf{M} \in \mathbb{R}^{m \times m}$, the generalized inertia matrix for the multi-body. $\boldsymbol{z}_I$ is the vector of contact force magnitudes and consists of strictly positive values.

Assume we permute and partition the rows of $\mathbf{N}$ into r linearly independent and $n - r$ linearly dependent rows, denoted by indices I and D, respectively, as follows:

$$\mathbf{N} = \begin{bmatrix} \mathbf{N}_I \\ \mathbf{N}_D \end{bmatrix} \tag{118}$$

Then the LCP vectors $\boldsymbol{q} = \mathbf{N}\boldsymbol{v}$, $\boldsymbol{z} \in \mathbb{R}^n$, and $\boldsymbol{w} \in \mathbb{R}^n$ and LCP matrix $\mathbf{Q} = \mathbf{N}\mathbf{M}^{-1}\mathbf{N}^\mathsf{T}$ can be partitioned as follows:

$$\begin{bmatrix} \mathbf{Q}_{II} & \mathbf{Q}_{ID} \\ \mathbf{Q}_{DI} & \mathbf{Q}_{DD} \end{bmatrix} \begin{bmatrix} \boldsymbol{z}_I \\ \boldsymbol{z}_D \end{bmatrix} + \begin{bmatrix} \boldsymbol{q}_I \\ \boldsymbol{q}_D \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_I \\ \boldsymbol{w}_D \end{bmatrix} \tag{119}$$

Given some matrix $\boldsymbol{\gamma} \in \mathbb{R}^{(n-r) \times r}$, it is the case that $\mathbf{N}_D = \boldsymbol{\gamma}\mathbf{N}_I$, and therefore that $\mathbf{Q}_{DI} = \boldsymbol{\gamma}\mathbf{N}_I\mathbf{M}^{-1}\mathbf{N}_I^\mathsf{T}$, $\mathbf{Q}_{ID} = \mathbf{N}_I\mathbf{M}^{-1}\mathbf{N}_I^\mathsf{T}\boldsymbol{\gamma}^\mathsf{T}$ (by symmetry), $\mathbf{Q}_{DD} = \boldsymbol{\gamma}\mathbf{N}_I\mathbf{M}^{-1}\mathbf{N}_I^\mathsf{T}\boldsymbol{\gamma}^\mathsf{T}$, and $\boldsymbol{q}_D = \boldsymbol{\gamma}\mathbf{N}_I\boldsymbol{v}$.

Lemma 1 *Since $rank(\mathbf{NM}) \leq \min(rank(\mathbf{N}), rank(\mathbf{M}))$, the number of positive components of $\boldsymbol{z}_I$ can not be greater than $rank(\mathbf{N})$.*

Proof The columns of $\mathbf{NM}$ have $\mathbf{N}$ multiplied by each column of $\mathbf{M}$, *i.e.*, $\mathbf{NM} = \begin{bmatrix} \mathbf{N}\boldsymbol{m}_1 & \mathbf{N}\boldsymbol{m}_2 & \ldots & \mathbf{N}\boldsymbol{m}_m \end{bmatrix}$. Columns in $\mathbf{M}$ that are linearly dependent will thus produce columns in $\mathbf{NM}$ that are linearly dependent (with precisely the same coefficients). Thus, $\text{rank}(\mathbf{NM}) \leq \text{rank}(\mathbf{M})$. Applying the same argument to the transposes produces $\text{rank}(\mathbf{NM}) \leq \text{rank}(\mathbf{N})$, thereby proving the claim.

We now show that no more positive force magnitudes are necessary to solve the LCP in the case that the number of positive components of $\boldsymbol{z}_I$ is equal to the rank of $\mathbf{N}$.

Theorem 1 *If $(\boldsymbol{z}_I = \boldsymbol{a}, \boldsymbol{w}_I = \boldsymbol{0})$ is a solution to the LCP $(\boldsymbol{q}_I, \mathbf{Q}_{II})$, then $(\begin{bmatrix} \boldsymbol{z}_I^\mathsf{T} = \boldsymbol{a}^\mathsf{T} & \boldsymbol{z}_D^\mathsf{T} = \boldsymbol{0}^\mathsf{T} \end{bmatrix}^\mathsf{T}, \boldsymbol{w} = \boldsymbol{0})$ is a solution to the LCP $(\boldsymbol{q}, \mathbf{Q})$.*

Proof For $(\begin{bmatrix} \boldsymbol{z}_I^\mathsf{T} = \boldsymbol{a}^\mathsf{T} & \boldsymbol{z}_D^\mathsf{T} = \boldsymbol{0}^\mathsf{T} \end{bmatrix}^\mathsf{T}, \boldsymbol{w} = \boldsymbol{0})$ to be a solution to the LCP $(\boldsymbol{q}, \mathbf{Q})$, six conditions must be satisfied:

1. $\boldsymbol{z}_I \geq \boldsymbol{0}$

2. $\boldsymbol{w}_I \geq \boldsymbol{0}$
3. $\boldsymbol{z}_I{}^\mathsf{T}\boldsymbol{w}_I = 0$
4. $\boldsymbol{z}_D \geq \boldsymbol{0}$
5. $\boldsymbol{w}_D \geq \boldsymbol{0}$
6. $\boldsymbol{z}_D{}^\mathsf{T}\boldsymbol{w}_D = 0$

Of these, (1), (4), and (6) are met trivially by the assumptions of the theorem. Since $\boldsymbol{z}_D = \boldsymbol{0}$, $\mathbf{Q}_{II}\boldsymbol{z}_I + \mathbf{Q}_{ID}\boldsymbol{z}_D + \boldsymbol{q}_I = \boldsymbol{0}$, and thus $\boldsymbol{w}_I = \boldsymbol{0}$, thus satisfying (2) and (3). Also due to $\boldsymbol{z}_D = \boldsymbol{0}$, it suffices to show for (5) that $\mathbf{Q}_{DI}\boldsymbol{z}_I + \boldsymbol{q}_D \geq \boldsymbol{0}$. From above, the left hand side of this equation is equivalent to $\boldsymbol{\gamma}(\mathbf{N}_I\mathbf{M}^{-1}\mathbf{N}_I{}^\mathsf{T}\boldsymbol{a} + \mathbf{N}_I\boldsymbol{v})$, or $\boldsymbol{\gamma}\boldsymbol{w}_I$, which itself is equivalent to $\boldsymbol{\gamma}\boldsymbol{0}$. Thus, $\boldsymbol{w}_D = \boldsymbol{0}$.

References

A. Ames (2013). 'Human-Inspired Control of Bipedal Robotics via Control Lyapunov Functions and Quadratic Programs'. In *Proc. Intl. Conf. Hybrid Systems: Computation and Control.*

M. Anitescu (2006). 'Optimization-based simulation of nonsmooth dynamics'. *Mathematical Programming, Series A* **105**:113–143.

M. Anitescu & F. A. Potra (1997). 'Formulating Dynamic Multi-Rigid-Body Contact Problems with Friction as Solvable Linear Complementarity Problems'. *Nonlinear Dynamics* **14**:231–247.

D. Aukes & M. R. Cutkosky (2013). 'Simulation-Based Tools for Evaluating Underactuated Hand Designs'. In *Proc. IEEE Intl. Conf. Robotics Automation (ICRA)*, Karlsruhe, Germany.

D. Baraff (1994). 'Fast Contact Force Computation for Nonpenetrating Rigid Bodies'. In *Proc. of SIGGRAPH*, Orlando, FL.

R. Bhatia (2007). *Positive definite matrices.* Princeton University Press.

R. W. Bisseling & A. L. Hof (2006). 'Handling of impact forces in inverse dynamics'. *J. Biomech.* **39**(13):2438–2444.

W. Blajer, et al. (2007). 'Multibody modeling of human body for the inverse dynamics analysis of sagittal plane movements'. *Multibody System Dynamics* **18**(2):217–232.

S. Boyd & L. Vandenberghe (2004). *Convex Optimization.* Cambridge University Press.

A. Chatterjee (1999). 'On the realism of complementarity conditions in rigid-body collisions'. *Nonlinear Dynamics* **20**:159–168.

A. Chatterjee & A. Ruina (1998). 'A New Algebraic Rigid Body Collision Law Based on Impulse Space Considerations'. *ASME J. Appl. Mech.* **65**(4):939–951.

S. H. Collins, et al. (2001). 'A Three-Dimensional Passive-Dynamic Walking Robot with Two Legs and Knees'. *Intl. J. Robot. Res.* **20**(2).

R. W. Cottle (1968). 'The Principal Pivoting Method of Quadratic Programming'. In G. Dantzig & J. A. F. Veinott (eds.), *Mathematics of Decision Sciences*, pp. 144–162. AMS, Rhode Island.

R. W. Cottle, et al. (1992). *The Linear Complementarity Problem.* Academic Press, Boston.

E. Drumwright & D. A. Shell (2010). 'Modeling Contact Friction and Joint Friction in Dynamic Robotic Simulation using the Principle of Maximum Dissipation'. In *Proc. of Workshop on the Algorithmic Foundations of Robotics (WAFR).*

P. L. Fackler & M. J. Miranda. 'LEMKE'. http://people.sc.fsu.edu/ burkardt/m_src/lemke/lemke.m.

R. Featherstone (1987). *Robot Dynamics Algorithms.* Kluwer.

R. Featherstone (2008). *Rigid Body Dynamics Algorithms.* Springer.

S. Feng, et al. (2013). '3D Walking Based on Online Optimization'. In *Proc. IEEE-RAS Intl. Conf. on Humanoid Robots (HUMANOIDS)*, Atlanta, GA.

H. Hatze (2002). 'The fundamental problem of myoskeletal inverse dynamics and its implications'. *J. Biomech.* **35**(1):109–115.

R. Hunger (2007). 'Floating Point Operations in Matrix-Vector Calculus'. Tech. rep., TU München.

M. Hutter & R. Siegwart (2012). 'Hybrid Operational Space Control for Compliant Quadruped Robots'. In *Proc. Dynamic Walking.*

A. P. Ivanov (1995). 'On multiple impact'. *J. Applied Mathematics and Mechanics* **59**(6):887–902.

S. Kuindersma, et al. (2014). 'An Efficiently Solvable Quadratic Program for Stabilizing Dynamic Locomotion'. In *Proc. IEEE Intl. Conf. Robot. Autom. (ICRA).*

A. D. Kuo (1998). 'A least-squares estimation approach to improving the precision of inverse dynamics calculations'. *Trans. American Society Mech. Engineers (ASME) J. Biomech. Engr.* **120**:148–159.

C. E. Lemke (1965). 'Bimatrix Equilibrium Points and Mathematical Programming'. *Management Science* **11**:681–689.

K. Lynch & M. J. Mason (1995). 'Pushing by slipping, Slip with infinite friction, and perfectly rough surfaces'. *Intl. J. Robot. Res.* **14**(2):174–183.

O. L. Mangasarian & S. Fromovitz (1967). 'The Fritz John Necessary Optimality Conditions in the Presence of Equality and Inequality Constraints'. *J. Mathematical Analysis and Appl.* **17**:37–47.

B. Mirtich (1996). *Impulse-based Dynamic Simulation of Rigid Body Systems.* Ph.D. thesis, University of

California, Berkeley.

M. Mistry, et al. (2010). 'Inverse Dynamics Control of Floating Base Systems'. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pp. 3406–3412.

J. J. Moreau (1983). *Standard inelastic shocks and the dynamics of unilateral constraints*, pp. 173–221. Springer-Verlag, New York.

K. G. Murty (1988). *Linear Complementarity, Linear and Nonlinear Programming*. Heldermann Verlag, Berlin.

J. Nocedal & S. J. Wright (2006). *Numerical Optimization, 2nd ed.* Springer-Verlag.

C. Ott, et al. (2011). 'Posture and Balance Control for Biped Robots based on Contact Force Optimization'. In *Proc. IEEE-RAS Intl. Conf. on Humanoid Robots (HUMANOIDS)*.

P. Painlevé (1895). 'Sur le lois du frottement de glissement'. *C. R. Académie des Sciences Paris* **121**:112–115.

M. Posa & R. Tedrake (2012). 'Direct Trajectory Optimization of Rigid Body Dynamical Systems through Contact'. In *Proc. Workshop on Algorithmic Foundations of Robotics (WAFR)*, Boston.

L. Righetti, et al. (2013). 'Optimal distribution of contact forces with inverse-dynamics control'. *Intl. J. Robot. Res.* **32**(3):280–298.

L. Righetti, et al. (2011). 'Inverse Dynamics Control of Floating-Base Robots with External Constraints: A Unified View'. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Shanghai, China.

L. Saab, et al. (2013). 'Dynamic Whole-Body Motion Generation Under Rigid Contacts and Other Unilateral Constraints'. *IEEE Trans. Robotics* **29**(2).

A. B. Sawers & M. E. Hahn (2010). 'The Potential for Error with Use of Inverse Dynamic Calculations in Gait Analysis of Individuals with Lower Limb Loss: A Review of Model Selection and Assumptions'. *J. Prosthetics & Orthotics* **22**(1):56–61.

L. Sciavicco & B. Siciliano (2000). *Modeling and Control of Robot Manipulators, 2nd Ed.* Springer-Verlag, London.

B. Smith, et al. (2012). 'Reflections on Simultaneous Impact'. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* **31**(4):106:1–106:12.

B. Stephens & C. Atkeson (2010). 'Dynamic Balance Force Control for Compliant Humanoid Robots'. In *Proc. IEEE/RSJ Intl. Conf. Intelligent Robots and Systems (IROS)*.

D. Stewart & J. C. Trinkle (2000). 'An Implicit Time-Stepping Scheme for Rigid Body Dynamics with Coulomb Friction'. In *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA)*, San Francisco, CA.

D. E. Stewart (1998). 'Convergence of a time-stepping scheme for rigid-body dynamics and resolution of Painlevé's problem'. *Arch. Ration. Mech. Anal.* **145**:215–260.

D. E. Stewart (2000a). 'Rigid-body Dynamics with Friction and Impact'. *SIAM Review* **42**(1):3–39.

D. E. Stewart (2000b). 'Time-stepping methods and the mathematics of rigid body dynamics'. In A. Guran, B. Feeny, A. Klarbring, & Y. Ishida (eds.), *Impact and Friction of Solids, Structures, and Intelligent Machines*. World Scientific.

D. E. Stewart & J. C. Trinkle (1996). 'An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and Coulomb friction'. *Intl. J. Numerical Methods in Engineering* **39**(15):2673–2691.

T. Sugihara & Y. Nakamura (2003). 'Variable Impedant Inverted Pendulum Model Control for a Seamless Contact Phase Transition on Humanoid Robot'. In *Proc. IEEE Intl. Conf. Robot. Autom. (ICRA)*.

E. Todorov (2011). 'A convex, smooth and invertible contact model for trajectory optimization'. In *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Shanghai.

E. Todorov (2014). 'Analytically-invertible dynamics with contacts and constraints: theory and implementation in MuJoCo'. In *Proc. IEEE Intl. Conf. Robot. Autom. (ICRA)*.

J. Trinkle, et al. (1997). 'On Dynamic Multi-Rigid-Body Contact Problems with Coulomb Friction'. *Zeithscrift fur Angewandte Mathematik und Mechanik* **77**(4):267–279.

A. J. Van Den Bogert & A. Su (2008). 'A weighted least squares method for inverse dynamic analysis'. *Comput. Methods Appl. Mech. Eng.* **11**(1):3–9.

J. Yang, et al. (2007). 'An Inverse Dynamical Model for Slip Gait'. In *Proc. First Intl. Conf. Digital Human Modeling*, Beijing, China.

S. Zapolsky & E. Drumwright (2014). 'Quadratic Programming-based Inverse Dynamics Control for Legged Robots with Sticking and Slipping Frictional Contacts'. In *Proc. IEEE/RSJ Intl. Conf. Intell. Robots & Systems (IROS)*.

S. Zapolsky, et al. (2013). 'Inverse Dynamics for a Quadruped Robot Locomoting on Slippery Surfaces'. In *Proc. Intl. Conf. Climbing Walking Robots (CLAWAR)*, Sydney, Australia.

H. Zhao, et al. (2014). 'Human-inspired multi-contact locomotion with amber2'. In *ACM/IEEE, International Conference on Cyber Physics System*.