Inverse Dynamics for a Quadruped Robot Locomoting Along Slippery Surfaces

Samuel Zapolsky[1], Evan Drumwright[1]
Ioannis Havoutis[2], Jonas Buchli[3], Claudio Semini[2]

[1] *Positronics Lab,*
The George Washington University,
samzapo@gwmail.gwu.edu, drum@gwu.edu

[2] *Dep. of Advanced Robotics,*
Istituto Italiano di Tecnologia (IIT)
ioannis.havoutis@iit.it, claudio.semini@iit.it

[3] *Agile & Dexterous Robotics Lab,*
ETH Zurich
buchlij@ethz.ch

We describe a method for computing inverse dynamics forces suitable for robots locomoting along (or manipulating) sticky and slippery surfaces. Our method assumes rigid body dynamics, rigid body contact, and Coulomb friction, and is sufficiently fast for computation in real-time servos. We assess our method's performance on the HyQ quadrupedal robot in both simulation and *in situ*.

Keywords: Inverse dynamics; contact modeling; slipping locomotion.

1. Motivation

Strategies for controlling robots locomoting across terrain are generally predicated on sufficient traction. Locomotion across slippery surfaces (ice, grease, sand, *etc.*) can endanger the robot and its surroundings. We address this problem using an inverse dynamics (ID) controller that models variable surface friction in order to both minimize slipping and leverage high friction conditions (as available). This controller is able to utilize surface data—whether provided *a priori* or measured via sensory perception—to maximize robot locomotion performance on mixed sticky-slippery surfaces; the controller is even able to account for surfaces varying at the robot's individual feet. Our method employs quadratic programming (QP) models yet runs sufficiently quickly for use in hydraulic robot control loops

Notation

$\boldsymbol{m}$ the number of rigid bodies contacting the robot

$\boldsymbol{n}$ the number of contact points between the robot and the rigid bodies

$\boldsymbol{w}$ the number of generalized coordinates in the *system* (the robot and contacting bodies)

$\boldsymbol{fb}$ designation for the wrench or twist on the robot's floating base

$\boldsymbol{cb}$ designation for the m wrenches or twists corresponding to the contacting bodies

$\boldsymbol{c}_N$ the impulses to be applied to the contact points in the normal direction

$\boldsymbol{c}_D$ the impulses to be applied to the contact points in the tangential directions

$\mathbf{M}$ the generalized inertia matrix of the *system*

$\boldsymbol{v}$ the current generalized velocity of the *system*

$\boldsymbol{v}^*$ the generalized velocity after the contact model and external wrenches are applied

$\dot{\boldsymbol{q}}$ the current robot joint velocity

$\ddot{\boldsymbol{q}}$ the desired robot joint acceleration

$\boldsymbol{\tau}$ the vector of ID torques to be applied by the robot's actuators

$\boldsymbol{f}_{\mathbf{ext}}$ the vector of generalized external forces of the *system*

$\mathbf{N}$ the $\mathbb{R}^{w \times n}$ matrix of normal generalized contact wrenches of the *system*

$\mathbf{D}$ the $\mathbb{R}^{w \times r \cdot n}$ matrix of tangential generalized direction contact wrenches of the *system*; each contact uses r vectors in the tangent plane[5]

μ the coefficient of friction

$h \ll 1$ first-order approximation scalar

and generates chatter-free torque profiles. We verify results using the HyQ quadrupedal robot platform.[1]

2. Inverse Dynamics Computation

We focus on the case where the desired robot acceleration is consistent with the contact constraints (*e.g.*, a robot is not commanded to accelerate its foot into a door). In computing ID, we assume that robots are modeled using rigid body dynamics with Coulomb contact friction. It is well known that—at the acceleration-level—rigid bodies in contact are inconsistent with the Coulomb friction model.[2] In order to admit solutions to arbitrary contact configurations, we will use the standard technique of moving to the velocity-level with *impulsive* forces (see, *e.g.*,[3]). We model rigid body contact using a convex contact model[4] with an approximate polygonal friction cone[5] in order to use fast pivoting QP algorithms. The process of determining the contact impulses that satisfy inverse dynamics and generate stable joint torques is a two-stage process, described by the convex QPs below.

$$\textbf{Stage I:} \min_{\boldsymbol{c}_N, \boldsymbol{c}_D} \ \frac{1}{2}\boldsymbol{v}^{*\mathsf{T}}\mathbf{M}\boldsymbol{v}^* \quad \text{(by principle of maximum dissipation)} \tag{1}$$

$$\mathbf{N}^{\mathsf{T}}\boldsymbol{v}^* \geq \mathbf{0} \quad \text{(non-interpenetration)} \tag{2}$$

$$\boldsymbol{c}_N \geq \mathbf{0}, \quad \boldsymbol{c}_D \geq \mathbf{0} \quad \text{(compressive force/polygon)} \tag{3}$$

$$\mu c_{N,i} \geq \mathbf{1}^{\mathsf{T}}\boldsymbol{c}_{D,i} \quad \text{(Coulomb friction)} \tag{4}$$

$$\boldsymbol{v}^*_{\text{robot.q}} = \dot{\boldsymbol{q}} + h\ \ddot{\boldsymbol{q}} \quad \text{(inverse dynamics)} \tag{5}$$

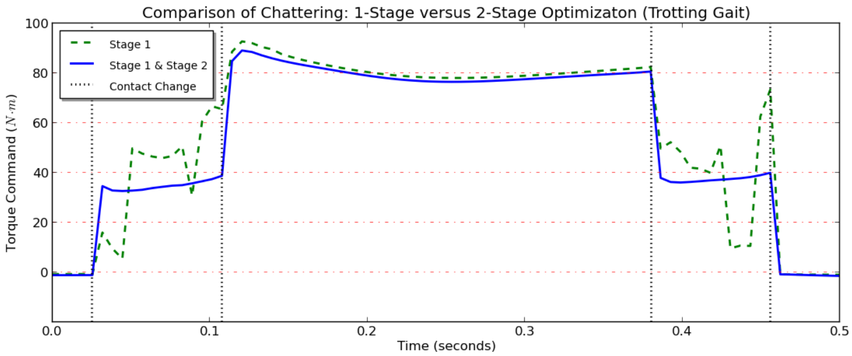


Fig. 1: *Comparison of torque profiles using only Stage I and both Stage I and Stage II. Proceeding chronologically, contact conditions are 1) three feet in contact; 2) four feet in contact; 3) two feet in contact; 4) four feet in contact; and 5) three feet in contact.*

$$\mathbf{M}(\boldsymbol{v}^* - \boldsymbol{v}) = \mathbf{N}\boldsymbol{c}_N + \mathbf{D}\boldsymbol{c}_D + h\ (\boldsymbol{f}_{\text{ext}} + \begin{bmatrix}\mathbf{0}\\ \boldsymbol{\tau}\end{bmatrix}) \tag{6}$$

$$\textbf{Stage II: } \min_{\boldsymbol{c}_N, \boldsymbol{c}_D} \ \frac{1}{2}\boldsymbol{\tau}^{\mathsf{T}}\boldsymbol{\tau} \quad \text{(chatter-free torque generation)} \tag{7}$$

$$\text{subject to} \ \ \textbf{Constraints } (2) - (6) \tag{8}$$

$$\text{and such that} \ \ \frac{1}{2}{\boldsymbol{v}^*}^{\mathsf{T}}\mathbf{M}\boldsymbol{v}^* = \textbf{stage 1 objective} \quad \text{(energy)} \tag{9}$$

where $\boldsymbol{v}^* = \left[{\boldsymbol{v}^*_{\text{robot.fb}}}^{\mathsf{T}}\ {\boldsymbol{v}^*_{\text{robot.q}}}^{\mathsf{T}}\ {\boldsymbol{v}^*_{\text{cb}}}^{\mathsf{T}}\right]^{\mathsf{T}}$ and $\boldsymbol{f}_{\text{ext}} = \left[{\boldsymbol{f}_{\text{fb}}}^{\mathsf{T}}\ \boldsymbol{\tau}^{\mathsf{T}}\ {\boldsymbol{f}_{\text{cb}}}^{\mathsf{T}}\right]^{\mathsf{T}}$.

2.1. *Computing inverse dynamics when the desired robot acceleration is consistent with contact model (Stage I)*

Our method simultaneously computes the coupled contact and ID forces of an articulated rigid body system. The time complexity of solving the resulting convex QP is $O(n^3)$ in the number of contact points.[6] The QP for the contact model provably is always solvable, and the QP for ID is always solvable as well (predicated on consistency of desired robot acceleration with the contact model). The resulting QP is described by Eqns. 1–6.

2.2. *Simplifying the computation*

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

$$\boldsymbol{v}^* = \boldsymbol{v} + \mathbf{M}^{-1}(\mathbf{N}\boldsymbol{c}_N + \mathbf{D}\boldsymbol{c}_D + h\boldsymbol{f}_{ext} + \begin{bmatrix}\mathbf{0}\\ h(\boldsymbol{f}_{ID} + \boldsymbol{x})\end{bmatrix}) \tag{10}$$

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:

$$\mathbf{R} \triangleq \begin{bmatrix}\mathbf{N}\ \mathbf{D}\end{bmatrix} \tag{11}$$

$$\boldsymbol{z} \triangleq \begin{bmatrix}\boldsymbol{c}_N\ \boldsymbol{c}_D\end{bmatrix}^{\mathsf{T}} \tag{12}$$

$$\mathbf{M} \triangleq \begin{bmatrix}\mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\mathsf{T}} & \mathbf{C}\end{bmatrix} \tag{13}$$

$$\mathbf{M}^{-1} \triangleq \begin{bmatrix}\mathbf{D} & \mathbf{E} \\ \mathbf{E}^{\mathsf{T}} & \mathbf{F}\end{bmatrix} \tag{14}$$

$$\boldsymbol{j} \triangleq \boldsymbol{v}_b + \begin{bmatrix}\mathbf{D}\ \mathbf{E}\end{bmatrix}(h\boldsymbol{f}_{ext} + \begin{bmatrix}\mathbf{0} \\ h\boldsymbol{f}_{ID}\end{bmatrix}) \tag{15}$$

$$\boldsymbol{k} \triangleq \boldsymbol{v}_q + \begin{bmatrix}\mathbf{E}^{\mathsf{T}}\ \mathbf{F}\end{bmatrix}(h\boldsymbol{f}_{ext} + \begin{bmatrix}\mathbf{0} \\ h\boldsymbol{f}_{ID}\end{bmatrix}) \tag{16}$$

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

$$\boldsymbol{v}_b^* = \boldsymbol{j} + \begin{bmatrix}\mathbf{D}\ \mathbf{E}\end{bmatrix}(\mathbf{R}\boldsymbol{z} + \begin{bmatrix}\mathbf{0} \\ h\boldsymbol{x}\end{bmatrix}) \tag{17}$$

$$\boldsymbol{v}_q^* = \boldsymbol{k} + \begin{bmatrix}\mathbf{E}^{\mathsf{T}}\ \mathbf{F}\end{bmatrix}(\mathbf{R}\boldsymbol{z} + \begin{bmatrix}\mathbf{0} \\ h\boldsymbol{x}\end{bmatrix}) = \boldsymbol{v}_q + h\boldsymbol{a}_q \tag{18}$$

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

$$\boldsymbol{x} = \frac{\mathbf{F}^{-1}(\boldsymbol{v}_q^* - \boldsymbol{k} - \begin{bmatrix}\mathbf{E}^{\mathsf{T}}\ \mathbf{F}\end{bmatrix}\mathbf{R}\boldsymbol{z})}{h} \tag{19}$$

Equation 19 indicates that once contact forces are computed, determining the actuator forces for ID requires solving only a linear equation. Substituting the solution for $\boldsymbol{x}$ into Eqn. 17 yields:

$$\boldsymbol{v}_b^* = \boldsymbol{j} + \begin{bmatrix}\mathbf{D}\ \mathbf{E}\end{bmatrix}\mathbf{R}\boldsymbol{z} + \mathbf{E}\mathbf{F}^{-1}(\boldsymbol{v}_q^* - \boldsymbol{k} - \begin{bmatrix}\mathbf{E}^{\mathsf{T}}\ \mathbf{F}\end{bmatrix}\mathbf{R}\boldsymbol{z}) \tag{20}$$

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

$$\mathbf{Z} \triangleq \begin{bmatrix}\mathbf{D}\ \mathbf{E}\end{bmatrix} - \mathbf{E}\mathbf{F}^{-1}\begin{bmatrix}\mathbf{E}^{\mathsf{T}}\ \mathbf{F}\end{bmatrix}\mathbf{R} \tag{21}$$

$$\boldsymbol{p} \triangleq \boldsymbol{j} + \mathbf{E}\mathbf{F}^{-1}(\boldsymbol{v}_q^* - \boldsymbol{k}) \tag{22}$$

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

$$\boldsymbol{v}_b^* = \mathbf{Z}\boldsymbol{z} + \boldsymbol{p} \tag{23}$$

We now represent the objective function (Eqn. 1) in block form as:

$$f(.) \triangleq \frac{1}{2}\begin{bmatrix}\boldsymbol{v}_b^* \\ \boldsymbol{v}_q^*\end{bmatrix}^{\mathsf{T}}\begin{bmatrix}\mathbf{A} & \mathbf{B} \\ \mathbf{B}^{\mathsf{T}} & \mathbf{C}\end{bmatrix}\begin{bmatrix}\boldsymbol{v}_b^* \\ \boldsymbol{v}_q^*\end{bmatrix} \tag{24}$$

which is identical to:

$$f(.) \triangleq \frac{1}{2}\boldsymbol{v}_b^{*\mathsf{T}}\mathbf{A}\boldsymbol{v}_b^* + \boldsymbol{v}_b^*\mathbf{B}^{\mathsf{T}}\boldsymbol{v}_q^* + \frac{1}{2}\boldsymbol{v}_q^{*\mathsf{T}}\mathbf{C}\boldsymbol{v}_q^* \tag{25}$$

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, we can ignore that term. Expanding remaining terms using Eqn. 17, the new objective function is:

$$\begin{aligned} f(.) &\triangleq \frac{1}{2}\boldsymbol{z}^{\mathsf{T}}\mathbf{Z}^{\mathsf{T}}\mathbf{A}\mathbf{Z}\boldsymbol{z} + \boldsymbol{z}^{\mathsf{T}}\mathbf{Z}^{\mathsf{T}}\mathbf{A}\boldsymbol{p} + \boldsymbol{z}^{\mathsf{T}}\mathbf{Z}^{\mathsf{T}}\mathbf{B}\boldsymbol{v}_q^* & (26)\\ &\triangleq \frac{1}{2}\boldsymbol{z}^{\mathsf{T}}\mathbf{Z}^{\mathsf{T}}\mathbf{A}\mathbf{Z}\boldsymbol{z} + \boldsymbol{z}^{\mathsf{T}}(\mathbf{Z}^{\mathsf{T}}\mathbf{A}\boldsymbol{p} + \mathbf{Z}^{\mathsf{T}}\mathbf{B}\boldsymbol{v}_q^*) & (27) \end{aligned}$$

subject to the following constraints:

$$\mathbf{N}^{\mathsf{T}}\begin{bmatrix}\mathbf{Z}\boldsymbol{z}+\boldsymbol{p}\\ \boldsymbol{v}_q^*\end{bmatrix} \geq 0 \tag{28}$$

$$\boldsymbol{z}_i \ \geq\ 0 \qquad (\text{for } i = 1\ldots n) \tag{29}$$

$$\mu\boldsymbol{z}_i \ \geq\ z_{n+i} + \ldots + z_{n+i+r-1} \qquad (\text{for } i = 1\ldots n) \tag{30}$$

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, $\boldsymbol{x}$ can be recovered. The actuator forces determined via ID are then $\boldsymbol{f}_{ID}+\boldsymbol{x}$.

2.3. *Recomputing Inverse Dynamics to Stabilize Actuator Torques (Stage II)*

The optimization perfomed by Stage I is prone to selecting an ID solution in which only two legs receive the majority of joint torque; this artifact arises from the redundancy inherent in robots with three or more points of support. Such solutions may be acceptable in simulations but can produce control torque discontinuities harmful to physically situated robots. Stage II addresses this issue by selecting the joint torque with minimum ℓ_2-norm. We define the optimization variables of Stage II within the null-space of Eq. 1 (*i.e.*, optimizing in the null-space of system kinetic energy) to obviate inclusion of a quadratic constraint. Once Stage II has been solved, the desired actuator forces are recovered with matrix-vector arithmetic. The two-stage optimization results in joint torques that are safe for the robot.

3. Experiments

Simulation data was collected using SL.[8] We evaluated our optimization procedure using multiple experimental setups including a high friction surface (the control) and a comparatively low friction experimental surface.

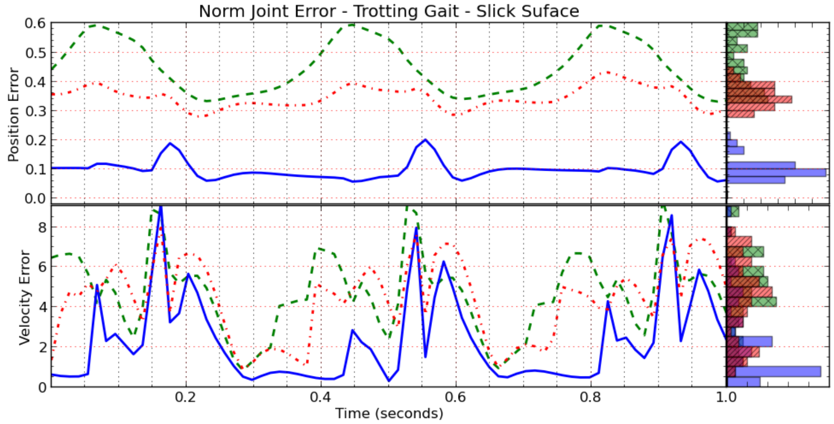
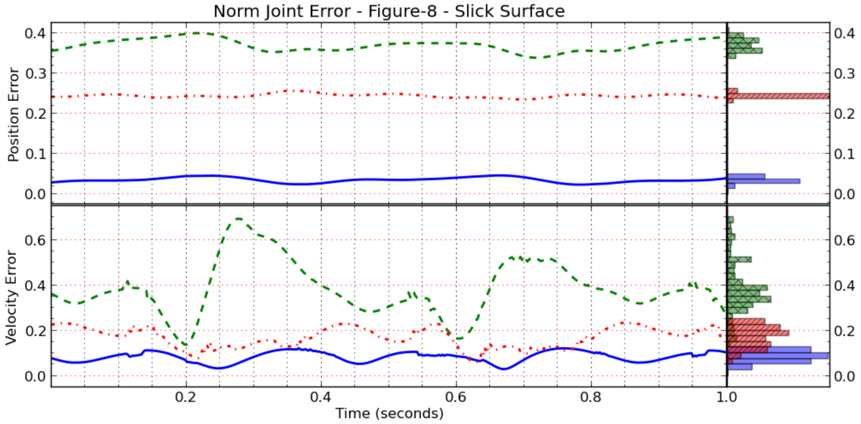


Fig. 2: *Norm of joint tracking error, comparing PID control* (green, double dash) *to our implementation of ID-based control* (blue, solid) *and an existing implementation*[7] *of non-slipping ID-based control* (red, dash-dot-dash) *on three steps of a walking trot (above) and one cycle of a Figure-Eight (below), both on surfaces with* $\mu = 0.1$. *A histogram of each plot's distribution is provided at right. Data was collected from the SL simulator.*



We used a trotting gait[9] as the reference trajectory for locomotion trials, and we also employed a Figure-Eight task to avoid the issue of impact (our model assumes only "resting" contact or inelastic impact). We compared the performance of the robot between three control methods: a feedback-based control strategy, an existing (non slipping) ID controller,[7] and our ID method. These trials aimed to validate both the inclusion of Stage II into the optimization process (evidenced by Fig. 1) and performance increases on slippery and sticky surfaces (evidenced by Figs. 2 and 3, respectively).

4. Results and Discussion

Mean position errors while performing a trot on simulated surface with $\mu = 0.1$ are reduced by 77.08% using our ID method over PID control. This performance differential exists even with the compliant (penalty-method-based) contact model used in SL (our ID model assumes rigid contact). Resulting tracking errors taken from the Figure-Eight reference trajectory on the slippery surface are even more compelling; position error was reduced 90.33% using our ID controller in place of PID control and 56.30% using our ID control in place of the existing (non slipping) ID control. Figure 3 shows that control accuracy on sticky surfaces is nearly identical between the two ID methods, as should be expected.

We also performed these experiments identically *in situ*, though these results were not as compelling. Neither ID controller was able to reduce position error substantially over PID control, though analysis indicates that the torques output by the three controllers are generally of the same direction and order of magnitude. Further analysis is likely to identify significant unmodeled phenomena, erroneous sensory data, or both.

Solving QPs and significant numerical linear algebra would seem to preclude the use of our approach in ID computation. Over sixty seconds of trotting, however, the two-stage optimization process completed in a mean time of 1ms and a worst-case time of 36ms. Thus, our method permits effective control rates of 250Hz or greater.

Acknowledgements

Zapolsky and Drumwright were funded by NSF award CMMI-110532.

References

1. C. Semini, N. G. Tsagarakis, E. Guglielmino, M. Focchi, F. Cannella and D. G. Caldwell, *Journal of Systems and Control Engineering* **225**, 831 (2011).
2. D. Stewart, *SIAM Rev.* **42**, 3(Mar 2000).
3. M. Anitescu and F. A. Potra, *Nonlin. Dyn.* **14**, 231 (1997).
4. E. Drumwright and D. A. Shell, Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation, in *Springer Tracts in Advanced Robotics: Algorithmic Foundations of Robotics IX: Selected Contributions of the Ninth International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2010.
5. D. Stewart and J. Trinkle, *Intl. J. Num. Meth. Engr.* **39**, 2673 (1996).
6. S. Boyd and L. Vandenberghe, *Convex Optimization* (Cambridge University Press, 2004).
7. M. Mistry, J. Buchli and S. Schaal, Inverse dynamics control of floating base

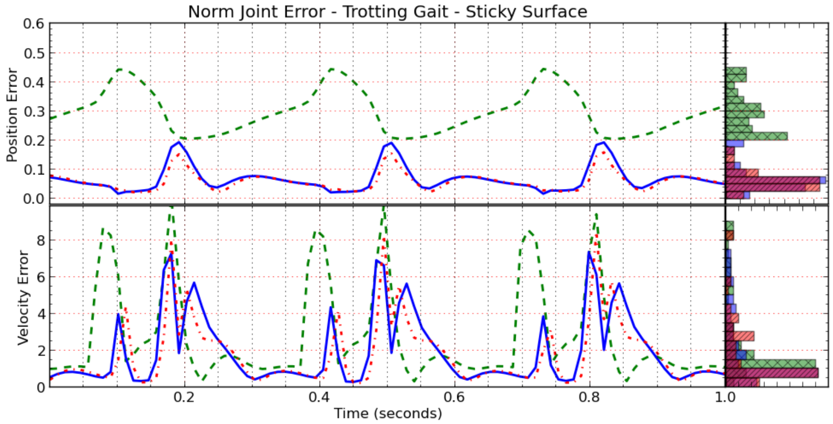
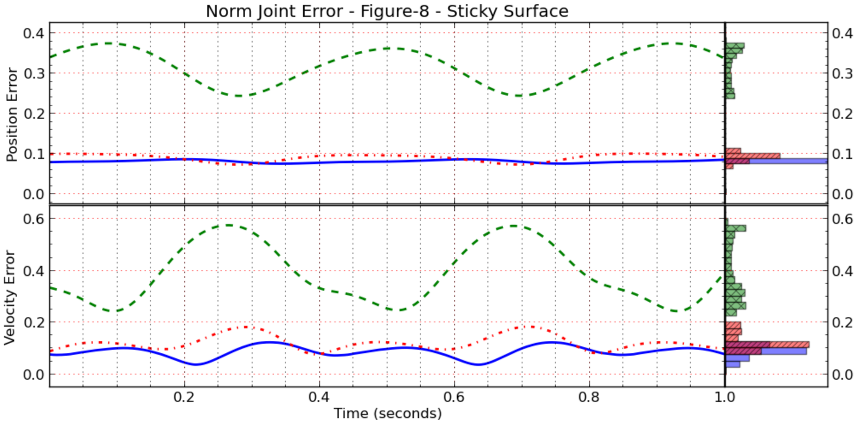


Fig. 3: *Norm of joint tracking error, comparing PID control* (green, double dash) *to our implementation of ID-based control* (blue, solid) *and an existing implementation*[7] *of non-slipping ID-based control* (red, dash-dot-dash) *on three steps of a walking trot (above) and one cycle of a Figure-Eight (below), both on surfaces with* $\mu = 1.0$. *A histogram of each plot's distribution is provided at right. Data was collected from the SL simulator.*



systems using orthogonal decomposition, in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2010.

8. S. Schaal, *The SL simulation and real-time control software package*, tech. rep., Univ. Southern California (2009).
9. V. Barasuol, J. Buchli, C. Semini, M. Frigerio, E. R. De Pieri and D. G. Caldwell, A reactive controller framework for quadrupedal locomotion on challenging terrain, in *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2013.