ESE 618: Learning for Dynamics and Control 1–13

Online Least Squares for Convex MPC of Biped Robots - Final Report

Brian Acosta BJACOSTA@SEAS.UPENN.EDU
Philadelphia, PA

Hersh Sanghvi HSANGHVI@SEAS.UPENN.EDU
Philadelphia, PA

Abstract

Contemporary bipedal robots such as Cassie have dozens of degrees of freedom which allow them to execute agile maneuvers and traverse rugged terrain. Planning these motions can be challenging, as the center of mass motion must result from proper foot placement and physically valid contact forces. To address this complexity, state of the art methods for online motion planning find trajectories for reduced order models of the robot. While these simple models reduce performance by limiting the degrees of freedom of the robot, venturing away from simple models leads to nonlinear trajectory optimization problems which are intractable to solve in real time. We seek to resolve this tension between model expressiveness and planning speed by planning with a global affine approximation of the nonlinear single rigid body dynamics model, and improving the accuracy of this model by online learning of time-varying residuals to the convex approximation. We analyze how perception error could affect controller performance by perturbing the switching time of the hybrid robot dynamics, and how online dynamics learning might aid in recovering from this error.

1. Introduction, Background, and Related Work

In this project, we aim to develop a Model Predictive Controller (MPC) that uses a reduced-order model of the dynamics of a legged robot and uses learning to make the model more accurate. We also incorporate perception into the controller, and analyze how errors in perception influence the dynamics estimator. This is an ambitious project which touches many areas of the robotics literature, so we will first give an overview of modern approaches to similar perception and control problems. We begin with an overview of dynamical systems models for legged locomotion, then explore how these models are used for real-time control, and then finally consider how perception is used to assist in choosing acceptable foot placement locations.

1.1. Models of Legged Locomotion

Biped robots are commonly modeled as hybrid Lagrangian systems which, during each contact mode, obey the manipulator equations

$$M(q)\dot{v} + C(q,v)v + g(q) = Bu + J^T\lambda. \tag{1}$$

Given the computational intractability of realtime trajectory optimization subject to (1), state of the art online planning methods generally use models like the linear inverted pendulum (LIP) (Kajita et al. (2001)) or spring loaded inverted pendulum (SLIP) (Blickhan (1989)) as reduced order models. These models assume that contact forces travel through the center of mass, and therefore the robot has no angular momentum about it's center of mass, restricting the class of achievable

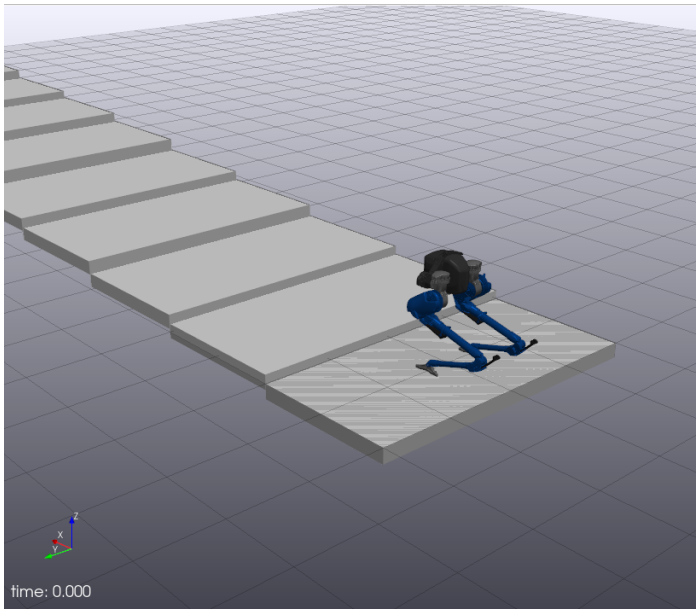

Figure 1: Our simulation environment for the Cassie biped

motions considerably. The single rigid body model commonly used for quadrupeds (Di Carlo et al. (2018)) allows more dynamic motions and can allow for convex trajectory optimization under mild assumptions, but applying such a model to biped robots with massive legs is an open area of robotics research.

1.2. Optimization-based Planning and Control for Legged Locomotion

Online motion planners for bipeds typically plan over foot placements using the step to step dynamics of the LIP or similar reduced order models (Gibson et al. (2021), Chen and Posa (2020)) using a quadratic program or very small nonlinear program. The continuous dynamics and swing foot location are then tracked using an operational space controller similar to the formulation introduced by Wensing and Orin (2013), which considers the full manipulator equations when solving for instantaneous accelerations subject to feasible contact forces. On the other hand, controllers using the single rigid body model often convert the MPC contact force solution directly to joint torques using the contact jacobian, or track the contact force using whole body control (Kim et al. (2019)). Here, we use the former technique, since tracking the pelvis orientation directly is needed to ensure stability of the robot's orientation.

1.3. Perception for Locomotion

While blind walking and running controllers have achieved highly robust legged locomotion over a variety of terrains (Green et al. (2020), Gong et al. (2018), Siekmann et al. (2021)), there are some scenarios in which perception is needed. For example, legged robots must avoid placing their feet in potholes, or raise their swing feet higher than normal to traverse steep steps. A number of approaches have been proposed in the literature to integrate planning into the legged controls stack. Kim et al. (2020) build a point cloud from onboard perception, and analyze the gradients of the

terrain to determine regions in which the robot cannot step. Using this information, they adjust their nominal gait so that footsteps do not fall in untraversable regions. Villarreal et al. (2020) use a similar approach, except with a learned cost map which incorporates heuristics such as terrain gradients and height. In addition to these, a variety of supervised- and reinforcement- learning based approaches have been proposed (Gangapurwala et al. (2020), Yu et al. (2021), Sanghvi and Taylor (2021)).

2. Methods

In this section, we outline the various methods we used in this project. We will cover the approximations we make of the true underlying dynamics, how we learn time-varying residual models to make those models more accurate, how we use those dynamics to formulate a convex optimization problem for Model Predictive Control (MPC), and how we incorporate perception into this framework.

2.1. Single Rigid Body Dynamics

We define the state variables $\boldsymbol{p}$, $\boldsymbol{\theta}$, $\dot{\boldsymbol{p}}$, $\boldsymbol{\omega}$, to be the floating base position (which we assume is coincident with the center of mass), body fixed Z-Y-X Euler angles, floating base translational velocity, and floating base angular velocity. We denote by $\boldsymbol{r}$ the position of the stance foot in each contact mode. We take the control input to be the contact force $\boldsymbol{f}$ and a moment ($\boldsymbol{\tau}_{yz}$) in the $y-z$ plane, as in Li and Nguyen (2021), since Cassie has no actuation about the foot $x-$axis. We make the approximations from Di Carlo et al. (2018) on small pitch and roll angles, and low angular velocity. We also assume for now that the robot is walking in a straight line so that we do not need to generate a yaw trajectory plan. Under these assumptions, the dynamics during stance are given by:

$$\ddot{\boldsymbol{p}} = \frac{1}{m}\boldsymbol{f} - \boldsymbol{g} \tag{2}$$

$$\dot{\boldsymbol{\theta}} = \boldsymbol{\omega} \tag{3}$$

$$\dot{\boldsymbol{\omega}} = \hat{\boldsymbol{\mathcal{I}}}^{-1}((\boldsymbol{r} - \boldsymbol{p}) \times \boldsymbol{f} + \boldsymbol{\tau}_{yz}) \tag{4}$$

Where m is the mass of the robot and $\hat{\boldsymbol{\mathcal{I}}}$ is the rotational inertia of the robot expressed in a global reference frame.

2.2. Linearization for Automatic Footstep Placement

In order to make (4) a linear constraint, quadruped controllers usually pre-specify a trajectory for foot position relative to the center of mass, $\boldsymbol{r} - \boldsymbol{p}$. Since foot adjustment is the primary method by which a biped robot can regulate its speed, and we wish to optimize jointly over footstep locations and state trajectories, we instead make (4) affine by linearizing the dynamics. We linearize around a nominal CoM position, $\boldsymbol{p}_\star$ stance foot location $\boldsymbol{r}_\star$, contact force $\boldsymbol{f}_\star$, and contact moment $\boldsymbol{\tau}_\star = 0$.

These affine dynamics during stance mode s are given by

$$\dot{\boldsymbol{x}}_s = \boldsymbol{A}\begin{bmatrix}\boldsymbol{x} \\ \boldsymbol{r}_s\end{bmatrix} + \boldsymbol{B}_s\boldsymbol{u} + \boldsymbol{b}_s \tag{5}$$

Where

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\theta} \\ \dot{\boldsymbol{p}} \\ \boldsymbol{\omega} \end{bmatrix}, \boldsymbol{u} = \begin{bmatrix} \boldsymbol{f} \\ \tau \end{bmatrix}, \boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \hat{\boldsymbol{\mathcal{I}}}^{-1}[\boldsymbol{f}_\star]_\times & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\hat{\boldsymbol{\mathcal{I}}}^{-1}[\boldsymbol{f}_\star]_\times \end{bmatrix}$$

$$\boldsymbol{B}_s = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \frac{1}{m}\boldsymbol{I} & \boldsymbol{0} \\ \hat{\boldsymbol{\mathcal{I}}}^{-1}[\boldsymbol{r}_{\star,s} - \boldsymbol{p}_\star]_\times & \hat{\boldsymbol{\mathcal{I}}}^{-1}\hat{\boldsymbol{e}} \end{bmatrix}, \boldsymbol{b}_s = -\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{g} \\ \hat{\boldsymbol{\mathcal{I}}}^{-1}((\boldsymbol{r} - \boldsymbol{p}) \times \boldsymbol{f} \end{bmatrix}$$

and $\hat{\boldsymbol{e}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^\top$. We choose the linearization point in each stance mode to be the foot and center of mass location when standing with a user specified height and foot spread, and $\boldsymbol{f}_*$ to be a force supporting the robot's weight and passing through the robot's center of mass.

2.3. Learning Time-Varying Residuals for Control

Since $\boldsymbol{A}, \boldsymbol{B}_s$, and $\boldsymbol{b}_s$ obtained through the SRBD do not perfectly match the real dynamics of the robot, we assume that there exist some time-varying residual matrices $\hat{\boldsymbol{A}}_s(t)$, $\hat{\boldsymbol{B}}_s(t)$, $\hat{\boldsymbol{b}}_s(t)$ for which

$$\dot{\boldsymbol{x}}_s = (\boldsymbol{A} + \hat{\boldsymbol{A}}_s)\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{r}_s \end{bmatrix} + (\boldsymbol{B}_s + \hat{\boldsymbol{B}}_s)\boldsymbol{f} + \boldsymbol{b}_s + \hat{\boldsymbol{b}}_s$$

accurately describes the dynamics of the SRBD model embedded on the real robot.

We adapt our approach from Sun et al. (2021) for an online least squares approach to learning these residual matrices from data. Our labels for least squares are the true derivatives minus the nominal derivatives: $\dot{\boldsymbol{x}} - \left(\boldsymbol{A}\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{r}_s \end{bmatrix} + \boldsymbol{B}_s\boldsymbol{u} + \boldsymbol{b}_s\right)$, where $\boldsymbol{A}$, $\boldsymbol{B}_s$, and $\boldsymbol{b}$ are the nominal single rigid body dynamics matrices (5). Our independent variables are vectors of the form $\begin{bmatrix} x_t & r_t & u_t & 1 \end{bmatrix}^\top$, which are the state, stance foot position, and center of mass force and torque respectively. Therefore, we solve the following least squares problem:

$$\min_{\hat{\boldsymbol{A}},\hat{\boldsymbol{B}},\hat{\boldsymbol{b}}} \left\| \dot{\boldsymbol{x}}_t - \left(\boldsymbol{A}\begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{r}_t \end{bmatrix} + \boldsymbol{B}\boldsymbol{u}_t + \boldsymbol{b}\right) - \left(\hat{\boldsymbol{A}}\begin{bmatrix} \boldsymbol{x}_t \\ \boldsymbol{r}_t \end{bmatrix} + \hat{\boldsymbol{B}}\boldsymbol{u}_t + \hat{\boldsymbol{b}}\right) \right\|^2$$

Which is actually several least squares problems stacked into one; each column of $\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{b}}$ is the solution to each problem.

When running the least squares online, we keep a history of N datapoints and labels as Cassie walks. The N datapoints and labels are maintained in a circular-buffer fashion, where at each timestep when a new state is received, the oldest datapoint and label are removed from the buffer, all other datapoints are rolled backward in time by one step, and the new state is added as the last state in the buffer. This allows us to run the least squares online at a high rate, obtaining a residual that is an accurate local approximation for the true dynamics.

In our experiments, we used two versions of this least squares estimator; one in which the entire $\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{b}}$ are free parameters (referred to as the "dense" residual), and one in which we leverage the

block structure of the dynamics to separate the problem into 4 smaller problems, and separately learn subblocks of $\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{b}}$. These subblocks are assembled such that $\hat{\boldsymbol{A}}, \hat{\boldsymbol{B}}, \hat{\boldsymbol{b}}$ have the same sparse block structure as $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{b}$, resulting in many fewer parameters. This version is called the "sparse" residual estimator.

For both the sparse and dense estimators, we add a regularization term to the normal equations, thereby turning our least squares problem into a ridge regression problem.

2.4. Online Single Rigid Body Trajectory Optimization

To find the optimal SRBD state trajectory over a given time horizon, the following trajectory optimization is solved in a receding horizon fashion, where $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x_{des}}$ and $\tilde{\boldsymbol{u}} = \boldsymbol{u} - \boldsymbol{u}_\star$, $\mathcal{F}$ is the friction cone, and $\mathcal{G}$ is the set of points which are on the ground:

$$\boldsymbol{x_{1:T}} = \underset{\boldsymbol{u}_{1\ldots T}, \boldsymbol{x}_{1\ldots T}, \boldsymbol{r}_i}{\arg\min} \quad \sum_{t=1}^{T} (\tilde{\boldsymbol{x}}_t^\top \boldsymbol{Q} \tilde{\boldsymbol{x}}_t + \tilde{\boldsymbol{u}}_t^\top \boldsymbol{R} \tilde{\boldsymbol{u}}_t) + \tilde{\boldsymbol{x}}_T^\top \boldsymbol{Q_f} \tilde{\boldsymbol{x}}_T \tag{6a}$$

$$\text{subject to} \quad \dot{\boldsymbol{x}} = \boldsymbol{A}_i \begin{bmatrix} \boldsymbol{x}^\top & \boldsymbol{r}_i^\top \end{bmatrix}^\top + \boldsymbol{B}_i \boldsymbol{u} + \boldsymbol{b}_i, \tag{6b}$$

$$\boldsymbol{f}_t \in \mathcal{F}, \tag{6c}$$

$$\boldsymbol{r}_i \in \mathcal{G} \tag{6d}$$

This optimal control problem has been implemented as a convex QP using the Drake interface to OSQP in C++ and a single rigid body model of the Cassie biped made by Agility Robotics. The dynamics constraint (6b) is enforced via direct collocation, which can be formualated as a linear equality constraint for affine dynamics.

2.5. Low-level Tracking Controller

We use an operational space (task space) controller formulated similarly to the impact-invariant controller presented by Yang and Posa (Yang and Posa (2021)). Because Cassie has 5 degrees of actuation in single stance, tracking the swing foot position, center of mass position, and pelvis orientation would lead to conflicting control objectives. Therefore we track only the pelvis orientation, center of mass height, and swing foot position.

2.6. Analyzing Perception for Legged Locomotion

The MPC controller assumes that the robot will place its steps according to some fixed gait sequence, which may not be optimal in the case of broken terrain. On legged robots, perception is commonly used to select regions in which it is safe for the robot to step (Kim et al. (2020), Villarreal et al. (2020)). We model our footstep planner off of that of Kim et al. (2020), in which depth images are received from onboard "stereo" cameras (the Drake simulator directly gives us depth images), and terrain regions are classified as "not steppable" if the terrain gradient in that region is too high, or the depth is outside of the kinematic reachability of the robot. If the nominal footstep calculated by the MPC falls into a "not steppable" region, then the footstep planner searches the image for the nearest steppable location to the nominal footstep, and returns the (x, y, z) coordinate. If the nominal footstep is in a "steppable" region already, the footstep planner acts as a query that returns the z height of the queried (x, y) location, since the MPC always assumes flat ground.

3. Theoretical Background

Here we restate some of the major assumptions from Sun et al. (2021). Since the primary result is that if the assumptions hold, the controller using the residual dynamics is stabilizing, we will focus mainly on the assumptions, and empirically validate whether these assumptions are valid in the following section.

Assumption 1 states, in our scenario, that the controller is stable (for some suitable notion of stability) if

$$\left\| \dot{\boldsymbol{x}} - (\boldsymbol{A} + \hat{\boldsymbol{A}}) \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{r} \end{bmatrix} - (\boldsymbol{B} + \hat{\boldsymbol{B}})\boldsymbol{u} - \boldsymbol{b} \right\| < \epsilon.$$

This essentially means that the controller is stable even if the dynamics model with the residual dynamics does not perfectly match the real system. This is a reasonable assumption, though for a complex robot like Cassie and the hierarchical control stack used here, computing a suitable value for ϵ is generally not feasible except perhaps empirically.

Assumption 2 states that the residuals are smooth over time:

$$\left\| \hat{\boldsymbol{B}}_{t+1} - \hat{\boldsymbol{B}}_t \right\| < \delta_B, \quad \left\| \hat{\boldsymbol{A}}_{\boldsymbol{t+1}} \boldsymbol{x}_{t+1} - \hat{\boldsymbol{A}}_{\boldsymbol{t}} \boldsymbol{x}_t + \hat{\boldsymbol{b}}_{t+1} - \hat{\boldsymbol{b}}_t \right\| < \delta_A$$

As we will see in the following section, δ_B and δ_A which satisfy Assumption 2 can be quite large for this controller, even when t and $t+1$ are very close together. Therefore, it may not be possible to find a δ_A, δ_B, and ϵ which simultaneously satisfy the assumptions, and the condition of Theorem 1. This condition requires that $C(\delta_A + \delta_B) < \epsilon$ where C is an expression depending on the conditioning of the OLS estimator used to construct the residual matrices.

4. Experimental Results

4.1. MPC Tracking Performance

Due to model error between the linearized SRBD model and the full robot under closed-loop feedback, the MPC without residuals is unstable after a few steps (Figure 2). This could be expected since control of bipedal robots using the SRBD model is an active research area, especially for robots such as Cassie, where the legs contribute a non-negligible amount to the total inertia of the robot. In order to quantify the error in this model, and whether adding residuals might help, we examine the improvement in dynamics error between the nominal dynamics and residual dynamics in the next section.

4.2. Least Squares Error

While adding residual estimates of the dynamics could improve the accuracy of the dynamics model in mid stride, the susceptibility of these residuals to noisy acceleration and force measurements impact events made using them for control a very brittle strategy (Figure 3), and we were still unable to achieve stable walking over the time-frame of this project. We see similar susceptibility to noise in both the sparse and dense estimations.

We also find that the assumption of smoothness in the dynamics residuals over time is a poor assumption. Figure 4 shows the magnitude of the residual $\hat{\boldsymbol{B}}$ over four consecutive steps. The same plot for $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{b}}$ are given in the Appendix for completeness.

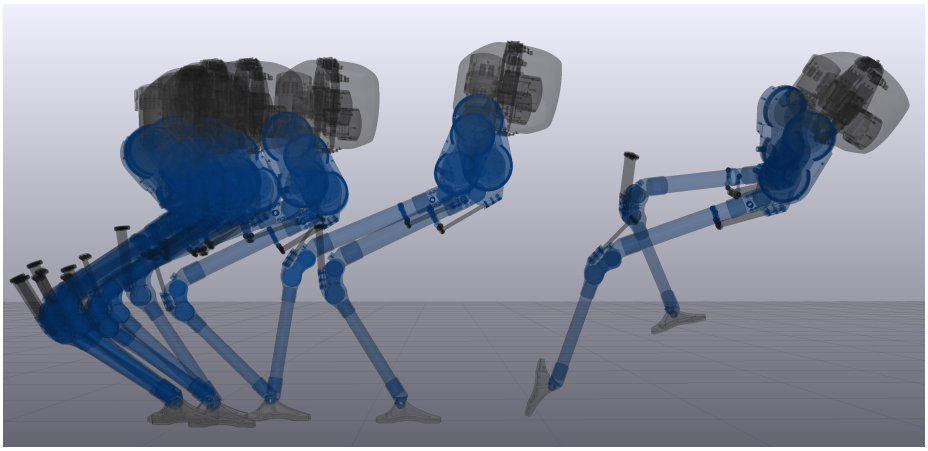

Figure 2: Model error leads the MPC to be unstable under the nominal linear SRBD model

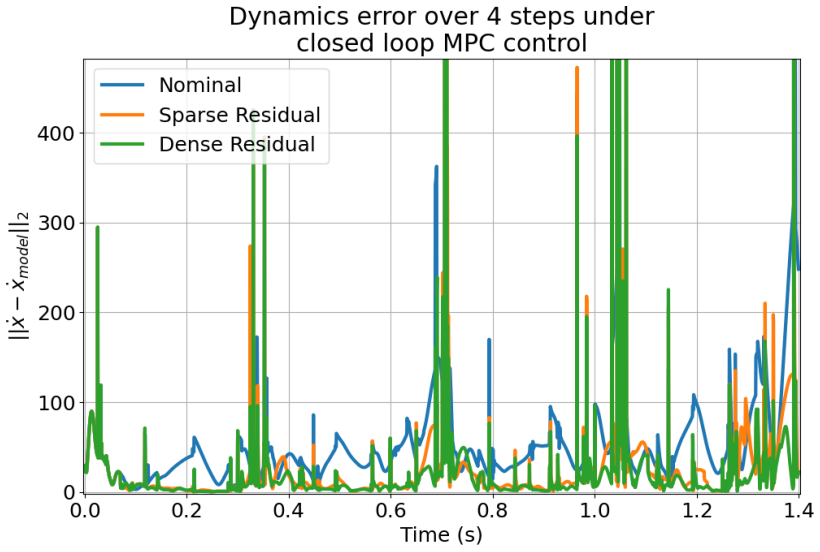

Figure 3: Adding residuals improves the accuracy of the dynamics model, though the residuals are susceptible to noise, especially near impact events. The estimations shown were generated using a window of 1 MPC time step (140 datapoints from the simulation broadcasting at 2000 hz.) and a regularization weight of 0.2.

4.3. Perception Error

There has been some prior work on developing theoretical bounds on perception-based estimators and controllers. Dean et al. (2019) treat perception as a noisy state estimator and develop robust guarantees for a linear controller that uses perceptual inputs. They show that robustness can be guaranteed with certain bounds on the state deviation from training data and the slope of the perception map. Due to the hybrid, nonsmooth, nature of even the SRB dynamics, and the fact that

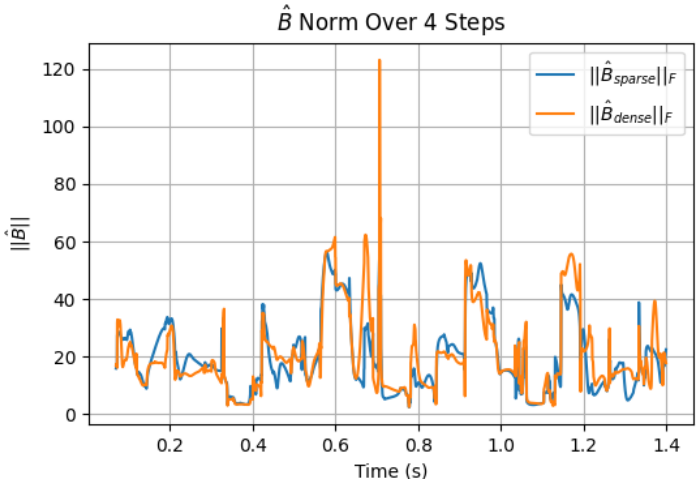

Figure 4: The residual estimates are nonsmooth, especially near impact events. This makes it difficult to use them for control, especially MPC with relatively coarse timesteps.

we use perception for planning rather than for state estimation, we focused most of our effort on experimental evaluations of the effects of perception errors rather than theoretical bounds.

Errors in depth estimation from RGB-D cameras can arise from multiple sources, including: sensor noise, erroneous intrinsic calibration, and erroneous extrinsic calibration.

Extrinsic calibration refers to the transformation matrix between the coordinate system of the camera frame and the body frame. We parameterize this transformation with the extrinsic roll-pitch-yaw euler angles, and the 3D vector describing the translation from the body origin to the camera origin.

In our experiments, we found that slight perturbations to the intrinsic parameters did not greatly affect depth estimation, and shot noise is usually not a limiting factor in depth estimation (Sweeney et al. (2019)). By contrast, a common problem with vision systems, especially on legged robots, is the difficulty of obtaining accurate extrinsic calibration (Reinke et al. (2019)), which can lead to incorrect conversion from a point's location in the camera frame to a point's coordinates in the body-fixed frame, causing inaccurate foot placement. Furthermore, due to the rugged nature of legged locomotion, it is far more likely that the camera gets jostled around, eventually causing deviations from the nominal extrinsic matrix.

Therefore, our first experiment was to characterize how increasing levels of noise in the extrinsic matrix affect the average z error of the points converted from the image to body frame. For this experiment, we separately add i.i.d zero mean Gaussian noise with increasing variance to each of the 3 components of translation and orientation, and measure the error of the transformed points compared to those transformed with the nominal transform. Results are shown in Figure 5. As expected, both the variance and expectation of the error increase linearly with increasing noise variance. While the amount of noise towards the right hand side of the plot are unrealistic, it shows that even with translational/ rotational error of 0.02m/0.02rad, it is altogether possible that the height estimation of the terrain could be off by many centimeters.

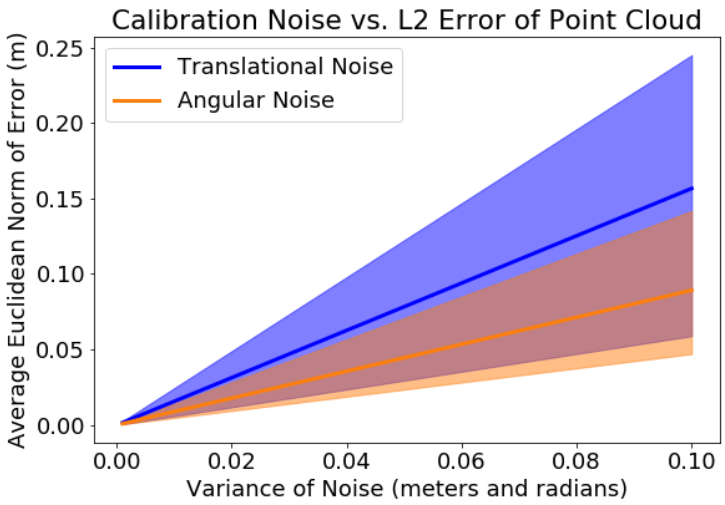

Figure 5: The average error of image points transformed to body frame plotted against calibration noise variance, for noise added to either the camera translation or rotation. The solid lines are the medians, while the shaded regions represent the middle two quartiles of the distribution

4.4. Perception Influence on Residual Estimator

Motivated by the results on how extrinsic calibration error affected perception, we conducted an experiment to measure how these perception errors would affect the residual estimator. We envisioned a scenario where the perception module informs the planner that the ground is actually a few centimeters higher than it actually is. Therefore, the planner would plan for a mode switch from the current stance leg to the next too early, causing the controller to use the wrong set of dynamics for the timesteps between the assumed footfall and the actual footfall. Results for this experiment are shown in Figure 6.

We can see that although the error for both the residual and nominal dynamics spikes at $t = 0.3$, due to using the wrong dynamics, the estimator is actually able to account for this and adapt the residual to match the true dynamics quickly, while the nominal (incorrect) model's error remains high until the real footfall occurs at $t = 0.35$. This matches the results shown in Sun et al. (2021), showing that the least squares estimator is able to adapt quickly to sudden changes in dynamics, adding a degree of robustness to this type of perception error.

5. Discussion and Conclusion

5.1. MPC Performance and Future Improvements

Most of Brian's efforts toward this project have been writing code to perform MPC subject to the nominal linearized SRBD model. The disagreement between Cassie and the SRBD model was challenging to overcome, and we were not able to successfully use residual dynamics to turn an unstable controller into a stable controller. Possible extensions which may help in the future include lineariz-

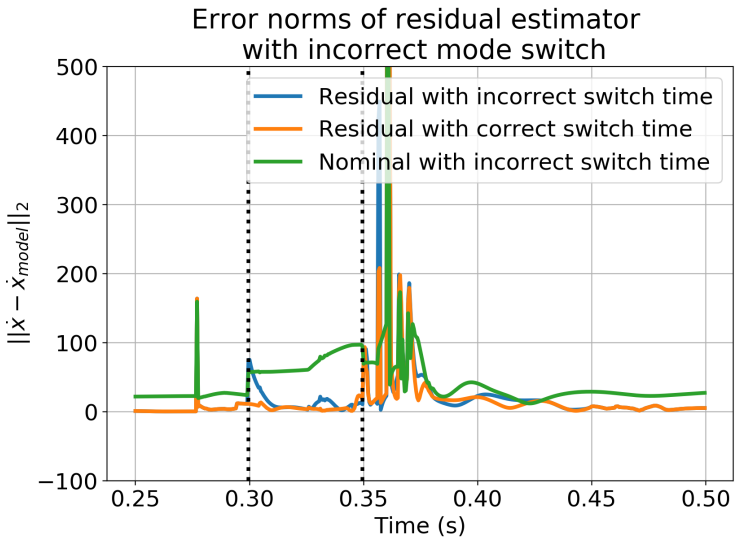

Figure 6: Norm of prediction error of residual (blue) and nominal (green) models given incorrect mode switch at $t = 0.3$s, and residual (orange) given the correct mode switch at $t = 0.35$s. We can see that even when given the incorrect dynamics at $t = 0.3$, the residual model (blue) is able to quickly recover.

ing about a reference gait rather than a static pose when generating the dynamics constraints, and adopting a less structured approach to learning the dynamics which is able to leverage significant offline simulation time, such as model-based RL. Finally, it may be worthwhile to consider formulations with fewer assumptions on the angular velocity of the floating base, since these assumptions limit the robot's motion to be similar to LIP based models.

5.2. Least Squares Residuals and Perception

We showed that using least squares to learn time varying linear residual models could significantly improve the accuracy of our SRBD model, and that with a frequent enough update rate, these residuals could mitigate errors in planning induced by an imperfect perception system. However, we note that the residuals are still extremely noisy during impact events. Future work includes integrating the residual estimator and proposed footstep planner with the SRBD MPC model, investigating how model noise can be mitigated during impact events, and investigating more complex perception systems that involve learning.

References

R. Blickhan. The spring-mass model for running and hopping. *Journal of Biomechanics*, 22(11):1217–1227, January 1989. ISSN 0021-9290. doi: 10.1016/0021-9290(89)90224-8. URL https://www.sciencedirect.com/science/article/pii/0021929089902248.

Yu-Ming Chen and Michael Posa. Optimal reduced-order modeling of bipedal locomotion. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8753–8760, 2020. doi: 10.1109/ICRA40945.2020.9197004.

Sarah Dean, Nikolai Matni, Benjamin Recht, and Vickie Ye. Robust guarantees for perception-based control, 2019.

Jared Di Carlo, Patrick M. Wensing, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9, October 2018. doi: 10.1109/IROS.2018.8594448. ISSN: 2153-0866.

S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis. Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control, 2020.

Grant Gibson, Oluwami Dosunmu-Ogunbi, Yukai Gong, and Jessy Grizzle. Terrain-aware foot placement for bipedal locomotion combining model predictive control, virtual constraints, and the alip, 2021.

Yukai Gong, Ross Hartley, Xingye Da, Ayonga Hereid, Omar Harib, Jiunn-Kai Huang, and Jessy Grizzle. Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway, 2018.

Kevin Green, Ross L. Hatton, and Jonathan Hurst. Planning for the unexpected: Explicitly optimizing motions for ground uncertainty in running. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1445–1451, 2020. doi: 10.1109/ICRA40945.2020.9197049.

S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa. The 3d linear inverted pendulum mode: a simple modeling for a biped walking pattern generation. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, volume 1, pages 239–246 vol.1, 2001. doi: 10.1109/IROS.2001.973365.

D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim. Vision Aided Dynamic Exploration of Unstructured Terrain with a Small-Scale Quadruped Robot. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2464–2470, May 2020. doi: 10.1109/ICRA40945.2020.9196777.

Donghyun Kim, Jared Di Carlo, Benjamin Katz, Gerardo Bledt, and Sangbae Kim. Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control, 2019.

Junheng Li and Quan Nguyen. Force-and-moment-based model predictive control for achieving highly dynamic locomotion on bipedal robots, 2021.

Andrzej Reinke, Marco Camurri, and Claudio Semini. A factor graph approach to multi-camera extrinsic calibration on legged robots. *2019 Third IEEE International Conference on Robotic Computing (IRC)*, Feb 2019. doi: 10.1109/irc.2019.00071. URL http://dx.doi.org/10.1109/IRC.2019.00071.

Hersh Sanghvi and Camillo Jose Taylor. Fast footstep planning on uneven terrain using deep sequential models, 2021.

Jonah Siekmann, Kevin Green, John Warila, Alan Fern, and Jonathan Hurst. Blind bipedal stair traversal via sim-to-real reinforcement learning, 2021.

Yu Sun, Wyatt L. Ubellacker, Wen-Loong Ma, Xiang Zhang, Changhao Wang, Noel V. Csomay-Shanklin, Masayoshi Tomizuka, Koushil Sreenath, and Aaron D. Ames. Online Learning of Unknown Dynamics for Model-Based Controllers in Legged Locomotion. *IEEE Robotics and Automation Letters*, 6(4):8442–8449, October 2021. ISSN 2377-3766. doi: 10.1109/LRA.2021.3108510. Conference Name: IEEE Robotics and Automation Letters.

Chris Sweeney, Greg Izatt, and Russ Tedrake. A supervised approach to predicting noise in depth images. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 796–802, 2019. doi: 10.1109/ICRA.2019.8793820.

O. Villarreal, V. Barasuol, P. M. Wensing, D. G. Caldwell, and C. Semini. MPC-based Controller with Terrain Insight for Dynamic Legged Locomotion. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2436–2442, May 2020. doi: 10.1109/ICRA40945.2020.9197312.

Patrick M. Wensing and David E. Orin. Generation of dynamic humanoid behaviors through task-space control with conic optimization. In *2013 IEEE International Conference on Robotics and Automation*, pages 3103–3109, 2013. doi: 10.1109/ICRA.2013.6631008.

William Yang and Michael Posa. Impact Invariant Control with Applications to Bipedal Locomotion. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

Wenhao Yu, Deepali Jain, Alejandro Escontrela, Atil Iscen, Peng Xu, Erwin Coumans, Sehoon Ha, Jie Tan, and Tingnan Zhang. Visual-locomotion: Learning to walk on complex terrains with vision. In *5th Annual Conference on Robot Learning*, 2021. URL https://openreview.net/forum?id=NDYbXf-DvwZ.

Appendix:

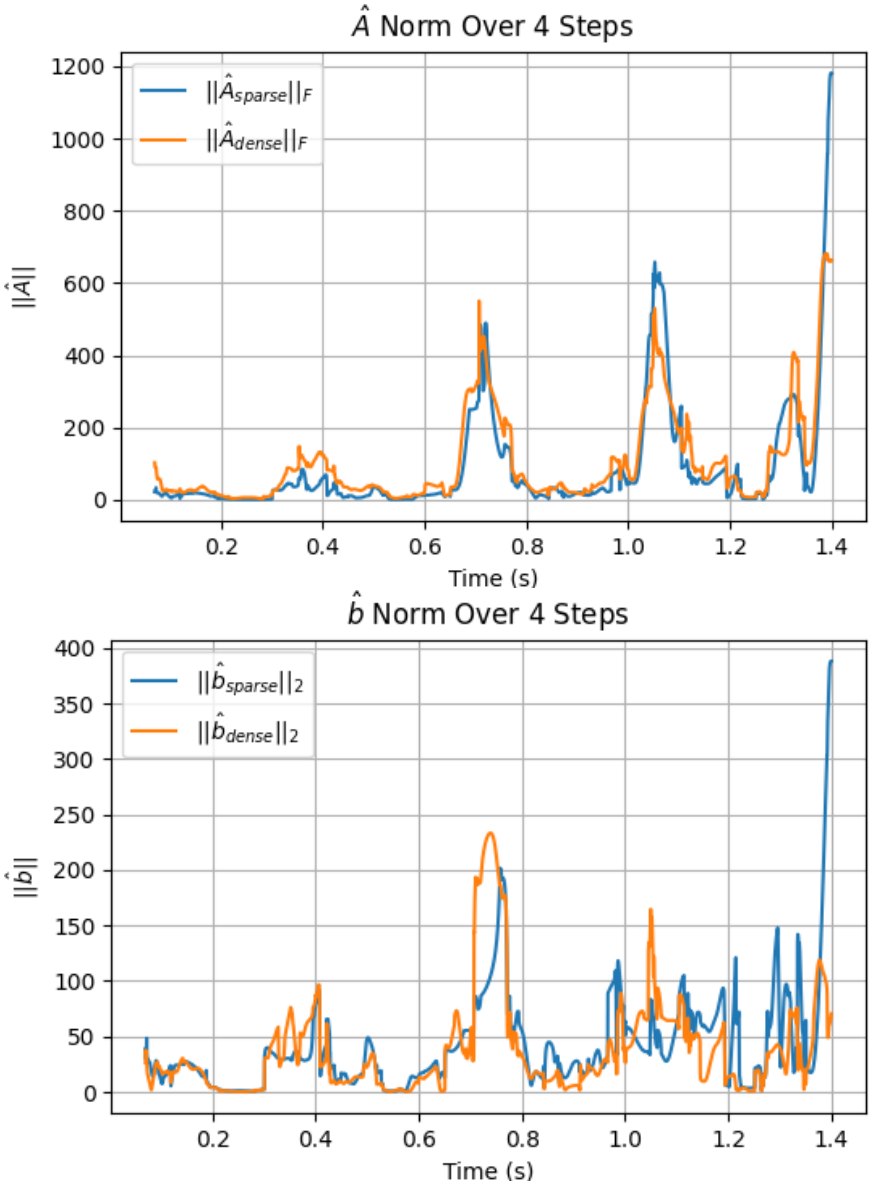

Figure 7: Magnitude of $\hat{\boldsymbol{A}}$ and $\hat{\boldsymbol{b}}$ over the duration of several steps. The $\hat{\boldsymbol{b}}$ estimate is especially susceptible to impacts as Cassie begins to fall over.