Time-Projection control on 3LP, a simple idea to deal with intermittent pushes online

Salman Faraji and Auke Jan Ijspeert
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
salman.faraji@epfl.ch

1 Summary

In this article, we propose a new controller for recovering intermittent pushes during bipedal locomotion. We use 3LP as a template model which can provide closed-form solutions for state evolution. The idea behind our controller is to project the perturbed state of current time-step back to the beginning of the hybrid phase, use the expertise of a discrete controller and then apply the resulting optimal policy to the system at the current time-step. Linear properties of 3LP makes such calculation very fast and effective. By optimizing a certain cost function, we find the most robust and generic projecting configuration which outperforms the discrete controller itself.

2 Introduction

Performing walking for bipedal robots is highly influenced be the characteristics of the hardware. Inspired by passive dynamic walkers (McGeer, 1990), the class of fixed-knee robots use inverted pendulum (IP) models to control and stabilize the motion. With the assumption of point-mass and the fact that swing and stance legs are massless, these models use attack angles and push-off forces to control the robot and compensate the energy loss in the heel strike (Collins et al., 2005). On the real hardware then, besides manufacturing light-weight legs, hip actuators try to track the final desired angle of attack to ensure stability. The underlying assumption in such control paradigm is therefore being able to impose the final attack angle without influencing overall dynamics.

In IP models, at certain discrete events like heel-strike or maximum apex, discrete controllers update the control inputs to deal with disturbances accumulated so far. Due to nonlinearity of the model however, these controllers have to use a linearized map of the system, called Poincaré map (Teschl, 2012) to predict the state and effect of control inputs in the next discrete event (Byl and Tedrake, 2008). Regardless of the control policy, unless using the same discrete event and the associated linearized map, these controller cannot react to intermittent pushes until the next discrete event. More advanced IP-based models that consider masses in the legs, torso or knees also suffer from the same nonlinearity properties.

3LP model however considers swing and torso dynamics and can already produce periodic gaits, without the need to impose footsteps or attack angles. Thanks to linearity, transition

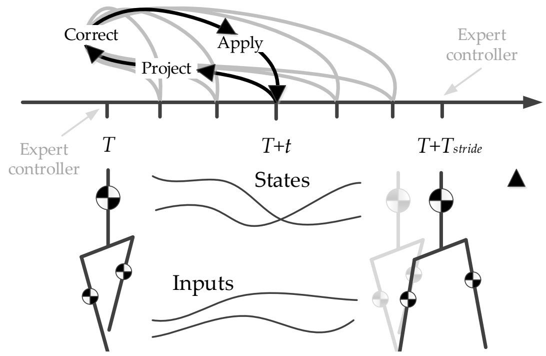

Figure 1: The overall idea of time-projection where the current state is mapped back in time to the beginning of the phase. It is then given to an expert controller whose output is used in the current time-step to stabilize the system online.

matrices can be obtained analytically which are not linearization around a specific gait anymore. In this work, we take advantage of these matrices and design a controller that refines control policies at every time-step, rather than discrete events. Such online controller is however essentially built upon a discrete LQR (DLQR), designed for touch-down events. We use the term touch down rather than heel-strike, because 3LP benefits from smooth weight transition without any impacts. In our approach, although the timing is fixed, the projecting controller modifies hip/ankle torques to change the final footstep location and provides online reaction to perturbations.

In the next section, we provide an overview of the proposed controller and how to use the expertise of a DLQR. Next, we demonstrate the ability of this controller in rejecting intermittent perturbations and finally, we discuss advantages of such computationally simple control policy.

3 Methods

Considering touch down events, we can simply form a discrete error system, describing the evolution of error and the effect of control inputs. This error system is valid in fact for any type of gait found for 3LP, regarding many actuation dimensions provided by the model. Using this error system and a choice of state and input costs, one can simply calculate a DLQR controller K (Ogata, 1995). However, we want to go beyond this controller and find a rule to update control inputs

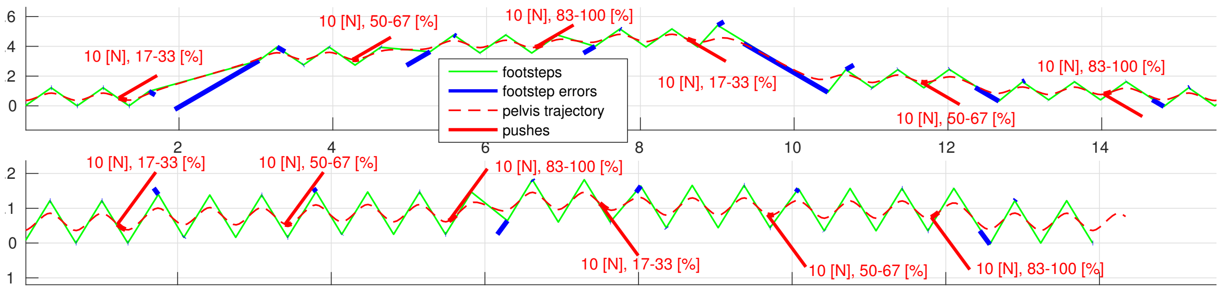

Figure 2: Footstep plan using DLQR (**top**) and projecting (**bottom**) controllers. The DLQR reacts to intermittent pushes only at the end of the phase while the projecting controller updates inputs continuously. Note the percentages that show force application timing.

at any time online. Given the expert DLQR controller and an online disturbance observer, we perform the following steps at any time t during a stride phase: (I) Map or project the currently observed state X_t back in time to the beginning of the phase, (II) Use the expert DLQR controller, (III) Apply the resulting control input at time t. This idea is demonstrated in Figure.1 where unlike the open-loop 3LP, control inputs are not linear functions of time anymore and they depend on disturbances. The important part is indeed time-projection. Although the transition matrices are available already, the control inputs applied from the beginning of the phase until time t are yet unknown.

The expert DLQR produces a control policy U_1 based on a projected state X_1 which are both unknown. Using feedback relation and state evolution matrices, one can simply solve a linear set of equations to find all unknown variables. However, we consider a second alternative system that generates another policy U_2 based on a projected state X_2. This architecture can in fact decouple the system dynamics from disturbance dynamics to provide a more robust time-projection.

The inputs U_1, U_2 and the currently observed disturbance W_t can have influence on X_1, X_2 and X_t. We explore interconnecting configurations with a grid search over all possibilities to find the best performance. The cost function considered for this optimization is similar to LQR, with a horizon of two steps. For each candidate configuration, we take a normal initial state and perturb each dimension separately to obtain a set of perturbed states. A two step simulation is then calculated for each member of this set by applying the candidate controller. We do the same simulation for a normal state also, but perturbed with intermittent pushes of certain magnitude and different timings. Then for each simulation, a separate LQR-like cost is calculated over states and inputs. The final cost for each configurations is the summation of individual costs calculated for adult-size and kid-size models. The resulting optimal combination with minimum cost is the most robust configuration against intermittent disturbances.

4 Results

The optimal configuration exploits both alternative systems and outperforms the DLQR controller alone, shown in Figure.2 over a simulation of multiple intermittent pushes. In other words, the DLQR controller produces large correcting steps in case of moderate but short-lasting pushes. The projecting controller however modulates the hip-torques online to capture the extra energy and therefore, it produces more normal footstep patterns. By further analysis, it also turns out that the optimal projecting architecture has slightly smaller eigenvalues and thus being faster, but with a more limited basin of attraction.

5 Discussion

The resulting projecting configuration is based on an expert DLQR controller, but more robust and responsive to the timing of external pushes. It updates control inputs online and requires only solving a system of $Ax = B$ equations with 16 unknowns. This can be computationally done in micro-seconds, compared to longer optimization times needed for model predictive controllers. However, we have to admit that MPC is more powerful as it can consider inequality constraints on torque limits and footstep lengths as well. The projecting controller is enough in case of slower walking speeds where limitations are rarely important. Thanks to calculation of the basin of attraction however, we can provide an emergency criteria that allows the algorithm to perform more complex calculations (like MPC) if the projecting controller violates limitations.

6 Acknowledgments

This work was funded by the WALK-MAN project (European Community's 7th Framework Programme: FP7-ICT 611832).

References

Katie Byl and Russ Tedrake. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1258–1263. IEEE, 2008.

Steve Collins, Andy Ruina, Russ Tedrake, and Martijn Wisse. *Science*, 307(5712):1082–1085, 2005.

Tad McGeer. *the international journal of robotics research*, 9 (2):62–82, 1990.

Katsuhiko Ogata. Prentice Hall Englewood Cliffs, NJ, 1995.

Gerald Teschl. American Mathematical Soc., 2012.