Neural Network Based Estimator and Controller
Simulation on Torque Actuated SLIP Monopod

Ahmet Safa Öztürk, Hasan Eftun Orhon, Ömer Morgül
Bilkent University, Department of Electrical and Electornics Engineering
sozturk@ee.bilkent.edu.tr

Abstract

We implemented an estimator and a controller for a torque actuated SLIP (spring loaded inverted pendulum) monopod. Our monopod model mathematically represents a running body which is actuated by the torque applied to the ground. In this study data generation and results verification is implemented with the simulation data. Our results shows that data driven type of controller is a good alternative to the conventional controllers and our approach gives promising results for further physical implementations.

Introduction

Spring Loaded Inverted Pendulum (SLIP) model is a common approach for the mathematical representation of legged locomotion. Running body is represented with a point mass actuated with a single spring. A visualization of the torque actuated SLIP model is given in **Figure 1.**[1]

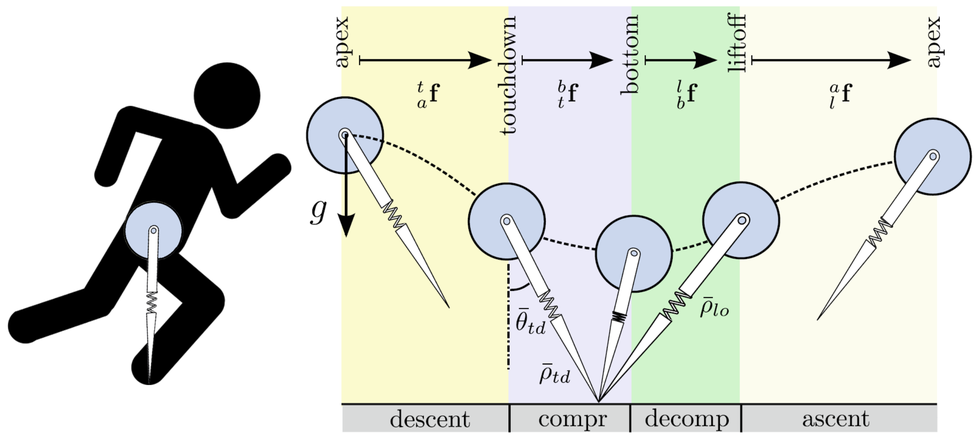

Figure 1: Physical comparison of SLIP model with a running body alongside with the apex return map phases

In this study legged locomotion is analyzed as consecutive single steps. One step is solved between two apex positions. SLIP dynamics are modelled with the following differential equations and the solutions for these equations require some mathematical approximations.[1]

$$Flight\,(descent\,\&\,ascend): \begin{bmatrix}\dot{y}\\ \dot{z}\end{bmatrix} = \begin{bmatrix}0\\ -1\end{bmatrix} \tag{1}$$

$$Stance\,(compr\,\&\,decompr): \begin{bmatrix}\dot{\rho}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\rho\dot{\theta}^2 - \kappa(\rho-1) - c\dot{\rho} - \cos\theta\\ (-2\dot{\rho}\dot{\theta} + \sin\theta)/\rho\end{bmatrix} \tag{2}$$

$$Apex\ State: X_a := [y_a, z_a, \dot{y}_a]^T \tag{3}$$

$$Control\ Inputs: u := [\tau, \theta]^T z_{a+1}, \dot{y}_{a+1} \tag{4}$$

In previous works control of the SLIP monopod is achieved by the controller which resulted from Equations 1 and 2. This controllers also used in the physical setup of the SLIP monopod. An important disadvantage of using approximate analytic controllers is that for complex tasks as trajectory planning or motion planning the equations become over complicated even some cases inconclusive. To introduce an alternative solution to these problems, we proposed a new type of controller which is based on a neural network. Our objective for this study to test the idea of data driven solution using data generated by our simulation which is previously developed by various contributors including the cited and acknowledged authors.

Since we took the running motion as single steps, we focused solely on the apex states. As a beginning point we took the apex state as $X_a := [z_a, \dot{y}_a]^T$ (Note that the y_a (horizontal position) is ignored since it is not important for our planned running trajectories.) We built a system to estimate the next apex state, $X_{a+1} := [z_{a+1}, \dot{y}_{a+1}]^T$ with given control inputs, $u := [\tau, \theta]^T$. Basically our estimator replaces the apex return map of forward kinematics. For the controller part we used the current and a desired apex state, X_a and X_{a+1}, to find the proper control inputs, u. So the controller replaces the apex return map of inverse kinematics. One can say that with the simulation all possible inputs and outputs can be generated so a new estimator and controller is unnecessary but as stated the future aim of this study to use the generated model on the physical setup.

Model

We used a simple Artificial Neural Network architecture (Multi-Layer Perception) for both estimator and controller models. Our inputs for the estimator is the current apex state and control input while the inputs for the controller is the current and next apex states. We used hyperbolic tangent function as the activations of the hidden and the output layers. The reason for the activation function selection is that we need to limit our outputs to comply with the properties of the physical setup. ANN architecture for estimator and the controller visualized in **Figure 2**.

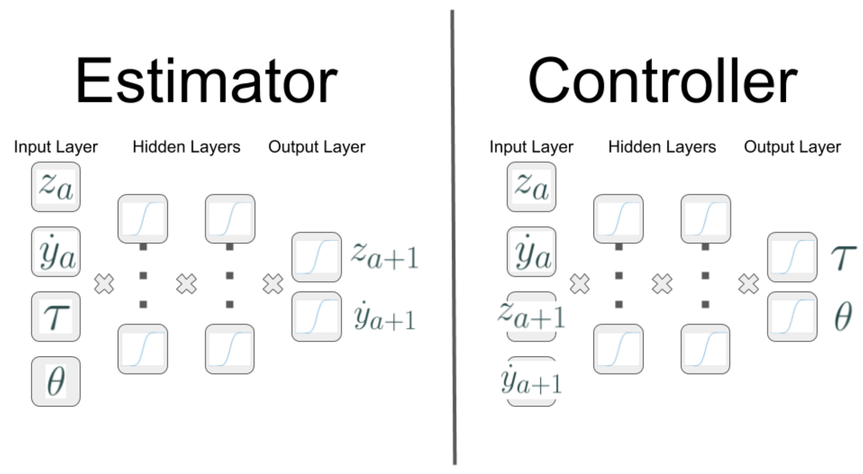

Figure 2: ANN Architectures

Results

Data Generation

Parameter	Range	Unit
Vertical Position (z)	1.25-2	Leg Rest Length(ρ)
Horizontal Velocity ($\dot{y}$)	0.5-4	m/s
Torque (τ)	0-10	nM
Touchdown Angle(θ)	0.175-0.7	Radians

Table 1: Parameter Range for Data Generation

Data acquisition done with our simulation and the parameter range for the generated data is shown in **Table 2**. In this ranges 10 points are selected linearly and total $\sim$10000 apex positions are generated. Also the cases where the monopod goes some impossible positions are avoided. Run-time of the simulation is $\sim$1 second for each apex generation.

Training

For training model complexity is kept low with just 2 hidden layers with 8 hidden units. Maximum number of epochs are set to 5000, the learning rate, (η) is 0.001 and the mini-batch size is 200. Stochastic mini-batch approach is used for training. Validation loss values for training is shown in **Figure 3** and the loss values for test set is shown in **Table 3**.

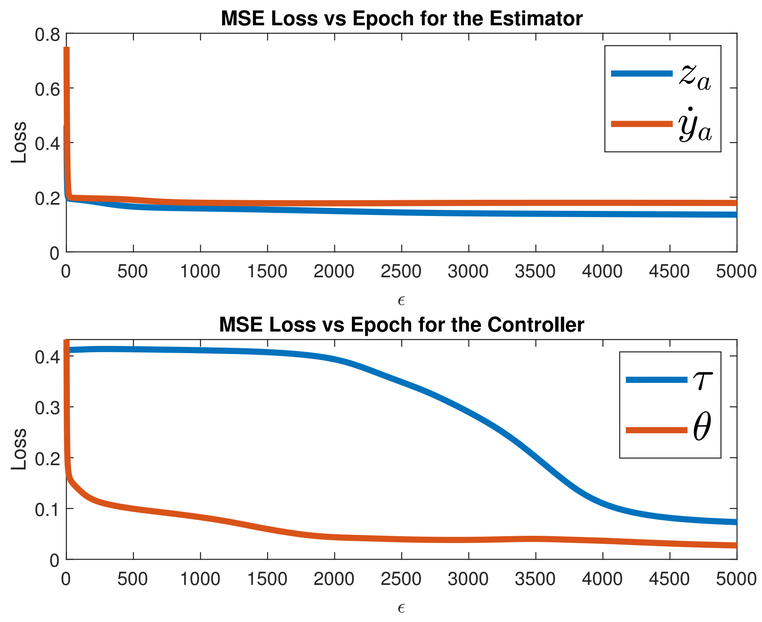

Figure 3: MSE Loss vs Epoch Number for Validation Set

Type	Loss for Output 1	Loss for Output 2
Estimator	0.1291	0.1923
Controller	0.0991	0.0351

Table 2: MSE Loss for Test Set

In **Figure 4** a sample trajectory is given to the trained controller and the resulted control inputs used back in the trained estimator to show the compliance of the estimator with the controller. Basically the red line is the input for the controller and the blue line is the output from the estimator.

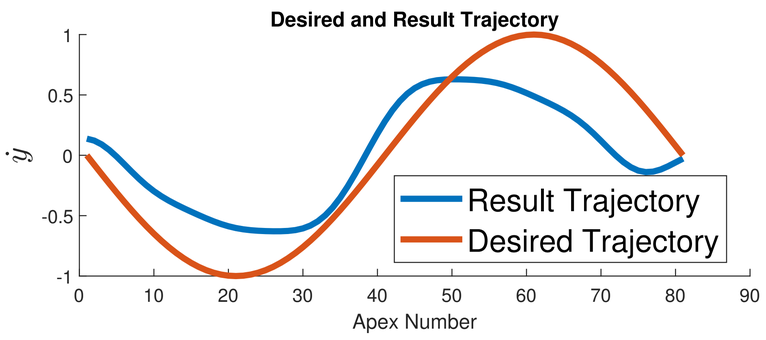

Figure 4: Sample Desired and Result Trajectory Comparison

Conclusions & Future Work

In this study we showed that the neural network based controllers can be a promising alternative to the current methods. As a future research point the same approach will be tested on our physical setup and the performance of the methods will be measured again in the physical world.

References

[1] Uluc Saranlı, Ömür Arslan, M. Mert Ankaralı, and Ömer Morgül. Approximate analytic solutions to non-symmetric stance trajectories of the passive spring-loaded inverted pendulum with damping. *Nonlinear Dynamics*, 62(4):729–742, Dec 2010.

Acknowledgements

Authors appreciate the fruitful discussions of İsmail Uyanık and Hasan Hamzaębi.