Two-Legged Robot System Identification With Artificial Neural Networks

Burak Çatalbaş, Bahadır Çatalbaş and Ömer Morgül

Department of Electrical and Electronics Engineering, Bilkent University

catalbas@ee.bilkent.edu.tr, cbahadir@ee.bilkent.edu.tr, morgul@ee.bilkent.edu.tr

IEEE Bilkent Graduate Research Conference 2020

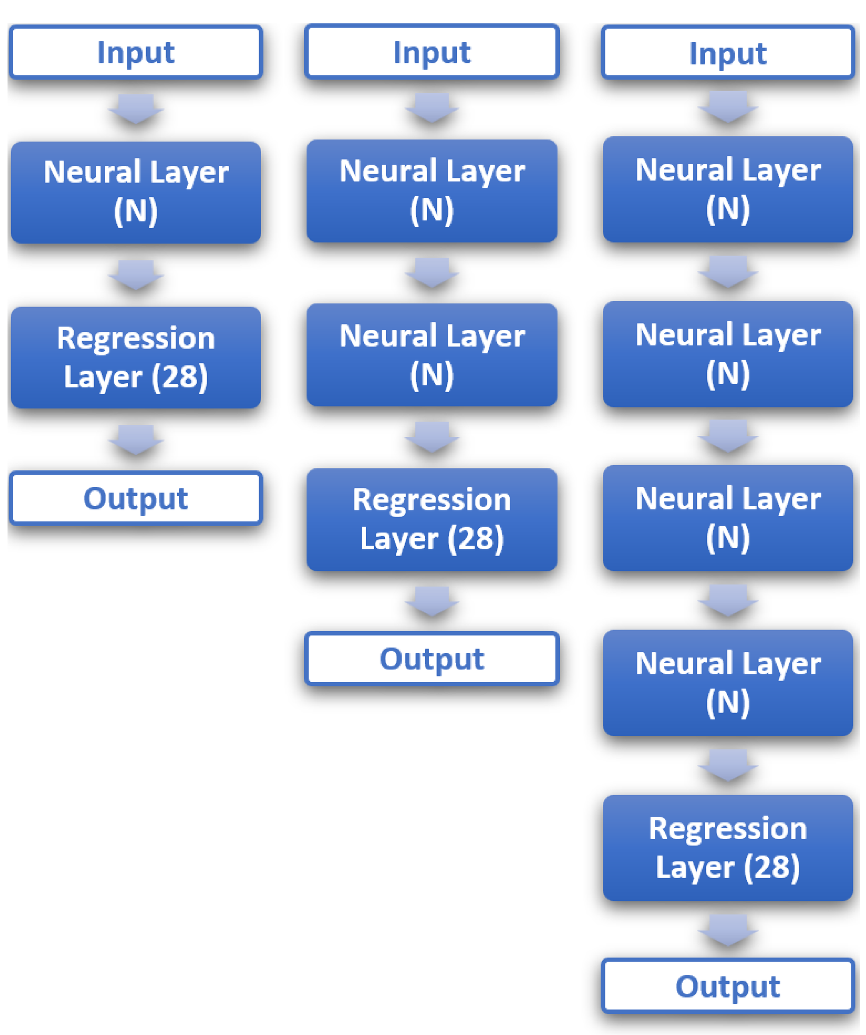
INTRODUCTION

- Two-Legged Robot System Identification is an important topic of research.
- The robot platforms have a dynamic structure with flight & contact phases.
- This hybrid dynamic structure and non-linear dynamics make it harder to calculate the system Jacobian needed for the adaptive control methods.
- Different neural networks are used in this study to make system identification.
- Supervised training is done with central pattern generator creating stable walking datasets, which are used by feedforward and recurrent networks.

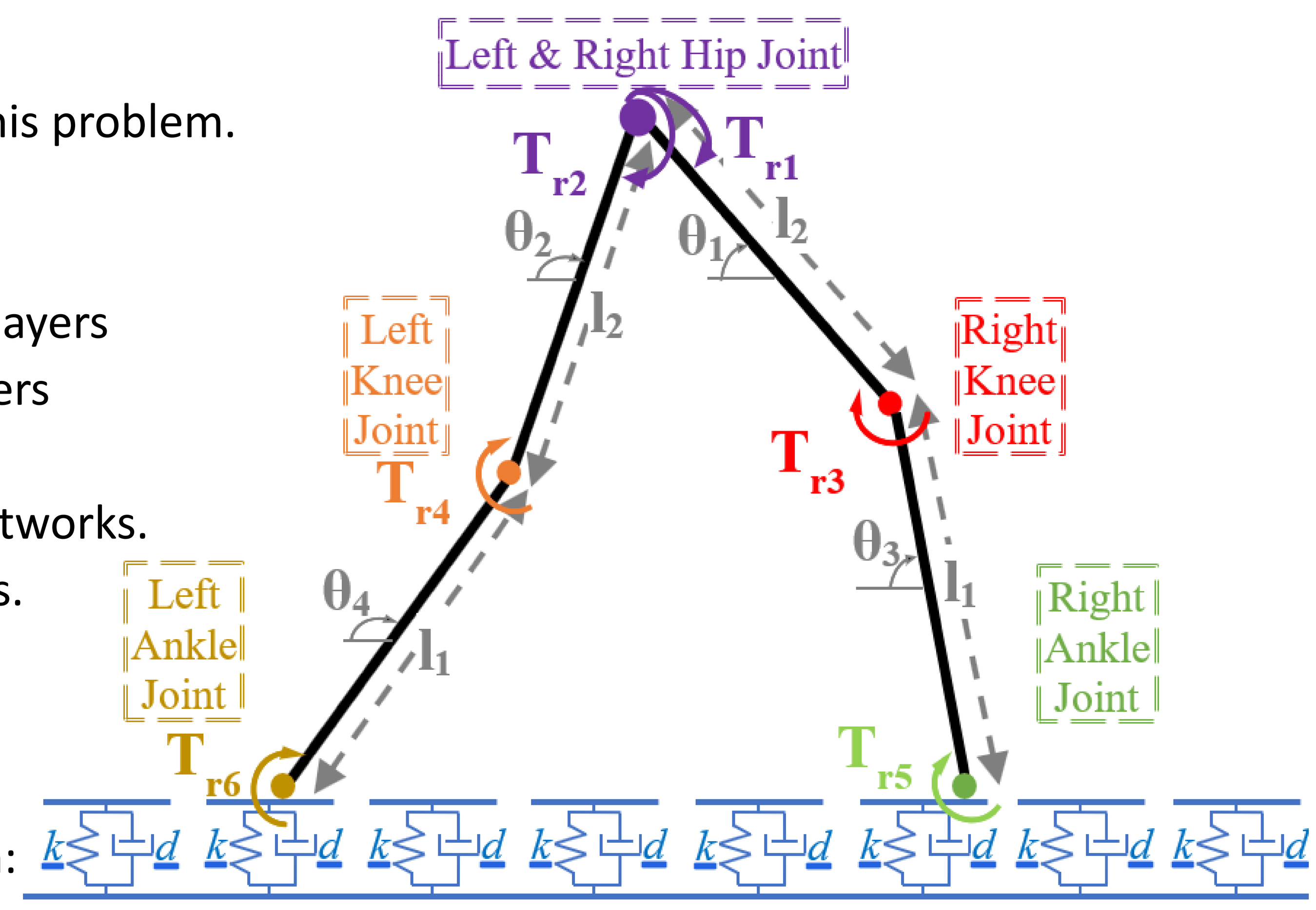
DATASET

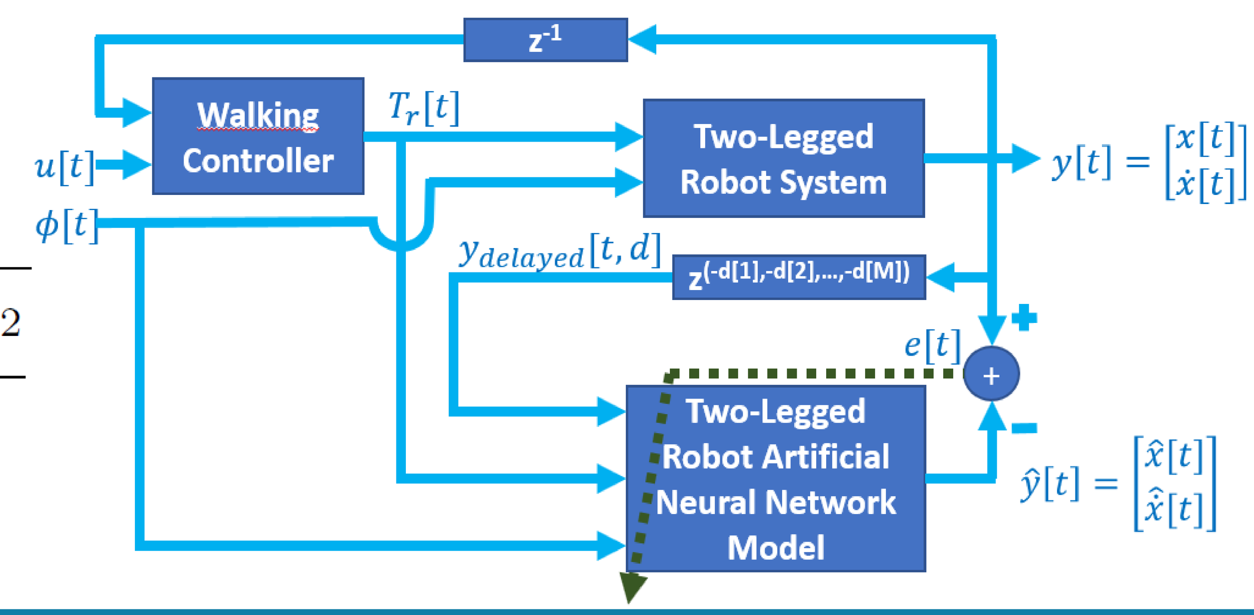
- Data is generated from a 2-legged robot system model with 4 parts & 6 joints.
- System models non-linear dynamics too, such as ground response and friction.
- Hybrid dynamic robot platform is combined with Central Pattern Generator:
  - → It consists of 6 Matsuoko oscillators with coupling.
  - → The samples are taken with 100 Hz frequency for a duration of 10 seconds.
  - → In total, 380 training, 189 verification and 141 test patterns are generated.

SYSTEM ARCHITECTURE



- Artificial Neural Networks (ANN) are proposed for solving this problem.
- Different types of neural networks are employed this work:
  - → Feedforward Neural Networks with different number of layers
  - → Recurrent Neural Networks with different number of layers
- 2, 3 and 5 layers-network formations are used for neural networks.
  - → All networks used linear & hyperbolic tangent activations.
- The networks are used to make the system identification.
- The serial-parallel architecture for the identification is given:





$$E = \sqrt{\sum_{t=1}^{N} \frac{(y[t] - \hat{y}[t])^2}{N}} = \sqrt{\sum_{t=1}^{N} \frac{(e[t])^2}{N}}$$

$$y_{delayed}[t, d] = \begin{bmatrix} y[t-d[1]] \\ y[t-d[2]] \\ \dots \\ y[t-d[M]] \end{bmatrix}, \quad d = [1, 2, \dots, M]$$

RESULTS

- Multiple tables are given as results for 2, 3 and 5-layer networks.
- Both recurrent and feedforward neural networks are compared:
  - → Recurrent networks are used with different neuron amounts.
  - → Feedforward networks are applied in this manner as well:
  - → In addition, different input delays are also applied to them.
- The best result is obtained by recurrent (LSTM) networks:
  - → The largest neuron amount and least number of layers is used.
  - → Reasons: Increased capability of network and less complexity.
  - → 1.77% test error is the best one, also for the training phases.

TABLO II: Single Hidden Layer Neural Networks

Number of Weights	HL Neuron Type	HL Neuron Number	Input Delay	System Characterization Error (RMSE) Training	Verification	Test
18628	LSTM	N=50	d=[1]	$1,61 \cdot 10^{-2}$	$1,99 \cdot 10^{-2}$	$2,38 \cdot 10^{-2}$
	FF (tanh)	N=291	d=[1]	$3,1 \cdot 10^{-2}$	$3,28 \cdot 10^{-2}$	$3,43 \cdot 10^{-2}$
	FF (tanh)	N=203	d=[1,2]	$2,88 \cdot 10^{-2}$	$3,07 \cdot 10^{-2}$	$3,23 \cdot 10^{-2}$
	FF (tanh)	N=155	d=[1,2,3]	$2,84 \cdot 10^{-2}$	$3,05 \cdot 10^{-2}$	$3,26 \cdot 10^{-2}$
	FF (tanh)	N=126	d=[1,2,3,4]	$2,83 \cdot 10^{-2}$	$3,02 \cdot 10^{-2}$	$3,17 \cdot 10^{-2}$
57228	LSTM	N=100	d=[1]	$1,17 \cdot 10^{-2}$	$1,62 \cdot 10^{-2}$	$1,97 \cdot 10^{-2}$
	FF (tanh)	N=894	d=[1]	$3,3 \cdot 10^{-2}$	$3,44 \cdot 10^{-2}$	$3,57 \cdot 10^{-2}$
	FF (tanh)	N=622	d=[1,2]	$2,89 \cdot 10^{-2}$	$3,09 \cdot 10^{-2}$	$3,22 \cdot 10^{-2}$
	FF (tanh)	N=477	d=[1,2,3]	$2,84 \cdot 10^{-2}$	$3 \cdot 10^{-2}$	$3,2 \cdot 10^{-2}$
	FF (tanh)	N=387	d=[1,2,3,4]	$2,78 \cdot 10^{-2}$	$2,98 \cdot 10^{-2}$	$3,16 \cdot 10^{-2}$
194428	LSTM	N=200	d=[1]	$\mathbf{0,81 \cdot 10^{-2}}$	$\mathbf{1,3 \cdot 10^{-2}}$	$\mathbf{1,77 \cdot 10^{-2}}$
	FF (tanh)	N=3038	d=[1]	$3,37 \cdot 10^{-2}$	$3,55 \cdot 10^{-2}$	$3,64 \cdot 10^{-2}$
	FF (tanh)	N=2114	d=[1,2]	$3,06 \cdot 10^{-2}$	$3,24 \cdot 10^{-2}$	$3,36 \cdot 10^{-2}$
	FF (tanh)	N=1620	d=[1,2,3]	$3,02 \cdot 10^{-2}$	$3,14 \cdot 10^{-2}$	$3,34 \cdot 10^{-2}$
	FF (tanh)	N=1314	d=[1,2,3,4]	$2,85 \cdot 10^{-2}$	$3,06 \cdot 10^{-2}$	$3,16 \cdot 10^{-2}$

TABLO III: Double Hidden Layer Neural Networks

Number of Weights	HL Neuron Type	HL Neuron Number	Input Delay	System Characterization Error (RMSE) Training	Verification	Test
38828	LSTM	N=50	d=[1]	$1,29 \cdot 10^{-2}$	$1,78 \cdot 10^{-2}$	$2,16 \cdot 10^{-2}$
	FF (tanh)	N=168	d=[1]	$2,28 \cdot 10^{-2}$	$2,57 \cdot 10^{-2}$	$2,8 \cdot 10^{-2}$
	FF (tanh)	N=156	d=[1,2]	$2,12 \cdot 10^{-2}$	$2,4 \cdot 10^{-2}$	$2,59 \cdot 10^{-2}$
	FF (tanh)	N=146	d=[1,2,3]	$2,08 \cdot 10^{-2}$	$2,35 \cdot 10^{-2}$	$2,55 \cdot 10^{-2}$
	FF (tanh)	N=137	d=[1,2,3,4]	$2,07 \cdot 10^{-2}$	$2,31 \cdot 10^{-2}$	$2,49 \cdot 10^{-2}$
137628	LSTM	N=100	d=[1]	$\mathbf{0,89 \cdot 10^{-2}}$	$\mathbf{1,44 \cdot 10^{-2}}$	$\mathbf{2,03 \cdot 10^{-2}}$
	FF (tanh)	N=340	d=[1]	$2,14 \cdot 10^{-2}$	$2,57 \cdot 10^{-2}$	$2,68 \cdot 10^{-2}$
	FF (tanh)	N=328	d=[1,2]	$1,98 \cdot 10^{-2}$	$2,2 \cdot 10^{-2}$	$2,47 \cdot 10^{-2}$
	FF (tanh)	N=316	d=[1,2,3]	$2,01 \cdot 10^{-2}$	$2,38 \cdot 10^{-2}$	$2,52 \cdot 10^{-2}$
	FF (tanh)	N=304	d=[1,2,3,4]	$1,96 \cdot 10^{-2}$	$2,34 \cdot 10^{-2}$	$2,42 \cdot 10^{-2}$

TABLO IV: Four Hidden Layer Neural Networks

Number of Weights	HL Neuron Type	HL Neuron Number	Input Delay	System Characterization Error (RMSE) Training	Verification	Test
79228	LSTM	N=50	d=[1]	$\mathbf{1,16 \cdot 10^{-2}}$	$\mathbf{1,72 \cdot 10^{-2}}$	$\mathbf{2,33 \cdot 10^{-2}}$
	FF (tanh)	N=152	d=[1]	$1,95 \cdot 10^{-2}$	$2,36 \cdot 10^{-2}$	$2,92 \cdot 10^{-2}$
	FF (tanh)	N=148	d=[1,2]	$1,78 \cdot 10^{-2}$	$2,28 \cdot 10^{-2}$	$2,42 \cdot 10^{-2}$
	FF (tanh)	N=144	d=[1,2,3]	$1,82 \cdot 10^{-2}$	$2,3 \cdot 10^{-2}$	$2,44 \cdot 10^{-2}$
	FF (tanh)	N=140	d=[1,2,3,4]	$1,86 \cdot 10^{-2}$	$2,3 \cdot 10^{-2}$	$2,54 \cdot 10^{-2}$

CONCLUSION & FUTURE WORK

- In this work, system identification is done by different neural networks using a synthetic dataset generated with CPG.
- In comparison, recurrent neural network is the best choice, while delayed inputs can improve feedforward ones better.
- High number of neurons with less number of layers seems as the best strategy to follow in this type of problems.
  - → As a future work, using other initialization, optimizer and regularization techniques can improve the performance.

ACKNOWLEDGEMENTS

Bahadır Çatalbaş was, and Burak Çatalbaş is supported by National Scientific and Technological Council of Turkey, TUBITAK.

IMPORTANT REFERENCES

[1] B. Çatalbaş, "Recurrent neural network learning with an application to the control of legged locomotion," Master's thesis, Bilkent University, 2015.

[2] S. Hochreiter, and J. Schmidhuber, "Long short-term memory.", *Neural computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[3] A. V. D. Schaft and H. Schumaher, *An introduction to hybrid dynamical systems.* Springer London, 2000, vol. 251.

[4] W. T. Miller, P. J. Werbos, and R. S. Sutton, *Neural networks for control*. MIT press, 1995.

[5] V. Kasparian and C. Batur, "Model reference based neural network adaptive controller," *ISA transactions*, vol. 37, no. 1, pp. 21-39, 1998.