Kinematic Modeling of a RHex-type Robot Using a Neural Network

Mario Harper[a], James Pace[b], Nikhil Gupta[b], Camilo Ordonez[b], and Emmanuel G. Collins, Jr.[b]

[a]Department of Scientific Computing, Florida State University, Tallahassee, FL, USA
[b]Department of Mechanical Engineering, Florida A&M - Florida State University, College of Engineering, Tallahassee, FL, USA

ABSTRACT

Motion planning for legged machines such as RHex-type robots is far less developed than motion planning for wheeled vehicles. One of the main reasons for this is the lack of kinematic and dynamic models for such platforms. Physics based models are difficult to develop for legged robots due to the difficulty of modeling the robot-terrain interaction and their overall complexity. This paper presents a data driven approach in developing a kinematic model for the X-RHex Lite (XRL) platform. The methodology utilizes a feed-forward neural network to relate gait parameters to vehicle velocities.

Keywords: Legged Robots, Kinematic Modeling, Neural Network

1. INTRODUCTION

The motion of a legged vehicle is governed by the gait it uses to move. Stable gaits can provide significantly different speeds and types of motion. The goal of this paper is to use a neural network to relate the parameters that define the gait an X-RHex Lite (XRL) robot uses to move to angular, forward and lateral velocities. This is a critical step in developing a motion planner for a legged robot.

The approach taken in this paper is very similar to approaches taken to learn forward predictive models for skid-steered robots. For example, this approach was used to learn a forward predictive model for the Crusher UGV (Unmanned Ground Vehicle).[1] RHex-type robot may be viewed as a special case of skid-steered vehicle as their rotating C-legs are always pointed in the same direction. The robot turns by the differential movements of the C-legs on opposing sides of the robot. It is hence not surprising that the dynamics of this type of robot bear resemblance to those observed for wheeled and tracked robots.[2]

For wheeled skid-steered robots analytical dynamic modeling has been empirically demonstrated.[3] However, analytical kinematic or dynamic modeling of RHex-type robots is far more difficult due in part to the leg compliance and the difficulty of modeling the robot-terrain interaction. Hence, in prior research, kinematic and dynamic models are obtained through curve fitting experimental data.[2] This approach was based on viewing the experimental data and choosing the functional forms of the curves representing the kinematics and dynamics of the platform. However, a more elegant approach that does not depend on recognizing these functional forms is to use neural network modeling as discussed in this paper. The approach is demonstrated using an XRL robot.

The remainder of this paper is organized as follows. Section 2 describes the XRL robotic platform used in the research, including the gait under investigation. Section 3 describes the experimental method and neural network procedure used in this research. Section 4 describes the resulting accuracy in motion prediction from the neural network. Finally, Section 5 summarizes the paper and proposes future work.

Further author information:
E. Collins: E-mail: ecollins@eng.fsu.edu, Telephone: 1 850 410 6373

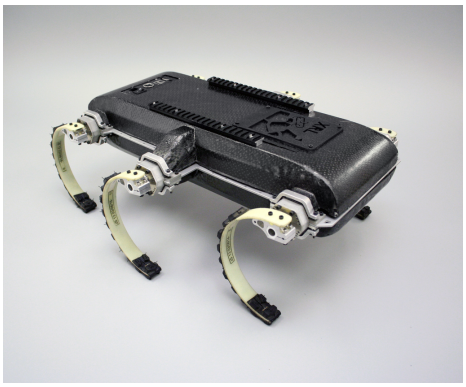

Figure 1. X-RHEX Lite (XRL) hexapedal robot used in this study.

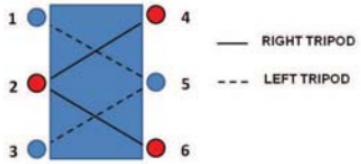

Figure 2. Leg scheme corresponding to each tripod and the leg numbering used throughout this paper for the XRL.

2. ROBOTIC PLATFORM

The XRL (X-Rhex Light)[4] robotic platform, shown in Figure 1, is a hexapedal robot that can use a number of diverse gaits for locomotion. For the research discussed in this paper, the robot uses an alternating tripod gait, with legs 1, 5, and 3 forming one tripod and legs 2, 4, and 6 forming the alternate tripod, as shown in Figure 2. The motion for each leg follows a trajectory generated by a Buehler Clock,[5] which is characterized by two phases, a stance phase and a flight phase 3. The legs in each tripod stay in the same phase throughout the trajectory.

The Buehler clock is defined by four parameters, frequency f, nominal duty factor $\overline{d_f}$, nominal leg offset $\overline{\phi_0}$, and the nominal leg sweep angle $\overline{\phi_s}$. Figure 3 shows a leg cycle for one of the six legs. Each leg in a tripod (see Figure 2) undergoes an identical cycle and the legs in different tripods undergo cycles that only differ in phase. A cycle begins with the stance phase, where the leg contacts the ground. The leg then swings through the flight phase and touches back down into the stance phase completing the cycle. The time spent in terms of this leg cycle is $t_c = \frac{1}{f}$ and the time spent in the stance phase is $t_s = \overline{d_f} t_c$. It is important to note that each leg of the XRL completes a cycle every t_c seconds irrespective of turn gain t_g. During a commanded turn, although t_c is constant for each leg, based on the value of t_g, each leg will alter the time spent in flight and stance phase allowing more time to push against the ground and causing the robot to turn.

To turn while in motion, the gait parameters for the inner (1,2,3) and outer (4,5,6) legs are perturbed. The perturbations are a function of the turn gain (t_g) and three constants (α, β, γ). The gait parameters for the inner legs become:

$$\phi_0^i = \overline{\phi_0} - t_g \alpha, \tag{1}$$

$$\phi_s^i = \overline{\phi_s} - t_g \beta, \tag{2}$$

$$d_f^i = \overline{d_f} - t_g \gamma. \tag{3}$$

The equations for the outer legs are similar, but with the sign for the perturbation flipped. For this research, the values were kept at $\alpha = 0.05$, $\beta = 0.7$, and $\gamma = 0.0$. The nominal duty factor, nominal leg offset, and nominal leg sweep angle are maintained at $d_f = 0.42$, $\overline{\phi_0}$ = -0.20, and $\overline{\phi_s} = 0.85$, respectively. The frequency and turn gain values are the values whose relationship to the forward and angular velocity are being learned by the neural

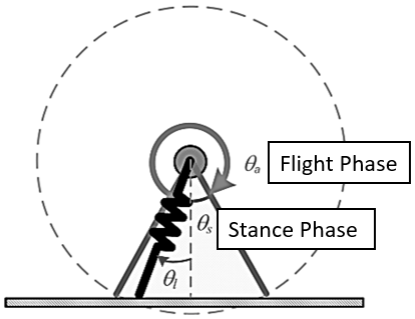

Figure 3. Diagram of a stride cycle, the leg begins in stance phase.

network. In this research the values for the frequency were in the range $[2.2, 2.6]\,Hz$ and the values for the turn gain were in the range $[-1.0, 1.0]$.

Prior work in relating the gait parameters to the vehicle motion have been done with a walking gait where the frequency varied between $0.5Hz$ and $1.1Hz$,[2] compared to the much faster gait under investigation here where the frequency varies between $2.2Hz$ and $2.6Hz$. In the previous work, the authors also found that the vehicle's forward velocity V was linearly related to the gait frequency f and that the turn radius R was related directly to the turn gain t_g of the gait. The authors were able to find functional relationships between frequency f and velocity V and between the turn gain t_g and radius R. However, it is in general difficult to find these functional relationships and hence neural network based modeling is pursued here. Also, it has been seen that at the higher frequency gaits under investigation here, it was found that the functional relationships described in the prior work did not accurately define the motion of the robot. Particularly, predictions of the angular velocity of the robot were poor. It should be noted that the neural network approach developed in this research is portable to other legged robots, for example, those with articulated legs.

3. METHOD

Data collection was necessary to obtain a sufficient breadth of data for neural network training. Field experiments were designed and executed to generate sufficient training data. A cycle filter was applied, the velocity data was averaged over a window of width t_c for each leg. This significantly reduced the influence of noise, while increasing network performance. The sections below describe in detail how the experiments were conducted and the process of selection and design of the neural networks.

3.1 Experimental Design

Data collection was conducted by commanding the XRL to follow twenty trajectories (differentiated based on start/goal location, number of turns, forward velocity of the vehicle, run-time) designed to utilize the full range of turn-gains and frequencies under consideration. The histograms of Figures 4 and 5 illustrate the variation in f and t_g for one of these trajectories, Trajectory 4, which was designed to have a relatively fast linear velocity with wide turns.

The data collection experiments were conducted on asphalt in an area of approximately 15 square meters. VICON motion capture cameras recorded the motion of the vehicle at 100Hz. Twenty independent runs were conducted, each lasting between 18-25 seconds. The total collected raw data points numbered over 30,000 and was reduced to just under 1000 points using the cycle filter described above.

The experiments were conducted as follows:

1. A pre-programmed trajectory is chosen by the operator
2. The trajectory is sent to the robot as a series of frequencies and turn-gains

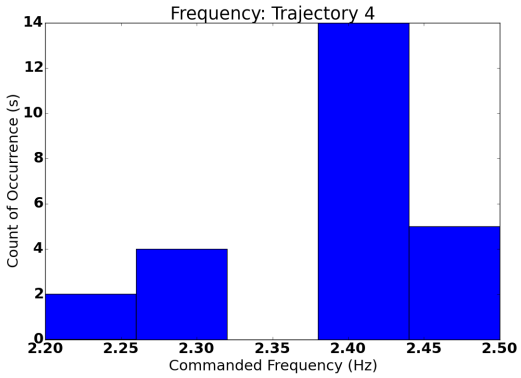

Figure 4. Histogram of the occurrence (in seconds) of frequencies commanded during the course of executing Trajectory 4.

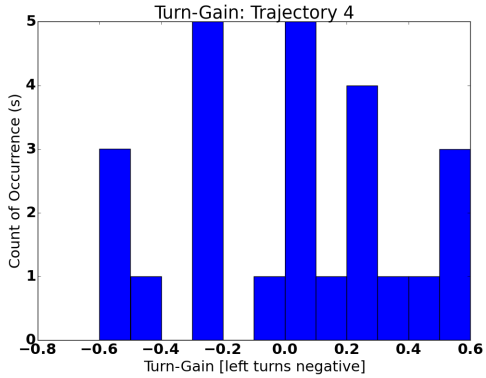

Figure 5. Histogram of the occurrence (in seconds) of commended turn-gain during the course of executing Trajectory 4.

3. During execution of the trajectory, received frequency and turn gain, leg position and current is recorded on-board. While the actual motion is recorded through the VICON motion capture system.

4. If at any time the vehicle needs to be stopped, the data collection is halted and the vehicle trajectory is terminated by sending a stop command.

3.2 Neural Network Design

All neural networks used in this research are feed-forward networks with a single hidden layer. The restriction to a single hidden layer network was heavily constrained by the desire to implement learning on-line in the future. This entails a significant limitation due to the vehicle size and power capacity. On-board computers have 512 MB of RAM and max processor speeds of less than 1.2 GHz. While training of the neural network was computed off-line, any true adaptability will require the network to be able to re-train mid trajectory, requiring the network to train on-line. To this end, although more complicated networks were considered and tested, they will not be given a detailed analysis in this paper as they are currently prohibitively expensive to train using on-board systems.

In Figure 6, we considered three different neural network configurations, with each configuration differing in required inputs but having the same outputs of forward, lateral and angular velocities. First, a simple mapping relating frequency and turn gain to the outputs was trained (Figure 6(a)). We expected (and found) that this network performed similarly to the previously mentioned curve fit model previously described in prior research. The network that corresponded to the experimental data improved with the increased complexity of inputs (Figure 6(b)) and again with (Figure 6(c)).

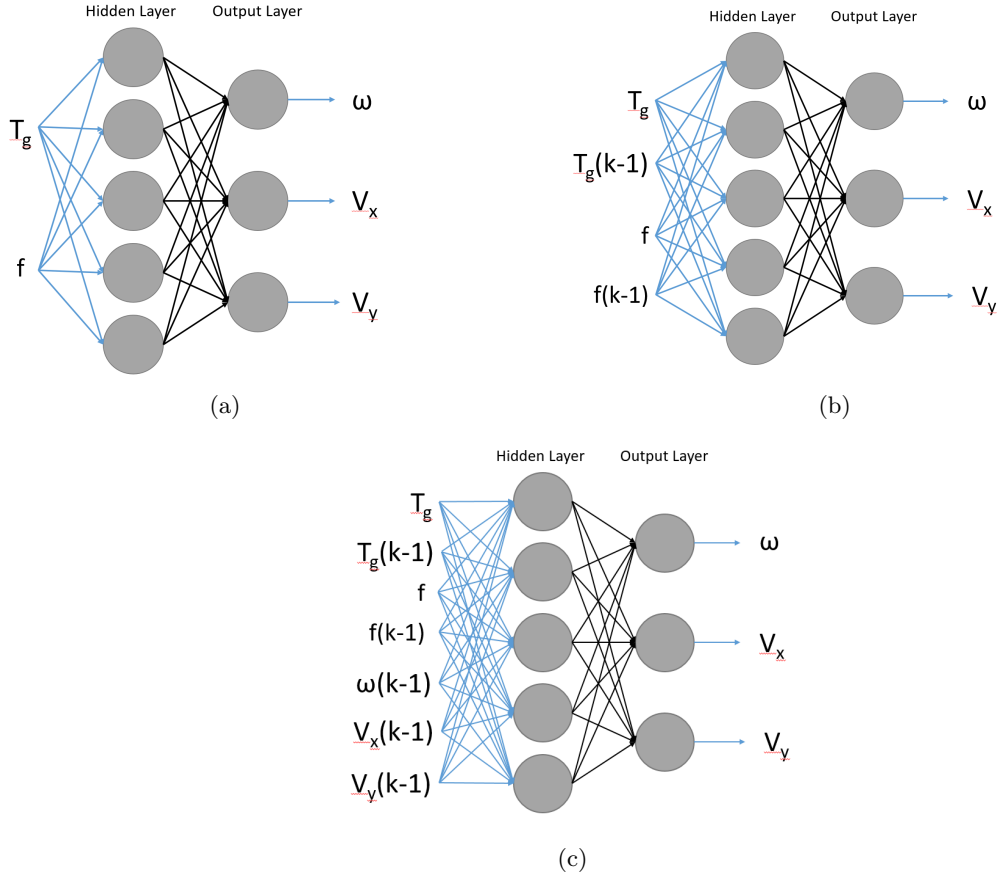

Figure 6. Structure of the three neural networks tested. While the inputs varied, the outputs remained the same. Starting with (a), we have the simple network ranging to the more complicated Input/Output network (c).

In fact, it was seen that it is critically important to take into account the dynamics of the vehicle, especially when operating at higher frequencies (higher forward velocity). As the neural networks of Figures 6(b) and 6(c) take into account information contained in past strides, they allow the network to have an idea of how past momentum will influence the predicted motion of the vehicle.

4. RESULTS

The performance of the neural networks corresponding to Figures 6(a), 6(b) and 6(c) will be expressed through:

1. Explanation of variance (R^2 value),
2. Mean Euclidean error (L_2) of our test data set between our predicted and actual velocities (angular, forward, lateral), defined as $\epsilon = \sqrt{\sum(Actual - Predicted)^2}$.

The R^2 values were obtained from the initial training of the network and are presented in Table 1. The more complicated neural network of Figure 6(c) was shown to be able to explain 99.85% of the variance in the predicted output compared to target output.

Table 1. R^2 Values for a Regression Fit Between the Target and Predicted Outputs.

	NN1	NN2	NN3
R^2 Training	0.94865	0.99727	0.99854
R^2 Testing	0.95483	0.99661	0.99783

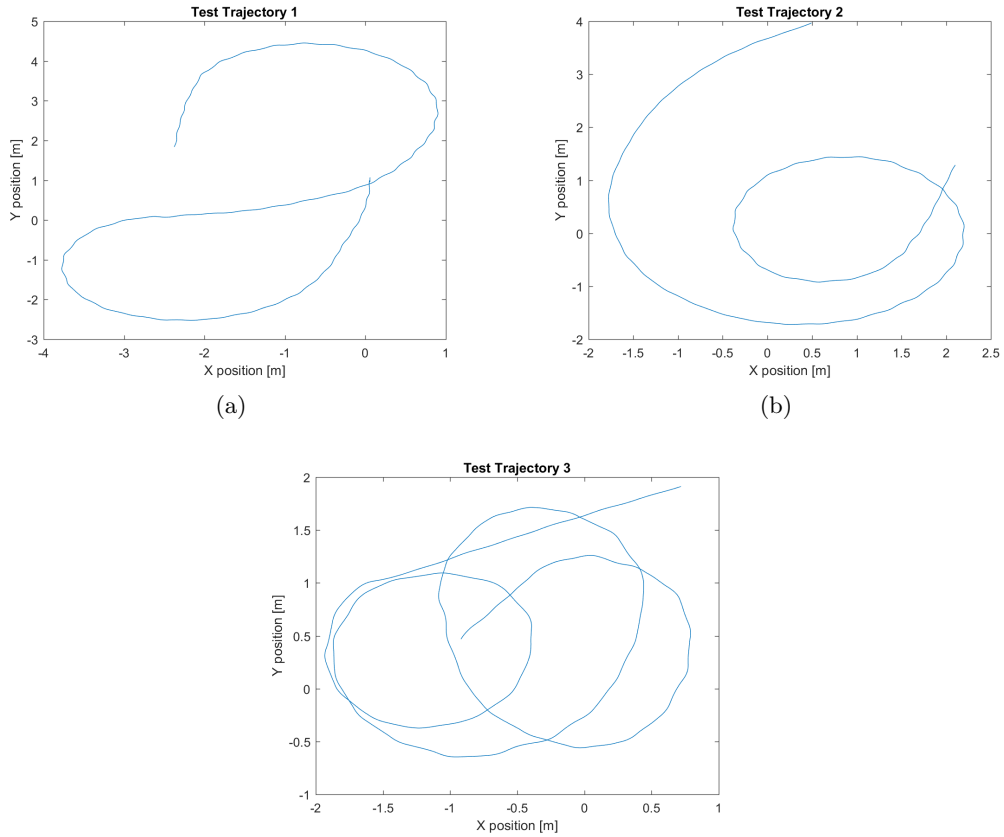

Figure 7. Three test paths in the X,Y plane. They were selected to test: a two-sided turn (a), a slow long turn (b) and a series of quick fast and slow turns (c). Note that along with the gaits, these paths define trajectories, i.e. time-dependent paths.

Table 2. Mean Error (L_2) for the Test Trajectory 1

	Angular Velocity	Lateral Velocity	Linear Velocity
NN1	2.59%	0.55%	1.93%
NN2	0.04%	0.49%	0.84%
NN3	0.04%	0.45%	0.32%

The three test cases shown in Figure 7 were used to verify the accuracy and predictive abilities of the networks. Each neural network was trained using data obtained from running experiments on twenty trajectories. The trajectories spanned multiple frequencies and turn-gains and ranged in duration from 18 to25 seconds. Tables 2-4 respectively show the resulting mean error (L_2) of all three networks with respect to test trajectories 6(a), 6(b), and 6(c).

It is seen that the errors monotonically decreased from NN1 to NN2 to NN3, indicating the importance of including information from past strides. The most dramatic improvements occurred in the angular velocity errors, which as whown in Table 2 reduced from 2.59% (NN1) to 0.04% (NN2 and NN3).

5. CONCLUSIONS AND FUTURE WORK

This research has shown the effectiveness of using a neural network to model the kinematics of a RHex-type robot. In performing this research a key to obtaining effective results was to express the experimental data in

Table 3. Mean Error (L_2) for the Test Trajectory 2

	Angular Velocity	Lateral Velocity	Linear Velocity
NN1	2.21%	0.54%	0.66%
NN2	0.04%	0.49%	0.36%
NN3	0.04%	0.38%	0.27%

Table 4. Mean Error (L_2) for the Test Trajectory 3

	Angular Velocity	Lateral Velocity	Linear Velocity
NN1	4.91%	0.73%	0.99%
NN2	0.16%	0.65%	0.75%
NN3	0.08%	0.64%	0.64%

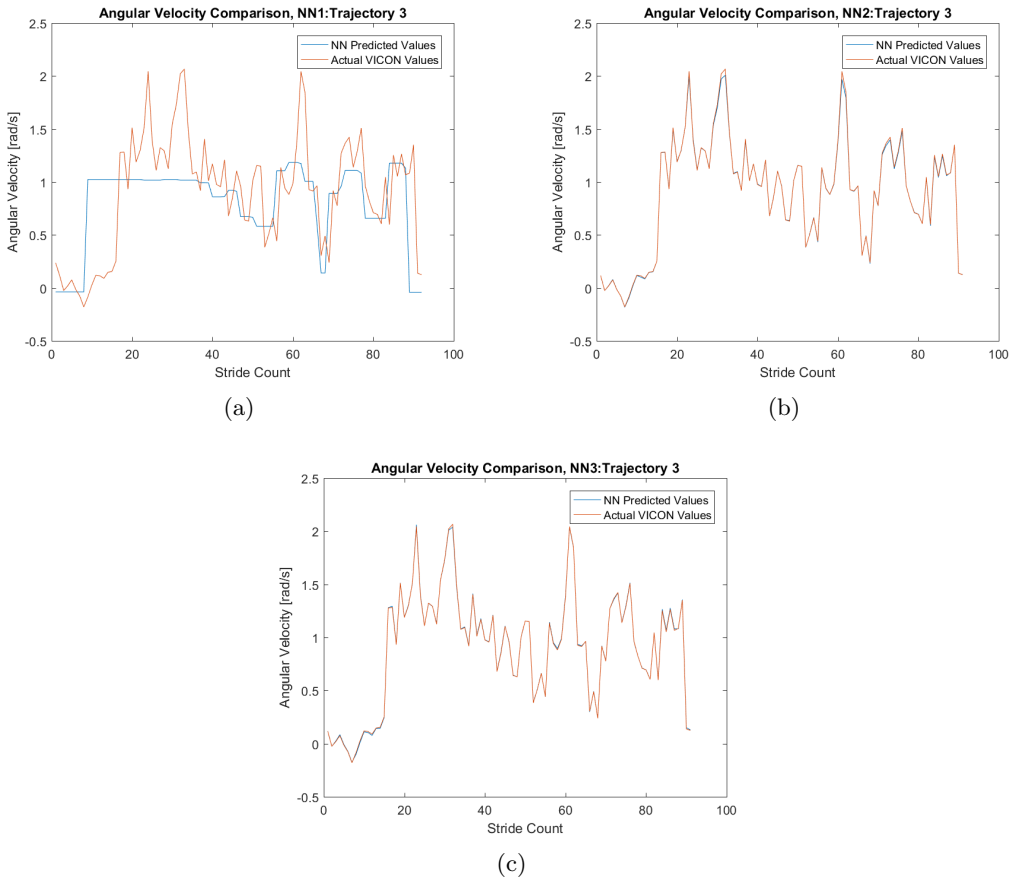

Figure 8. Angular velocity prediction of (a) NN1, (b) NN2 and (c) NN3.

terms of the robot stride. This led to a dramatic reduction in the data, from 30,000 data points to less than 1,000 data points, but more importantly enabled the neural networks to better match the experimental data.

Also, it was found that inclusion of past stride data as inputs to the neural network dramatically improved performance. Tables 2- 4 and Figures 8 and 9 illustrate this improvement and also show that the improvement from NN1 to NN2 was far greater than the improvement from NN2 to NN3. As NN2 is simpler to implement in real time than NN3, it is conjectured here that it is the preferred network for motion planning.

Work has already commenced on using a trained neural network as the kinematic model with Sampling-Based Model Predictive Optimization (SBMPO),[6] which enables generation of optimal trajectories by taking advantage

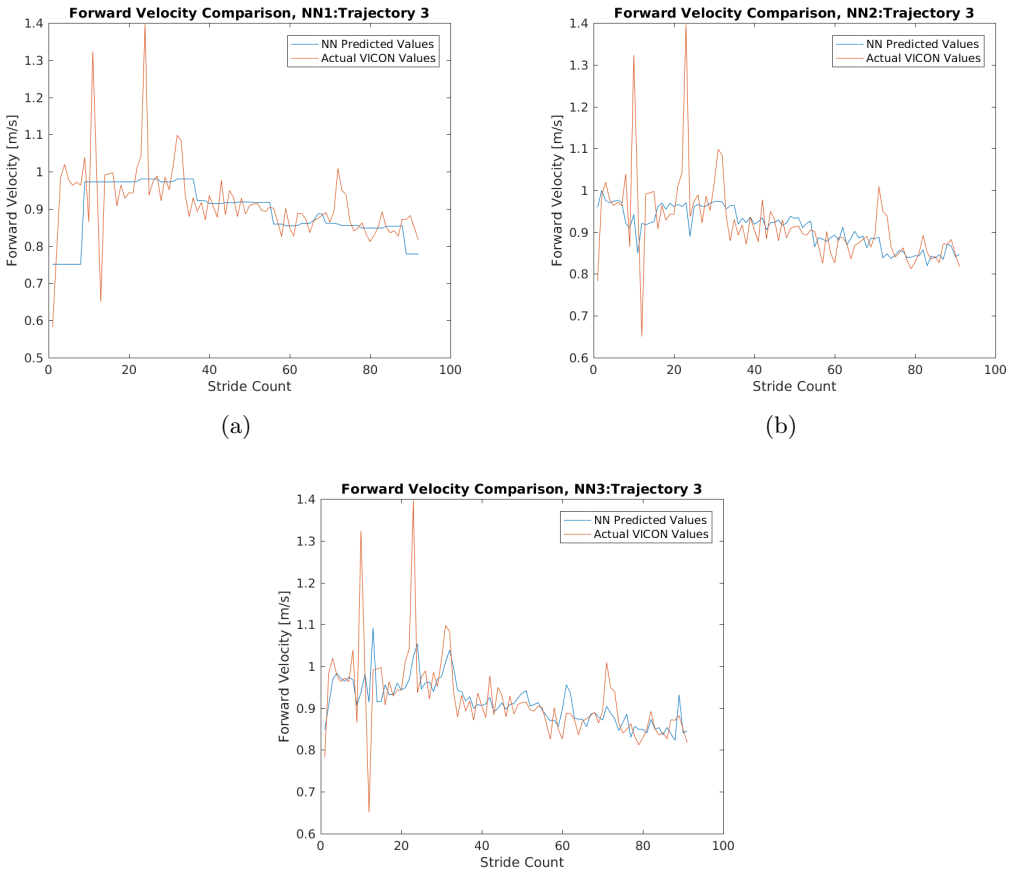

Figure 9. Forward velocity prediction of (a) NN1, (b) NN2 and (c) NN3.

of kinematic or dynamic models of the system under consideration. Furthermore, work will be done to allow the vehicle to periodically adapt and learn the neural network model while it is in regular operation.

Also under consideration is the use of more complicated neural networks. Preliminary results show that deep learning networks can fit the data better than the simple networks employed in these initial findings. "Memory" based networks such as LSTM networks are also being considered to obtain better predictions of the robot motion.

ACKNOWLEDGMENTS

This work was supported by the collaborative participation in the Robotics Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD 19-01-2-0012. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation thereon.

REFERENCES

[1] Bode, M. W., "Learning the forward predictive model for an off-road skid-steer vehicle," Tech. Rep. CMU-RI-TR-07-32, Robotics Institute, Pittsburgh, PA (September 2007).

[2] Ordonez, C., Gupta, N., Collins, E. G., Clark, J., and Johnson, A. M., "Power modeling of the XRL hexapedal robot and its application to energy efficient motion planning," in [*Proceedings of the International Conference on Climbing and Walking Robots*], 689–696 (July 2012).

[3] Yu, W., Chuy, O., Collins, E., and Hollis, P., "Analysis and experimental verification for dynamic modeling of a skid steered vehicle," *IEEE Transactions on Robotics* **26**, 340–353 (April 2010).

[4] Haynes, G. C., Pusey, J., Knopf, R., Johnson, A. M., and Koditschek, D. E., "Laboratory on legs: an architecture for adjustable morphology with legged robots," in [*Unmanned Systems Technology XIV*], Karlsen, R. E., Gage, D. W., Shoemaker, C. M., and Gerhart, G. R., eds., **8387**, 83870W, SPIE (2012).

[5] Saranli, U., Buehler, M., and Koditschek, D. E., "Rhex: A simple and highly mobile hexapod robot," *International Journal of Robotics Research* **20**, 616 – 631 (July 2001).

[6] Dunlap, D., Caldwell, C., and Collins, E., "Nonlinear model predictive control using sampling and goal-directed optimization," in [*Proceedings of the Multi-Conference on Systems and Control*], (September 8-9 2010).