Plant Modeling for an Autonomous Vehicle

Nick Nauman, Hannah Grady, Advisor: Dr. Suruz Miah

Department of Electrical and Computer Engineering, Bradley University, Peoria IL

Objective and Contributions

Objective

- Provide accurate plant models of each autonomous vehicle subsystem to be used for designing controllers

Contribution

- Determine if System Identification or Neural Network modeling produces better models
- Non-linearity modeling

Applications

- Use in testing to help develop more accurate vehicle controllers
- Create a guide for modeling future vehicle subsystems

Problem Setup

- Conducted a literature review to look for existing solutions • System Architecture

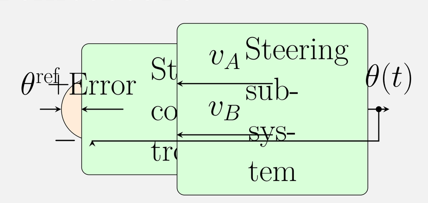


Figure 1:Steering subsystem block diagram.

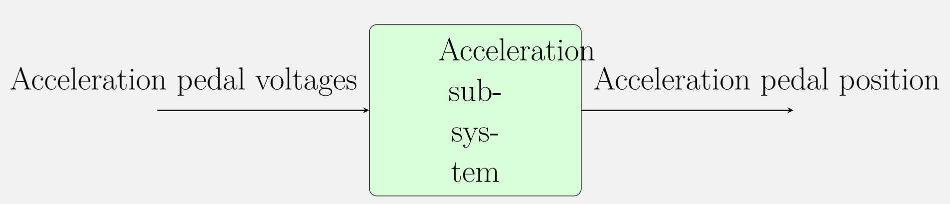


Figure 2:Acceleration subsystem block diagram

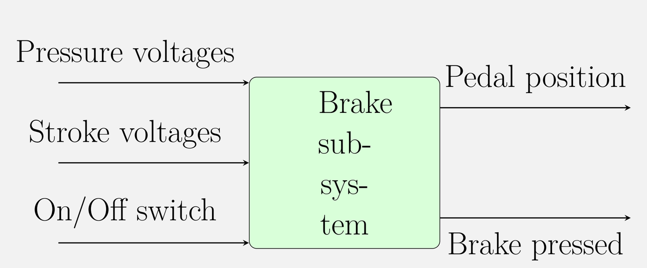


Figure 3:Brake subsystem block diagram

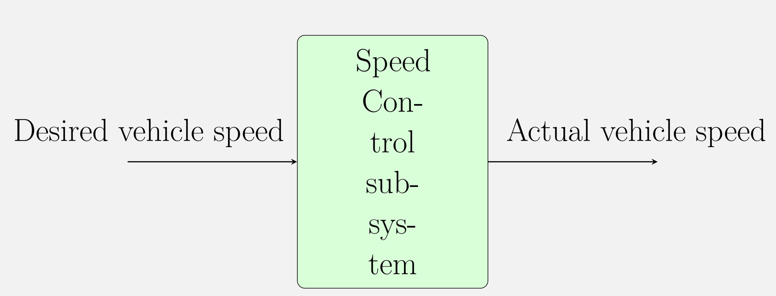


Figure 4:Speed Control subsystem block diagram

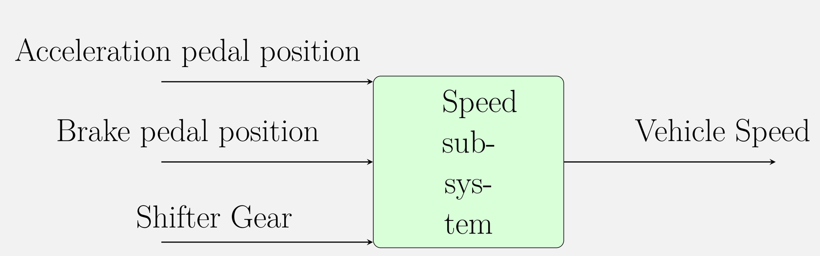


Figure 5:Speed subsystem block diagram

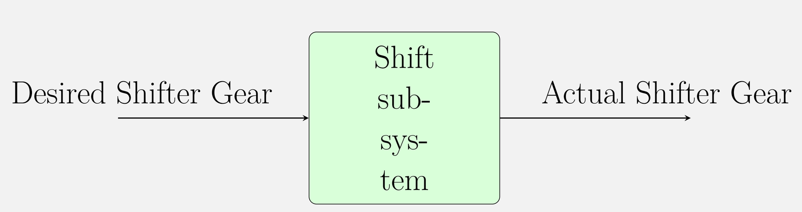


Figure 6:Shift subsystem block diagram

- Collected data using a Lexus RX450H vehicle platform

Transfer Function Modeling

- MATLAB's System Identification Toolbox used to create models
- Models needed to meet best fit and error requirements

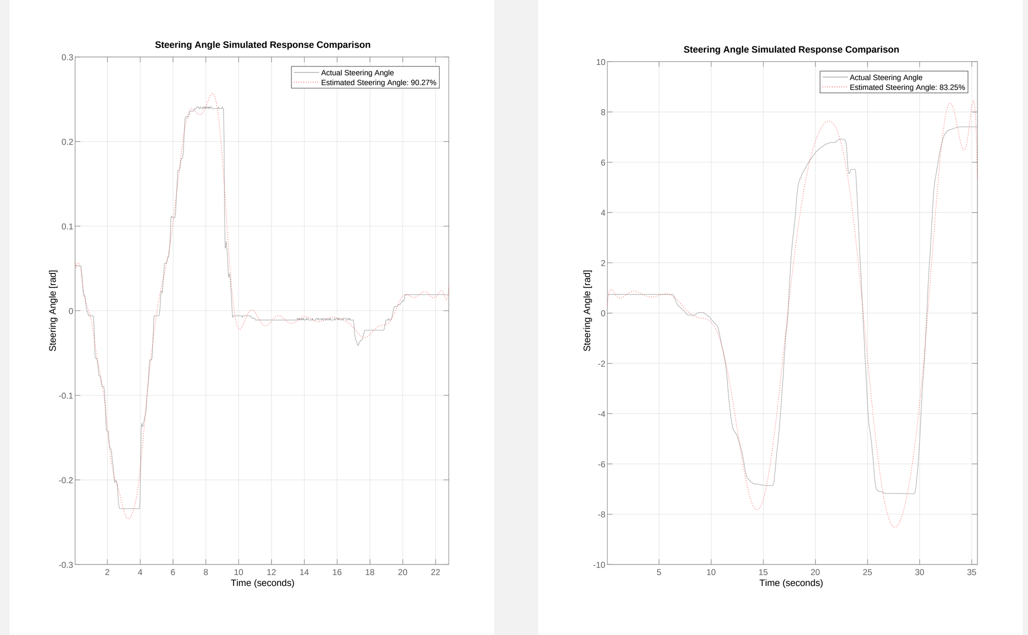


Figure 7:Steering System Estimated Steering Angle Comparison

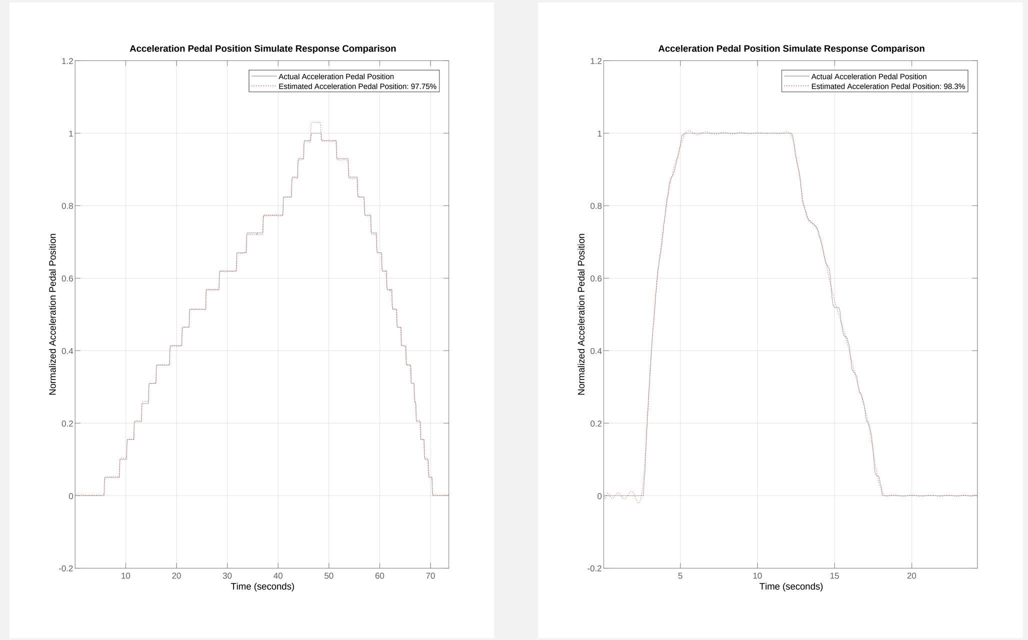


Figure 8:Acceleration System Estimated Pedal Position Comparison

Neural Network Modeling

- Used MATLAB's Neural Network Time Series App
- Generated models using the Bayesian Regularization Algorithm
- Models trained using collected log data

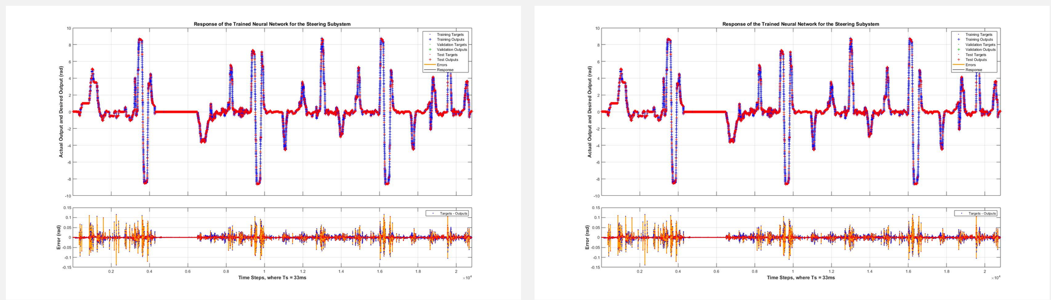


Figure 9:Steering System Training Plots

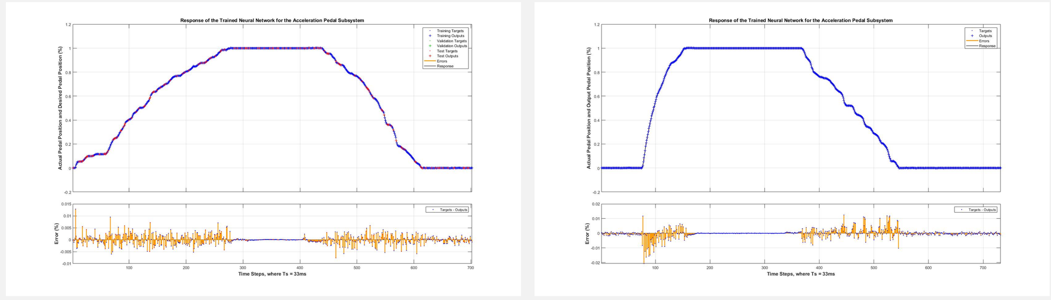
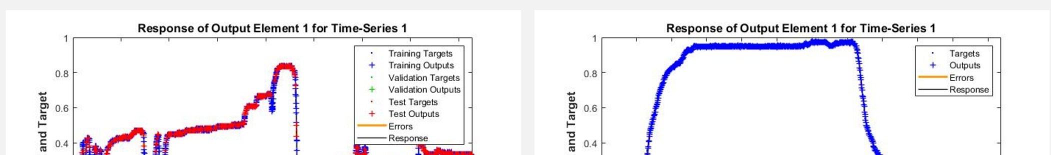


Figure 10:Acceleration System Training Plots



Neural Network Algorithm

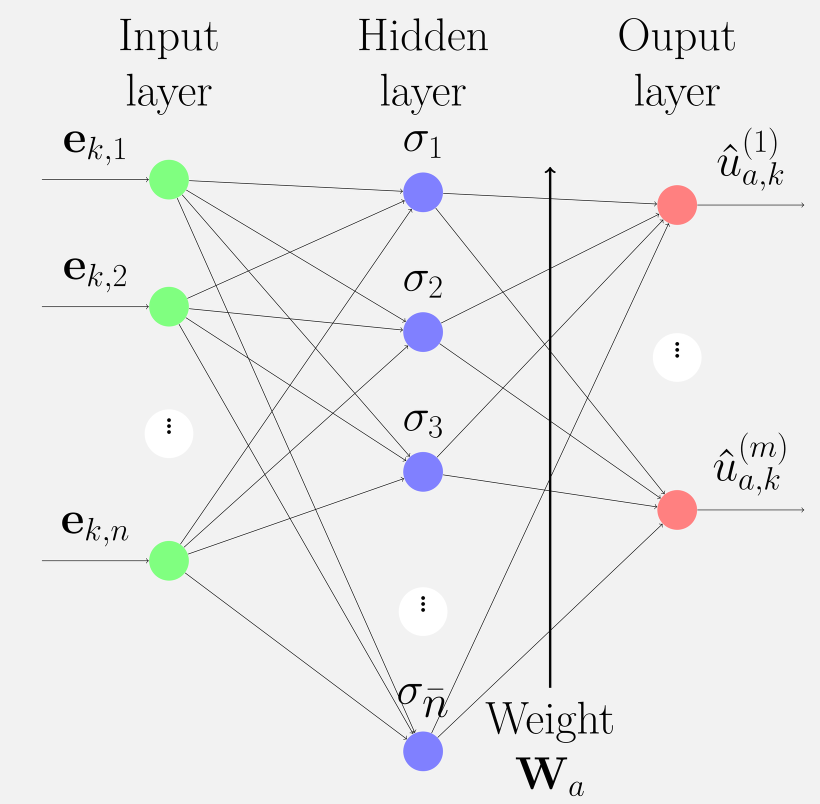


Figure 12:Actor neural network structure for approximating control input.

1. Approximate neural network signal
$$\hat{\mathbf{u}}_{a,k} = \mathbf{W}_a^T \boldsymbol{\sigma}(\mathbf{e}_k),$$
2. Define the weight matrix
$$\mathbf{W}_a = \begin{bmatrix} w_a^{1,1} & w_a^{1,2} & \dots & w_a^{1,m} \\ w_a^{2,1} & w_a^{2,2} & \dots & w_a^{1,m} \\ \vdots & \ddots & \dots & \vdots \\ w_a^{\bar{n},1} & w_a^{\bar{n},2} & \dots & w_a^{\bar{n},m} \end{bmatrix}.$$
3. Compute the sum-of-squared error
$$\boldsymbol{\Sigma}\mathbf{W}_a \approx \mathbf{U}^{[d]}, \qquad \text{with}$$
$$\boldsymbol{\Sigma}\mathbf{W}_a \equiv \hat{\mathbf{U}}$$
$$\delta_a = \frac{1}{2}\|\mathbf{U}^{[d]} - \hat{\mathbf{U}}\|^2$$
4. Find the actor weight by minimizing the sum-of-square error
$$\delta_a = \frac{1}{2}\mathrm{Tr}\left[(\mathbf{U}^{[d]} - \boldsymbol{\Sigma}\mathbf{W}_a)^T(\mathbf{U}^{[d]} - \boldsymbol{\Sigma}\mathbf{W}_a)\right].$$
5. Take the derivative of δ_a with respect to $\mathbf{W}_a$ and set $\frac{\partial \delta_a}{\partial \mathbf{W}_a}$ to zero
$$\frac{\partial \delta_a}{\partial \mathbf{W}_a} = -\boldsymbol{\Sigma}^T\mathbf{U}^{[d]} + \boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\mathbf{W}_a$$
$$\mathbf{W}_a = \left(\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\right)^T \boldsymbol{\Sigma}^T\mathbf{U}^{[d]}.$$
6. Use the gradient descent approach to minimize the least square order alter weights when training the neural network
$$\begin{aligned}\mathbf{W}_a^{(r+1)} &= \mathbf{w}_a^{(r)} - \ell_a \frac{\partial \delta_a}{\partial \mathbf{W}_a} \\ &= \mathbf{W}_a^{(r)} - \ell_a\left(-\boldsymbol{\Sigma}^T\mathbf{U}^{[d]} + \boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\mathbf{W}_a^{(r)}\right) \\ &= \mathbf{w}_a^{(r)} - \ell_a\boldsymbol{\Sigma}^T\left(\boldsymbol{\Sigma}\mathbf{W}_a^{(r)} - \mathbf{U}^{[d]}\right)\end{aligned}$$

Experimental Results

- Preliminary model testing conducted before sending models to AutonomouStuff
- Official testing conducted using dSPACE software and HIL Bench

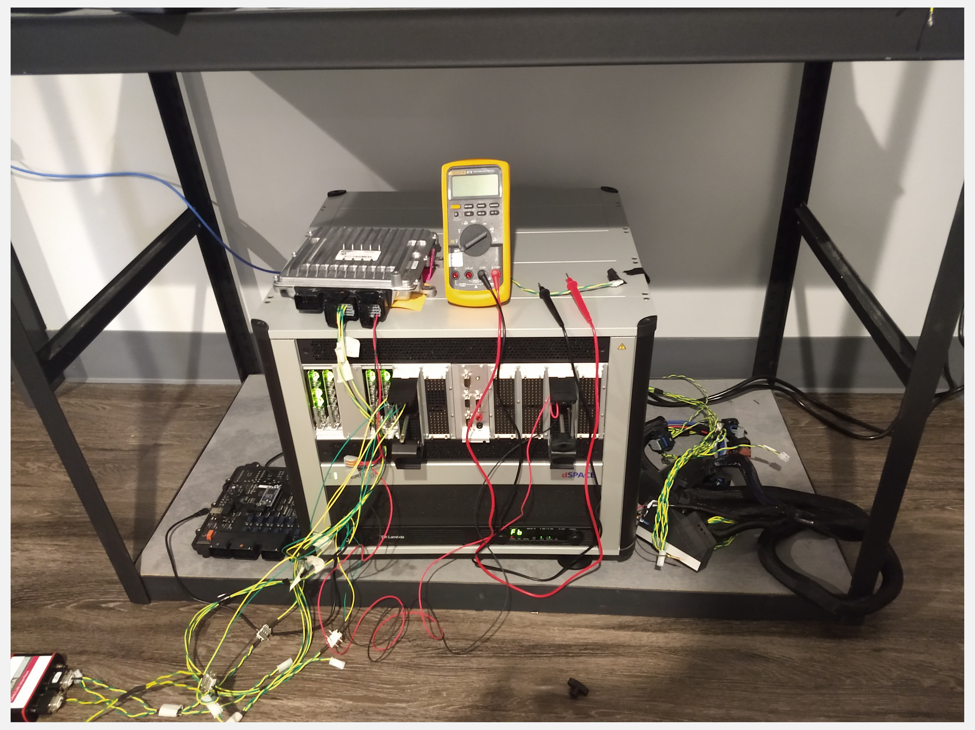


Figure 13:Experimental Setup

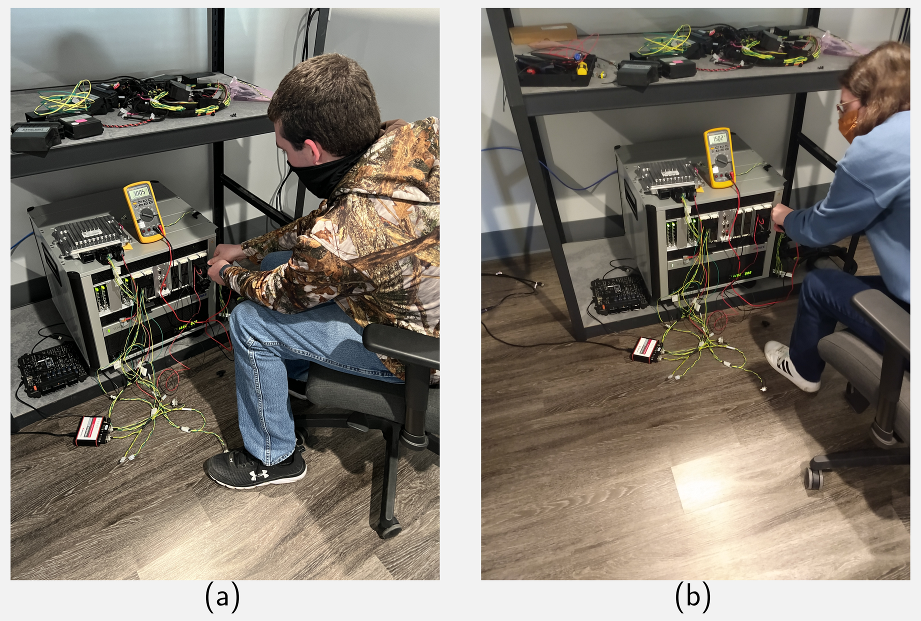


Figure 14:Verifying voltage measurements on the HIL Bench

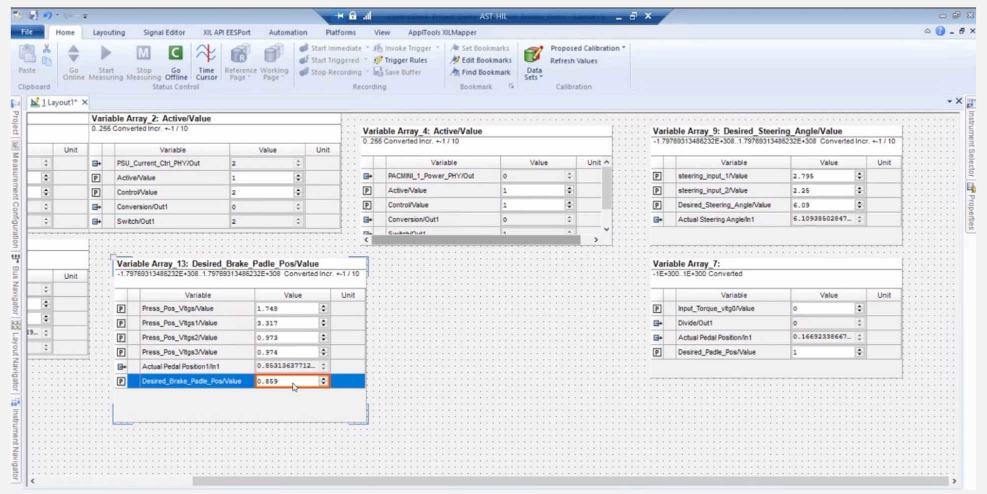


Figure 15:Model validation setup

Conclusion and Future Work

- Using Neural Networks produced more accurate models than System Identification
- Test models using Hardware-in-the-Loop
- Create new vehicle controllers