NONLINEAR CONTROL OF A GROUND VEHICLE USING DATA-DRIVEN DYNAMIC MODELS

A Thesis

by

SAMIR HAMED HASSEN

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Swaminathan Gopalswamy
Committee Members,	Srikanth Saripalli
	Dezhen Song
Head of Department,	Andreas Polycarpou

May 2020

Major Subject: Mechanical Engineering

ABSTRACT*

As autonomous vehicles continue to grow in popularity, it is imperative for engineers to gain greater understanding of vehicle modeling and controls under different situations. Most research has been conducted on on-road ground vehicles, yet off-road ground vehicles which also serve vital roles in society have not enjoyed the same attention. The dynamics for off-road vehicles are far more complex due to different terrain conditions and 3D motion. Thus, modeling for control applications is difficult. A potential solution may be the incorporation of empirical data for modeling purposes, which is inspired by recent machine learning advances, but requires less computation. This thesis presents results for empirical modeling of an off-road ground vehicle, Polaris XP 900. As a first step, data was collected for 2D planar motion by obtaining several velocity step responses. Multivariable polynomial surface fits were performed for the step responses. Sliding mode control layered on top of pure pursuit guidance is then used to drive the vehicle for waypoint following, using the empirical model. Simulation and experimental results show that the vehicle can perform waypoint following for a circular and sinusoidal with minimal error. Furthermore, more experimental data was collected to show the effects of adaptive velocity and adaptive look-ahead for path tracking. A comparison of the controller's performance was also explored between on-road and off-road terrain.

ACKNOWLEDGMENTS

I would like to thank my advisor Dr. Swaminathan Gopalswamy for his constant support throughout my graduate academic career. I would also like to thank Dr. Srikanth Saripalli for letting me use some of his equipment and his guidance on how to use them. Also, DJ Franklin, Kenny Chour, and Andrew Weaver were a huge help with this project. I appreciate everyone mentioned for helping me on this journey!

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Swaminathan Gopalswamy, my advisor; Professor Srikanth Saripalli in the Department of Mechanical Engineering; and Professor Dezhen Song in the Department of Computer Science and Engineering.

For experiments, DJ Franklin, Kenny Chour, and Andrew Weaver were safety drivers and helped collect data for this project.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

The work reported in this paper was partially funded by Army Research Labs through DCIST CRA Award W911NF-17-2-0181, subaward 576433

NOMENCLATURE

RNN	Recurrent Neural Network
ROS	Robot Operating System
PID	Proportional, Integral, and Derivative control
LQR	Linear Quadratic Regulator
$f(\hat{x})$	The approximation of a function $f(x)$ based on measured data
PACMOD	A software and hardware suite that enables drive-by-wire capabilities on the Polaris Ranger

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

1. INTRODUCTION AND LITERATURE REVIEW

Autonomous vehicle development has been making rapid strides, and they promise many benefits such as eliminating emissions, reducing road accidents, and improving traffic conditions. Yet most autonomous research has only been involved in on-road vehicles owing to their ubiquitous use in daily life. Many important vehicle operations take place on unpaved off-road tracks made of materials such as dirt, sand, gravel, snow, rocks, and other natural terrain. Compared to traditional on-road vehicles which can only operate on standardized road ways, off-road vehicles serve far more versatile roles in many different fields of applications. Such applications include military, agricultural, recreational, transportation, forestry, recreational and tourism. Given the important roles filled by off-road vehicles, it warrants further study to improve their reliability and efficiency. Also compared with traditional vehicles, off-road vehicles bring greater engineering challenges in the form of modeling and controls. With off-road vehicles, there are additional variables to consider, such as terrain type and off-road vehicle structure.

The first step in vehicle analysis is to obtain an accurate motion model, typically derived from energy conservation, first principles, or geometry. For on-road vehicles, the kinematic bicycle model is commonly used to model a 4-wheeled ground vehicle without capturing inertial information. Yet, at higher longitudinal speeds, the kinematic bicycle model suffers from path deviation due to nonlinearities. On the other hand, the dynamic bicycle model is used to model the same ground vehicle moving at higher speeds, but may perform poorly at very low speeds due to singularity issues with tire force calculations [1]. Clearly, for off-road vehicles, these models cannot adequately describe the complex tire-terrain interaction or 3D motion. On the other hand, highly complex models have been derived which describe one aspect or several aspects of the vehicle. 6-DOF models which describe the full vehicle dynamic behavior serve as useful simulation tools in [2], [3], [4], [5]. Also simpler models which omit the suspension are also common [6]. Important for the 6-DOF model is the calculation of tire forces, which are arguably the most important consideration in modeling. Utilizing different tire force models may yield drastically different motion

results. Tire-terrain interaction forces for different soil conditions are considered in [7]. A key idea is the inclusion of empirical data to aid in the modeling process. Nonlinear and non-steady state tire modeling for dynamic simulation was presented in [8]. The authors term their model "semi-physical" which includes empirical data for modeling purposes. Clearly, the examination of the presented models reveal they are complex and highly involved, yet only describe a limited aspect of the vehicle.

As an alternative, research efforts have begun to shift towards a new paradigm; the so called model-free approach. Under this regime, tools from the field of machine learning such as neural networks and its variants are used along side measured input-output data to produce black-box models. A survey of techniques can be found in [9]. In [10], a real-time recurrent neural network (RNN) is described which is used to generalize lateral vehicle motion. However, longitudinal dynamics were not considered. In [11], suspension and tire dynamics were also simulated using RNN. The model was able to predict vertical forces with sufficient accuracy, but had issues with longitudinal forces. Finally, in [12], radial basis function networks were trained on pure slip conditions, but tested on combined slip, which provided results within 1% accuracy.

The focus of machine learning approaches is on producing a working model with less emphasis placed on understanding the physics underlying the system's behavior. Additionally, in many cases, an extremely large data set is required for training. A good compromise between the analytical and model-free approaches may come in the form of empirical data-fitting. Using data-driven dynamics, collected data is fit using simpler mathematical models such as n-th degree multivariable polynomials. A famous example is Pacejka's empirical tire model.

Once an acceptable vehicle model is found, control theory can then be applied to drive the system to a desired output(s). For model-based controls, linear control theory such as LQR can be used. Additionally, model-less techniques such as PID and advanced variants can produce good results, though they become cumbersome with high degree of freedom systems. Alternatively, typical nonlinear control design methodologies include feedback linearization, the backstepping method, or control-lyapunov functions. However these methods depend on having a highly accu-

rate analytical model. Robust control techniques such as sliding mode control, however, can prove useful in the face of uncertainties. In [13], a combination of an analytical physics-based model for the lateral control and a model-free controller for longitudinal control was used. The method was simple to implement resulting in securing first place in the DARPA Grand Challenge in 2005. This method may be improved by using data-driven models to capture complex nonlinearities in the system to aid in control system design.

In this paper, data-driven approaches were utilized to empirically derive a model for an off-road vehicle, a drive-by-wire enabled Polaris XP 900. As a first step of study, planar motion is studied. After collecting motion data through sensors, the data is fit using high order multivariable polynomials. Finally, to validate the fit, path following is observed using sliding mode control inspired by pure pursuit. A few works have used this concept ([14], [15], [16]), but none to date– to the best of our knowledge–have used data-driven models for control of a ground vehicle. This control methodology is adopted to both overcome imperfections in the modeling phase and specify targets along a path. Simulation and experimental results show that vehicle modeling and control through data-driven approaches is viable, with good results.

2. DATA-DRIVEN DYNAMIC MODEL

In this paper, for any function $f(.)$, the function $\hat{f}(.)$ will denote that this is an approximation of the original function based on measured data.

2.1 Dynamic Model

First, the vehicle motion model in the body frame is represented, as below:

$$\begin{aligned} \dot{v_x} - \dot{\theta} v_y &= f_1(v_x, v_y, \alpha, \beta) \\ \dot{v_y} + \dot{\theta} v_x &= f_2(v_x, v_y, \delta) \\ \dot{\theta} &= f_3(v_x, v_y, \delta) \\ \begin{bmatrix} \dot{X_g}, \dot{Y_g} \end{bmatrix}^T &= {}^G R(\theta)_B \begin{bmatrix} v_x, v_y \end{bmatrix}^T \end{aligned} \tag{2.1}$$

where v_x, v_y and $\dot{\theta}$ are the longitudinal, lateral, and angular velocities, respectively, of the vehicle in the body frame, and X_g, Y_g represents the velocity of the vehicle in the global frame. ${}^G R(\theta)_B$ is the rotational transformation matrix associated with the vehicle or the orientation of the vehicle frame, called B, with respect to ground coordinate frame G. Additionally, $\alpha \in [0, 1]$ is the throttle command, $\beta \in [0, 1]$ is the brake command, and $\delta \in [-7.854, 7.854]$ is the steering wheel command. The key point to note in (2.1) is that the right hand side of the equation represents quantities that can be measured through experiments (first three lines) or are known analytically (fourth line). Specifically the first two items represent the longitudinal and lateral accelerations, and the third item represents the angular velocity, and all these quantities can be measured through standard sensors such as an inertial measurement unit, or by using kinematics and knowing the actual path traversed by the vehicle.

2.2 Assumptions

There are some assumptions about the model presented in eq. (2.1):

- As a first step, all work performed have the assumption of no side-slip condition, i.e. $|v_y| << |v_x|$, hence dynamics associated with v_y in (2.1) can be ignored. Additionally, the throttle and braking commands are combined as one input u such that $u \in [-1, 1]$ where $u < 0$ for braking and $u \geq 0$ for throttle. Thus, eq. (2.1) becomes:

$$\begin{aligned} \dot{v_x} &= f_1(v_x, u) \\ \dot{\theta} &= f_3(v_x, \delta) \\ \begin{bmatrix} \dot{X}_g & \dot{Y}_g \end{bmatrix}^T &= {}^G R(\theta)_B \begin{bmatrix} v_x & 0 \end{bmatrix}^T \end{aligned} \tag{2.2}$$

- Explicit representation of the suspension dynamics are neglected. However, since the model will be data driven (i.e. the RHS of eq. (2.2) is obtained through experimentation), this assumption translates to neglecting some transient dynamics between actuation and realized accelerations or angular velocities.

- Actuator dynamics are also neglected, i.e. the relationship between the actuation inputs (namely α, β and δ) as directly impacting the achieved accelerations and angular velocity. Effectively this translates to neglecting the transient dynamics, similar to the previous assumption.

3. NONLINEAR CONTROL STRATEGY

In this section, a control strategy is developed that utilizes the data-driven dynamic model established in the prior section. The control strategy involves a high level outer loop that determines desired quantities to track, and a low-level controller using sliding mode control. Figure 3.1 visually represents the overall architecture of the control strategy.

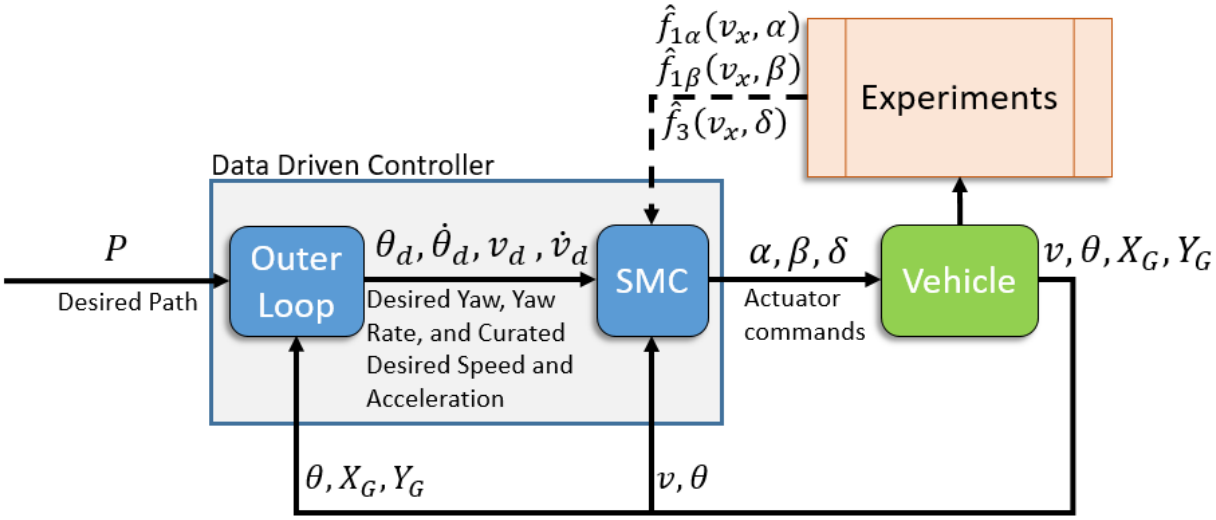

Figure 3.1: Overall diagram for the nonlinear control strategy.

3.1 High Level Strategy

The overall objective is to track a desired path $P(s)$ parameterized by s, at a desired velocity v_d. Inspired by the simplicity of the pure-pursuit approach [17] [18], its look-ahead manner was combined with the robustness of sliding mode control. First, the target position is identified $(X_{g,d}, Y_{g,d})$ to steer towards, using a look-ahead distance d and based on the current position (X_g, Y_g):

$$\begin{bmatrix} X_{g,d}, Y_{g,d} \end{bmatrix}^T = \text{LookAhead}(P, \begin{bmatrix} X_g, Y_g \end{bmatrix}^T, d) \tag{3.1}$$

Defining $\vec{r_g} := (X_g, Y_g) \rightarrow (X_{g,d}, Y_{g,d})$ as the vector from the current location to the look ahead point on the path, in the global reference frame, the desired heading θ_d as well as the desired yaw rate $\dot{\theta_d}$ are defined as below:

$$\begin{aligned} \hat{r} &= \vec{r_g}/||r_g|| \\ \hat{r_b} &= {}^{B}R(\theta)_{G}\vec{r_g} \\ e_{lat} &= (\hat{r_b})_y \\ \theta_d &= \tan 2^{-1}(\hat{r}_y, \hat{r}_x) + k_p e_{lat} \\ \kappa &= \frac{2\sin(\theta_d - \theta)}{d} \\ \dot{\theta_d} &= \kappa v_d \end{aligned} \tag{3.2}$$

Here r_g is used to calculate the desired vehicle heading θ_d using simple geometric relationships. Additionally, the desired heading vector $\hat{r}$ is projected into the body frame $\hat{r_b}$, in order to calculate a lateral error e_{lat}. Together, the two quantities form the total desired vehicle yaw θ_d. Using pure pursuit relations, curvature κ can also be derived, which defines a circular arc passing through the target point and the vehicle's current position. Finally, the desired angular velocity $\dot{\theta}_d$ can be found via curvature kinematics.

Thus, the quantities θ_d, $\dot{\theta}_d$, and v_d form the desired trajectories to track for the low level sliding mode controller. For added comfort, v_d may be defined as follows

$$v_d = \min\left(v_{max}, \max\left(v_{min}, \sqrt{(A_{ylim}/\kappa)}\right)\right) \tag{3.3}$$

Equation (3.3) effectively bounds v_d above and below with v_{max} and v_{min} respectively, while also ensuring that commanded velocity is lowered during sharp corners (high κ values). This is due to the lateral acceleration limit, A_{ylim}, that the designer selects in the control strategy.

3.2 Nonlinear Robust Control

Recalling the assumption on $|v_y| << |v_x|$, the following control objectives is presented:

$$\begin{aligned} v_x &\longrightarrow v_d \\ \theta &\longrightarrow \theta_d \end{aligned} \tag{3.4}$$

3.2.1 Error Dynamics

Based on the lower level control objectives, two error terms are defined as below:

$$\begin{aligned} e_v &= v_x - v_d \\ e_\theta &= \theta - \theta_d \end{aligned} \tag{3.5}$$

The error terms can be used to define linear sliding surfaces upon which we will enforce first order behavior on the linear system. In other words, we wish to define a sliding surface per degree of freedom as follows:

$$s_i = \left(\frac{d}{dt} + \lambda_i\right)^{\rho-1} e_i \quad \text{where} \quad i = v, \theta \tag{3.6}$$

Here, ρ is the relative degree of the state. Since, the relative degree ρ is 1, $s_i = e_i$. Now, taking the derivatives of these error terms and using eq. (2.2), the following is obtained:

$$\begin{aligned} \dot{e_v} &= \hat{f}_{1u}(v_x, u) - \dot{v_d} + \Gamma_v \\ \dot{e_\theta} &= \hat{f}_3(v_x, \delta) - \dot{\theta}_d + \Gamma_\theta \end{aligned} \tag{3.7}$$

where Γ_v and Γ_θ represent the unstructured uncertainty in the modeling because of the approximations in the surface fits of the data, the errors in the data measurements, and any unmodeled dynamics.

3.2.2 Controller Assumptions

The key assumption made is that the inaccuracies can be bounded:

$$\begin{aligned} |\Gamma_v| &< \Gamma_{vmax} \\ |\Gamma_\theta| &< \Gamma_{\theta max} \end{aligned} \tag{3.8}$$

3.2.3 Core Control Strategy

Inverse dynamics are utilized for a known model in order to implement sliding mode control. The control strategy is defined as below:

$$\begin{aligned} u &= \hat{f}_{1u}^{-1}(v_x, \dot{v}_d + \phi_v(e_v)) \\ \delta &= \hat{f}_3^{-1}(v_x, \dot{\theta}_d + \phi_\theta(e_\theta)) \end{aligned} \tag{3.9}$$

where ϕ_v and ϕ_θ represent the robust feedback terms. Using eq. (3.9) with eq. (2.2), the following is obtained:

$$\begin{aligned} \dot{v_x} = \dot{v}_d + \phi_v(e_v) \quad &\Rightarrow \dot{e_v} - \phi_v(e_v) = 0 \\ \dot{\theta} = \dot{\theta}_d + \phi_\theta(e_\theta) \quad &\Rightarrow \dot{e_\theta} - \phi_\theta(e_\theta) = 0 \end{aligned} \tag{3.10}$$

Several options have been studied for these in literature in order to impose the so-called Sliding Conditions [19]. A common approach is to use signum function on error as follows:

$$\begin{aligned} \phi_v(e_v) &= -\lambda_v \,\mathrm{sgn}(e_v) \\ \phi_\theta(e_\theta) &= -\lambda_\theta \,\mathrm{sgn}(e_\theta) \end{aligned} \tag{3.11}$$

where λ_v and λ_θ are control constants to be chosen.

However, using the signum function may introduce undesirable chattering phenomena due to its non-continuous switching feature. Thus, we will use a piece-wise continuous function such as the saturation or sigmoidal function such that $f_i : \mathcal{R} \rightarrow [-1, 1]$:

$$\begin{aligned} f_i &= \left(\frac{2}{1 + \exp\left(-\kappa_i e_i\right)} - 1 \right), \quad \text{for} \quad i = v, \theta \\ \phi_v(e_v) &= -\lambda_v \, f_v \\ \phi_\theta(e_\theta) &= -\lambda_\theta \, f_\theta \end{aligned} \tag{3.12}$$

3.2.4 Controller Feasibility

The inverse functions $\hat{f}_{1u}^{-1}$ and $\hat{f}_3^{-1}$ must exist in order for the controller defined in eq. (3.9) to be feasible. Analytical solutions are presented later in *Inverse Functions*.

3.2.5 Stability and Robustness

In the simplified dynamics of eq. (2.2) with the assumption of zero side slip, we have to concern ourselves with only two states when discussing stability, namely v_x and θ. Alternatively, we can consider the corresponding errors e_v and e_θ as defined in eq. (3.5).

We can expand the error dynamics in eq. (3.7) using the control definitions in eq. (3.9) and eq. (3.12) to get

$$\begin{aligned} \dot{e_v} &= \phi_v(e_v) + \Gamma_v = -\lambda_v \, f_v + \Gamma_v \\ \dot{e_\theta} &= \phi_\theta(e_\theta) + \Gamma_\theta = -\lambda_\theta \, f_\theta + \Gamma_\theta \end{aligned} \tag{3.13}$$

Considering eq. (3.8), if we choose $\lambda_v > \Gamma_{vmax}$ and $\lambda_\theta > \Gamma_{\theta max}$, eq. (3.13) leads to $(e_v, e_\theta) \longrightarrow \epsilon > 0$ in finite time, where ϵ is some non-zero residual about the origin. Within the boundary layer, the switching function is approximated by a smooth sigmoidal function as described before. Thus, instead of driving the system error states to zero and risk chattering, the states are instead driven to remain inside the boundary layer to remove chattering. We then obtain acceptable trajectory

tracking independent of the modeling errors Γ_v and Γ_θ.

4. DATA COLLECTION EXPERIMENTS

For our experiments, the off-road vehicle is a drive-by-wire enabled Polaris Ranger XP 900. As a result, the control inputs for throttle, braking, and steering are fully programmable via PC interface. The software interface called PACMOD has been designed with the Robot Operating System (ROS) architecture, allowing for a publish and subscribe mechanism. The vehicle was subjected to open loop commands in order to map steady-state outputs to the commands. The vehicle is shown in figure 4.1.

Figure 4.1: Polaris Ranger with autonomous features

The vehicle was driven on relatively flat paved roads during these experiments to ensure consistency and as a first step towards validating the methodology. Future experiments will be conducted on off-road terrain which will expand the applicability of this approach.

4.1 Longitudinal Dynamics

To model longitudinal dynamics, the vehicle was allowed to reach a specified steady-state velocity and then a throttle or brake command was given. The resulting transient response was used to estimate the vehicle's acceleration for the specified command. In these experiments, the vehicle's initial velocity was varied from 0-25 mph in 5 mph increments. Step responses were obtained for these initial velocities with the throttle command in the 0-70% range and braking in the 0-100 % range. Afterwards, the data was smoothed out as a surface by fitting it with a 3rd order polynomial function. A 3rd order polynomial surface fit was chosen to avoid overfitting and underfitting. Simple evaluations on the sensitivity of the coefficients were done by changing one or several points to different values and evaluating how the polynomial equation varied. While arriving at the best fit, the coefficients for each term were allowed to be unbounded except for constant coefficient. The constant coefficient was set to 0 to reflect the physical reality that the vehicle will not accelerate if both the input and velocity are 0, on flat ground.

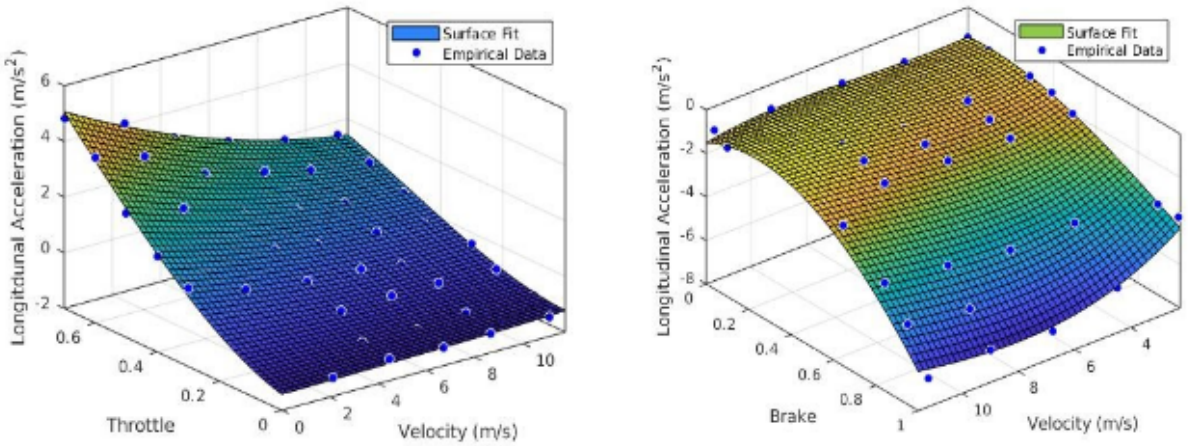

Figure 4.2: Data-driven longitudinal dynamics due to throttle and brake

Figure 4.2 refers to the longitudinal dynamics with throttle and brake separately.

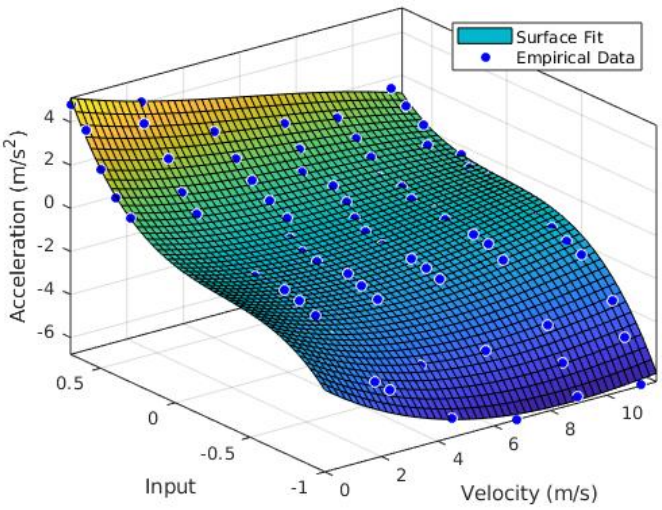

Figure 4.3: Data-driven longitudinal dynamics due to throttle and brake combined as a synthetic input

Figure 4.3 refers to the longitudinal dynamics with throttle and brake combined as a synthetic input.

Equation (4.1) is a 3rd order polynomial that represents the longitudinal dynamics surface fit due to throttle action.

$$\begin{aligned}\hat{f}_{1\alpha}(v_x,\alpha) = \; & -0.4876v_x - 2.385\alpha + 0.0536v_x^2 \\ & + 0.4953v_x\alpha + 19.65\alpha^2 - 0.0018v_x^3 \\ & - 0.0207v_x^2\alpha - 0.8346v_x\alpha^2 - 8.448\alpha^3\end{aligned} \tag{4.1}$$

Equation (4.2) is a 3rd order polynomial that represents the longitudinal dynamics surface fit due to brake action.

$$\begin{aligned}\hat{f}_{1\beta}(v_x,\beta) = \; & -0.7403v_x + 0.8255\beta + 0.1211v_x^2 - 0.2619v_x\beta \\ & - 3.704\beta^2 - 0.0059v_x^3 + 0.0571v_x^2\beta - 0.6992v_x\beta^2 + 1.551\beta^3\end{aligned} \tag{4.2}$$

Equation (4.1) and eq. (4.2) can be combined to obtain the longitudinal dynamics in eq. (2.2) as below:

By defining a synthetic input u as

$$u = \alpha * (\alpha >= 0) - \beta * (\beta > 0) \tag{4.3}$$

The combined dynamic equation can be written as the following polynomial fit:

$$\begin{aligned} \hat{f}_{1u}(x,u) :=& \hat{f}_1(v_x, \alpha, \beta) \\ =& -0.6062 v_x + 1.937 u + 0.08625 v_x^2 + 0.2565 v_x u + 4.283 u^2 \\ & - 0.00339 v_x^3 - 0.03248 v_x^2 u - 0.3862 v_x u^2 + 4.878 u^3 \end{aligned} \tag{4.4}$$

4.2 Lateral Dynamics

Similar experiments were conducted to obtain descriptions for the lateral dynamics. In accordance with the assumptions made earlier, we will restrict the discussion to angular velocity and ignore lateral acceleration data. For each empirical data point, the vehicle was set to a constant forward velocity and steering angle. In this experiment, the steering wheel angle was varied between -450° and 450° in 90° increments, while the forward velocity was varied between 2-10 m/s in 2 m/s increments. However, the tests were limited to a total lateral acceleration magnitude of less than 5.5 m/s^2 for safety reasons.

Figure 4.4 refers to the vehicle lateral dynamics.

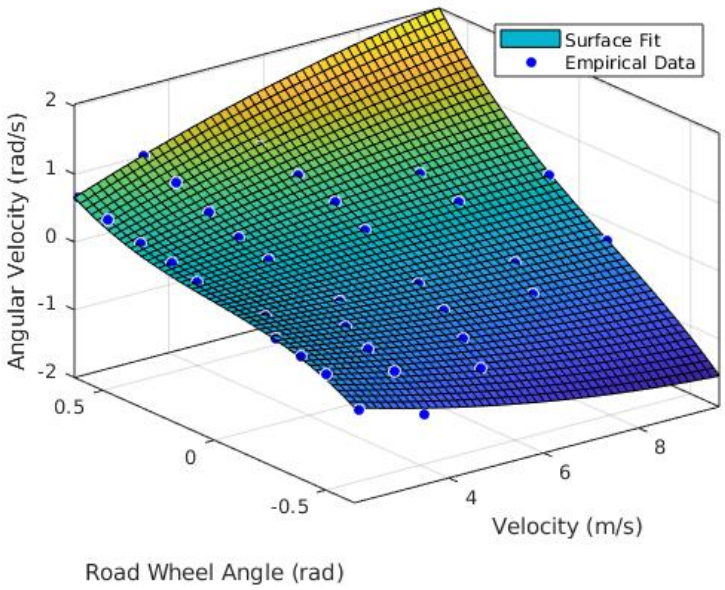

Figure 4.4: Data-driven angular velocity dynamics due to road wheel angle

Equation (4.5) is a 3rd order polynomial that represents the lateral dynamics surface fit due to steering action.

$$\begin{aligned}\hat{f}_3(v_x,\delta) = \dot{\theta} = 0.0264v_x - 0.0067v_x^2 + 0.3911v_x\delta \\ + 0.0013v_x^3 - 0.1934v_x^2\delta \\ + 0.1626v_x\delta^2\end{aligned} \tag{4.5}$$

4.3 Inverse Functions

As discussed in *Core Control Strategy*, the main requirement for sliding mode control is dynamic invertilibity. Thus, inverse mapping functions are found that are also parameterized as polynomial surfaces.

First, the synthetic input will be described as a function of desired acceleration and current velocity. This can be achieved from using the model derived from before. Second, the road wheel angle will be described as a function of current velocity and desired yaw rate. Figure 4.5 illustrates the surface fits which represent the synthetic input and road wheel angle, respectively.

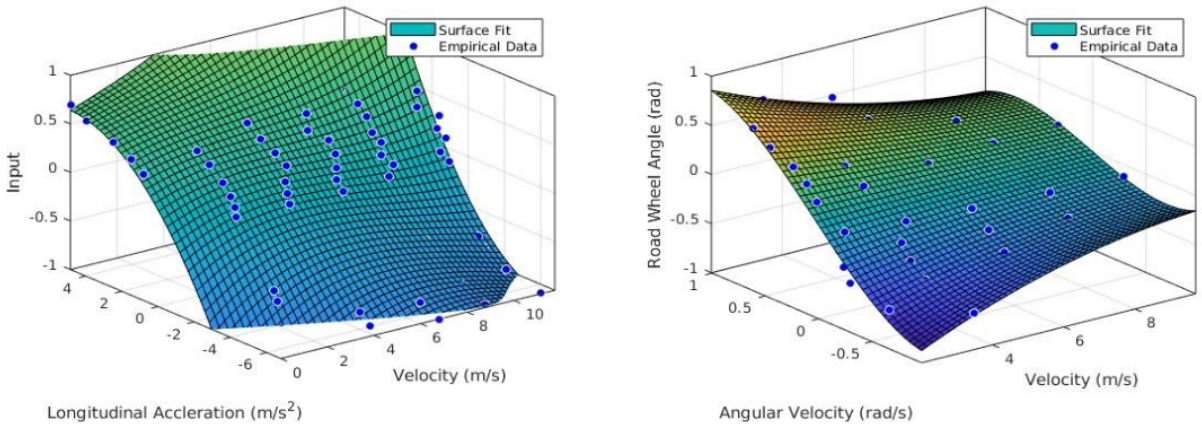

Figure 4.5: Synthetic input mapping from inverse data-driven longitudinal dynamics $\hat{f}_{1u}^{-1}$ and road wheel angle mapping from inverse data-driven angular velocity dynamics $\hat{f}_3^{-1}$

The synthetic input surface fit can be described as the following equation:

$$\begin{aligned}\hat{f}_{1u}^{-1}(v_x, \dot{v}_{xdes}) = u = 0.0997v_x + 0.2586\dot{v}_{xdes} - 0.0164v_x^2 - 0.0260v_x\dot{v}_{xdes} \\ - 0.0260\dot{v}_{xdes}^2 + 0.0009v_x^3 + 0.0034v_x^2\dot{v}_{xdes} + 0.0049v_x\dot{v}_{xdes}^2\end{aligned} \tag{4.6}$$

The road wheel angle input surface fit can be described as the following equation:

$$\begin{aligned}\hat{f}_3^{-1}(v_x, \dot{\theta}_{des}) = \delta = -0.0303v_x + 1.966\dot{\theta} + 0.0077v_x^2 - 0.3524v_x\dot{\theta}_{des} - 0.1374\dot{\theta}_{des}^2 \\ - 0.0005v_x^3 + 0.0211v_x^2\dot{\theta}_{des} + 0.0296v_x\dot{\theta}_{des}^2 - 0.2045\dot{\theta}_{des}^3\end{aligned} \tag{4.7}$$

5. RESULTS AND DISCUSSION

To test the sliding mode controller, the vehicle was driven on two paths consisting of a circular shape and a sinusoidal shape. The circular path meant to test if the controller is tracking the desired path correctly, while the sinusoidal path is meant to test the controller at extremes.

For both simulation and initial implementation experiments, the look-ahead distance was 5 m and the desired velocity was 5 m/s. This was done to keep the simulation and physical experiments consistent with each other.

After initial testing, more implementation experiments were conducted during wet conditions for both on-road and off-road terrain. With adaptive velocity command mentioned in section 3.1 and adaptive look-ahead distance of $v_x + 1$, the vehicle was able to track the path and maintain desired velocity with some error even during extremities.

5.1 Simulation Results

Figure 5.1 presents the simulated data for the circular path.

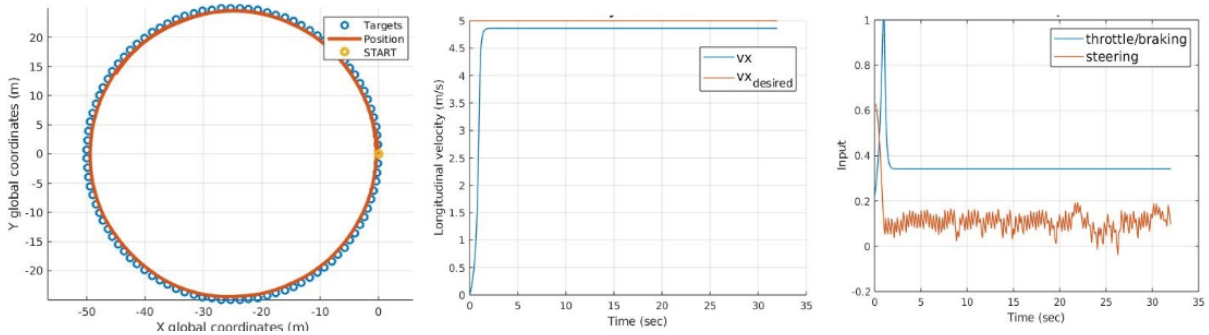

Figure 5.1: Circular path simulated data for path tracking, velocity tracking, and controller input behaviors.

Figure 5.2 presents the simulated data for the sinusoidal path.

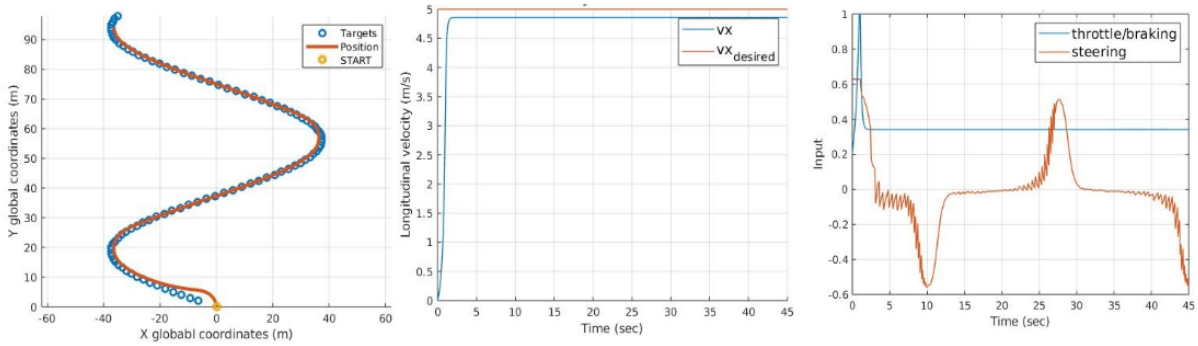

Figure 5.2: Sinusoidal path simulated data for path tracking, velocity tracking, and controller input behaviors.

With both of these simulations to verify the control strategy, the next step is to validate the control strategy through implementation.

5.2 Implementation Results

The implementation of the sliding mode controller on the vehicle yielded similar results to the simulation. The real-world paths attempted to emulate the simulated paths on paved roads. They are not perfectly identical, but the general behaviour of the vehicle can be compared between the results of the simulation and the physical implementation.

Figure 5.3 presents the data collected from real-world implementation into a vehicle for the circular path.

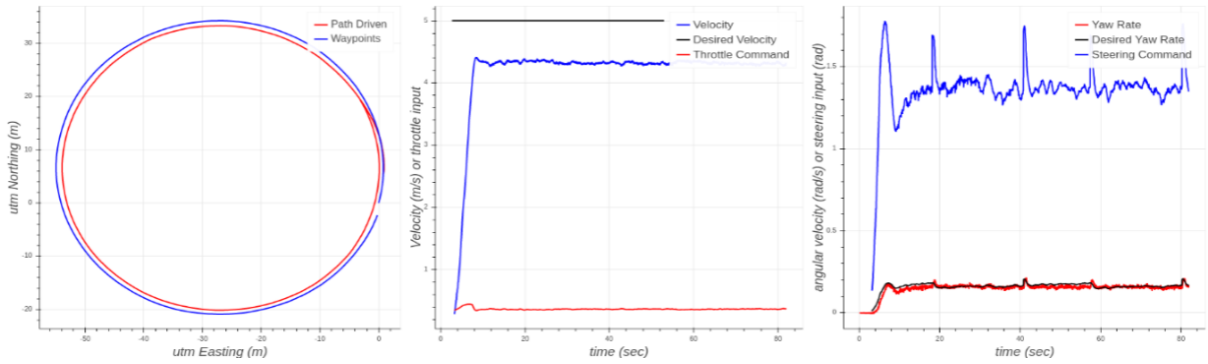

Figure 5.3: Circular path implemented data for path tracking, velocity tracking, and controller input behaviors.

Figure 5.4 presents the data collected from real-world implementation into a vehicle for the sinusoidal path.

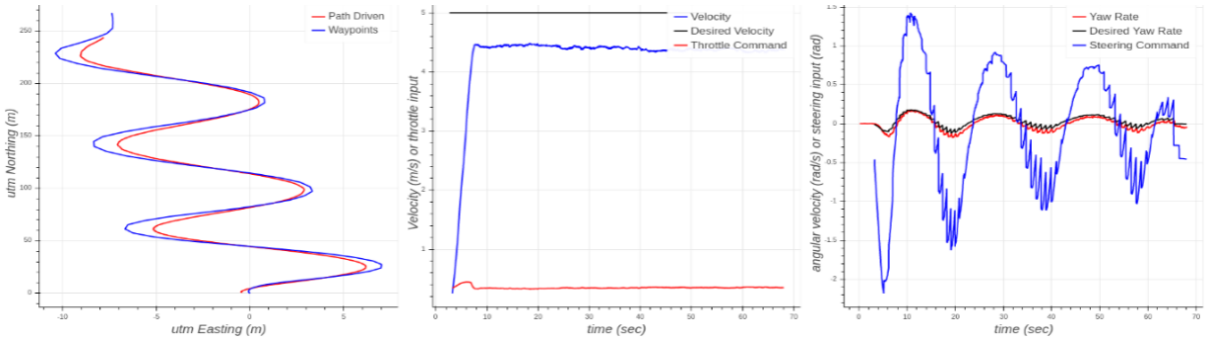

Figure 5.4: Sinusoidal path implemented data for path tracking, velocity tracking, and controller input behaviors.

Moving forward with cases that involve harsher condition, the ground vehicle was subjected to rainy conditions to track the path given and velocities. This includes wet paved roads and muddy off-road terrain. These experiments were to show the controller's robustness. Figure 5.5 shows the terrain's physical features.

Figure 5.5: On-road track (left) and off-road track (right).

Figure 5.6 presents the data collected from real-world implementation into a vehicle on saturated paved roads with both adaptive velocity and adaptive look-ahead algorithms executed.

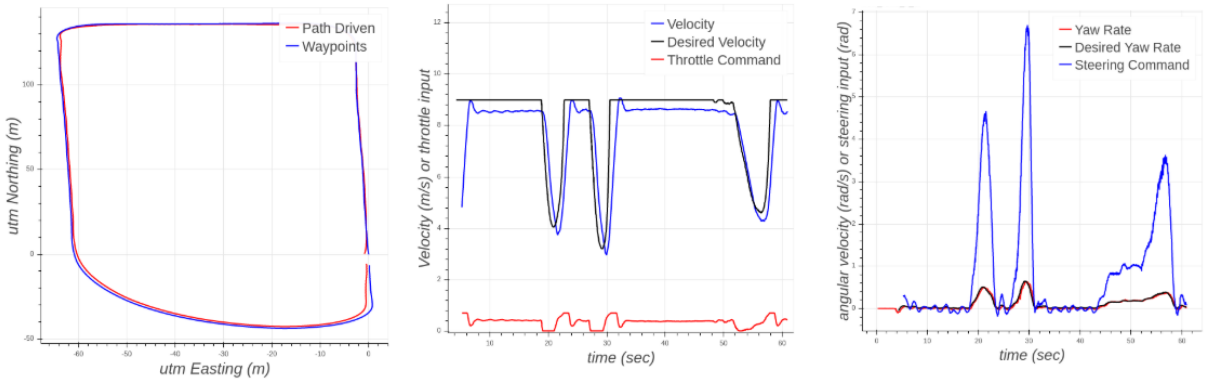

Figure 5.6: Wet conditioned on-road implemented data for path tracking, velocity tracking, and controller input behaviors.

Figure 5.7 presents the data collected from real-world implementation into a vehicle on saturated soil terrain with both adaptive velocity and adaptive look-ahead algorithms executed.

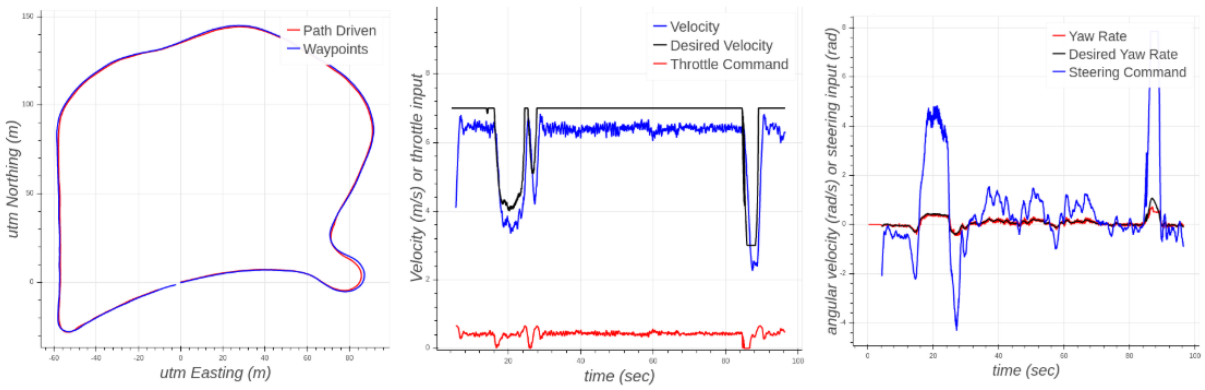

Figure 5.7: Wet conditioned off-road implemented data for path tracking, velocity tracking, and controller input behaviors.

Four experiments were completed for both the on-road and off-road tracks during saturated conditions. Figure 5.8 shows the results for average lateral error and maximum lateral error for each run.

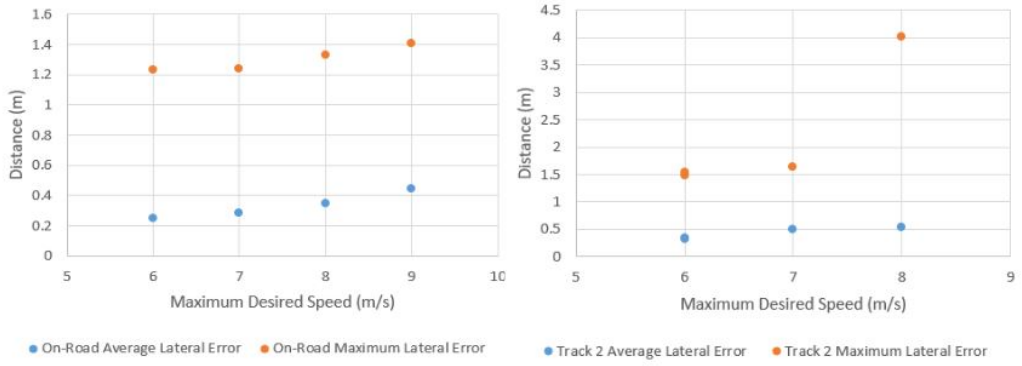

Figure 5.8: On-road lateral error data (left) and off-road lateral error data (right).

With the results presented, data analysis is needed to understand more about the path and

velocity tracking performance.

5.3 Data Analysis

In the initial results, the controller performed reasonably well. In the simulated circular path tracking, the vehicle's center of gravity was slightly inside the waypoints of the circular path. For velocity tracking, there was a steady-state error of 0.139 m/s which is 2.78% off of the desired value. There are some discrepancies regarding the control inputs that happened in the simulation that should not happen in real life. First, the throttle command should not go to 1 immediately from rest or an engine misfire will happen. Second, the chattering of the steering control will cause the riders to feel uncomfortable in the vehicle.

In the simulated sinusoidal path tracking, the vehicle was offset 7 m and the vehicle's initial heading was not aligned with the path. Even when this happened, the vehicle was able to follow the path. The results for the sinusoidal velocity tracking are the same as the circular velocity tracking. The throttle command needs to gradually increase instead of immediately going to 1, which will lead to an engine misfire. The steering command still has a chattering effect which would be uncomfortable for passengers.

For the implemented circular path tracking, the vehicle was slightly inside the circle. This result is very similar to the simulated circular path tracking. For implemented velocity tracking, the steady-state error is 0.62 m/s which is 12.4% off of the desired value. The implemented throttle input gave an instantaneous acceleration while smoothly approaching the desired velocity without any chattering. The implemented steering input has clear chattering.

For the implemented sinusoidal path tracking, the vehicle turned earlier than the path intended. For implemented velocity tracking, the steady-state error is 0.56 m/s which is 11.2% off of the desired value. The implemented throttle and steering input are similar to the results from the circular path.

For the saturated paved road experiment, adaptive velocity and adaptive look-ahead algorithms helped improved performance for path tracking. The steady-state error when approaching 9 m/s is 0.45 m/s which is 5% off of the desired value. The average lateral error ranges from 25 cm to 45

cm, and the maximum lateral error ranges from 1.23 m to 1.41 m.

For the saturated soil terrain experiment, adaptive velocity and adaptive look-ahead algorithms helped improved performance for path tracking. The steady-state error when approaching 7 m/s is 0.48 m/s which is 6.8% off of the desired value. The average lateral error ranges from 30 cm to 55 cm, and the maximum lateral error ranges from 1.49 m to 4.02 m.

6. CONCLUSION

Ultimately, a data-driven model was derived using simple experiments which did not require a large amount of data. This model was split into two categories: longitudinal dynamics and lateral dynamics. The models were surface fitted into 3rd order polynomials which avoided overfitting and underfitting. It should be noted that these models were only calculated and implemented offline in this paper but could be used and updated online in order to improve any discrepancies.

Afterwards, a nonlinear closed-loop control strategy was developed. This control strategy was tested in simulation for a circular path and a sinusoidal path. Then, this strategy was implemented into a real vehicle to follow both a circular path and a sinusoidal path similar to the simulation. To test the robustness of the controller, the implementation was tested in conditions that the model did not reflect.

The main issues with the implementation is that the path tracking cuts corners, velocity tracking approximately has a 0.5 m/s steady-state error for desired velocities explored, and steering input has chattering. This shall be addressed in future work.

Overall, this method displayed simple modeling with data-driven dynamics. The control strategy was developed offline which led to fast computational speed. Initial implementation results can easily be improved with a few more algorithms and gain tuning. This method can easily be applied to off-road terrain.

7. FUTURE WORK

Future work for this thesis includes collecting experimental data in off-road terrain. This way, more aggressive maneuvers will produce side slip and the data from these maneuvers will help produce a more complete lateral dynamic model. Then the same control strategy will be explored for off-road driving.

Once the offline approach is improved, using machine learning for this project will be feasible. Neural networks can be developed to solve for the inverse dynamics [20]. Then the nonlinear control strategy proposed in this paper can be used for velocity and path tracking. This will most likely not be explored during my thesis, but this could be a future direction for this project in order to reduce inaccuracies in the inverse dynamic model while driving.

REFERENCES

[1] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, “Kinematic and dynamic vehicle models for autonomous driving control design.” *2015 IEEE Intell. Veh. Symp.*, pp. 1094 - 1099, 2015.

[2] Yang, S. M., and J. H. Kim. "Validation of the 6-dof vehicle dynamics model and its related VBA program under the constant radius turn manoeuvre." *International Journal of Automotive Technology* 13.4 (2012): 593-605.

[3] R. K. Douglas, D. P. Malladi, R. H. Chen, D. L. Mingori, and J. L. Speyer, “Fault Detection and Identification for Advanced Vehicle Control Systems,” *IFAC Proc.*, vol. 29, no. 1, pp. 7897–7902, 1996.

[4] Stone, Matthew R., and Michael A. Demetriou. "Modeling and simulation of vehicle ride and handling performance." *Proceedings of the 2000 IEEE International Symposium on Intelligent Control. Held jointly with the 8th IEEE Mediterranean Conference on Control and Automation (Cat. No. 00CH37147)*. IEEE, 2000.

[5] S. B. Lu, S. B. Choi, Y. N. Li, M. S. Seong, and J. S. Han, “Global integrated control of vehicle suspension and chassis key subsystems,” *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.*, vol. 224, no. 4, pp. 423–441, 2010.

[6] U. Kiencke and L. Nielsen, Automotive control systems: For engine, driveline, and vehicle: Second edition. Springer, Germany, 2005.

[7] C.-Y. Liang, R. W. Allen, T. J. Rosenthal, J. P. Chrstos, and P. Nunez, “Tire Modeling for Off-Road Vehicle Simulation,” SAE Tech. Pap. Ser., vol. 1, no. 724, 2010.

[8] K. Guo, D. Lu, S. K. Chen, W. C. Lin, and X. P. Lu, “The UniTire model: A nonlinear and non-steady-state tyre model for vehicle dynamics simulation,” *Vehicle System Dynamics*, vol. 43, no. SUPPL., pp. 341–358, 2005.

[9] S. J. Rutherford and D. J. Cole, “Modelling nonlinear vehicle dynamics with neural networks,” *Int. J. Veh. Des.*, vol. 53, no. 4, p. 260, 2010.

[10] Y. U. Yim and S. Y. Oh, “Modeling of vehicle dynamics from real vehicle measurements using a neural network with two-stage hybrid learning for accurate long-term prediction,” IEEE Trans. Veh. Technol., vol. 53, no. 4, pp. 1076–1084, 2004.

[11] P. Guarneri, G. Rocca, and M. Gobbi, “A Neural-Network-Based Model for the Dynamic Simulation of the Tire/Suspension System While Traversing Road Irregularities,” IEEE Trans. Neural Networks, vol. 19, no. 9, pp. 1549–1563, Sep. 2008.

[12] J. Bastiaan, “Estimation of tyre forces using smart tyre sensors and artificial intelligence,” Int. J. Veh. Des., vol. 76, no. 1/2/3/4, p. 110, 2018.

[13] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo and S. Thrun, "Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing," 2007 American Control Conference, New York, NY, 2007, pp. 2296-2301. doi: 10.1109/ACC.2007.4282788

[14] Yamasaki, Takeshi, et al. "Advanced pure pursuit guidance via sliding mode approach for chase UAV." AIAA Guidance, Navigation, and Control Conference. 2009.

[15] Shima, Tal, Moshe Idan, and Oded M. Golan. "Sliding-mode control for integrated missile autopilot guidance." Journal of Guidance, Control, and Dynamics 29.2 (2006): 250-260.

[16] Kim, Mingu, and Youdan Kim. "Lyapunov-based pursuit guidance law with impact angle constraint." IFAC Proceedings Volumes 47.3 (2014): 2509-2514.

[17] Conlter, R. Craig. "Implementation of the Pure Pursuit Path’hcking Algorithm." Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-92-0 (1992).

[18] Snider, Jarrod M. "Automatic steering methods for autonomous automobile path tracking." Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-09-08 (2009).

[19] Slotine, Jean-Jacques E., and Weiping Li. Applied Nonlinear Control. Prentice Hall, Englewood Cliffs, NJ, 1991.

[20] Pham, D. T., & Oh, S. J. (1999). Identification of plant inverse dynamics using neural networks. Artificial Intelligence in Engineering, 13(3), 309–320. https://doi.org/10.1016/S0954-1810(99)00003-5

[21] Hassen, S., Chour, K., Weaver, A., and Gopalswamy, S., "Nonlinear Control of a Ground Vehicle using Data-Driven Dynamic Models," SAE Technical Paper 2020-01-0171, 2020, doi:10.4271/2020-01-0171.