Electric bike modeled with a Hybrid PID Controller

Team #5: David Nguyen, Theo Bailly-Maitre, Micaela Kapp
CMPE 149/249 Final Project

{All Members}

***Abstract*—In this paper we propose and develop a model of a Cyber Physical System to control a set distance from a vehicle with an electric bike using hybrid dynamics to represent the change in velocity based on the input voltage with respect to time. Using a hybrid proportional-integral-derivative (PID) controller, the electric bike system will transition between different inputs with better reaction time.**

I. INTRODUCTION {Nguyen}

In order to understand the electric bike model, a grasp of a cyber-physical system (CPS) is necessary. A CPS combines physical and computer components to represent a feedback system. The physical component consists of systems that can be found in nature, such as the growth rate of plants with different sunlight and humidity levels or those developed by humans, such as the effect that changing gears has on a velocity of a car. The cyber component consists of systems that involve converting, processing, and controlling information via computational means. An example of a cyber component would be a state machine, PID Controllers, and Discrete-time Algorithms. For the systems, physical model (plant) and cyber model (controller) components, subsystems are needed to interconnect the two main components. These are done with analog-to-digital and digital-to-analog converters.

As described in [4], a cyber physical system is can be represented as:

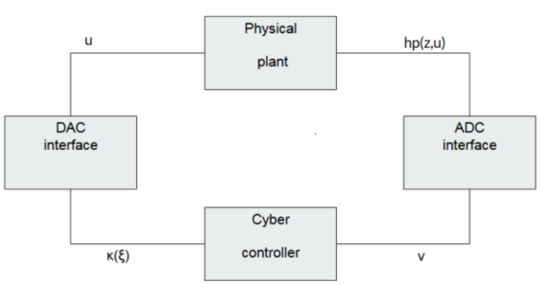

Figure 1: Conceptual diagram of a CPS

A. Example of a CPS

A Cyber-Physical System is used to model the temperature of a room under the effect of a heater that can be turned On or Off.

1) Physical Component

The Physical aspect of the CPS can be shown as:

$$\dot{T} = -aT + T_r + qT\Delta$$

Where:

- $T \in$ R, (temperature of room)
- $a \in$ R (decay rate of temperature)
- $T_r \in$ R (outside room temperature)
- $q \in \{0,1\}$ (state and output of state machine)
- $T_\Delta \in$ R ≥ 0 (heater capacity)

In this model, the temperature of the room is controlled by the heater which is represented by the output of the state machine (q) and the heater capacity (T_Δ).

2) Cyber Component

The cyber aspect of the CPS can be demonstrated for our example can be demonstrated with a finite state machine that keeps the room within a certain temperature range:

A finite-state machine model describing the effect of a heater to the temperature of the room is defined as follows [4]:

- The input alphabet is Σ = R $\geq$ 0and $v \in \Sigma$ is connected to T.
- The finite set of states is $Q = \{0,1\}$, where $q \in Q$ represents that the heater is ON when $q = 1$ and OFF when $q = 0$.
- The set of output symbols is $\Delta = \{0,1\}$ and $\zeta \in \Delta$.
- The output function $\kappa : Q \rightarrow \Delta$ assigns ζ, i.e., $\zeta = \kappa(q)$. Here we define it as the identify, and thus $\zeta = q$.
- The transition function $\delta : Q \times \Sigma \rightarrow Q$ is defined for each $(q,v) \in Q \times \Sigma$

$$(q, v) = \begin{cases} 0 \text{ } if \text{ } q = 1 \text{ } and \text{ } v \geq Tmax \\ 1 \text{ } if \text{ } q = 0 \text{ } and \text{ } v \leq Tmin \end{cases}$$

3) ADC interface

An analog-to-digital converter (ADC), or sampling device is used to provide the cyber component measurements from the physical system. This measures the output v_s from the physical component at a periodic rate T_s^*. A basic model for a sampling device consists of a timer state(T_s) and a sample state ($m_s \in R^{rp}$). When the timer reaches the value of the sampling time $T_s^* \in R \geq 0$, the timer is reset to zero and the sample state is updated with the inputs ($v_s \in R^{rp}$) to the sampling device [4].

The model of this sampling device is listed as

$$\begin{aligned} T_s^* &= 1\,, m_s = 0 \quad when \quad T_s \in [0, T_s^*] \\ T_s^+ &= 0, m_s = v_s \quad when \quad T_s \geq T_s^* \end{aligned} \qquad (1)$$

4) DAC interface

A digital to analog convertor (DAC) converts digital systems into analog equivalent to be used by the physical complement. One of the most common models for a DAC is the zero-order hold model (ZOH). The output of this model is updated at discrete time and held constant in between updates at the next sampling time ($T_h^+ \in R \geq 0$). When $T_h \geq T_h^*$, the timer state is reset to zero and the sample state ($m_h \in R^{rc}$) is updated with inputs ($v_h \in R^{rc}$) of the DAC [4].

The model of this device is

$$\begin{aligned} T_h^* &= 1\,, \dot{m}_h = 0 \quad when \quad T_h \in [0, T_h^*] \\ T_h^+ &= 0, \dot{m}_h = v_h \quad when \quad T_h \geq T_h^* \end{aligned} \qquad (2)$$

With this explanation of a completed cyber-physical system for the effect of a heater on the temperature of a room, a CPS of an electric bike can be understood if physical and cyber components were derived or provided.

II. MODELING OF AN ELECTRIC BIKE {Bailly-Maitre }

A. *First order model*

The bike that is the subject of this study can be considered as an energy converter. The onboard battery feeds a motor which transforms electrical energy into mechanical energy. First it is a rotation of the shaft and the driving wheel, then is become a translation of the whole bike. In the case where the bike is following a moving car, the distance between the bike and the car will be impact by the voltage inputting in the motor. It is this link that the following model is describing.

The input of the model is a voltage apply to the motor, the output is the distance between the bike and the car. The velocity of the car is a non-deterministic parameter that impacts the output. Therefore, it can be considering as a perturbation of the system.

The following list define all the parameter used in the model:

- U and u: voltage input
- Vbike: velocity of the bike
- Vcar: velocity of the car
- Dcar: distance traveled by the bike
- Dbike: distance traveled by the car
- Dr and dr: distance between the car and the bike

The strategy used to model the bike is empirical therefore, it include already all perturbation and energy dissipation. The experiment shows that the correlation between the input voltage (U) and the velocity (V) of the bike is modeled with a first order transfer function.

$$\frac{V_{bike}(s)}{U(s)} = \frac{K}{1+\tau s} \qquad (3)$$

The coefficient K is the gain of the system and $\boldsymbol{\tau}$ is the time constant. From experience, when starting from 0, the bike reaches a velocity maximum of 15.6 meters per second in 6 seconds when the max voltage of 48 volts is inputting. These data provide the numerical value of the coefficient.

$$\tau = \frac{t_r}{\ln(9)} = \frac{6}{3.3} = 2.73 \qquad (4)$$

$$K = \frac{V_{max}}{U_{max}} = \frac{15.6}{48} = 0.325 \qquad (5)$$

B. *Concept of relative distance*

The velocity of the bike is sufficient to calculate the traveled distance with an integrator. However, for this application the model should outputting the relative distance between the bike and the car.

(

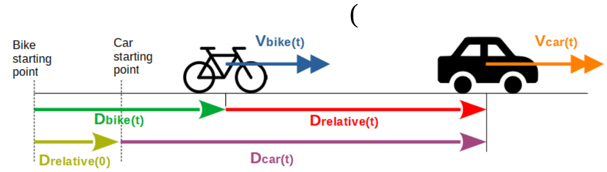

Figure 2: Conceptual diagram of bike model

The above figure shows that the relative distance is define as the initial relative distance added to the traveled distance from the car minus the traveled distance from the bike.

$$D_r(t) = D_r(0) + D_{car}(t) - D_{bike}(t) \qquad (6)$$

Moreover, the traveled distance from a vehicle is the derivative of its speed for both vehicles, is given by

$$D_{car} = \int_0^t V_{car}(x)dx \quad (7)$$

$$D_{bike} = \int_0^t V_{bike}(x)dx \quad (8)$$

Next, the expression of the relative distance is

$$D_r(t) = Dr(0) + \int_0^t V_{car}(x) - V_{bike}(x)dx \quad (9)$$

It is used to measure the distance between the electric bike and the vehicle.

C. *Simulation of the model*

This equation can be implemented in Simulink with a summation and an integrator. After adding the first order transfer function, the complete model of the bike can be simulating with Simulink using the following block diagram.

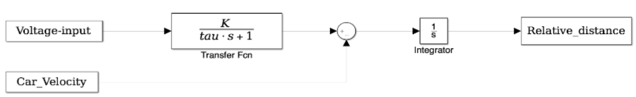

Figure 3: Simulink block diagram of the physical model

In order to get the state model of the system, the inverse transformation of Laplace have to be apply to the complete transfer function of the bike. To get the equation without perturbation, the car velocity is considering as null in a first time.

$$\frac{D_r(s)}{U(s)} = \frac{K}{s(1+\tau s)} \quad (10)$$

The result is a second order differential equation linking the input voltage (U) to the first and second derivative of the relative distance (*dr*).

$$u(t) = \frac{1}{K} \times \dot{d_r}(t) + \frac{\tau}{K}\ddot{d_r}(t) \quad (11)$$

Therefore, two state variables (Z1 and Z2) are necessary to model the physical behavior of the bike. A good choice is to take the relative distance and its derivative.

This concludes the model of the bike. The block diagram can be used for direct continuous simulation in Simulink. The state space model is given by

$$\dot{Z} = \begin{bmatrix}\ddot{dr}\\ \dot{dr}\end{bmatrix} = \begin{bmatrix} u(t)\frac{K}{\tau} - Z_1\frac{1}{\tau} \\ Z_1\end{bmatrix} = \begin{bmatrix}-\frac{1}{\tau} & 0\\ 1 & 0\end{bmatrix}\begin{bmatrix}Z_1\\ Z_2\end{bmatrix} + \begin{bmatrix}\frac{k}{\tau}\\ 0\end{bmatrix}u \quad (12)$$

The state variables and their derivative can be used for simulation using the Hybrid equation toolbox [1].

III. CPS OF BIKE WITH SIMPLE STATE MACHINE FOR CONTROLLER {Nguyen}

A. *Physical Component*

1) The physical aspect of the CPS is represented from the following equation (11), the following are key components to the bike model:

- $Z \in$ R (output state of model)
- $K \in$ R ≥ 0 (gain of system)
- $\boldsymbol{\tau} \in$ R ≥ 0 (time constant)
- $q \in \{0,1\}$, (state of controller)
- $Z_1 \in$ R (Relative Velocity)
- $Z_2 \in$ R (Relative Position)

Having derived the physical model of the bike in the section above, a cyber component and interfaces are needed for a complete CPS.

B. *Cyber Component*

2) The cyber aspect of the CPS can be demonstrated with a finite state machine model describing the effect of a motor given an input voltage in an electrical bike system.

A control strategy for this component is to turn the motor on if the relative distance ($dr = Z_2$) is less than the set distance (D) and to turn the motor off if *dr* is greater than D. This state machine to follow the control strategy is defined as follows:

- The input alphabet is Σ = R ≥ 0 and $v \in \Sigma$ is connected to *dr*. *dr* is set to Z_2 from equation (11).
- The finite set of states is $Q = \{0,1\}$, where $q \in Q$ represents that the motor is ON when $q = 1$ and OFF when $q = 0$.
- The set of output symbols is $\Delta = \{0,48\}$ and $\zeta \in \Delta$.
- The output function $\kappa : Q \rightarrow \Delta$ assigns ζ, i.e., $\zeta = \kappa(q)$

$$\kappa(q) = \begin{cases} 48 \; if \; q = 1 \\ 0 \; if \; q = 0 \end{cases} \quad (13)$$

- The transition function $\delta : Q \times \Sigma \rightarrow Q$ is defined for each $(q,v) \in Q \times \Sigma$.
  (D is the set distance the bike keeps away from the vehicle set by the user)

$$(q, v) = \begin{cases} 0 \; if \; q = 1 \; when \; dr \leq D \\ 1 \; if \; q = 0 \; when \; dr \geq D \end{cases} \quad (14)$$

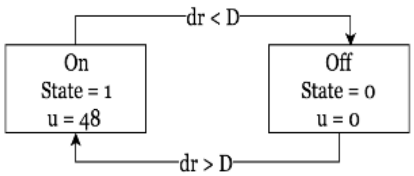

Figure 4: Representation of the simple state machine with appropriate variables

3) Similar ADC and DAC interfaces are used as from the example of tracking the temperature of the room under effect of a heater.

For the ADC, the input is dr (relative distance of bike compared to vehicle) from the physical model and outputs the result at intermittent time. The DAC input is the output voltage (u) of the state machine and outputs the result at intermittent time for the physical model.

C. *Simulations*

Using Hybrid Equation toolbox from MATLAB, there are 3 cases that are simulated where *K*, τ, starting distance, and D are set at constants [1].

- *K* and τ are set based on the equation (5) and (6)
- The starting distance is set to 100m, meaning the bike starts 100m behind the vehicle.
- D is also set to 80m meaning the bike is aiming to stay 80m behind the vehicle.

Velocity of the car is the variable that can be modified to test different cases

1). The first case is when the vehicle is at a velocity of 0m/s or is stopped.

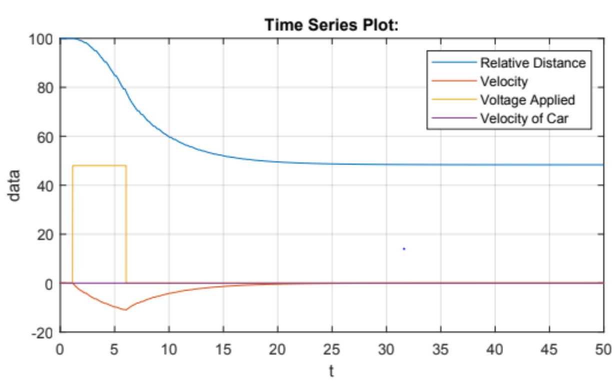

Figure 5: Bike with stopped vehicle

It is obvious the bike turns on the voltage input to 48v until the relative distance is under 80m which then turns the motor off. The bike does not converge to 80m because the bike

2). The keeps momentum while slowing down as fast as it can slow down. second case is when the vehicle has a constant velocity of 5m/s.

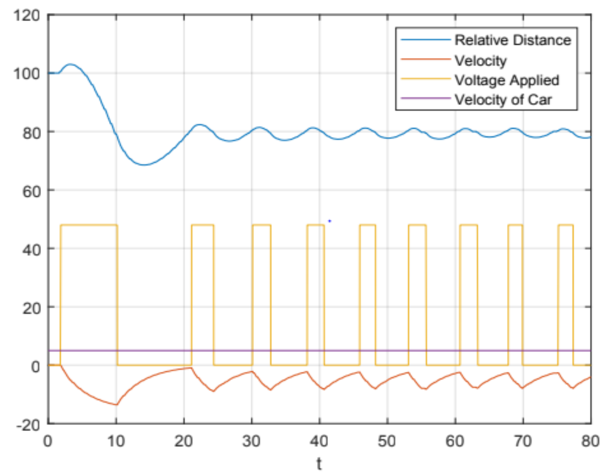

Figure 6: Bike with vehicle moving with constant velocity

Figure 6 shows the relative distance starts to grow because the vehicle is moving while the bike must speed up to reduce the difference in distance. Once a relative distance of 80m is reached, the bike turns off the motor but then overshoots the set D because the bike is now moving faster than the car. The relative distance then stays around 80m and the velocity of the bike stays around 5m/s but very variable due to the nature of the state machine.

3). The third case is a bike with changing velocity.

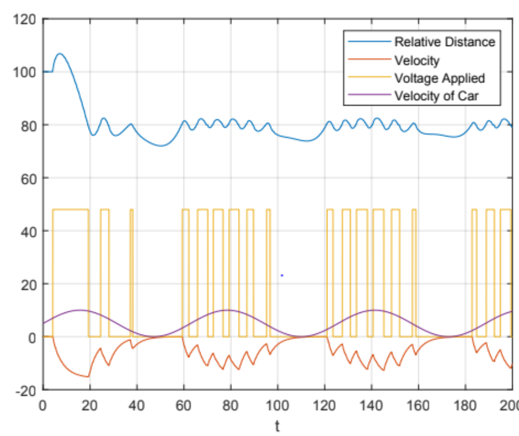

Figure 7: Bike with vehicle moving with sine wave velocity

In Figure 7, the velocity of the car is a sine wave with magnitude and bias of 5 meaning the velocity ranges from 0m/s – 10m/s. The beginning of the graph shows the same behavior with the bike speeding up and overshooting the distance D. As the vehicle slows down, the bike attempts to match the velocity of the car but is not as accurate because the simple state machine. The bike then follows the same pattern trying to keep the same distance of 80m and turns the motor on and off accordingly.

These simulations show that the control stagey implemented with simple state machine has room for improvement. In particular, there is notable amount of overshoot when switching in between states due to the motor only having 2 states. This problem can be corrected with a PID controller which calculates the overshoot into the system

IV. PID CONTROLLER THEORY {Kapp}

A proportional-integral-derivative, or PID, controller is mechanism that uses feedback control loop to continuously monitor a system process. In control engineering it is among the most accurate and precise type of controller that is used in many industries from automobile and manufacturing to industrial and robotics. The PID controller equation for a continuous time system where $t \geq 0$, is given by

$$u(t) = K_P e(t) + K_I \int_0^t e(s)ds + K_D \dot{e}(t), \quad (15)$$

where u is system input, K_P, K_I, and K_D are the proportional, integral, and derivative gains (or parameters), and e represents the error between the state and the set point. The gains are the components to be designed to a reach desired system response. Some of the most common design parameters include settling time, rise time and overshoot of the system response. The settling time of a second order system, t_s, is described as "the time required to a system to reach and stay within approximately 2% of the final value and outlined by

$$t_s = \frac{4}{\zeta \omega_n}, \quad (16)$$

Where ζ is the damping ration, and ω_n represents the natural frequency. The rise time of a second order system given by

$$t_r = \frac{\pi}{2\omega_n}, \quad (17)$$

and representative of the time required for the system to go from approximately 10% to 90% of its final value. The steady state error can be derived from the open or closed loop transfer function with unity feedback by the Final Value Theorem, defined by

$$e(\infty) = lim_{s \to 0} sE(s) = lim_{s \to 0} \frac{sR(s)}{1+G(s)}. \quad (18)$$

In (4), $E(s)$ represent the error of the system, $R(s)$ is the system input, and $G(s)$ is the transfer function.

A. *PID Example*

A common example of a system improved by a PID controller is a mass-spring-damper system given in Figure 8.

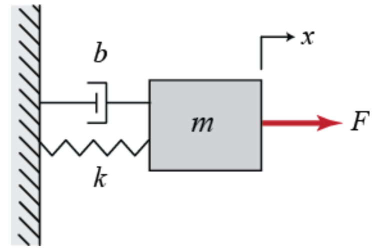

Figure 8: Mass-spring-damper system diagram

The physical, or governing, equation is given in (15),

$$m\ddot{x} + b\dot{x} + kx = F, \quad (19)$$

is a second order differential equation with b representing the damping coefficient, m the block mass, and k depicting the spring coefficient. The Laplace Transform of (15) is given by (16),

$$ms^2X(s) + bsX(s) + kX(s) = F(s), \quad (20)$$

and the transfer function between the input force and output displacement outlined in (4),

$$\frac{X(s)}{F(s)} = \frac{1}{ms^2+bs+k} \quad (21)$$

From this point, the contents or known variables can be entered and the result can be simulated. Figure 9 illustrates the response of the mass-spring-damper system when the input is a step response, mass of the block is 1 kg, the damper coefficient is 10 $\frac{N*s}{m}$, and the spring coefficient is 15 $\frac{N*s}{m}$.

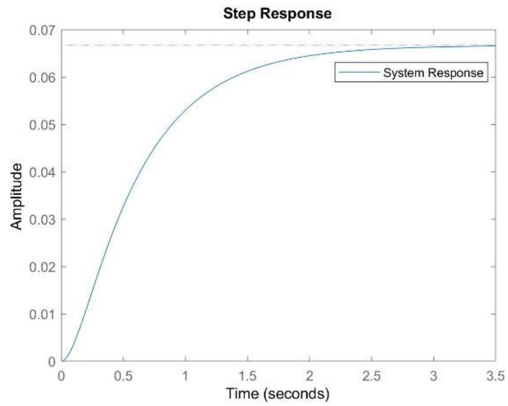

Figure 9: Step Response of spring-mass-damper system

For a step input, we desire the response to reach 1. However, 0.07 is the final value which corresponds to a steady-state error of 0.93, an absurdly large amount of error. Additionally, the rise time is about 1.5 seconds, and the settling time is about 2.5 seconds. By designing a PID controller for this system, the rise time and settling time can be reduced, and the steady state error eliminated. Figure 10 shows the improved system response with an added PID controller.

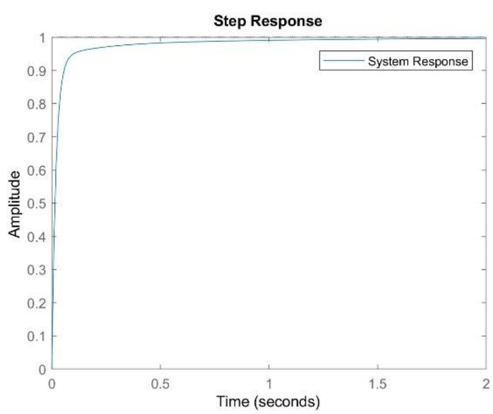

Figure 10: Step Response of spring-mass-damper system with PID controller

The above figure illustrates there is zero steady state error as the system reaches the desired value of 1. The rise time has been reduced to 0.01 seconds, and the settling time has also been reduced to approximately 0.5 seconds.

V. ELECTRIC BIKE MODEL WITH PID CONTROLLER {Kapp}

PID controllers, are famously used for controlling systems with great accuracy and flexibility for continuous time models. For the given electric bike model, the aim for adding a PID controller is to reduce the time taken for the bike to react and allow for smoother transitions while maintaining a set distance away from a vehicle. However, real world situations do not always allow for ideal model situations, thus the addition of a hybrid PID controller will allow this bike model to become more adaptable.

One example for the inclusion of a hybrid PID with friction from the ground causing the proximity sensor to fire in different directions causing unreliable data to be feedback into the system. Other examples include the event that the sensors become blocked by or if the target is much smaller at a given distance than another (think of a large truck carrying a pipe that protrudes the length of the truck bed making the vehicle closer). In any case, the addition would allow the system to continue in a controlled manner in the event the sensors are unable to record reliable data on a continuous interval.

A. *Notation*

Using the electric bike physical model, which is a linear time-invariant system with state $z \in \mathbb{R}^n$ and input $u \in \mathbb{R}^m$ the input is represented by

$$\dot{z} = Az + Bu, \tag{22}$$

where A and B matrices are characterized as,

$$A = \begin{bmatrix} \frac{-1}{\tau} & 0 \\ 1 & 0 \end{bmatrix} \text{ and } B = \begin{bmatrix} \frac{k}{\tau} \\ 0 \end{bmatrix}, \tag{23}$$

the same matrices as listed in (11). The output equation associated with (21) is

$$y = Hz \in \mathbb{R}^p \tag{24}$$

where the H matrix can represent either relative position or relative velocity. Relative velocity matrix is listed as

$$H = [1 \quad 0]. \tag{24}$$

As a reminder, the state given by $z = (z_1, z_2)$ is defined by z_1 = relative velocity and z_2 = relative position. To allow for impulsive measurements of the plant output, a timer state is created satisfying the sequence of times

$$T_1 \leq t_{s+1} - t_s \leq T_2 \quad \forall s \in \mathbb{N}\ \{0\} \tag{26}$$

$$t_{s1} \leq T_2$$

such that T_1 parameter represents the minimum time for samples and T_2 the maximum time between samples. This timer state is denoted at $\tau \in [0, T_2]$ indicating a continuous decrease then a jump when the timer reaches zero to a location between $[T_1, T_2]$. This time is modeled as the hybrid inclusion

$$\begin{cases} \dot{\tau} = -1 & \tau \in [0, T_2] \\ \tau^+ \in [T_1, T_2] & \tau = 0 \end{cases}. \tag{27}$$

Note that (26) is from [2]. The τ is not the time constant from (5).

Then, the input of the plant is denoted as u from (22) which is also the output of the PID controller. This output evolves according to the zero-order hold in which u is held constant until the controller receives a new measurement. Thus, the controller is indicated as

$$\begin{cases} \dot{u} = 0 & \tau \in [0, T_2] \\ u^+ = v_P + v_I + v_D & \tau = 0 \end{cases} \tag{28}$$

where v_P, v_I and v_D are the components of the PID controller. The closed loop system with the plant from (22) and (22), the controller output of (28) and the timer in (27) provides the following for f, C, d, G equations

$$f(x) := \begin{bmatrix} A_f \\ -1 \end{bmatrix}$$

$$\forall x \in C := \mathbb{R}^n \times \mathbb{R}^p \times [0, T_2] \tag{29}$$

$$G(x) := \begin{bmatrix} A_g x_1 \\ [T_1, T_2] \end{bmatrix}$$

$$\forall x \in D := \mathbb{R}^n \times \mathbb{R}^p \times \{0\}.$$

where the matrices A_f and A_g are delivered by

$$A_f = \begin{bmatrix} A & 0 & B & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, A_g = \begin{bmatrix} I & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ -\tilde{K}_P - \tilde{K}_D & -\tilde{K}_I & 0 & 0 \\ H & 0 & 0 & 0 \end{bmatrix} \tag{30}$$

and the design parameters are given by

$$\begin{aligned} \tilde{K}_P &= K_P H - (I + K_D H B)^{-1} K_D H B K_P H \\ \tilde{K}_I &= K_I - (I + K_D H B)^{-1} K_D H B K_I \\ \tilde{K}_D &= (I + K_D H B)^{-1} K_D H A \end{aligned} \tag{31}$$

which are dependent on $K_P,\ K_I,\ and\ K_D$. Additional important relationships include $x = (x_1, x_2)$ with $x_1 = (z, z_I, u, m_s)$ and $x_2 = \tau$ from (27) with z_I being the integral state and m_s the memory state as listed in [2].

Equations (26)-(31) were also used from [2] to model the hybrid PID.

B. PID Design

The design of the hybrid PID was modeled after the design in [2]. The key difference is the model of the continuous time plant from (11). The overall goal was to maintain the set distance and allow the bike to have better transitions between states. Figure 11 show the improved response.

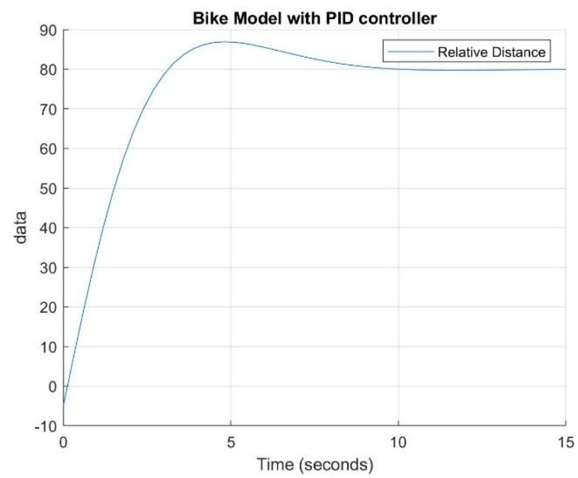

Figure 11: Improved response with PID controller

The PID parameters or gains were found using the PID Tuner from [3] with the closed loop system. The PID gain values are as follows

$$K_P = 4.449,\ K_I = 3.209,\ and\ K_D = -0.580.$$

The improved response significantly decreased the settling time of the system by reaching 80 meters in approximately 50% less time than illustrated in Figures 6 and 7.

C. As Simulation Results

The addition of the hybrid PID proved impactful for all three instances when compared to the original simulations using the state machine model; Figures 5-7. The first instance, illustrated in Figure 12, shows the addition of the hybrid PID controller. As a result, the PID controller decreased the settling time of the system from approximately 17 seconds to 10 seconds.

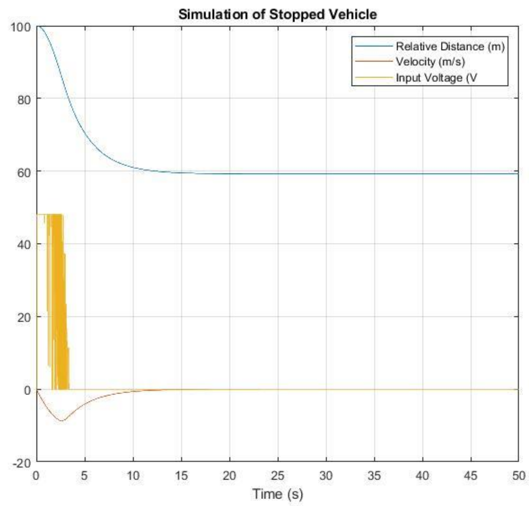

Figure 12: Stopped Vehicle with Hybrid PID controller

However, the bike did not immediately stop. This is because the model does not take other physics concepts, such as gravity and momentum into consideration for simplicity. Furthermore, we can see the input voltage oscillating due the simplicity of the model. The bike motor is either on or off.

The second instance is when the bike is following behind a vehicle moving at a constant velocity. Due to the PID, we expect the bike to reach the desired distance away from the vehicle in a much faster manner as well as maintain that distance in a much cleaner way. Figure 13 shows that this is indeed true. The bike reaches the desired distance within approximately 15 seconds.

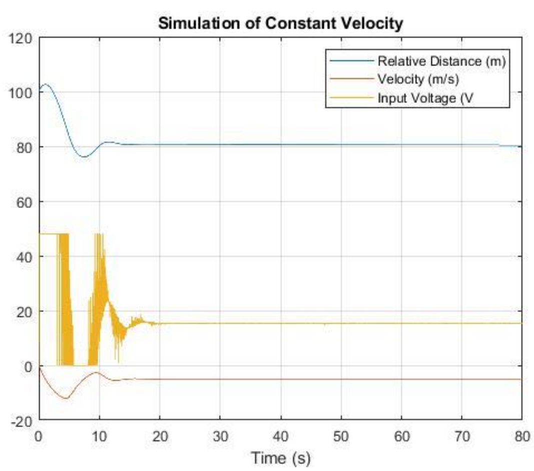

Figure 13: Vehicle model with constant velocity and Hybrid PID controller

Furthermore, we can see while maintaining the desired distance, the oscillations that were formerly present in Figure 6 have smoothed out. The input velocity also shows oscillations as in Figure 12.

The last case is where the bike is following behind a vehicle with changing velocity. The changing velocity in our model, although unrealistic, is an ideal sine wave. Figure 14 shows the same parameters as in Figures 11 and 13 but with the added sine wave in purple.

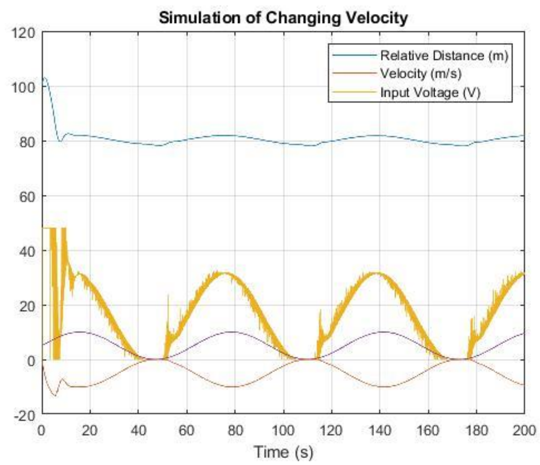

Figure 14: Vehicle model with changing velocity and Hybrid PID controller

Although the bike does not maintain a strict distance of 80 meters, the interval around 80 is much more desirable. The extreme oscillations around the set point have significantly decreased when compared to Figure 7. Last, response time of the system also remains approximately the same value as in Figure 12 which is also significantly decreased comparted to the state machine model of Figure 7. The input velocity appears to oscillate on and off the most in this case compared to the other.

VI. CYBER PHYSICAL SYSTEM PROPERTIES {Bailly-Maitre}

A. *Attractivity*

Attractivity or equivalently, convergence is an asymptotic property or a finite-time property that specifies where executions reaches. In other word it is the convergence or not to a desired point or region.

The goal of the project is to maintain the relative distance constant. An interesting property to study is the attractivity of the system. For example, if the goal distance is 80 m, the bike has to match the speed of the car thus the relative velocity should converge to 0. Therefore, the point of study for the physical state vector [relative distance, relative speed] would be {0}x{80}. However, there is another parameter which influence the attractivity, the velocity of the car. If the car goes faster than the max velocity of the bike, the relative distance will increase to infinity. If the car stops, the bike stops as well but with an overshoot. Thus, the system is not attractive to this point of study. However, if we include the velocity of the car in the state vector then we can define a set for which the system is attractive.

$$z = \begin{bmatrix} d_r \\ \dot{d}_r \\ V_{car} \end{bmatrix} \qquad (32)$$

$$set = \{SetDistance\} \times \{0\} \times]0; V_{max}[\qquad (33)$$

The point {0}x{80}x{0} could be include in the set if the bike could go backward. Indeed, when the relative distance is below the set distance, the error is negative then the PID create a negative command but the input voltage of the motor cannot be negative. Thus, the real voltage stays null and instead of going backward to converge to the set distance, the bike stop. This means than the convergence from relative distances above the set distance is enabled by the bike forward movement but he convergence from relative distances below the set distance is enabled by the car forward movement. This is the reason for the point with Vcar = 0 is excluded from the set of attractivity.

VII. FUTURE WORK {All Members}

Future improvements for the physical bike model include incorporating more traditional concepts such as, breaking and additional motor stages. Additionally, physics concepts such as momentum and the bike on unlevel ground can be looked at. This will make the model more practical and complete.

Hybrid PID controller future work includes refining the sampling interval and redesigning the PID gains with the improved plant model. The possibility of scheduling gains according to the motor stage is also of interest.

VIII. CONCLUSION {All Members}

This paper demonstrated a model of an electric bike that stays a set distance away from a vehicle with a Hybrid PID controller as a cyber-physical system. This system first modeled with a simple state machine that only had 2 states based on the relative distance.

The next evolution was the implementation of a Hybrid PID to smooth out system transitions and sudden changes to velocity by accounting for error which is represented as the difference between the relative distance and set distance. As a result, the system did indeed react in a more controlled manner. The settling time was decreased and the transitions between maintaining velocity and distance were smoother in comparison to the state machine model when plotted. The purpose of implementing a hybrid PID was to simulate instances where the proximity sensors would not be able to continuously deliver reliable data.

An additional note about this bike model is that if the bike were following the vehicle at max speed, a minimum set distance of greater than 300m would be necessary in order for the vehicle to not crash into the car if the car velocity were to suddenly reach 0m/s i.e. a crash. This is because the system does not account for user breaking which will increase the response time of decreasing velocity.

References

[1] R.G. Sanfelice, Hybrid Equations (HyEq) Toolbox, Matlab Toolbox.

[2] D. Lavell, S. Phillips, and R. G. Sanfelice "A Hybrid PID Design for Asymptotic Stabilization with Intermittent Measurements", *Proceedings of the 2018 IEEE Conference on Decision and Control*, pp. 737-742, 12/2018.

[3] MathWorks, (2012). *Control Systems Toolbox: PID Tuner (R2017b).*

[4] R.G. Sanfelice, CMPE149/249 Introduction to Cyber-Physical Systems, 2019 R. G. SANFELICE, 02 February, 2019