2012 NDIA GROUND VEHICLE SYSTEMS ENGINEERING AND TECHNOLOGY SYMPOSIUM
MODELING & SIMULATION, TESTING AND VALIDATION (MSTV) MINI-SYMPOSIUM
AUGUST 14-16, MICHIGAN

SOFTWARE-BASED ELECTRONIC STABILITY CONTROLLER FOR TACTICAL WHEELED VEHICLES

Marcus Mazza
Matthew Rhoads
Peilin Song, PhD
Army Materiel Systems
Analysis Activity
Aberdeen Proving Ground, MD

Christopher Way
Mechanical Simulation Int.
Yorktown, VA

Ross Brown
Motile Robotics, Inc.
Joppa, MD

Mark Reiter
ManTech International Corp.
Belcamp, MD

ABSTRACT

Electronic Stability Control (ESC) is a wheeled vehicle safety system through which active yaw and roll control are realized. ESC is quickly making its way onto tactical wheeled vehicles (TWV) and a source to independently analyze the performance of such systems is becoming increasingly necessary. This paper outlines the efforts of the Army Materiel Systems Analysis Activity (AMSAA) and the Army Research Laboratory (ARL) in the development of a modeling tool capable of replicating typical ESC control systems and algorithms. Created within the MATLAB®/Simulink® environment, the fully-adaptable and reconfigurable tool features corrective yaw control via differential braking and throttle management, along with a detailed braking sub-model capable of capturing brake actuation dynamics. A multi-degree of freedom TruckSim® vehicle model is utilized in conjunction with the tool to accurately model the complexity state and replicate the onboard sensor signals typically available to an ESC controller.

INTRODUCTION

Electronic Stability Control (ESC) is a safety system that offers active yaw and roll control through differential braking and throttle management. A well established, commercially available technology, ESC has proven to be substantially effective in mitigating the loss of vehicle control in the passenger vehicle sector. The benefits of such systems in improving vehicle stability have been documented extensively in OEM technical documents [1]. Having reached this level of maturation, ESC is quickly making its way onto over-the road and tactical wheeled vehicles (TWVs) alike. While over-the-road, commercial truck ESC systems are the foundation for tactical vehicle ESC systems; many issues remain in their adaptation.

A fundamental issue with integration of ESC is that many of the solutions currently available exist in a "black box" format for the end customer; and often times, also for the end supplier. Due to the proprietary nature of the software employed, individuals adapting these systems to TWVs have traditionally had little insight into the algorithms and control strategies used within the controller beyond the tuning parameters "allowed" by a given system. As such, proper integration of ESC remains difficult without a more fundamental understanding of the underlying control strategies and the ability to independently test, model, and simulate these systems.

In order to fill this gap, engineers with the Army Materiel Systems Analysis Activity (AMSAA) and the Army Research Laboratory (ARL) have developed a fully functional ESC controller within the MATLAB/Simulink [2] software environment. The software emulation is capable of replicating typical ESC control strategies used by industry, allowing the controller to parallel a given commercial controller in order to assist in the tuning and or testing and evaluation thereof. It can also be used to optimize the overall control strategy for a given vehicle; which may lead to larger improvements than those available through tuning alone.

This paper outlines the modular layout of the ESC controller, with attention given to each level of the control scheme as well as a discussion of the integration between Simulink and TruckSim with respect to parameter formatting. Vehicle state parameters, provided by the multi-degree of freedom TruckSim model, are traced through the system as controller functionality is discussed. Each of the main subsystem figures depicting controller functionality is accompanied by a brief description of the specific purpose of the block.

ESC MODEL ARCHITECTURE

The ESC controller was created in the MATLAB/Simulink simulation environment. Figure 1 summarizes the controller's top level in conceptual flow chart form. Figure 2 depicts the actual Simulink block diagram of the ESC controller, consisting of multiple blocks which perform the major functions of the stability control process. Many of the blocks visible at this level contain subsystems to streamline the controller functions and ensure signal traceability.

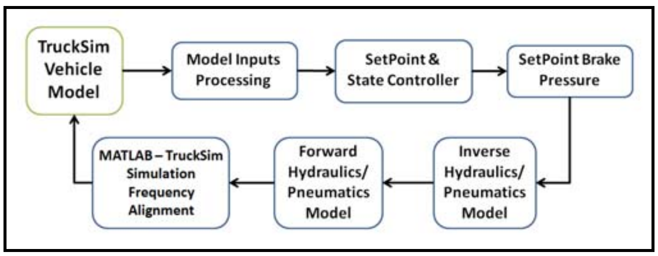

Figure 1: Conceptual Flow Chart of ESC Controller.

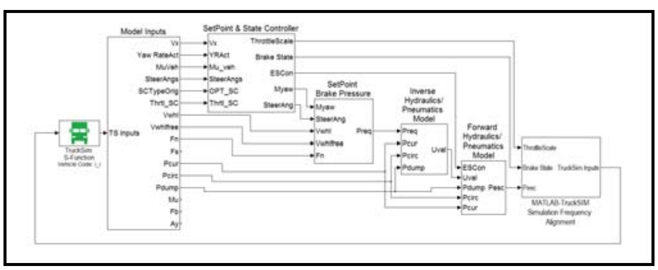

Figure 2: Top Level ESC Controller – Simulink Diagram.

Typically an ESC controller requires vehicle sensor input to supply data defining the vehicle's dynamic state during a given maneuver. However in the software environment, a surrogate simulation tool must be used. The vehicle simulation tool selected for this study was TruckSim [3] which integrates with the Simulink model through a TruckSim S-function block.

Multiple channels representing the dynamic state of the TruckSim vehicle at a given simulation time step are sent to the Model Inputs block for organization, filtering, and sampling rate changes. The data channels are then sent to the other main blocks of the ESC controller, including the Setpoint and State Controller, SetPoint Brake Pressure, Inverse Hydraulic/Pneumatic, and Forward Hydraulic/Pneumatic blocks.

The Setpoint and State Controller calculates the yaw moment and throttle adjustments necessary for keeping the vehicle as close as possible to its intended path. Next, the Setpoint Brake Pressure block calculates individual brake pressures within the braking scheme needed to create the desired corrective yaw moment. The Inverse Hydraulic/Pneumatic block calculates how long pressure control valves would have to cycle in order to generate the requested pressure increase or decrease, allowing for a detailed brake system model. Then the Forward Hydraulic/Pneumatic block utilizes the previously estimated brake cycling time, along with a pulse-width modulation strategy, to calculate a current state brake pressure for each wheel cylinder. The brake dynamics modeling accurately emulates actual brake dynamics depicted in SAE professional educational documents (i.e., time lag associated with pressure build and dump) [4]. Finally, the new brake pressures result in brake application in the TruckSim vehicle, causing desired reactions at the tire/road interface which, in conjunction with the adjusted throttle setting, influence the path and speed of the vehicle.

All of the separate subsystems are described in more detail in the following sections.

IMPORTED AND EXPORTED VARIABLES

Imported channels provided *to* the vehicle model include normalized engine throttle position, scale factor for throttle (in open-loop throttle control case), and brake pressure at each wheel location. These signals are the means through which the ESC algorithm controls the speed and yaw trajectories of the TruckSim vehicle during a given maneuver.

Exported channels provided *from* the TruckSim vehicle model include several channels that are typically sensed and recorded on the vehicle as well as some that are calculated by a typical ESC controller. Table 1 contains the output variables of interest, along with a brief description of each variable. Some of these variables could not be obtained directly by an actual ESC control system. For example, TruckSim supplies the road coefficient of friction under each tire at a given time step. This feature is utilized in order to simplify the Simulink implementation of the ESC controller, since the additional complexity of the logic required by an actual ESC system to estimate the road friction in real time is beyond the scope of this study.

Some output channels are obtained as a single value related to the chassis such as vehicle longitudinal speed. Other outputs are obtained for each tire such as the longitudinal speed of each wheel center. These multi-channel outputs are typically grouped as vectors throughout the controller. The convention used on these types of channels is always Axle 1 left (L1), Axle 1 right (R1), Axle 2 left (L2), Axle 2 right (R2), and so on.

Table 1: TruckSim Outputs (Data sent to ESC Controller)

Name	Units	Type	Full Name
Vx_SM	km/h	Longitudinal speed	Long. speed, vehicle SM CG
AVZ	deg/s	Angular rate	Yaw rate (body-fixed), vehicle
AyBf_SM	g's	Lateral acceleration	Ay (body-fixed), vehicle SM CG
Steer_L1	deg	Angle	Wheel steer L1
Steer_R1	deg	Angle	Wheel steer R1
Vx_XX	km/h	Longitudinal speed	Vx (equivalent), at each wheel location
Vx_Wc_XX	km/h	Longitudinal speed	Vx at each wheel center
PbkCh_XX	MPa	Pressure	Brake whl. cylinder pressure at each wheel location
Pbk_Con	MPa	Pressure	Brake control input (M/C)
Fz_XX	N	Force	Wheel XX vertical force
MuX_XX	-	Friction coefficient	Ground friction for tire XX
VxTarget	km/h	Longitudinal speed	Target longitudinal speed
Thrtl_SC	-	Normalized Position	Closed-Loop Speed Control normalized throttle position
OPT_SC	-	Custom	Steering Control Type

(In the above table, XX is representative of L1, L2, R1 or R2)

MODEL INPUTS SUBSYSTEM

The Model Inputs subsystem receives a vector signal from the TruckSim vehicle model and processes and repackages the channels for use within the downstream blocks. Figure 3 shows the contents of the Model Input block. The vector input is passed through an optional low pass filter. The purpose of the filter is to eliminate any numerical noise that may occur in the simulation and create artificial spikes in the input data.

Some of the individual signals listed in Table 1 would be sensed in an actual vehicle where the output rate of the signal would be constrained by the onboard sensors. To include this constraint in the controller, the corresponding channels are run through a zero order hold block that uses the *SensorFreq* variable from the workspace. This resampling can be seen in the green 'SensorRate' blocks towards the top of Figure 3. The channels appear stepped in the simulation, providing a more realistic representation of the actual vehicle state data being measured. The channels that are resampled this way include vehicle forward velocity, yaw rate, lateral acceleration, steering angles, and wheel velocities.

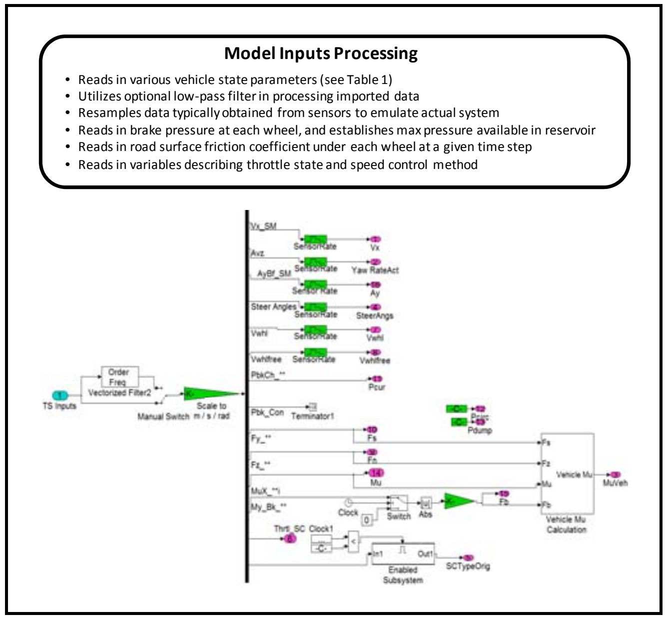

Figure 3: Model Inputs Subsystem.

Continuing down the large demux block, *PbkCh_XX* represents the current wheel cylinder brake pressures. These pressures are controlled by the ESC controller in the active state. *Pbk_Con* represents the current master cylinder pressure from TruckSim. The ESC controller can utilize the maximum pressure made available from the brake system and so *Pbk_Con* is currently terminated at this point and replaced by variable defined in the MATLAB workspace as P_{circ}. P_{circ} is then defined as the maximum pressure available to the ESC system. The current implementation assumes that the driver is not applying the brakes, and that the only braking during the ESC simulation is due to the ESC controller.

The next channels are *Fz_XX*, the normal load under each wheel at a given time step. These tire forces are currently only used in an optional calculation of the road friction coefficient.

TruckSim also reports the road friction coefficient under each tire throughout the simulation. This information is not readily available in an actual vehicle; however the simplified implementation of the ESC controller takes advantage of the available data. As mentioned earlier, the complex algorithms used in an actual system to estimate tire-road friction coefficient are beyond the scope of this modeling tool, and so the TruckSim friction coefficients are used to calculate a single vehicle friction coefficient in the Vehicle Friction Calculation block. The details of this derivation follow in the next section.

The next two inputs are constants that are provided to the ESC controller. *Thrtl_SC* represented the current closed loop speed control throttle position. This variable is used in the ESC Throttle Control block. *OPT_SC* determines the type of speed controller used within the TruckSim model. This information is used in the Throttle Control block to control how the throttle managed when ESC is activated.

VEHICLE FRICTION CALCULATION

A vehicle friction coefficient is required by the Setpoint and State Controller block. The calculation of the friction coefficient is a difficult procedure in an actual vehicle. Scenarios such as split-µ, low-µ, and abrupt surface changes (dry surface to ice) contribute to this challenge. Here TruckSim provides a distinct advantage as it calculates the friction of the tire-road surface interface. Thus the tires of the TruckSim model can export the unique road friction coefficients at each wheel station as the vehicle travels on a normal or split-µ terrain. This is analogous to an actual vehicle controller correctly calculating the road friction 100% of the time.

From the individual coefficient of friction at each wheel station, a single vehicle friction coefficient is calculated via one of the three methods discussed below.

Method 1: Minimum

This option automatically selects the minimum of the coefficient of friction under each of the tires at any given time step, and assigns the minimum as the vehicle overall friction coefficient. This is the most conservative method and would likely not perform as well as an actual ESC controller.

Method 2: Average

This method averages the friction coefficients for all four tires at each time step. This method is less conservative than method 1, but may not represent certain conditions well.

Method 3: Weighted Average

This method takes the coefficient of friction for each wheel and weights them based on the tire normal force. Therefore, the tires with more normal force have greater influence on the vehicle friction coefficient.

SETPOINT & STATE CONTROLLER

The SetPoint & State Controller calculates the difference between the driver's intended path and the vehicle's actual path. A significant deviation between these two quantities results in an intervention from the stability control system. This block focuses on the calculation of driver intent, the comparison of the intended and the actual yaw rates, and the throttle control performed to supplement the later-requested brake application.

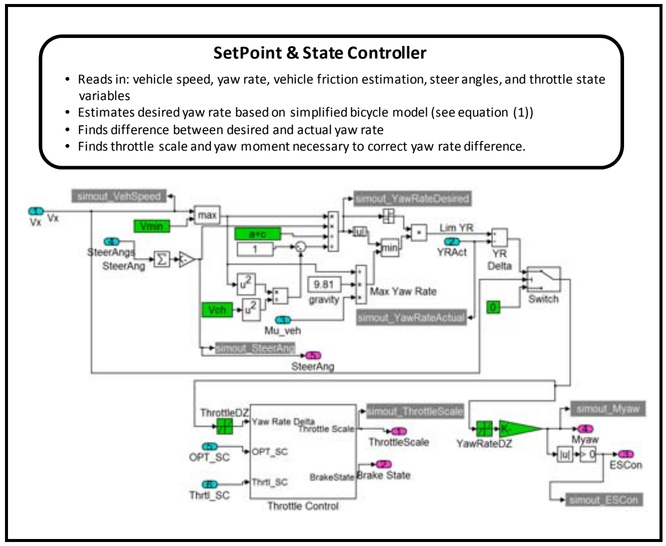

Figure 4: SetPoint & State Controller.

Figure 4 illustrates the details of the SetPoint & State Controller subsystem. The first calculation is the intended path of the driver. A single-track bicycle model is used to calculate the desired yaw rate. This calculation, summarized in equation (1), is a function of the steer angles, vehicle forward velocity, track width, wheel base, and vehicle characteristic velocity [5]. Of these inputs, steer angles and vehicle forward are obtained directly from TruckSim, where the additional inputs are specified as MATLAB workspace variables. Vehicle forward velocity is defined as the velocity between a global fixed axis system and the vehicle's CG in the forward direction of an axis system fixed to the vehicle chassis. The steer angles, not steering wheel angle, of the wheels on the first axle are averaged to obtain a single steer angle for the vehicle. In order to calculate the intended yaw rate, geometric information about the vehicle is required.

$$\frac{d\psi_{No}}{dt} = \frac{v_x \delta_w}{(a+c)\left(1+\frac{v_x}{v_{CH}^2}\right)} \quad (1)$$

Where a is the distance from the front wheels to the vehicle center of gravity, c is the distance from the rear wheels to the vehicle center of gravity, v_x is the vehicle forward velocity, v_{CH} is the characteristic velocity of the vehicle (found via Skid Pad test data or simulation), δ_w is the average front wheel steering angle, and the time derivative of ψ_{No} is the nominal yaw rate.

The desired yaw rate is limited by the friction coefficient of the road surface. Obviously, a high friction surface sustains higher lateral accelerations than a low friction surface. This limit is imposed in the controller using the relationship shown in equation (2) [5]. Vehicle side slip is also considered. The equation states that the max desired yaw rate increases with increasing friction but decreases with increasing forward velocity.

$$\frac{d\psi_{No}}{dt} \leq \mu_{HF}\frac{g}{v_F} \qquad (2)$$

Where the time derivative of ψ_{No} is the nominal yaw rate, μ_{HF} is the coefficient of friction of the road at the current location of the vehicle, g is the acceleration due to gravity, and v_F is the linear velocity of the vehicle.

Figure 5 shows a depiction of the nominal yaw rate limit imposed by the available friction at the tire-road interface [6]. The middle curve represents a scenario in which the surface friction coefficient limits the nominal yaw rate available to the vehicle. ESC intervention allows the vehicle to follow the nominal path to the extent made available by the friction at the tire-road interface before large deviations in vehicle slip angle occur [6].

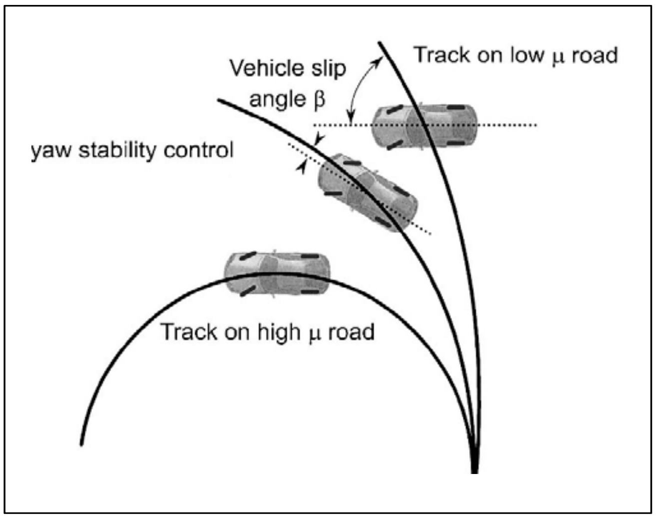

Figure 5: Yaw Stability Control Intervention [6].

While the desired yaw rate is calculated, the actual yaw rate of the vehicle is obtained from TruckSim. This is similar to a yaw rate being measured by a sensor on an actual vehicle. The difference between desired yaw rate and actual yaw rate is calculated. This difference then diverges into two separate dead-zones designed to prevent ESC from actuating during driving situations that would not necessitate ESC.

For example, there is no need to correct the path of the vehicle if yaw rate error signal is very low. Therefore, a dead zone block is used to allow for ESC activation only at a considerable difference in actual and intended yaw rate. The ESC algorithm also has a speed-actuated feature, in that the vehicle speed must be greater than or equal to a specified minimum before the yaw rate difference be passed through to create corrective action.

A second yaw rate difference branch goes through a dead zone that then leads to a throttle control block. This dead zone is set to a lower threshold than the dead zone threshold for the yaw rate difference so as to not command the vehicle to apply brakes while the speed controller commands more throttle to maintain a given target speed.

Once the yaw portion of the ESC event is finished, the application of throttle is restored from the reduced position to the current throttle requested by the TruckSim speed controller. The reapplication of throttle is delayed for a specified duration after the yaw control portion of the ESC event has finished. This delay is meant to mimic the throttle delay seen in a physical vehicle system.

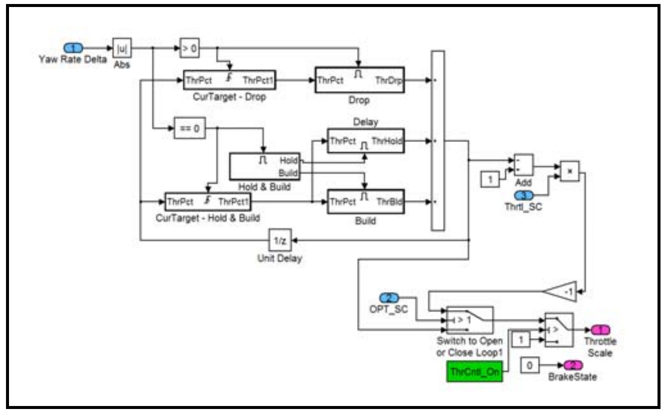

Figure 6: Throttle Control Subsystem.

After the yaw rate difference passes through the throttle dead zone in the SetPoint & State controller, it enters the throttle control subsystem, shown in Figure 6. When the magnitude of the difference is nonzero, a trigger gets the current throttle scale factor, originally initialized to 1, and enables a "Drop" block. ESC throttle correction in this particular model is approximated by a first-order dynamic system, and the rate of decrease or increase in throttle is determined by a tunable time constant.

The "Drop" block remains enabled until the yaw rate difference goes to zero. If given sufficient time, the scale factor decreases to almost zero. Once the yaw rate difference going into the throttle control subsystem reaches zero, a "Hold and Build" timer block is enabled, and a trigger holds the current throttle scale factor. Once the delay time is reached, the "Build" block is enabled. The "Build" block begins building the throttle scale factor from the current value back to 1. This cycle repeats once per given ESC activation.

The scale factor is always multiplied by the TruckSim model specified open loop throttle position.; therefore, when ESC is off, the throttle coming from the ESC controller is (1*TruckSim Internal Throttle). When ESC is activated, the scale factor is decreased in order to model a drop in throttle. The throttle scale factor in the Throttle Control block is constrained as:

$$0 < Throttle\ Scale\ Factor \leq 1.0 \quad (3)$$

TruckSim features both open and closed loop speed control. In open loop control, there is no desired speed defined and only a prescribed throttle position can be requested. In closed loop control, a PI controller is used to maintain a target speed that can be specified as a constant, a target speed versus time, or a target speed versus station. The ESC tool allows for manipulation of both open- and closed-loop throttle strategies such that vehicle speed may be controlled in either case.

SETPOINT BRAKE PRESSURE MODEL

The SetPoint Brake Pressure block calculates the brake pressures required at each wheel in order to create the corrective yaw moment found in the SetPoint and State Controller. This block uses the sign of the requested yaw moment along with the steering angle direction to detect whether the vehicle is in an oversteer or an understeer condition. It then distributes the brake pressure to the appropriate wheels in order to create the requested yaw moment. Figure 7 summarizes the functionality of the SetPoint Brake Pressure sub-model.

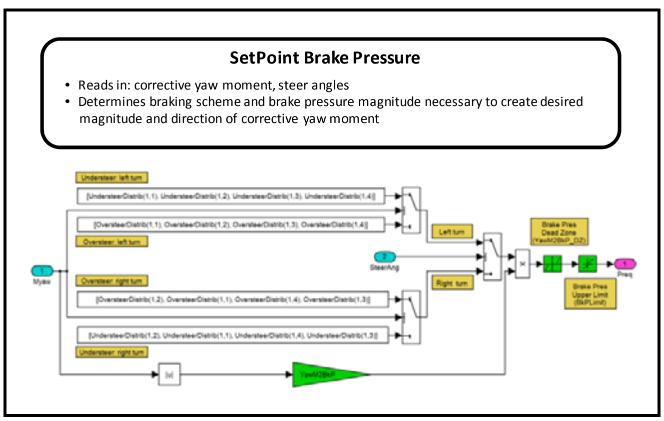

Figure 7: SetPoint Brake Pressure Block.

The inputs to this sub-model block are yaw moment (M_{yaw}) and steer angle (*SteerAng*) while the requested pressures for each wheel brake cylinder are provided as outputs. *SteerAng* is used to detect if the vehicle is turning left or right. The sign of M_{yaw} is then used to detect whether the vehicle is in an understeer or oversteer condition. Based on this logic, a wheel station or wheel stations are selected for brake application.

The order convention throughout the ESC model is that the arrays or vector signals are always (Axle1 Left, Axle1 Right, Axle2 Left, Axle2 Right). However, for this particular part of the model, “inside” and “outside” is used. "Inside" indicates the inside tire in a turn, whereas "outside" indicates the corresponding outside tire.

For oversteer events, the vehicle yaws more than desired and therefore the outside brakes apply the corrective yaw moment. In an understeer scenario, the vehicle is yawing less than the desired rate, therefore the inside brakes apply the correction. The severity to which each wheel station is braked is tunable within the MATLAB code. Furthermore, the overall yaw control strategy that determines which wheel is delivered brake pressure in a given handling scenario is completely modifiable within the Simulink controller and is the subject of further research.

Once the steering direction and the sign of M_{yaw} are evaluated and the brake distribution is selected, the array of the brake distribution is multiplied by a gain. This gain turns the requested yaw moment into a corresponding brake pressure level. The brake pressure level is then multiplied by the distribution array in order to achieve a brake pressure at each wheel. The array is then passed through a dead zone to eliminate very small pressure requests. The requested brake pressure is also limited at maximum pressure by a saturation block, in order to model the limitations of the vehicle’s braking system.

INVERSE HYDRAULICS/PNEUMATICS MODEL

The Inverse Hydraulics/Pneumatics model receives the requested brake pressures from the SetPoint Brake Pressure controller and calculates the valve cycling time response, given the current brake pressures in the vehicle. The subsystem simulates the vehicle brake hydraulics/ pneumatics and the associated control valve characteristics. It takes into account the current brake pressures and the available reservoir pressure to calculate a valve cycle time referred to as U_{val}. The subsystem determines whether pressure needs to build or dump and how long the valve has to stay open in order to achieve the pressure differential.

The Inverse Hydraulics/Pneumatics block is divided into two separate subsystems in order to allow the brake pressure to build and dump at different rates. Figure 8 shows these two subsystems labeled “Build Pressure” and “Dump Pressure”. If ESC is requesting a pressure that is greater than the current pressure, the U_{val} from the Build Pressure block is used. Likewise, if ESC is requesting a pressure less than the current pressure, U_{val} from the “Dump Pressure” block is used. Both the build and dump subsystems were

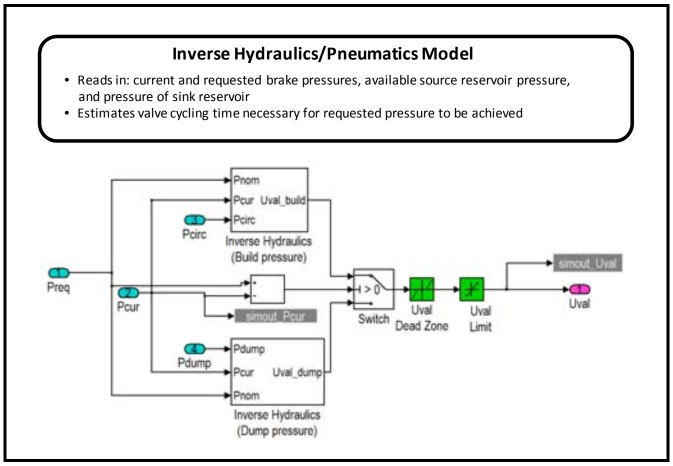

Figure 8: Inverse Hydraulics/Pneumatics Subsystem.

modeled using the Bernoulli theorem for fluid flow and a pressure/volume relationship. Equation (4) and Equation (5) show the calculation used to obtain U_{val} [5].

$$U_{val} = \frac{P_{new} - P_{cur}}{(c_{1build} + c_{2build} P_{cur})\sqrt{|P_{circ} - P_{cur}|}} \quad (4)$$

$$U_{val} = \frac{P_{new} - P_{cur}}{(c_{1dump} + c_{2dump} P_{cur})\sqrt{|P_{dump} - P_{cur}|}} \quad (5)$$

Where U_{val} is the valve cycle time, P_{circ} is the max brake system pressure available, P_{dump} is the pressure in the wheel cylinder in a zero brake condition, P_{new} is the new brake pressure, P_{cur} is the current brake pressure, and c_1 and c_2 are the tunable parameters to control the first order time delay in the brake engagement (for both increasing and decreasing pressure.)

FORWARD HYDRAULICS/PNEUMATICS MODEL

The Forward Hydraulics/Pneumatics model utilizes all of the information regarding valve cycle times as well as the current pressure in each wheel and cycles the valves to raise or lower brake pressure while ESC is activated. The model receives input from the Setpoint and State Controller, Inverse Hydraulic block, and the Model Inputs block, and it outputs wheel cylinder brake pressures.

The forward hydraulics/pneumatics block must be able to handle multiple scenarios. If ESC is active, the forward hydraulics/pneumatics model must be able to build pressure, reduce pressure, or hold pressure for the prescribed time U_{val} found via the inverse hydraulics/pneumatics model. This occurs at each wheel station during the ESC event in a manner consistent with the actual vehicle. If ESC was triggered on and then off, the forward hydraulics/pneumatics model must also ensure that any brake pressure due to the ESC system returns to the zero brake condition. For this reason, this block must utilize strategies to simulate cycles of opening and closing valves, the mechanical and electrical delays due to the valves, and the delays due to fluid flow.

Figure 9 shows the contents of the Forward Hydraulics/Pneumatics block. The subsystem takes valve cycle time, current pressure in the wheel brake cylinder, available system pressure, dump pressure, and a flag to signal the on/off state of the ESC controller. It outputs the individual brake pressures for each wheel as commanded by the ESC system. The values for U_{val} are also sent back to the MATLAB workspace for additional reporting.

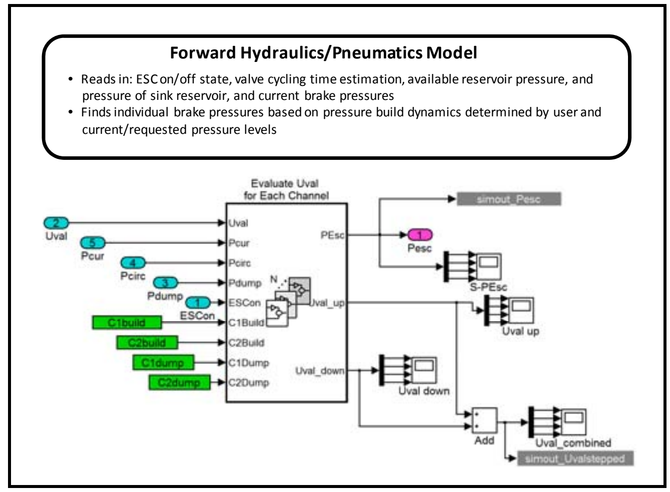

Figure 9: Forward Hydraulics/Pneumatics Subsystem.

The Forward Hydraulics/Pneumatics block requires Simulink-enabled subsystems for each braked wheel to be individually controlled. To accomplish this, the Simulink 'ForEach' block was used. Therefore, the valve times and pressures for each wheel brake cylinder are separated internally in Simulink and sent into the 'ForEach' block. The results of this block are automatically regrouped much like a bus or mux signal. Figure 9 shows the layout of the 'ForEach' block.

Starting from the left, U_{val} is split and sent into two pulse width modulation (PWM) algorithms, one for pressure build (+U_{val}) and one for pressure dump (-U_{val}). The objective of the PWM blocks is to produce an on/off signal that triggers a build pressure, dump pressure or hold pressure command. While a typical PWM control strategy cycles the actuator in rapid succession, multiple times per time step, in this case the base rate of the signal is set equal to that of the time step, and a single pulse is then used to control the period of valve actuation.

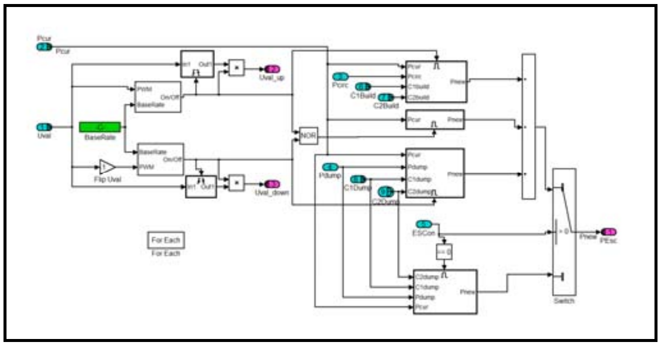

Figure 10: Forward Hydraulics/Pneumatics 'ForEach' Subsystem.

Once the on/off signals are generated by the two PWM blocks shown in Figure 10, they are routed to the enable ports on the forward hydraulics/pneumatics build, dump and hold pressure blocks. An "on" or "1" signal from the top PWM block causes the "Build For U_{val}" triggered subsystem to enable. This block then builds an output pressure. If the build block is on, the hold and dump blocks are deactivated.

If both PWM blocks are sending an "off" or "0" signal and *ESCon* is greater than zero, ESC is still active but it is not requesting any change in the pressure state. In this case, the hold pressure block is enabled. This block simply passes the current pressure as the new pressure. All of the enable blocks reset to zero when not enabled. Therefore, the build, hold and dump pressure blocks can be safely summed to achieve a new wheel cylinder pressure.

When the build pressure block is enabled, a new pressure is calculated based on Equation (6). This is the same relationship used in the inverse hydraulics/pneumatics build pressure block. Similarly, when the dump pressure block is enabled, a new pressure is calculated using Equation (7). They are simply rearranged to solve for P_{new} [5].

$$P_{new} = P_{cur} + U_{val}(c_{1build} + c_{2build} P_{cur})\sqrt{|P_{circ} - P_{cur}|} \quad (6)$$

$$P_{new} = P_{cur} + U_{val}(c_{1dump} + c_{2dump} P_{cur})\sqrt{|P_{dump} - P_{cur}|} \quad (7)$$

Where U_{val} is the valve cycle time, P_{circ} is the max brake system pressure available, P_{dump} is the pressure in the wheel cylinder in a zero brake condition, P_{new} is the new brake pressure, P_{cur} is the current brake pressure, and c_1 and c_2 are the tunable parameters to control the first order time delay in the brake engagement (for both increasing and decreasing pressure.)

There is one last scenario that the forward hydraulics/pneumatics model must handle. This is the case where ESC has been triggered on and then shuts off. This can occur because the yaw rate of the vehicle has returned back to the desired yaw rate or the speed of the vehicle has decreased below the threshold where ESC is enabled. This scenario requires the Simulink model to reduce any remaining brake pressure created by the ESC event to the zero brake state. The block below the build, dump, and hold blocks is referred to as the "Dump All" block. This block is tasked with dumping any remaining ESC brake pressure.

The *ESCon* channel that originates in the SetPoint and State Controller is compared to zero and the result is used by the enable port of the Dump All block. Therefore, if *ESCon* is "0" the Dump All block is enabled and a switch is triggered to pass the pressures from it to the output channel instead of the build, hold, or dump addition block. The drop in pressure is controlled by the same equation for the dump block in the forward hydraulics/pneumatics model. The same tunable parameters, c_{1dump} and c_{2dump}, define how quickly the current pressure can return to the zero brake state.

CONCLUSION

The ESC controller presented in this report provides the ability to design, evaluate, and study the effects of adding electronic stability control to existing and prototype tactical wheeled vehicles. AMSAA and ARL engineers now have access to an ESC simulation model that can replicate the level of complexity in vehicle state estimation and brake dynamic modeling seen in real ESC systems. With this capability, engineers can also explore the interconnectivity of such a system with the human driver or other driver-assist technologies such as Traction Control or Antilock Brake systems once added to the model. The controller's architecture supports expansion to vehicles with more than four wheels and to simulation packages other than TruckSim.

REFERENCES

[1] *Safety, Comfort, and Convenience Systems*. Robert Bosch GmbH. 2006: Bentley Publishers.

[2] MathWorks. "www.mathworks.com." 5 April 2012. Online. Available: ("http://www.mathworks.com/products/simulink/")

[3] Mechanical Simulation. "Mechanical Simulation." 5 April 2012. Online. Available: ("http://www.carsim.com/products/trucksim/index.php").

[4] Walker Jr., J., *Introduction to Brake Control Systems: ABS, TCS, and ESC Seminar.* Aug 2011.

[5] Bosch. *Bosch Automotive Handbook*. 7th ed. Robert Bosch GmbH. 2007.

[6] Rajamani, R., *Vehicle Dynamics and Control,* Springer, New York, 2000