Homework Part 2: Modeling the Guidance Navigation & Control System for a Skid-Steer UGV

Instructions:

The following questions for Part 2 are related to the ugvGNC\_complete.slx Simulink Model you will find in the Class Sept 30.zip file available to you in the UGV Modeling and Simulation Modules section of the class CANVAS site.

Please download and unzip the Class Sept 30 file on your computer. Point MATLAB to that extracted folder. Open up the ugvGNC\_complete.slx file. This should open up the Model in Simulink. Please save this model again under a new name of your choosing. This will be your working copy of the GNC Model. You can now make changes and play with this model. If you “Break It” you can always reopen the original and create another working model for yourself.

The following questions are related to this ugvGNC\_complete model.

6.0 **The following questions are related to the GNC components of a Skid-Steer UGV platform. Dr. Dogan’s Powerpoint Slides “UGV simulation with Simulink (1).pdf” will also be useful when answering questions in Part 2 of this homework. Important Note: The Simulink model in the Powerpoint Sides is slightly different from the ugvGNC\_Complete model. Please base your answers on the ugvGNC\_Complete version of the GNC implementation.**

ok

6.1 **In your own words, what function or job does the Navigation Component of the GNC subsystem do?**

Purpose: estimate current vehicle state (position, orientation, velocity)

6.2 **In your own words, what function or job does the Guidance Component of the GNC subsystem do?**

Purpose: figures out desired behavior to achieve task

6.3 **In your own words, what function or job does the Control Component of the GNC subsystem do?**

Purpose: generate motor commands to achieve desired behavior

6.4 **How does the GNC subsystem interact with the UGV subsystem? What does the GNC subsystem give to UGV subsystem? What information does the GNC subsystem get form the UGV?**

The GNC subsystem gives the UGV subsystem the DCR and DCL. The GNC subsystem gets the “CountR” and “CountL.” I believe this is the “eCount(K)” in the powerpoint, which is the encoder count for the current sample time.

6.5 **In class we said we were using Dead Reckoning to support the Navigation Function within our system. Please provide a general description of the Dead Reckoning process. Then briefly describe specifically how we are going to implement this within our ugvGNC simulation?**

Dead Reckoning: navigation method to estimate position using direction, speed, elapsed time.

6.6 **To support Dead Reckoning in our ugvGNC simulation we had to add an additional component to our UGV vehicle simulation subsystem. Where is this new model component located in the overall Simulink Model? What physical hardware is this component simulating? Provide a picture of the Simulink Blocks that are implementing this functionality. In your own words, explain how it works.**

A screenshot of a computer

AI-generated content may be incorrect.

A diagram of a computer program

AI-generated content may be incorrect.

The dead recoking system is in the “ugvGNC\_complete->GNC System->Nagivation block. It is simulating the APU, arithmetic processing unit, or more generally, the CPU. So it gets the estimated forward velocity (V), then it calculates the theta-dot, aka turn-rate. Then from there it splits it into the x-dot and y-dot, the x and y-component of the velocity. And it integrates the theta-dot, the turn-rate to get the current angle. Then you integrate the x-dot and y-dot to get current x and current y positions. Then you do Pythagorean theorem on the x-dot and y-dot to get the instantaneous forward velocity.

6.7 **What role does the eTick parameter play in this UGV simulation component?**

Ticks per meter. We use the ticks per meter to find the estimated wheel speed.

7.0 **The following set of Simulink Blocks are found in the Speed Estimation component of the Navigation Sub-System. Please answer the following questions related to the Speed Estimation component.**

ok

Diagram, schematic

Description automatically generated

7.1 **What is the “big picture” job of all of these blocks.**

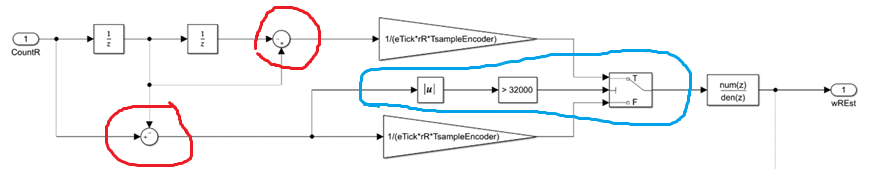
It uses the # of ticks in a sampling period to get the estimated speed. If the most recent sampling period is close to overflowing, or if it is reaching max speed, aka 32000, then use the previous sampling period: eCount(k-1) – eCount(k-2) instead of eCount(k)-eCount(k-1)

7.2 **The logic implemented in this set of blocks is designed to take care of a common problem found when using encoders to support navigation. What is this problem? What causes this problem?**

The common problem is over and underflow. What causes this problem is when your number becomes larger than the number of bits you allocated for that number.

7.3 **How does the logic expressed in these blocks handle this problem. Refer to the specific blocks to answer this question. Pictures or diagrams might be helpful.**

If the most recent sampling period is close to overflowing, or if it is reaching max speed, aka 32000, then use the previous sampling period: eCount(k-1) – eCount(k-2) instead of eCount(k)-eCount(k-1)



So the top red is the eCount(k-1)-eCount(k-2) and the lower is the current sampling period. Then the blue block does the calculation to see if the current sampling period is too large or not. If it’s >32000, “switch” to the top path and use that number.

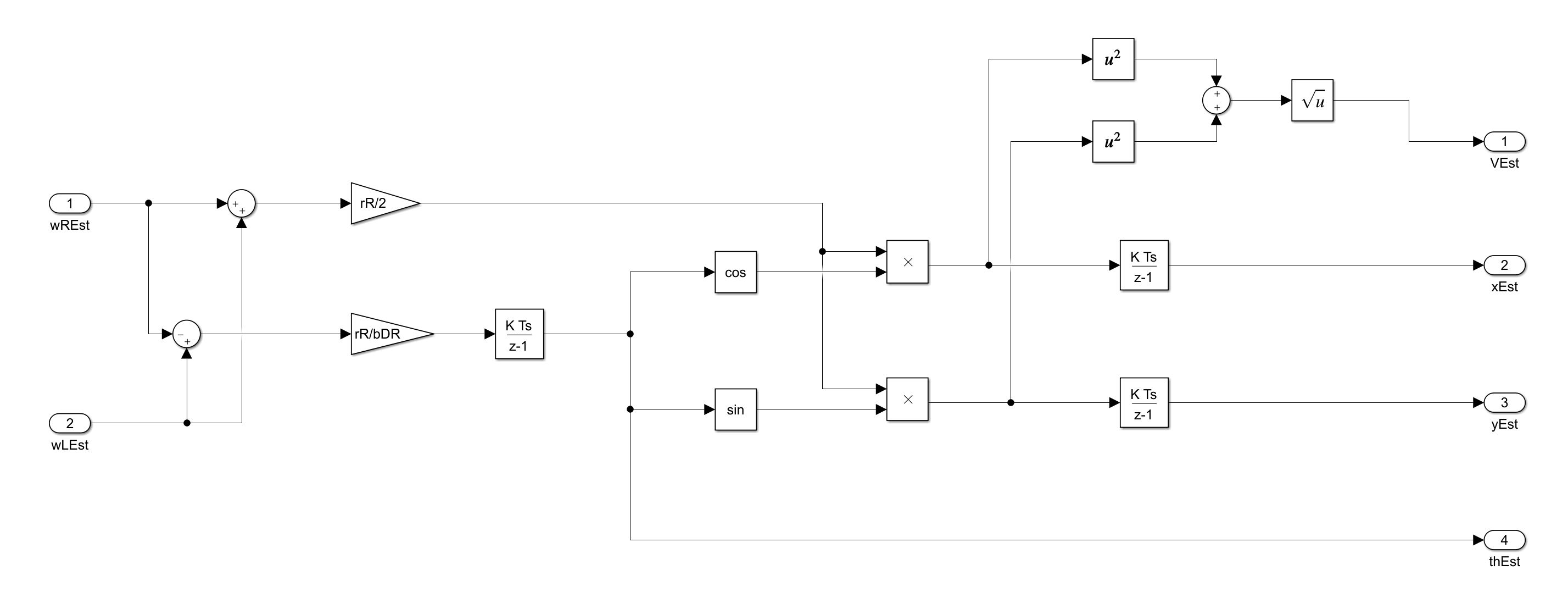
7.4 **What job does the following Simulink Block perform in this process?**

Diagram

Description automatically generated

This is the discrete 1st order filter to smooth the calculation

8.0 **Explain in your own words what is happening in the following series of Simulink Blocks**



So it gets the left and right wheel velocity then it calculates the theta-dot, aka turn-rate and forward velocity. Then from there it splits it into the x-dot and y-dot, the x and y-component of the velocity. And it integrates the theta-dot, the turn-rate to get the current angle. Then you integrate the x-dot and y-dot to get current x and current y positions. Then you do Pythagorean theorem on the x-dot and y-dot to get the instantaneous forward velocity.

9.0 **The following questions are associated with the Guidance function within the GNC system.**

ok

9.1 **How does a “User” of our ugvGNC simulation tell the simulated UGV where the User wants the vehicle to go?**

You change the waypoints in the parameters:  
A screenshot of a computer program

AI-generated content may be incorrect.

9.2 **What role does the “rp1” parameter play in the Guidance Subsystem?**

Radius from waypoint at which point to start slowing down

9.3 **What role does the “rp2” parameter play in the Guidance Subsystem?**

Radius from waypoint at which point to switch to next waypoint

9.4 **What is happening in these Simulink Blocks?**

Diagram, schematic

Description automatically generated

So we are getting the current x and y position, we are then converting it into a continuous value, then we subtract how far we are from the x and y of the waypoing. Then we combine those 2 values into 1 vector

9.5 What is the job of this set of Simulink Blocks? Explain how this logic works. What information is being input to this set of blocks?

Diagram

Description automatically generated

Ok so here, in “<=rp1”, we are checking how far we are from the waypoint. If we are “close”, aka “<=”, then “do stuff”, aka do “triggered subsystem”, and then “lookup” what the “x” and “y” values of the next waypoint are. Then, in the “1/z” block, we are “waiting” and “holding” the values for 1 sample period, and then we release them. Why? Most likely to prevent waypoints from changing in the middle of a calculation.

9.6 What is the purpose of the set of Simulink Blocks?

Diagram

Description automatically generated

This checks if it is the last waypoint, and if it is, then stop. The block is just seeing where in the array is the next waypoint.

9.10 What is the function of this set of Simulink Blocks? In your own words explain how it works.

Diagram

Description automatically generated

Ok so this is taking in “r,” aka how close we are to the waypoint. If we are closer than the constant, “rp2”, then change “desired” speed to 0.1. So, slowdown. If we are still far, maintain the course. The manual switch to the right, I guess, is a manual “override.”

9.11 **There are two locations within the Guidance Subsystem where the Simulation “User” has the ability to manually control how the Guidance Subsystem behaves. Please explain where these two “Manual Controls” are located and what behavior they control in the Guidance Subsystem. If pictures help, use them.**

In the speedCommand block, manually switch to vcom1. So instead of actually modifying speed depending on proximity to a waypoint, just maintain speed.A diagram of a circuit

AI-generated content may be incorrect.

And in this spot below where you can choose whether to loop again or stop  
Diagram

Description automatically generated

10.0 **The following questions are related to the Control function within GNC.**

ok

10.1 **What is the role of this set of Simulink Blocks in the Control Subsystem?**

Diagram

Description automatically generated

Ok so we smooth the calculation in the ZOH and ZOH1. We modify the VelocityEst to be continuous, and then we subtract the computed vs estimated velocity to get errorV. And in the bottom one, we subtract theta computed vs theta estimated to get error theta.

10.2 **How does the Control Subsystem automatically drive the system to match the Commanded Velocity and Commanded Heading?**

Using the PID Control block at the top and bottom.

10.3 **Our Control Subsystem has two PID controllers embedded inside the subsystem. What are each of these PID controllers doing?**

The top one is for the velocity the bottom one is for the angle

10.4 **A PID Controller has three components: a Proportional component, a Integral component, and Differential component. Use the reference material provided to you (Powerpoint, and PDF doc) to describe the role of each of these components within a PID controller.**

P control – low steady state error, high respnsonse speed, high instability. Increases stiffness and decreases damping coefficient, increasing vibrational frequency and amplitude

I control – very low steady state error, small increase in instability

D control – increases damping and improves stability, decreasing vibration amplitude.

10.5 The Control Subsystem Block within our model outputs Angular Velocity commands (wR and wL) to the two wheels/tracks. Within our model this is converted into % PWM Duty Cycles that are the control inputs to our simulated H-Bridge Amplifiers that drive the motors attached to the left and right tracks/wheels. In our GNC model WHERE and HOW is the conversion from Angular Velocity commands to % PWM Duty Cycles performed?

This is performed using a lookup table. This is located after the control block.

A diagram of a diagram

AI-generated content may be incorrect.