2025 Control System Description

3/16/2025

The control system on the robot is broken into systems. Each system contains the control logic for reading sensors, processing commands, performing closed loop control logic, writing values to motors, and writing values to network tables.

Some systems, such as Human, Trajectory, Vision, and Autonomous have specialized functions.

The control system is broken into the following systems: Drive, Climb, Intake, Elevator, Wrist, Human, Autonomous, Vision, and Trajectory.

The programming/control system team this year consisted of:

Zeke Simpson (programming lead), Mya Mitchell, Zach, Will, Ezra, Liam, Jack, Tim, James Sheridan.

Drive

The drive system reads a chassis demand setpoint sent from other systems such as Human or Autonomous. The setpoints are rate limited to help prevent the drivers from tipping the robot. The control system calculates the angle and travel speed demand for each of the 4 swerve modules. The angle of each module is controlled by a closed loop position control algorithm. This is a proportional only controller with an error deadband and output limiting. The travel speed is controlled with a feedforward and a PID (proportional, integral, derivative) controller. Values for each module are written to network tables so the dashboard can use this information for diagnosis and tuning. There is separate code that characterizes the feedforward by driving the robot at various constant motor outputs and records the velocity. There is also separate code that accelerates and decelerates the robot at a constant rate to determine the lag and calculate the acceleration lead feedforward term (Ka). There is also separate code that drives the robot at a constant speed then makes a step change to calculate the PID tuning. This routine uses the LabVIEW PID tuning functions.

Climb

The climb control logic responds to commands sent from other systems such as Human and Autonomous. The following commands are implemented: Reach (out), Pull (up), Cancel, Reach Increment, Pull Increment. Magnetic limit switches monitor the position of the climber. The control logic uses a finite state machine to control the motor output. This includes logic to pull the robot up if it starts to drop. Information is written to Network Tables so it can be viewed on the driver station.

Intake

The intake control logic responds to commands sent from other systems such as Human and Autonomous. The following commands are implemented: Intake Coral, Intake Algae, Deposit Coral, Deposit Algae, and Cancel. An Allen-Bradley proximity sensor (light reflection) detects the presence of a coral. A beam break sensor can detect the presence of an algae. (Currently the mechanism cannot hold an algae.) The control logic uses a finite state machine to control the left, right, and top motor outputs. Information is written to Network Tables so it can be viewed on the driver station.

Elevator

There are two separate elevators that move independently, an inner and outer elevator. The elevator control logic responds to commands sent from other systems such as Human and Autonomous. Commands for each preset position, and Cancel are implemented. The preset and cancel commands set the height demand for both elevators. There are also increment and decrement commands for the inner elevator. Commands for wrist position are sent to correspond with the requested elevator demand. The height of each elevator is determined from the scaled value of the Neo brushless motor's built in encoder. Each elevator also has upper and lower limit switches. The control logic position and motor controller position values are reset when a limit switch is encountered. The height of each elevator is controled by a closed loop position control algorithm. This is a proportional only controller with an error deadband and output limiting. Inner elevator demand is limited to protect the wrist when it's desired or actual position is pointing down. Information is written to Network Tables so it can be viewed on the driver station.

Wrist

The wrist control logic responds to commands sent from other systems such as Elevator or Human. Commands for each preset position, and Cancel are implemented. The angle of the wrist is determined by reading a through bore absolute encoder. The wrist angle position is controlled by a closed loop position control algorithm. This is a proportional and derivative controller with an error deadband and output limiting. The controller also includes gravity compensation. The wrist angle demand is limited to protect the wrist when the elevator is too low to safely lower the wrist. Limiting is also used to help the wrist not get entangled with bumpers and elevator limit switches while the elevator is traveling. Information is written to Network Tables so it can be viewed on the driver station.

Autonomous

The autonomous system processes a list of instructions. Each instruction contains an action and up to 4 parameters. A time out parameter is also provided to force the list processor to go to the next step even if hasn't determined it is done. Some of the commands are: drive forward/backwards, turn, follow absolute trajectory, follow relative trajectory, position elevator, intake coral, deliver coral, wait for elevator, wait. There are auto instructions for everything that the robot is capable of doing. The auto lists are a memory data structure that is read from from a file when the robot code starts. Currently 30 auto files are configured on the robot for use. The desired auto routine is chosen from the dashboard prior to the match. These files are formatted as CSV (command separated values) files. A separate auto editor program, that runs on a PC, can read, edit, and write these auto files. They can then be transferred to the robot. A PC based robot simulator can be used to fully simulate robot operation, including executing auto routines.

Human (Teleop)

The human system executes during teleop. It responds to inputs from the human players from joysticks and USB button boxes. It sends commands to other systems. It determines the drive mode, either joystick controlled, or trajectory controlled by one of the destination buttons. For the driver there is robot or field oriented driving. There is also a slow mode button for accurate robot positioning.

Vision (Odometry)

The vision system continuously calculates the robot's absolute position on field. It uses swerve module distances and angles, the robot's gyro reading to calculate a field position. It statistically merges this with position data calculated from one or more cameras detecting April Tags. The robot has 2 vision co-processors, running PhotonVision. Each has 2 cameras looking for April Tags. One has a third camera for the drivers to watch. When the either of the two front facing cameras detect an April Tag, this data is used. If both front cameras detect an April Tag, these positions are averaged. If neither front cameras see an April Tags then the rear and side cameras are used. Similarly if both the rear and side cameras see an April Tag this data is averaged. Information is written to network tables, allowing camera detection status and robot position to be displayed on the dashboard.

Trajectory

This system reads 30 absolute trajectory and 30 relative trajectory files when the robot code starts. (These numbers may change.) These can be used for autos. This system contains the code for following relative and absolute trajectories. These use the WPI trajectories with the addition of code to orient swerve robot separate from the direction of travel. There are also routines for creating straight line trajectories from the current location to another location. This is used by the human system for auto positioning at the reef, feeder stations, or climbing station.