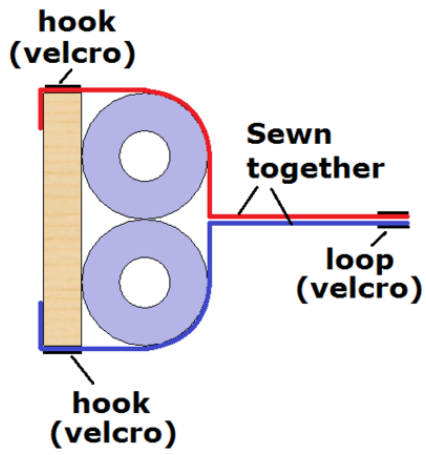
THE VARMINT WEEKLY

Third Edition February 5 2018

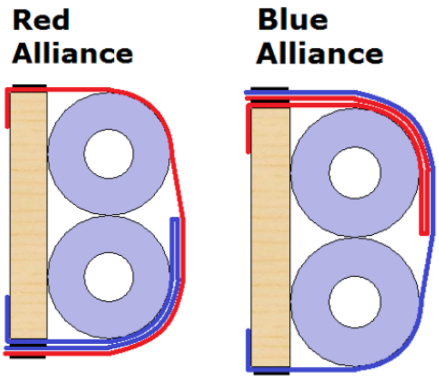
Written and edited by Ben Chandra Contributing Writers: Dave Beers and Ryan Feldbush

The Bumpers

Bumpers are an important implement for a team. For the drive team, the 386 emblazoned on the fabric allows for easy identification through the chaos of gameplay, for spectators the fiery red or cool blue distinguishes its affiliation, but perhaps most importantly, the bumpers wrap the robot, shielding its delicate components from the rigor of a match.



Bumpers need to be either red or blue depending on which team the bot is on. This means that a pit crew needs to work fast to secure the new bumper to the bot after each game while still allowing additional time to perform routine maintenance. Teams work hard to make bumpers that can be installed quickly while still being strong enough to weather the brunt of a hundred pound bot crushing into it; to handle three minutes of sustained wear and tear, yet be pulled off in the span of seconds. It's because of this, there is a movement in the *FIRST* community to use a more intuitive form of bumpers. More and more teams have instead began using two-in-one bumpers which can switch between red and blue team via a hook and loop on a strap. These reversible bumpers have been in use by our team for around 6 years. They allow our team to quickly change our alliance color between matches so our robot is always ready for its next game. To the right are schematics from a similar bumper uploaded to Chief Delphi , a popular community site for *F.I.R.S.T.*



Software Team Update

With the robot chassis not yet made and the wiring not quite finished, software has been focusing on the autonomous part of the match. In the first 15 seconds of the match, the robot is scored on what it can do without user input. This can mean anything from driving in a straight line to picking up a cube and placing it on a mechanism. In order to match the high expectations set by earlier years, our robot must be able to find a yellow cube, identify it, then move a set distance to pick it up and place it on a lever. Our software team faces two main problems in achieving this. Obviously, ou can't just tell your robot to move "x feet." For a robot, precision autonomous entails telling the motors to turn the wheel "x" amount of rotations, incorporating the circumference of the wheels. The robot must also use various sensors to adjust for outside factors.

In regards to basic movement, the software team has written code connected to the motor's encoder, which records how many times the motor has turned, to tell the robot to stop after a set distance. More difficult, of course, is in the turning. The software team has incorporated Proportional Derivative (PD) loop to govern the robots turning. This means using the various speeds throughout the turn and using the rate of change to adjust for better handling. This is in contrast to a Proportional Integral Derivative (PID) loop which uses integrals to account for outside noise. The PID loop, while more precise, was deemed unnecessary due to the brevity of the tasks. The software team plans to implement further tuning to both the forward and turning code once the chassis is built and additional variables such as weight can be considered. As for foreign variables software has been focusing on incorporating robots gyro sensors based on a proportional control system. In layman's terms, if an unexpected value is received by the gyro sensor an inverse but equivalent force is applied.

Of course, that's only half of the equation. In regards to vision processing, we have completed an algorithm to differentiate between adjacent cubes. The algorithm uses OpenCV to take an image from our camera and create a submat from the yellow portions of the original image, then uses canny edge detection to find the boundaries between cubes, these edges are then dilated, inverted, and superimposed upon the yellow submat to get clear, distinct rectangles to control the robot. The algorithm has been tested and shown to be effective (although the sheer complexity of it only allows for 3 processed frames per second) but currently has not been tested on a moving robot.