2018 FLYSET FTC Workshop

- **Chassis Study Test Design**

(9/3/2018)

Brandon

- Joined 2011
- Junior at Plano West Senior High School
- Software lead
- Career aspirations: CS
- When not doing robotics: tennis, reading

technicbots 8565

Chassis Study Test Design

Chassis Categories

- Basic Chassis: designed for comparison
- Experimental Chassis: designed for exploring

technicbots 8565

Test Assumptions for Basic Chassis

- All chassis use AndyMark NeveRest 40 motors
- All chassis use 4 inch wheels with material noted
- Test on same surface (gray FTC mat)
- Same battery level (13.5±0.25 V) based on display on driver station
- FTC SDK 3.7
- Conduct each test with 0 lb, 15 lb, 30 lb additional weight load on top
- Each test (with the same load) should be run three times and calculate the average value
- Record battery voltage for all tests

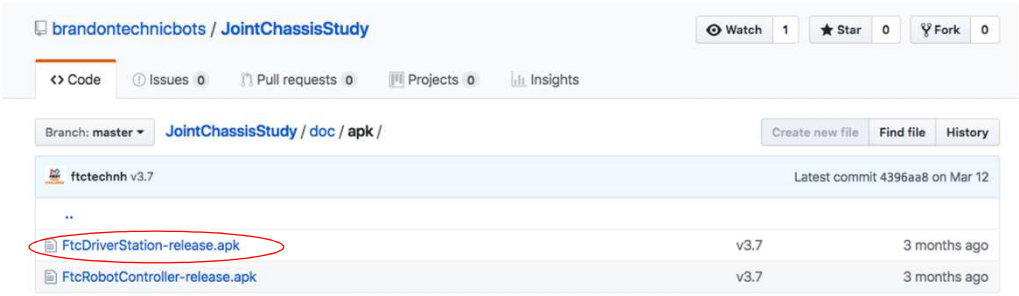
technicbots 8565

Test Assumptions for Experimental Chassis

- Chassis can choose appropriate Motor and Wheel size to fit the study purpose
- Test on same surface (gray FTC mat)
- Same battery level (13.5±0.25 V) based on display on driver station
- FTC SDK 3.7
- Conduct each test with 0 lb, 15 lb, 30 lb additional weight load on top
- Each test (with the same load) should be run three times and calculate the average value
- Record battery voltage for all tests

technicbots 8565

Install Compatible Driver Station App



technicbots 8565

Test Code Assumptions

In the robot configuration on phone, name the motors/imu accordingly (without the quotes):

- Left Front motor: **"left_front"** (with encoder attached)
- Left Back motor: **"left_back"**
- Right Front motor: **"right_front"**
- Right Back motor: **"right_back"**
- IMU/Gyro: **"imu"**

For two wheel/three wheel drive chassis, adjust the motor name according to your configuration.

technicbots 8565

Testing Code Template

Github link found here: https://tinyurl.com/FLYSETChassisStudyTestCode

Just download, modify to fit your robot configuration (under teamcode), and run!

- DriftTest.java
- Forward.java
- GyroTurn180.java
- GyroTurn90.java
- Robot.java
- Teleop.java
- Turn.java

technicbots 8565

Testing Code Template

```java
// This code executes the first test (Driving forward for 5 seconds).
// The full specification is in the Engineering Notebook.
@Autonomous(name = "Test 1: Forward Speed Test", group = "Linear Opmode")
public class Forward extends LinearOpMode {

    private Robot robot  = new Robot();
    private ElapsedTime runtime = new ElapsedTime();

    @Override
    public void runOpMode() {
        robot.init(hardwareMap);

        waitForStart();
        runtime.reset();

        // Run the robot forward for 5000 ms
        while (runtime.milliseconds() < 5000 && opModeIsActive()) {
            robot.leftFrontMotor.setPower(1);
            robot.leftBackMotor.setPower(1);
            robot.rightFrontMotor.setPower(1);
            robot.rightBackMotor.setPower(1);
        }
        // Stop the robot
        robot.stopRobot();
    }
}
```

```java
/**
 * This class defines the hardware for the robot and its initialization
 */

public class Robot {

    DcMotor leftFrontMotor, rightFrontMotor, leftBackMotor, rightBackMotor = null;

    /* local OpMode members. */
    private HardwareMap hwMap = null;

    /* Constructor */
    Robot() {

    }

    /* Initialize standard Hardware interfaces */
    void init(HardwareMap ahwMap) {
        // Save reference to Hardware map
        hwMap = ahwMap;

        //If you need to change the configuration, do it here
        leftFrontMotor = hwMap.get(DcMotor.class, "left_front");
        leftBackMotor = hwMap.get(DcMotor.class, "left_back");
        rightFrontMotor = hwMap.get(DcMotor.class, "right_front");
        rightBackMotor = hwMap.get(DcMotor.class, "right_back");
```

technicbots 8565

Test 1: Forward Speed Test

- Class name: **Forward.java**
- Drive forward for 5 seconds at full power (100%)
- Measure distance traveled in meters

	No load	15 lb load	30 lb load
Distance traveled			

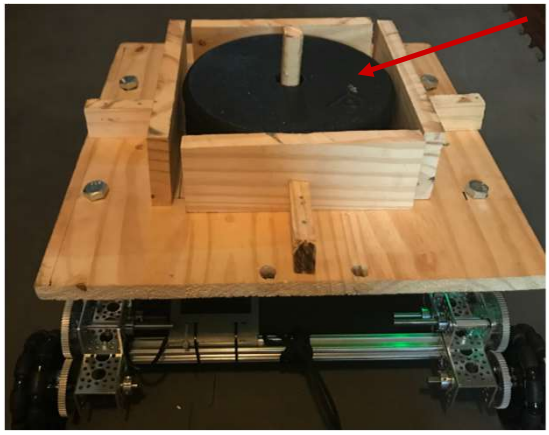
technicbots 8565

Test 1: Forward Speed Test Code Snippet

```
// Run the robot forward for 5000 ms
while (runtime.milliseconds() < 5000 && opModeIsActive()) {
    robot.leftFrontMotor.setPower(1);
    robot.leftBackMotor.setPower(1);
    robot.rightFrontMotor.setPower(1);
    robot.rightBackMotor.setPower(1);
}
```

technicbots 8565

Load Testing Setup



Test 2: 3 Second Turn Test

- Class name: **Turn.java**
- Turn for 3 seconds at full power (one side 100% power forward, other side 100% power backward)
- Record: can it turn freely? Does it get stuck? (Yes/No)
- If it can turn freely, record how far it turned in degrees using IMU sensor

	No load	15 lb load	30 lb load
Degree turned			

Test 2: 3 Second Turn Test Code Snippet

```
// Turn the robot for 3000 ms
while (runtime.milliseconds() < 3000 && opModeIsActive()) {
    robot.leftFrontMotor.setPower(-1);
    robot.leftBackMotor.setPower(-1);
    robot.rightFrontMotor.setPower(1);
    robot.rightBackMotor.setPower(1);
}
```

technicbots 8565

Test 3: Balance Stone balancing ability

- Class name: **Teleop.java**
- Drive the robot on to the Relic Recovery Balancing stone, and then attempt to balance.
- Balancing Stone — 2" above playing field, 23"x23" surface
- Record: can the robot drive on to the balance stone, and then balance (Yes/No)?

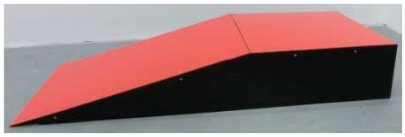
	No load	15 lb load	30 lb load
Can Balance?			

technicbots 8565

Test 4: Driving up/down a ramp

- Class name: **Teleop.java**
- Manually drive up the ramp, then drive back down
- Ramp is 36 in long with 20 degree slope
- Record: can it drive up the ramp and then drive back down? (Yes/No)
- Use the ramp from FTC Cascade Effect Challenge

	No load	15 lb load	30 lb load
Can drive on/off ramp?			



technicbots 8565

Test 3: Teleop Code Snippet

```
// Basic Tank Drive to control the robot.
@Override
public void loop(){
    leftThrottle = gamepad1.left_stick_y;
    rightThrottle = gamepad1.right_stick_y;

    robot.leftFrontMotor.setPower(leftThrottle);
    robot.leftBackMotor.setPower(leftThrottle);
    robot.rightFrontMotor.setPower(rightThrottle);
    robot.rightBackMotor.setPower(rightThrottle);
    }
```

technicbots 8565

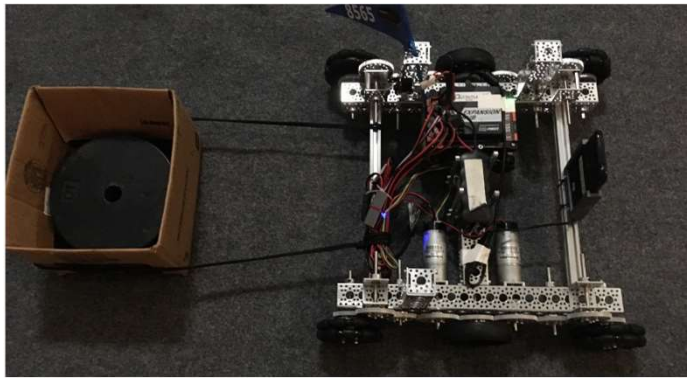
Test 5: Pull Strength Test

- Class name: **Drift.java**
- Determine pulling force over distance of 10ft
- Tie increasing weights behind robot on a cardboard sled (10 lb, 20 lb, 30 lb additional) with 1 foot behind robot
- Drive the robot at full power
- Measure time taken in seconds to travel the 10 ft for each weight

	No load	15 lb load	30 lb load
10 lb pull weight			
20 lb pull weight			
30 lb pull weight			

technicbots 8565

Test 5: Pull Strength Test



technicbots 8565

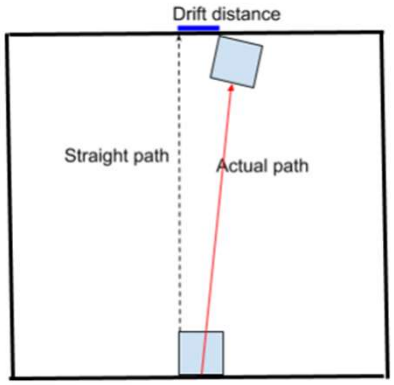
Test 6: Autonomous Straight Line Drift Test

- Class name: **DriftTest.java**
- Motor encoder needed
- Important to use matched motors
- Travel forward distance of 10 ft autonomously
- Measure horizontal drift in inches (perpendicular to the robot's direction of movement)

	No load	15 lb load	30 lb load
Horizontal drift			

technicbots 8565

Test 6: Autonomous Straight Line Drift Test



technicbots 8565

Test 6: Autonomous Straight Line Drift Test Code Snippet

```
private static final double     COUNTS_PER_MOTOR_REV    = 1120 ;    // For Neverest 40
private static final double     DRIVE_GEAR_REDUCTION    = 1.0 ;     // This is < 1.0 if geared UP
private static final double     WHEEL_DIAMETER_INCHES   = 4.0 ;     // For figuring circumference
private static final double     COUNTS_PER_INCH         = (COUNTS_PER_MOTOR_REV * DRIVE_GEAR_REDUCTION) /
        (WHEEL_DIAMETER_INCHES * 3.1415);

private double targetTicks = 120 * COUNTS_PER_INCH; // 120 in converted to counts
```

technicbots 8565

Test 6: Autonomous Straight Line Drift Test Code Snippet (cont)

```
// Reset the encoder on front left motor
robot.leftFrontMotor.setMode(DcMotor.RunMode.STOP_AND_RESET_ENCODER);

// Run the robot forward for the target number of ticks
while (robot.leftFrontMotor.getCurrentPosition() < targetTicks && opModeIsActive()) {
    robot.leftFrontMotor.setPower(1);
    robot.leftBackMotor.setPower(1);
    robot.rightFrontMotor.setPower(1);
    robot.rightBackMotor.setPower(1);
}
```

technicbots 8565

Test 7: Autonomous 90/180 degree turn offset

- Class name: **GyroTurn90.java** and **GyroTurn180.java**
- Use built in IMU in REV Expansion Hub to measure turning angle
- Use smaller power such as 0.6
- Turn 90 and 180 degrees using IMU (sample code is provided)
- Tune P controller using 15 lb load
- Measure drift (in degrees from target angle) as read from IMU displayed on the driver station phone

	No load	15 lb load	30 lb load
Degree drift (90° target)			
Degree drift (180° target)			

technicbots 8565

Test 7: Autonomous 90/180 Degree Turn Code Snippet

```
robot.init(hardwareMap);
imu = hardwareMap.get(BNO055IMU.class, "IMU");
imuInit();
```

```
private float getHeading() {
    return imu.getAngularOrientation(AxesReference.INTRINSIC, AxesOrder.ZYX, AngleUnit.DEGREES).firstAngle;
}


/* Initializes Rev Robotics IMU */
private void imuInit() {
    // Set up the parameters with which we will use our IMU.
    BNO055IMU.Parameters parameters = new BNO055IMU.Parameters();
    parameters.angleUnit = BNO055IMU.AngleUnit.DEGREES;
    parameters.accelUnit = BNO055IMU.AccelUnit.METERS_PERSEC_PERSEC;
    parameters.calibrationDataFile = "BNO055IMUCalibration.json";
    parameters.loggingEnabled = false;
    parameters.loggingTag = "IMU";
    imu.initialize(parameters);
}
```

technicbots 8565

Test 7: Autonomous 90/180 Degree Turn Code Snippet (cont)

```
private void gyroTurn(double deg) {
    double target_angle = getHeading() - deg;
    while (Math.abs((target_angle - getHeading())% 360) > 1 && opModeIsActive()) {
        double error_degrees = (target_angle - getHeading()) % 360; //Compute Error
        double motor_output = Range.clip(error_degrees * TURN_P, -.6 ,.6); //Get Correction
        // Send corresponding powers to the motors
        robot.leftFrontMotor.setPower(-1 * motor_output);
        robot.leftBackMotor.setPower(-1 * motor_output);
        robot.rightFrontMotor.setPower(motor_output);
        robot.rightBackMotor.setPower(motor_output);
    }
```

technicbots 8565

Test 7: Autonomous 90/180 Degree Turn Code Snippet

```
// This code executes the 7th test (90 degree turn offset).
// The full specification is in the Engineering Notebook.
@Autonomous(name = "Test 7: 90 degree turn offset", group = "Linear Opmode")
public class GyroTurn90 extends LinearOpMode {

// This code executes the 7th test (180 degree turn offset).
// The full specification is in the Engineering Notebook.
// This code was written for the Rev Robotics IMU.
// The Modern Robotics system will use slightly different gyroscope code.
@Autonomous(name = "Test 7: 180 degree turn offset", group = "Linear Opmode")
public class GyroTurn180 extends LinearOpMode {
```

technicbots 8565

Test 8: Chassis Specific Test

- If the chassis has a special feature, it can be tested individually for the chassis.
- Examples on these kinds of tests: sideways strafe test for omnidirectional chassis

	No load	15 lb load	30 lb load
Defined feature			

technicbots 8565

GrabCAD

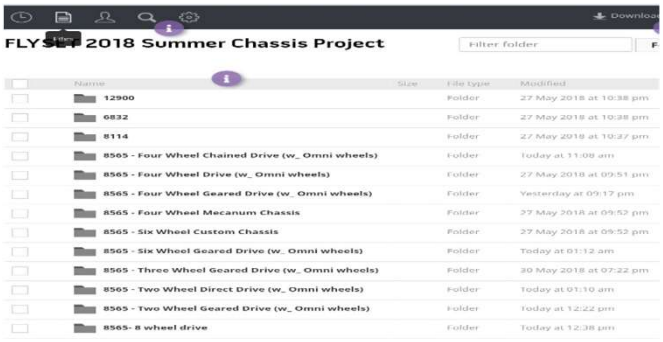
Justin

- Joined 2008
- Senior at Hebron High School
- CAD lead
- Driver Coach
- Career aspirations: Designer
- When not doing robotics: eating, sleeping

technicbots 8565

GrabCAD

This summer, we utilized GrabCAD as an organization tool for chassis development

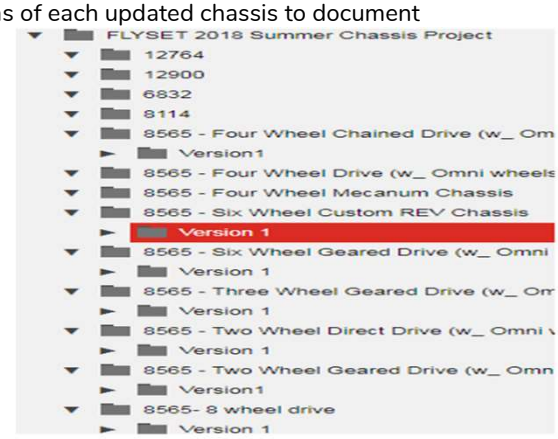


technicbots 8565

GrabCAD

Under each folder, we stored versions of each updated chassis to document iterations

- Allowed us to see version history and development between versions
- Organized 9 different projects into a comprehensive list

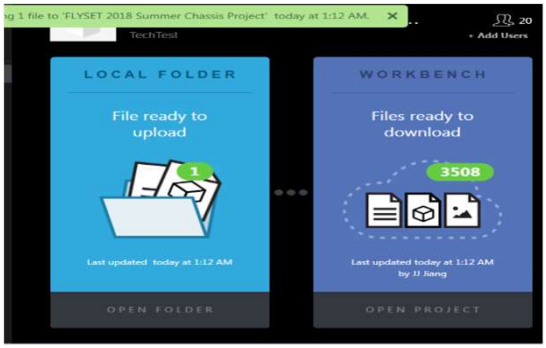


technicbots 8565

GrabCAD

Benefits-

- Instantaneous upload and automatic updating of any changed files
- Increased collaboration between team members
- Easily shows who edited a file and how that file was changed
- Dropbox of CAD



technicbots 8565