# Sphero RVR Control System Manual

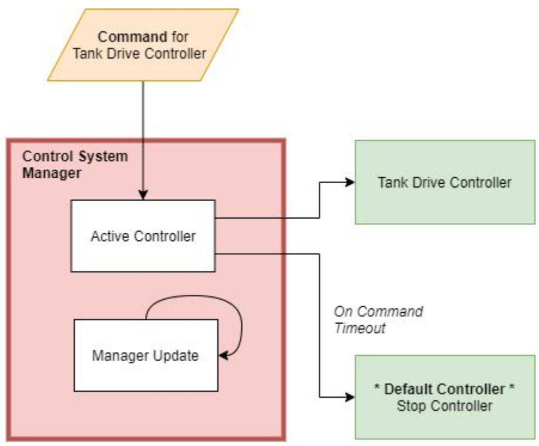
Applicable Firmware Versions:
Nordic SoC v8.3.432
ST SoC v8.6.448

# Introduction

The objective of this documentation is to help the end user understand and use all of the control system features of their Sphero RVR.

## Control System Manager

The control system manager manages all of the control systems or controllers that can be active on a unit. Only one controller can be active at any time. When a unit boots up or a command timeout occurs for the active controller the manager always sets the default controller of type “stop” as active.




## Control System Types

There are various types of control systems that can be enabled on a unit.

| Type ID | Controller type | Description |
|---|---|---|
| 0 | Stop | Controls how the wheels come to a complete stop |
| 1 | Raw motor | Sends duty cycle to the wheels |
| 2 | Tank drive | Driving is controlled by two independent wheel velocity inputs |
| 3 | Drive with yaw | Driving is controlled by yaw and linear velocity inputs |
| 4 | RC drive | Driving is controlled by angular and linear velocity inputs |
| 5 | XY position drive | X and Y position waypoints are set as targets that the unit drives towards |

| | | |
|---|---|---|
| 6 | Infrared drive | Used by infrared modules to direct driving |
| 7 | Magnetometer drive | Used by magnetometer modules to direct driving |

## Control System ID

Each controller installed on the unit has a unique control system ID. There can be multiple control systems of the same type.

| **Control System ID** | **Type ID** | **Controller Description** |
|---|---|---|
| 0 | 0 | Decelerating stop |
| 1 | 1 | Raw motor |
| 2 | 2 | Tank drive |
| 3 | 3 | Drive with yaw advanced mode (default for type) |
| 4 | 3 | Drive with yaw basic mode |
| 5 | 4 | RC drive rate mode (default for type) |
| 6 | 4 | RC drive slew mode |
| 7 | 5 | XY position drive |
| 8 | 6 | Infrared follow and evade |
| 9 | 7 | Magnetometer calibration |

Only one controller of the same type can receive commands at a time. This controller is the default of its type. The user has the ability to change the default of each type of controller. The changes made by the user last until a power cycle occurs.

| **DID** | **CID** | **Description** |
|---|---|---|
| 0x16 | 0x0E | Set default control system for type |
| 0x16 | 0x44 | Get active control system ID |
| 0x16 | 0x45 | Restore initial default control system |
| 0x16 | 0x46 | Get default control system for type |

## Normalized vs. SI units

There are two types of commands that exist that you can use to run a controller. The normalized commands have certain inputs that have been normalized between -127 and 127. The SI commands take inputs in standard units or SI units, as shown in the table below:

| **Quantity** | **Unit for SI commands** |
|---|---|
| Linear Velocity | Meters per second (m/s) |
| Angular Velocity | Degrees per second (deg/s) |
| Linear Position/Distance | Meters |
| Angular Position | Degrees |
| Linear Acceleration | Meters per second per second ($m/s^2$) |

## Maximum Velocity Limits

For SI commands, the robot will limit setpoints to stay within the bounds in the table below. Values outside of these bounds will saturate at the closest bound. For example, a target linear velocity of 4 m/s will be internally limited to 2 m/s.

| **Quantity** | Bounds |
|---|---|
| Linear Velocity (m/s) | -2..2 |
| Angular Velocity (degrees/s) | -624..624 |

Variations in operating conditions can prevent the robot from achieving the full ranges. These conditions include but are not limited to:

- Payload mass
- Payload center of gravity position
- Battery voltage
- Terrain (surface condition, surface inclination)
- Obstacles

## Command timeout

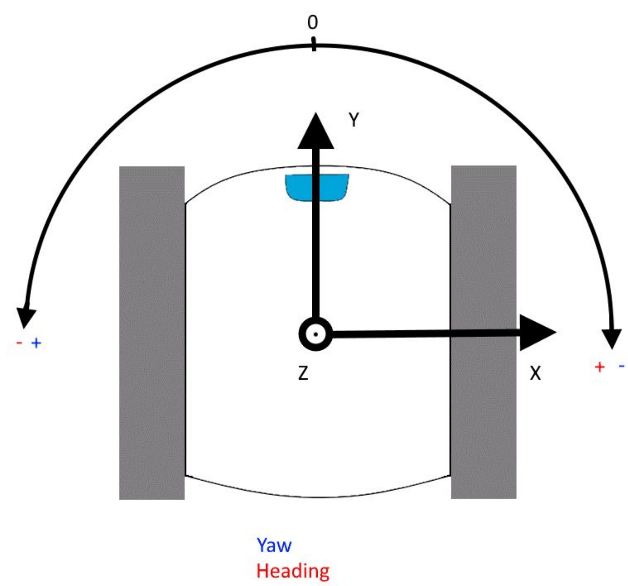
Most commands used to run a controller timeout in 2000 milliseconds by default. The timeout may be adjusted through the below API commands. A custom timeout will remain in effect until overwritten by API command, or the robot is turned off.

| DID | CID | Description |
|---|---|---|
| 0x16 | 0x22 | Set custom control system timeout |
| 0x16 | 0x43 | Restore default control system timeout |

## Right hand rule

The orientation of the IMU follows the right hand rule. In the coordinate space of the unit the X axis is to the right, the Y axis is forward and the Z axis is up. The inputs related to the unit also follow this orientation. For example with regards to yaw inputs in controller commands, 0 degrees is forward, the degrees ranges from 0 to 180 degrees to left and it ranges from 0 to -180 degrees to the right.

The only command that does not follow this model is the *drive with heading* command which uses heading as an input rather than yaw. Heading follows the left hand rule.




| DID | CID | Description |
|---|---|---|
| 0x16 | 0x07 | Drive with heading |

## Update rate

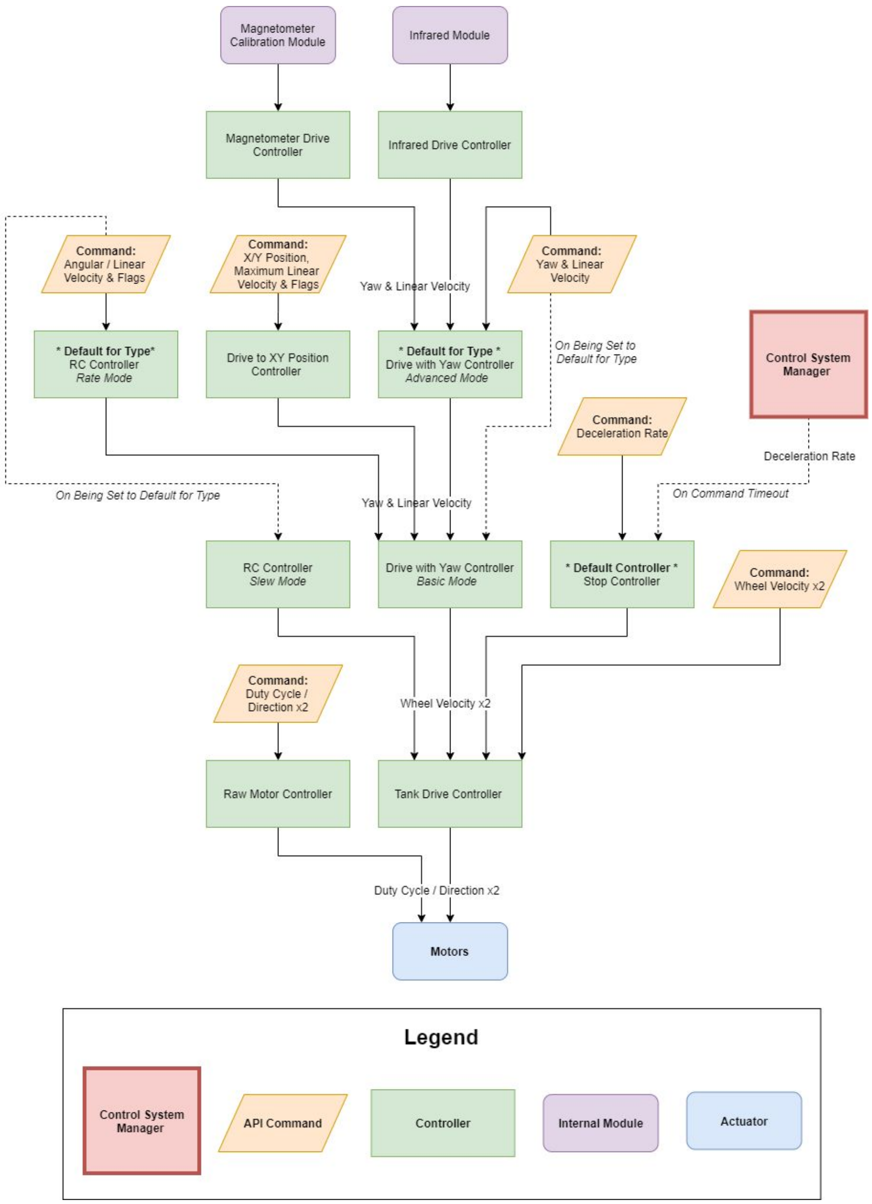
All controllers update at a rate of 400Hz except for the tank drive controller which updates at the rate of 1kHz. The 400 Hz update rate matches the IMU update frequency, as these controllers depend on knowing the unit’s orientation.

## Driving upside down

The unit can gracefully handle flipping over and still achieve the drive targets.

# Control system architecture

RVR's control system is constructed in hierarchical layers, with upper layers providing targets to layers beneath them.  In addition, each individual controller may be accessed through API interfaces specific to its own capabilities.  This arrangement is designed to maximize flexibility for a wide variety of applications.



# Raw motor

The raw motor controller does not contain any actual control system logic but was plugged into the control system manager so that one system controls access to the drive motors. The raw motor controller allows for a constant duty cycle to be applied to each drive motor.

| DID | CID | Description |
|---|---|---|
| 0x16 | 0x01 | Raw motors |

# Tank drive

The tank drive controller allows for the left and right tread velocities to be controlled independently. Two wheel velocity inputs are applied to each wheel independently.

| DID | CID | Description |
|---|---|---|
| 0x16 | 0x32 | Drive tank SI |
| 0x16 | 0x33 | Drive tank normalized |

# Drive with yaw

The drive with yaw controllers take a yaw and linear velocity input to drive the unit.

| DID | CID | Description |
|---|---|---|
| 0x16 | 0x36 | Drive with yaw SI |
| 0x16 | 0x37 | Drive with yaw normalized |

## Basic mode

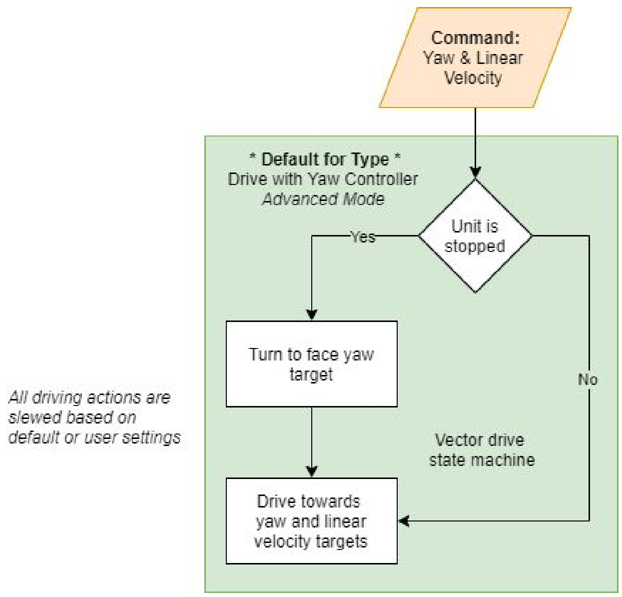
The basic mode controller takes the inputs of yaw and linear velocity and converts them to two linear velocities that get plugged into the lower level tank drive controller. No additional processing occurs on the inputs to this controller.

## Advanced mode

The advanced mode controller stacks on top of the basic mode controller. This controller also takes the inputs of yaw and linear velocity but performs additional processing on both inputs before plugging them into the basic mode controller. The outputs of this controller are processed yaw and linear velocity. The processing that is applied includes vector drive and slewing.

## Vector Drive

Depending on if the unit is at rest or driving the inputs of the controller are differently applied. If the unit is at rest, the unit will first go to the target yaw and then drive forward with the linear velocity input. If the unit is already driving, yaw and linear velocity will be applied at the same time.



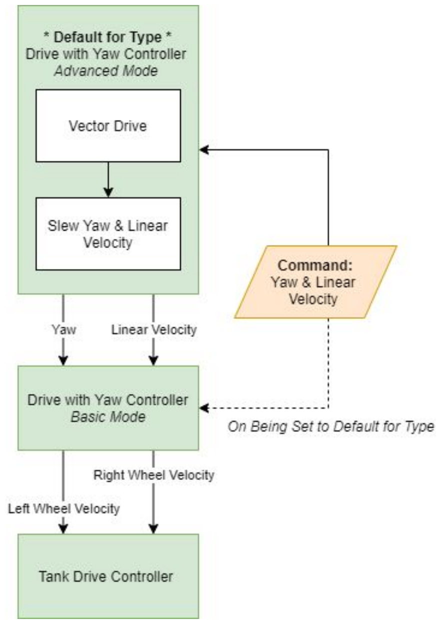

## Slewing

An algorithm is used to determine turn speed based on the current velocity of the unit. The turn speed will be different if the unit is at rest or driving at a slow velocity or driving at a fast velocity, etc. The user has the ability to adjust the parameters in the algorithm to change the turn speed characteristics at various linear velocity targets. Additional details can be found in Drive target slew parameters.

# Architecture

Both the advance and basic mode controllers are of the drive with yaw controller types. Only one of these controllers can receive commands and be set as the active controller. The *set default control system for type* command can be used to change which controller is the default for the drive with yaw controller type.

Architecturally, the advanced controller stacks on top of the basic mode controller which stacks on top of the tank drive controller. The advanced mode controller takes yaw and velocity inputs and applies processing before plugging these inputs into the basic mode controller. The basic mode controller takes the target yaw and linear velocities, whether from a command or the processed values from an advanced mode controller and derives two target wheel velocities that get plugged into the tank drive controller.




# RC drive

The RC controllers take an angular and linear velocity input to drive the unit.

| DID | CID | Description |
|---|---|---|
| 0x16 | 0x34 | Drive RC SI |
| 0x16 | 0x35 | Drive RC normalized |

## Rate mode

The rate mode controller takes the inputs of angular and linear velocity and converts them to two linear velocities that get passed to the lower level tank drive controller. The rate mode controller is very user friendly to drive with joystick type controllers, with one tied to angular velocity and another tied to linear velocity. It provides the end user with a smoother drive experience then the slew mode controller.

## Slew Mode

The slew mode controller stacks on top of the drive with yaw basic mode controller. This controller also takes the inputs of angular and linear velocity and converts them to yaw and linear velocity. The slew mode controller is ideal for programmatic control as it has lower drift error over time when compared to the rate mode controller.

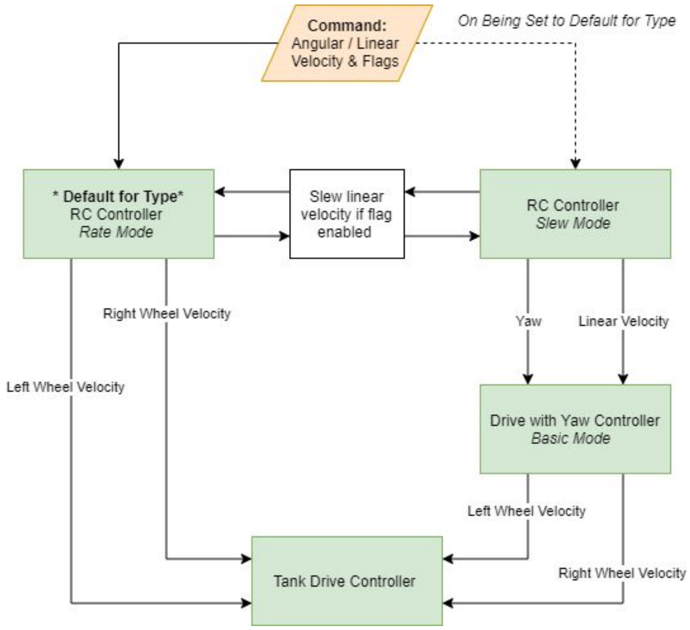
## Flag to slew linear velocity

When the *slew linear velocity* flag is enabled in either controller the linear velocity is further processed. The linear velocity will be slewed between the current and target linear velocity. Additional details can be found regarding the algorithm used in Drive target slew parameters. For example if the unit is currently at rest and is commanded to go at a linear velocity of 0.5 m/s, without the flag being set the unit will try and achieve this velocity as fast as possible and with the flag set the unit will ease into the target linear velocity, avoiding jerky movements.

## Architecture

Both the slew and rate mode controllers are of the RC drive controller types. Only one of these controllers at a time can receive commands and be set as the active controller. The *set default control system for type* command can be used to change which controller is the default for the RC drive controller type.

Architecturally, the rate mode controller stacks on top of the tank drive controller.  The inputs of angular and linear velocity are used to derive left and right tread velocity set points which are then passed to the tank drive controller.  The slew mode controller stacks on top of the drive with yaw basic mode controller. Inputs of angular and linear velocities are converted into yaw position and linear velocity set points and passed to the drive with yaw basic mode controller.




# Drive to XY position

The drive to XY position controller provides the unit with waypoints to drive to with end yaw along with additional configuration settings.

| DID | CID | Description |
|---|---|---|
| 0x16 | 0x38 | Drive to XY position SI |
| 0x16 | 0x39 | Drive to XY position normalized |
| 0x16 | 0x3A | Reached target XY position async |

## Turning in place

This controller stacks on top of the Drive with Yaw Basic Mode but includes the vector drive and yaw slewing features of the Drive with Yaw Advanced Mode. On starting at reset, the unit will turn towards the waypoint before starting to drive in that direction. On reaching the waypoint the unit will use the same slewing algorithm to turn to face the end yaw.

## Reverse modes

This controller provides a lot of flexibility with regards to reverse behavior when driving to a waypoint through command flags. Bits 0 and 1 are a two-bit field only used for driving direction settings.

| Value | Description |
|---|---|
| 0 | Default: drive forward |
| 1 | Drive reverse, robot drives backwards |
| 2 | Auto reverse, drive the robot in the direction that requires the least amount of turning |
| 3 | Reserved, command will return error if this value is set |

## Drive to relative position

By setting bit 2 in the flags, the robot drives to the waypoint value relative to its current position. In this scenario the robot's current position is considered to be (0, 0).

## Relationship with the IMU and Locator modules

The way points that the unit drives to are in the same space as the locator and the end yaw that the unit points to is in the same space as the yaw reported by the IMU. Using reset commands associated with either of these modules results in driving and end heading behavior as expected of the reported sensor data.

| DID | CID | Description |
|---|---|---|
| 0x18 | 0x13 | Reset locator X and Y |
| 0x16 | 0x06 | Reset Yaw |

# Stopping

The stop controller decelerates the unit until it comes to a complete stop. The default maximum deceleration is 2 $m/s^2$. This controller stacks on top of the tank drive controller architecturally.

| DID | CID | Description |
|---|---|---|
| 0x16 | 0x3E | Stop with custom deceleration |
| 0x16 | 0x42 | Stop with default deceleration |
| 0x16 | 0x3F | Active controller has stopped async |
| 0x16 | 0x41 | Get stop controller state |

# Other controllers

The magnetometer calibration and infrared follow and evade codes stack on top of the drive with yaw advanced mode code. Features associated with these controllers can be accessed through commands associated with these modules.

| DID | CID | Description |
|---|---|---|
| 0x18 | 0x25 | Magnetometer calibrate to North |
| 0x18 | 0x51 | Magnetometer calibration complete async |
| 0x18 | 0x28 | Start robot to robot infrared following |
| 0x18 | 0x32 | Stop robot to robot infrared following |
| 0x18 | 0x33 | Start evading |
| 0x18 | 0x34 | Stop evading |

# Drive target slewing

The drive target slew algorithm controls the turning behavior of the unit. This slew algorithm affects multiple controllers, including drive with yaw advanced mode, both rc drive controllers, and drive to XY position.

| DID | CID | Description |
|---|---|---|
| 0x16 | 0x3C | Set drive target slew parameters |
| 0x16 | 0x3D | Get drive target slew parameters |
| 0x16 | 0x40 | Restore default drive target slew parameters |

The table below shows which target parameters are slewed for different controllers.

| Controller | Slewed Targets |
|---|---|
| Drive with yaw advanced mode | Yaw, Linear Velocity |
| RC drive slew mode | Linear Velocity |
| RC drive rate mode  you | Linear Velocity |

| | |
|---|---|
| Drive to XY position | Yaw (during spin states) |

# Linear velocity slewing

Linear velocity target slewing has 2 selectable modes, constant acceleration and proportional acceleration. These are selectable via the *Linear Velocity Slew Method* parameter.

## Constant acceleration

Constant acceleration mode simply applies a fixed acceleration limit to the robot's linear velocity target. Note that this limit does NOT apply to individual tread velocity targets, which may change as quickly as upstream control systems or user commands request. The robot's behavior in this mode is simple and easily predictable.

Until the internal target linear velocity is equal to the user-specified target linear velocity, the internal target will be slewed at the configured acceleration rate:
$TargetVelocity[t] = TargetVelocity[t - \Delta t] \pm Acceleration * \Delta t$

## Proportional acceleration

The proportional acceleration mode is designed to provide a more fun and responsive driving feel than constant acceleration mode. At zero linear velocity, the target velocity slews at the configured linear acceleration rate. As the velocity error decreases, so does the acceleration. The configured linear acceleration rate is always used under braking conditions.

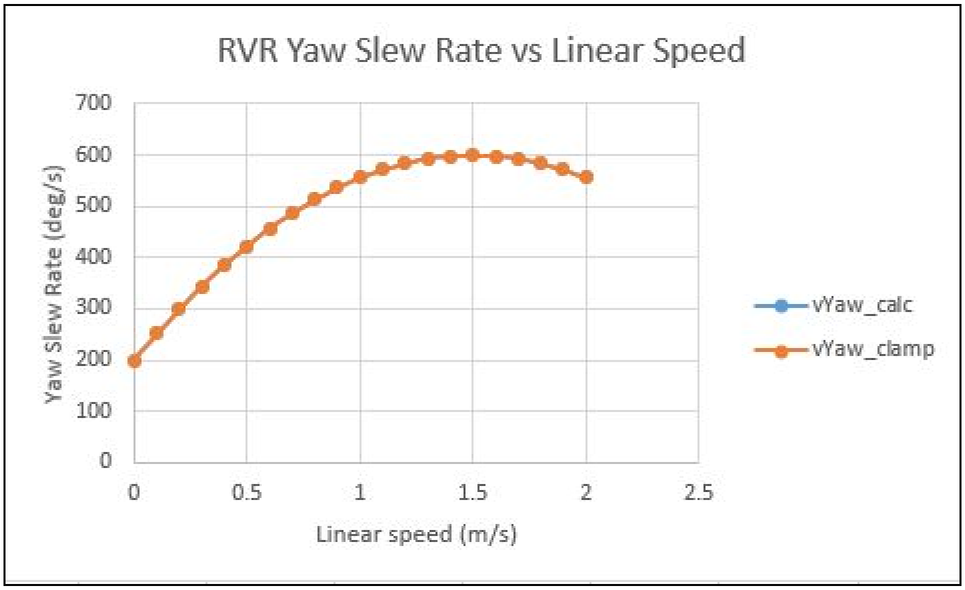
# Yaw target slewing

Yaw targets employ a variable slew rate, dependent on the current linear velocity of the robot. This follows the following formula:

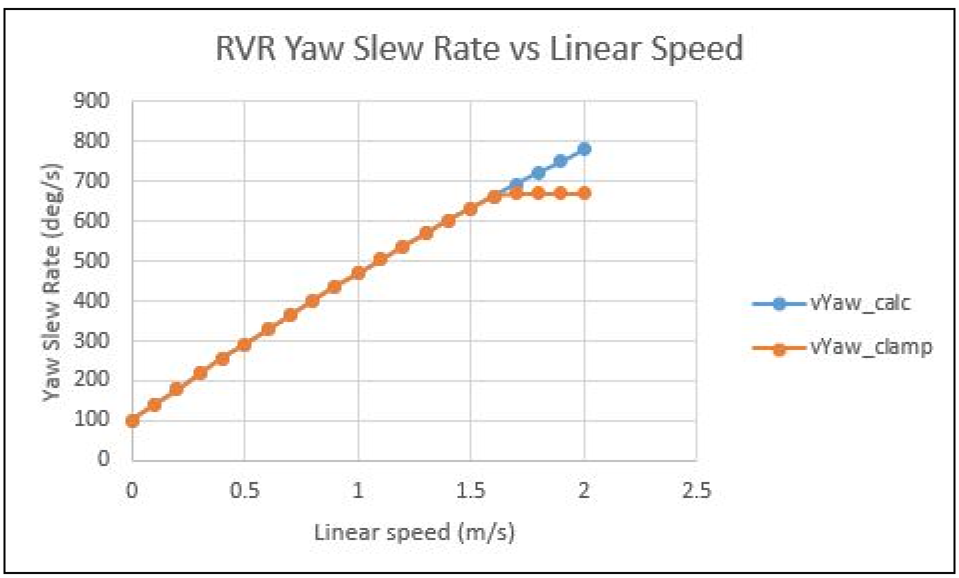
Let $\frac{dy}{dt}$ be the maximum absolute yaw slew rate, and let $v$ be the current linear velocity. The yaw slew rate follows this equation:

$$\frac{dy}{dt} = av^2 + bv + c$$

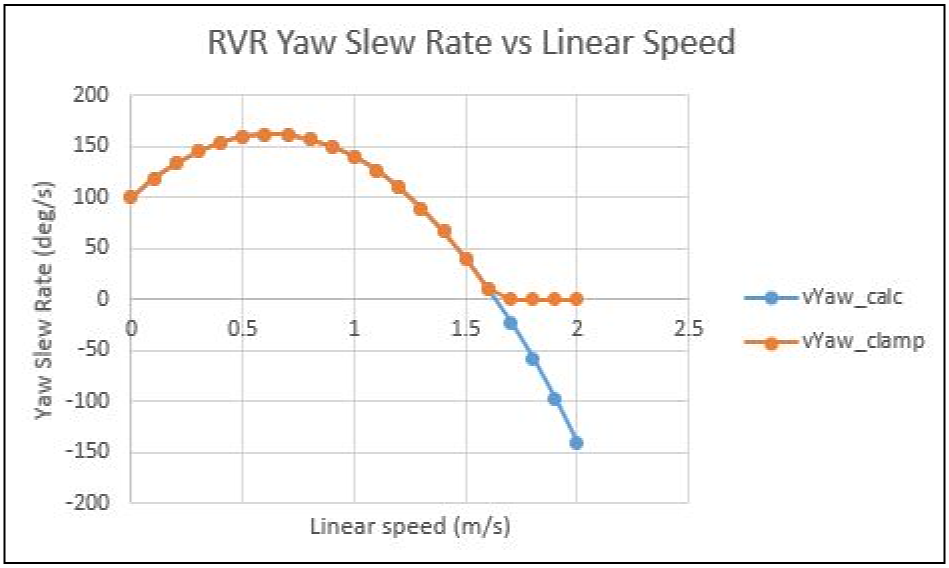
The default curve is plotted below, with {a,b,c} = {-177.78,533.33,200}




The coefficients $a,b,c$ are user-adjustable, and allow for a parabolic, linear, or constant relationship between the maximum absolute yaw slew rate and the linear velocity. Unachievable rotation rates will be clamped to keep them in range of the robot's capabilities. See an example in the next plot, {-30, 400, 100}:




Parameters that produce a maximum yaw slew rate of zero will prevent turning for affected operating points. In the following example, the robot will not turn after it exceeds ~1.6 m/s. Parameters are {-160, 200, 100}:




Here’s an example of a constant yaw slew rate with no speed dependence {0, 0, 60}: