Swerve Drive Second Order Kinematics

FRC Team 449 - The Blair Robot Project
Rafi Pedersen, Mentor and Class of 2019 Alum
November 2022

1 Introduction

As swerve drives become more prevalent in FRC, more advanced control schemes may become helpful to improve robot control. The most basic part of the swerve control system is its kinematics - that is, the equations relating the states of the individual modules to the state of the whole robot. The current kinematics were detailed back in 2011 by Ether and do not seem to have changed since. They provide the relationship between the robot's x, y, and angular velocities v_x, v_y, and ω to the n modules' linear speeds $\langle v_1, v_2, ...v_n\rangle$ and headings $\langle \theta_1, \theta_2 ...\theta_n\rangle$. This will be referred to as the first order kinematics.

Unfortunately, the first order kinematics are limited in the way that they do not solve for the modules' linear accelerations and angular velocities that may be needed to perform a maneuver. With a traditional tank drive, making a constant maneuver (e.g. drive straight while turning with constant linear and angular velocities) only requires constant wheel velocities, so the desired path trajectory can be achieved only using first order kinematics. In contrast, consider a swerve drive that is driving straight while turning with constant velocities - while the overall robot has constant velocities, each individual module's linear velocity and heading are constantly changing. This means that in practice, a swerve drive performing this maneuver using the first order kinematics will skew in the direction of rotation[1].

2 Derivation

The kinematics assume that the relationship between the robot state and the one module's state is independent from the state of all of the other modules. We use the reference frame of the field, so the x and y direction are relative to the field, not the robot, and do not change as the robot moves and rotates. In this derivation, we will focus on just one module. This module is positioned at $\vec{r} = \langle r_x, r_y, 0\rangle$ relative to the center of rotation. When the robot is moving with linear velocity $\vec{v} = \langle v_x, v_y, 0\rangle$ and angular velocity $\vec{\omega} = \langle 0, 0, \omega\rangle$, we know the module is moving with velocity

$$\vec{v}_m = \vec{v} + \vec{\omega} \times \vec{r} \tag{1}$$

This equation can be written in matrix form as

$$\begin{bmatrix} v_{mx} \\ v_{my} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -r_y \\ 0 & 1 & r_x \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{2}$$

[1]This skew is also caused by the response delay between when the controller commands a state and when the modules achieve that state, but resolving this aspect is outside the scope of this whitepaper.

In order to find the rate of change of the module's velocity vector, we take the first derivative and applying the chain rule and find

$$\dot{\vec{v}}_m = \dot{\vec{v}} + \dot{\vec{\omega}} \times \vec{r} + \vec{\omega} \times \dot{\vec{r}} \tag{3}$$

Since $\dot{\vec{r}} = \vec{v}_m - \vec{v} = \vec{\omega} \times \vec{r}$, we know

$$\begin{aligned}
\vec{\omega} \times \dot{\vec{r}} &= \vec{\omega} \times (\vec{\omega} \times \vec{r}) \\
\vec{\omega} \times \dot{\vec{r}} &= \vec{\omega}(\vec{r} \cdot \vec{\omega}) - \vec{r}(\vec{\omega} \cdot \vec{\omega}) \\
\vec{\omega} \times \dot{\vec{r}} &= -\omega^2 \vec{r}
\end{aligned}$$

Equation 3 now simplifies to

$$\begin{aligned}
\dot{\vec{v}}_m &= \dot{\vec{v}} + \dot{\vec{\omega}} \times \vec{r} - \omega^2 \vec{r} \\
\begin{bmatrix} a_{mx} \\ a_{my} \\ 0 \end{bmatrix} &= \begin{bmatrix} a_x \\ a_y \\ 0 \end{bmatrix} + \begin{bmatrix} -\alpha r_y \\ \alpha r_x \\ 0 \end{bmatrix} - \begin{bmatrix} \omega^2 r_x \\ \omega^2 r_y \\ 0 \end{bmatrix} \\
\begin{bmatrix} a_{mx} \\ a_{my} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & -r_x & -r_y \\ 0 & 1 & -r_y & r_x \end{bmatrix} \begin{bmatrix} a_x \\ a_y \\ \omega^2 \\ \alpha \end{bmatrix}
\end{aligned} \tag{4}$$

Where a_{mx} and a_{my} are the module's linear acceleration components, a_x and a_y are the robot's linear acceleration components, and α is the robot's angular acceleration. Note how when $\vec{v}$ and $\vec{\omega}$ are constant (so $a_x = a_y = \alpha = 0$), the equation evaluates to $\vec{a_m} = -\omega^2 \vec{r}$, which is the known centripetal acceleration formula. Now that we can find v_{mx} from the first-order kinematics in Equation 2 and a_{mx} from the second order kinematics in Equation 4, we just need to convert these values to be in terms of linear velocity v_m and heading θ_m. Note that here, θ_m is measured relative to the field x-axis, not the robot x-axis.

We know

$$v_m = |\vec{v}_m| = \sqrt{v_{mx}^2 + v_{my}^2} \tag{5}$$

$$\theta_m = \operatorname{atan2}(v_{my}, v_{mx}) \tag{6}$$

Dobule check your programming language's ordering for atan2 before implementing Equation 6. We can differentiate Equation 5 (remembering to apply the chain rule) to find

$$\begin{aligned}
\dot{v}_m &= \frac{d}{dt}\sqrt{v_{mx}^2 + v_{my}^2} \\
\dot{v}_m &= \frac{\partial}{\partial v_{mx}}(\sqrt{v_{mx}^2 + v_{my}^2})\frac{dv_{mx}}{dt} + \frac{\partial}{\partial v_{my}}(\sqrt{v_{mx}^2 + v_{my}^2})\frac{dv_{my}}{dt} \\
\dot{v}_m &= \frac{1}{2}(v_{mx}^2 + v_{my}^2)^{-1/2}(2v_{mx})\dot{v}_{mx} + \frac{1}{2}(v_{mx}^2 + v_{my}^2)^{-1/2}(2v_{my})\dot{v}_{my} \\
a_m &= \frac{v_{mx}\dot{v}_{mx} + v_{my}\dot{v}_{my}}{\sqrt{v_{mx}^2 + v_{my}^2}} \\
a_m &= \frac{v_{mx}}{v_m}a_{mx} + \frac{v_{my}}{v_m}a_{my} = \cos(\theta_m)a_{mx} + \sin(\theta_m)a_{my}
\end{aligned} \tag{7}$$

Next, we differentiate Equation 6. Since $\frac{d}{dt}$atan2(y,x) $= \frac{d}{dt}\arctan(y/x)$

$$
\begin{aligned}
\dot{\theta}_m &= \frac{d}{dt}\arctan(v_{my}/v_{mx}) \\
\dot{\theta}_m &= \frac{\partial}{\partial v_{mx}}(\arctan(v_{my}/v_{mx}))\frac{dv_{mx}}{dt} + \frac{\partial}{\partial v_{my}}(\arctan(v_{my}/v_{mx}))\frac{dv_{my}}{dt} \\
\omega_m &= (\frac{1}{1+(v_{my}/v_{mx})^2})(-\frac{v_{my}}{v_{mx}^2})a_{mx} + (\frac{1}{1+(v_{my}/v_{mx})^2})(\frac{1}{v_{mx}})a_{my} \\
\omega_m &= -\frac{v_{my}}{v_{mx}^2+v_{my}^2}a_{mx} + \frac{v_{mx}}{v_{mx}^2+v_{my}^2}a_{my} \\
\omega_m &= -\frac{v_{my}}{v_m^2}a_{mx} + \frac{v_{mx}}{v_m^2}a_{my} = -\frac{\sin(\theta_m)}{v_m}a_{mx} + \frac{\cos(\theta_m)}{v_m}a_{my} \qquad (8)
\end{aligned}
$$

Equation 7 and Equation 8 can be combined in the following matrix forms

$$
\begin{bmatrix} a_m \\ v_m\omega_m \end{bmatrix} = \begin{bmatrix} \cos(\theta_m) & \sin(\theta_m) \\ -\sin(\theta_m) & \cos(\theta_m) \end{bmatrix} \begin{bmatrix} a_{mx} \\ a_{my} \end{bmatrix}
$$

$$
\begin{bmatrix} a_{mx} \\ a_{my} \end{bmatrix} = \begin{bmatrix} \cos(\theta_m) & -\sin(\theta_m) \\ \sin(\theta_m) & \cos(\theta_m) \end{bmatrix} \begin{bmatrix} a_m \\ v_m\omega_m \end{bmatrix}
$$

We have now solved for the module's instantaneous linear acceleration and heading angular velocity needed to help keep the robot tracking correctly, alongside the already known first-order kinematics' linear velocity and heading angle. Simply command v_m, θ_m, a_m, and ω_m to the swerve modules.

Since the modules are controlled relative to the robot's orientation, the final step is to convert these values to be relative to the robot. The magnitudes of the velocity and acceleration are unaffected, but the module heading angle and angular velocity need to be offset by the robot's orientation and angular velocity, respectively, before assigning the values to the modules, so $\theta_{m,robot} = \theta_m - \theta_{robot}$ and $\omega_{m,robot} = \omega_m - \omega_{robot}$.

Finally, you may find it helpful to perform these calculations in the robot reference frame instead of the field reference frame - this can be done with no issue, simply multiply or divide by the robot's rotation matrix as appropriate. Just make sure that everything is consistent - for example, don't mix $\vec{r}$ measured in the robot-frame with $\vec{v}$ measured in the field frame.

These second order kinematics will not solve all issues that can lead to skewing - they compensate for the time gap between sequential commands from the controller, but they do not fully compensate for the delay between when a state is commanded and when the swerve modules actually achieve that state, so some form of closed loop control using the robot's desired pose and its current pose may still be necessary.