REV A 2/24/2018

# Team 3176 White Paper

| | |
|---|---|
| Title: Inverse Kinematics of a Skid Steer Robot | Author(s): Jonathan Heidegger, Julian Lewis, Jackson Fellers |
| Subteam/Function: Programming | Subject: Controls |
| Date/Season: 17/18 | White Paper Number: 1 |

## Abstract

This paper documents the process and proof for the inverse kinematics of a skid steer robot. The paper is oriented at people who have an understanding of geometry and algebra, calculus is mentioned but the proof is based in geometry. The goal of the algorithm is to find the needed wheel commands for a robot to drive it from a starting point to any point (a,b) in quadrants 1 and 2 of robot coordinates.

## Definitions

Skid steer robot - Refers to a robot that commands two series of wheels parallel to one another where turning is accomplished by different wheel speeds. Often referred to as “tank drive”.

## Coordinate Systems

Field coordinates - points in the form of (x,y) relative to a static origin point.

Robot coordinates - points in the form (a,b) in which all points are relative the robots position and rotation. Used for the inverse kinematics algorithm.

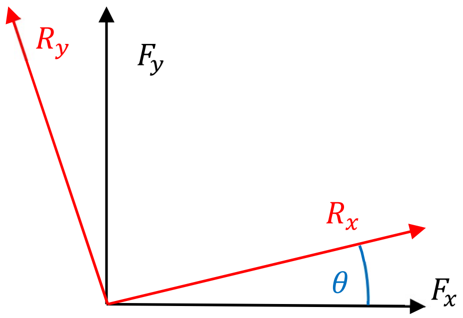
The base of the algorithm is in coordinates and coordinate system transfers. This is the process of translating between field coordinates and robot coordinates. This can be accomplished using a transform matrix as follows

$$T_{gr} = \begin{bmatrix} R_x & \cos\theta & \sin\theta \\ R_y & -\sin\theta & \cos\theta \\ 0 & 0 & 1 \end{bmatrix}$$

The coordinate transform is a matrix multiplication of the field coordinates by the transform $T_{gr}$ to result in the coordinates in robot coordinates. The operation is as follows

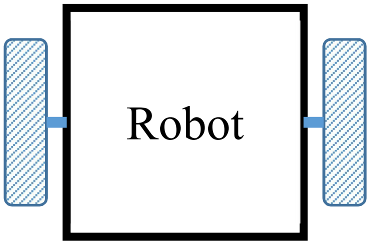
$$\begin{bmatrix} a \\ b \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} T_{gr}$$

This allows for any point (x,y) to be transformed into robot coordinates if the robots position and rotation is known. Note: Theta is the rotation of the robot coordinates from the field X axis counter clockwise.

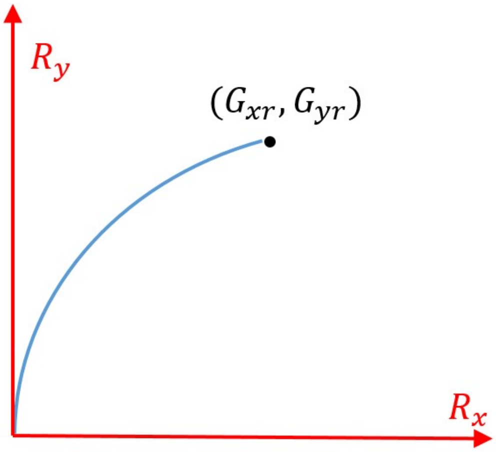


## Inverse Kinematics Theory

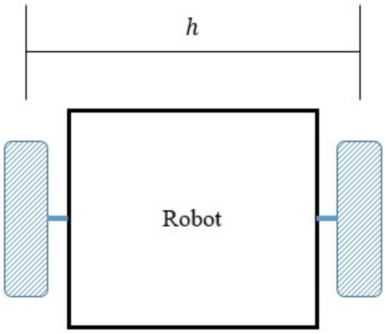
The skid steer robot is modeled as a robot with only two wheels, one on each side. This model ignores friction caused by robots having multiple wheels on each side. The more wheels on each side the less accurate the model becomes. See Limits for more information

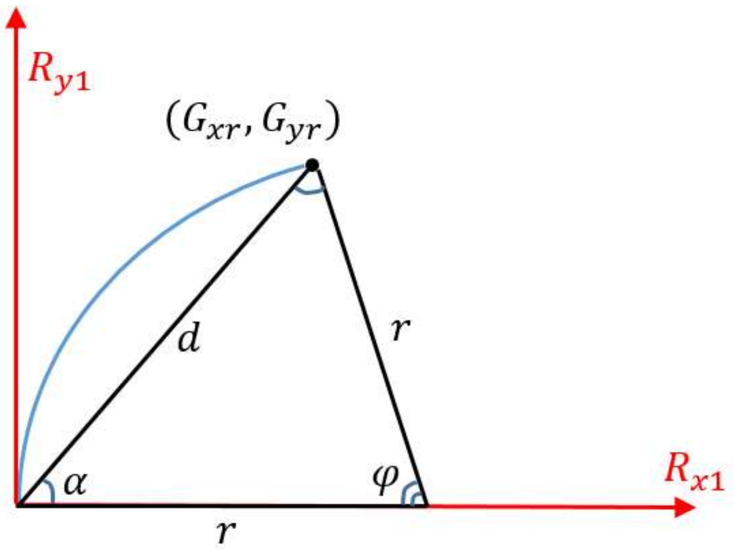


The Inverse Kinematics finds an arc that connect the robot from its current position to the goal position and calculates the needed wheel distances to achieve this curve. An example is below.



The following variables will be used throughout to calculate the wheel commands for the inverse kinematics model.





**Inverse Kinematics Equations**

First the distance $d$ is found with the distance formula which in robot coordinates is just a modified Pythagorean Theorem.

$$d = \sqrt{{G_{xr}}^2 + {G_{yr}}^2}$$

The angle $\alpha$ is calculated using the right triangle with $G_{xr}$ and $G_{yr}$. Using the inverse tangent restricts the domain to be between $\pm\pi/2$ . This means that the goal point is restricted to the first quadrant. If $G_{xr} < 0$ then $|G_{xr}|$ is used in its place, which will be corrected later by switching the wheel commands.

$$\alpha = \tan^{-1}\frac{G_{yr}}{G_{xr}}$$

Then $r$ can be calculated using the Law of Sines.

$$\frac{\sin A}{a} = \frac{\sin B}{b}$$

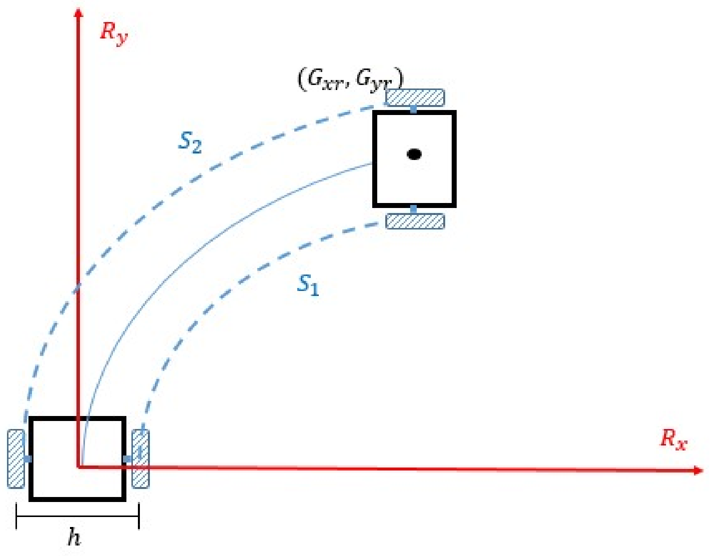
$$\frac{\sin\alpha}{r} = \frac{\sin\varphi}{d}$$

$$r = \frac{d\sin\alpha}{\sin\varphi}$$

Once the angle is known the arc distance of each wheel can be found. The inside wheel is called $S_1$ and the outside is called $S_2$. These distances can be calculated by multiplying the angle $\varphi$ by the effective radius of each arc.

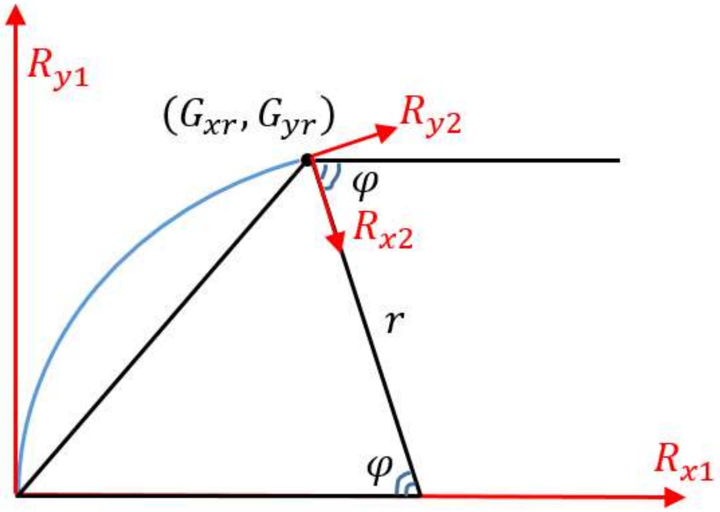
$$S_1 = \varphi(r - \frac{h}{2})$$

$$S_2 = \varphi(r + \frac{h}{2})$$



If $G_{xr} < 0$ then $S_1$ and $S_2$ will be switched so that the curve is into the 2nd quadrant. This is to correct for the restricted domain of the inverse tangent function used to find $\alpha$

The angle change of the robot is given as a the negative $\varphi$. The angle is congruent and is in the clockwise direction so the change in theta is negative. However if $G_{xr} < 0$ then the value is in the counter clockwise direction and therefore positive $\varphi$



The curvature of the path is determined by the ratio between the two distances. Velocity commands can be determined using any two velocities that match the ratio of the distances. This algorithms output is the two distances $S_1$ and $S_2$

## Limits of the Algorithm

This algorithms main error is generated from the friction of the wheels dragging while turning. This means that if the output of this is run on an open loop system the system will understeer and not reach the goal point. The output of this algorithm in conjunction with closed loop control of the wheels will lead to a higher accuracy. To reduce the wheel base the robot should be limited. Designs such as drop center or omni wheels can help reduce the effect. Another option if using a high friction system such as tank treads is to add a correction term that modifies the distance values such that $S_1$ is smaller and $S_2$ is larger.