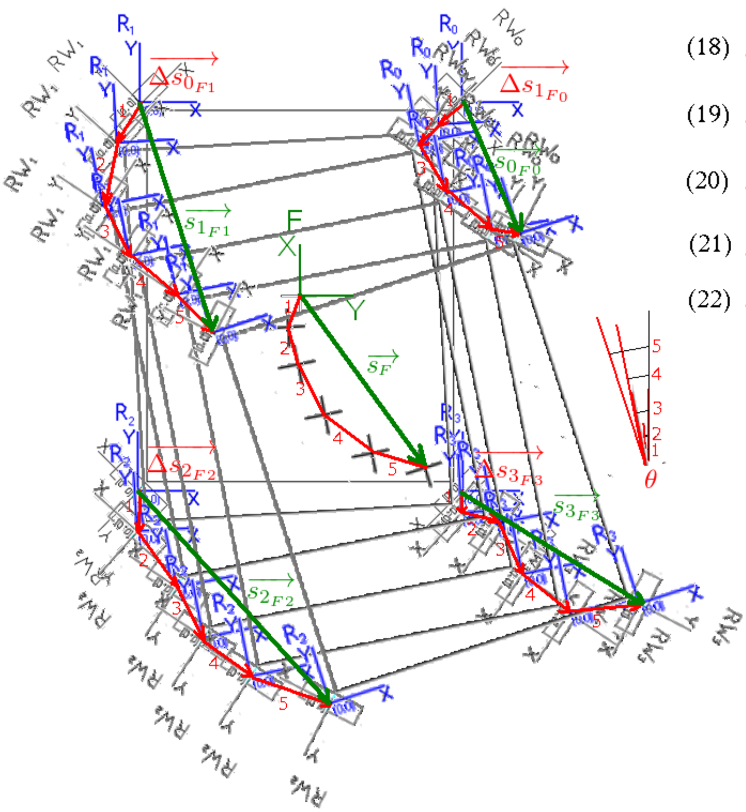
Gear Mashers' Holonomic Drive Navigation and Control



$$(18)\quad \overrightarrow{s0_{F0}} = \sum_{i=1}^{n} \overrightarrow{\Delta s0_{F0}}(i)$$

$$(19)\quad \overrightarrow{s1_{F1}} = \sum_{i=1}^{n} \overrightarrow{\Delta s1_{F1}}(i)$$

$$(20)\quad \overrightarrow{s2_{F2}} = \sum_{i=1}^{n} \overrightarrow{\Delta s2_{F2}}(i)$$

$$(21)\quad \overrightarrow{s3_{F3}} = \sum_{i=1}^{n} \overrightarrow{\Delta s3_{F3}}(i)$$

$$(22)\quad \overrightarrow{s_F} = (\overrightarrow{s0_{F0}} + \overrightarrow{s1_{F1}} + \overrightarrow{s2_{F2}} + \overrightarrow{s3_{F3}})/4$$

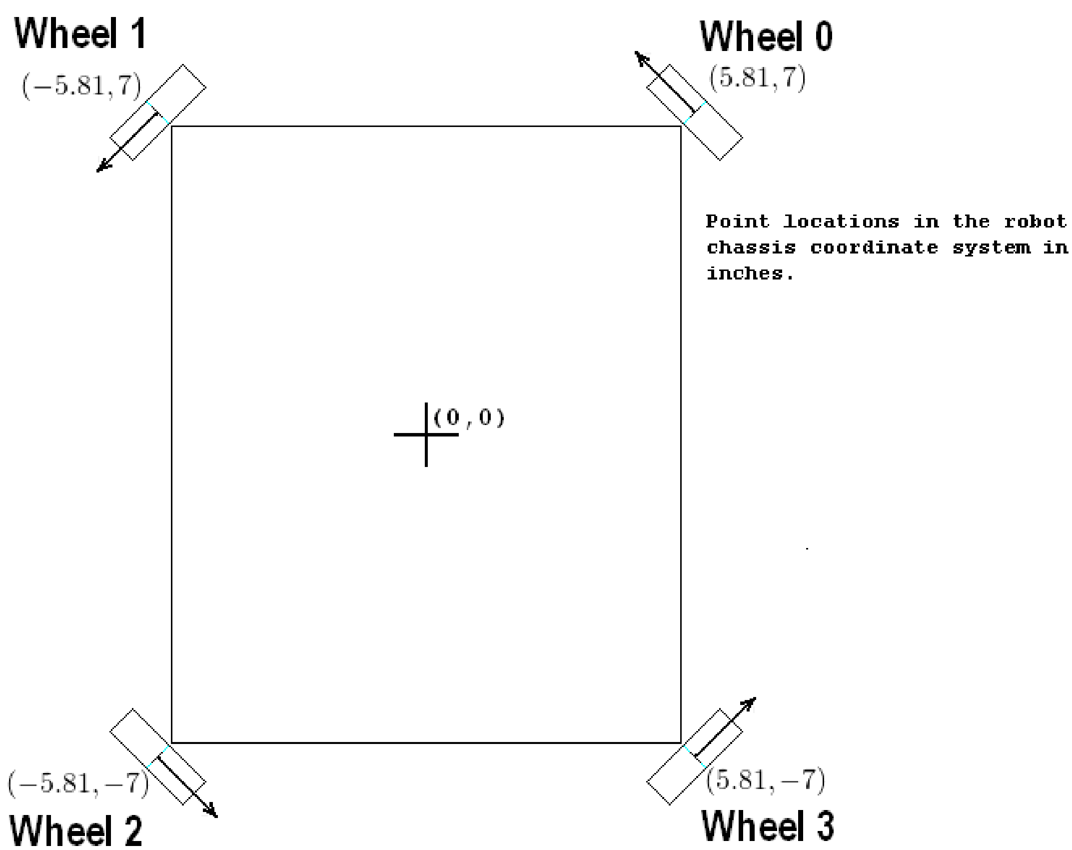
By: Kevin Van Sickle and The Gear Mashers

1. Introduction:

The Gear Mashers' holonomic drive consists of four "VEX 2-Wire Motor 393" [ref 1] gear motors, each equipped with a "4inch Omni-Directional Wheel" [ref 2] and an "Integrated Encoder Module" [ref 3]. The gear motors are mounted at the extreme corners of the robot chassis with the Omni-wheels at a 45 degree angle.

The source code is configured to assign the positive wheel direction to rotate in the motor in the clockwise direction. We also ensure encoder variable is mapped with the positive direction as clockwise.

The program has a main loop, which is regulated by a real time clock to twenty five milliseconds per loop. The encoders are read and cleared on each loop. This differential value is the amount the corresponding wheel rotates during one loop of the program. We represent this number with the variables "s_0", "s_1", "s_2", and "s_3" representing wheel-0, wheel-1, wheel-2, wheel-3 respectively.



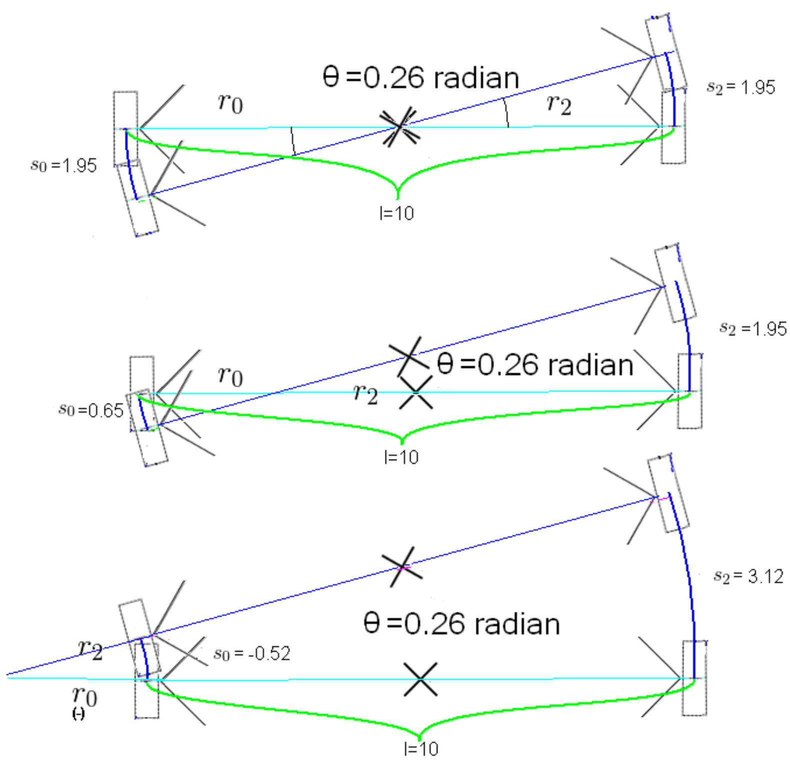
2. Angle Tracking:(θ)

(Please examine the links in Reference section. This stuff will make better sense)

Using these values captured from the encoders, we can calculate the change in rotation of the robot chassis during one loop of the program (25mS). The angle can be tracked, in the absence of wheel slip, by the difference in motion between "s_0" and "s_2", or the difference between "s_1" and "s_3". With algebra and the arc length formula [ref 4] , we determine a formula for the angle of the robot chassis from the motion of opposing wheels (s_0 and s_2, or s_1 and s_3). See formula (1) below.

Examining the arc length formula, we find there are three variables. The variables are, theta for Angle (θ), "r" for radius (r_0, r_1, r_2, r_3), and "s" for arc length (s_0, s_1, s_2, s_3). The function as it stands appears to be non-linear due to the product of the variables "1/r" and "s". ([ref 5] Linearity is a very important concept in mathematics. Linear problems are typically solvable problems.) In the diagram below, the system is shown in several conditions with different values for "r_0" and "r_2". Notice how "r_0" decreases by the same quantity that "r_2" increases. The length between the wheels "ℓ" is a constant with $\ell = r_0 + r_2$. Using algebra, we use substitution [ref 6] to replace "r_0" and "r_2" with their equivalent from the arc length formula ($\ell = s_0/\theta + s_2/\theta$). With another algebraic operation we "factor"[ref 7] out "1/θ". Next we multiply both sides of the equation by "θ" and "$1/\ell$". Now, since "$1/\ell$" is a constant, formula (1) below is a linear formula for angle "θ" in terms of two arc lengths for opposite wheels "s_0" and "s_2".

θ=15°=0.26 radian



$$\theta = \frac{ArcLength}{r}$$

$$\theta = \frac{s_0}{r_0} = \frac{s_2}{r_2}$$

$$l = r_0 + r_2$$

$$r_0 = \frac{s_0}{\theta}$$

$$r_2 = \frac{s_2}{\theta}$$

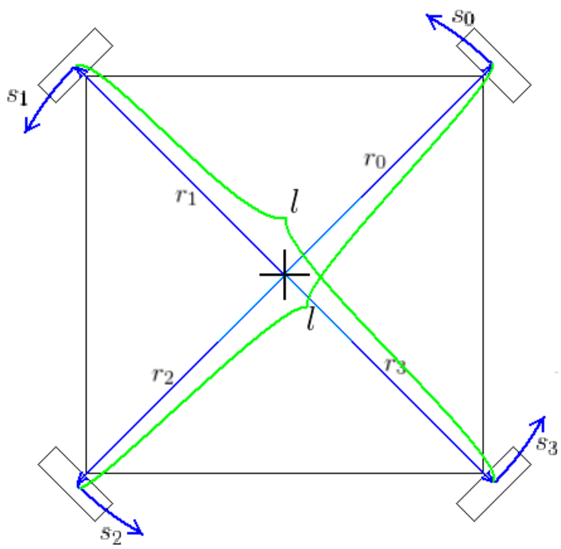
$$l = \frac{s_0}{\theta} + \frac{s_2}{\theta}$$

$$l = \frac{1}{\theta}(s_0 + s_2)$$

$$\boxed{\theta = \frac{1}{l}(s_0 + s_2)}$$

(1)

Our robot has an encoder on each wheel, so we combine formula (1) for wheels S_0 and S_2, with the same formula for wheels S_1 and S_3 giving us formula (2) below. The effect of this is an average of the measurement of the two wheel pairs. Finally, since the robot is not a true square and there may be imperfection in the placement of the wheels, it is most practical to determine a calibration constant "K" experimentally. Formula (2) below tracks the angle of the robot.



$$ArcLength = \theta r$$

$$\theta = \frac{ArcLength}{r} = \frac{s_0}{r_0} = \frac{s_1}{r_1} = \frac{s_2}{r_2} = \frac{s_3}{r_3}$$

$$l = r_0 + r_2 = r_1 + r_3$$

$$\theta = \frac{1}{l}(s_0 + s_2) = \frac{1}{l}(s_1 + s_3)$$

$$2\theta = \frac{1}{l}(s_0 + s_2) + \frac{1}{l}(s_1 + s_3)$$

$$\theta = \frac{1}{2l}(s_0 + s_2 + s_1 + s_3)$$

$$K = CalibrationFactor$$

$$\boxed{\theta = K(s_0 + s_2 + s_1 + s_3)}$$

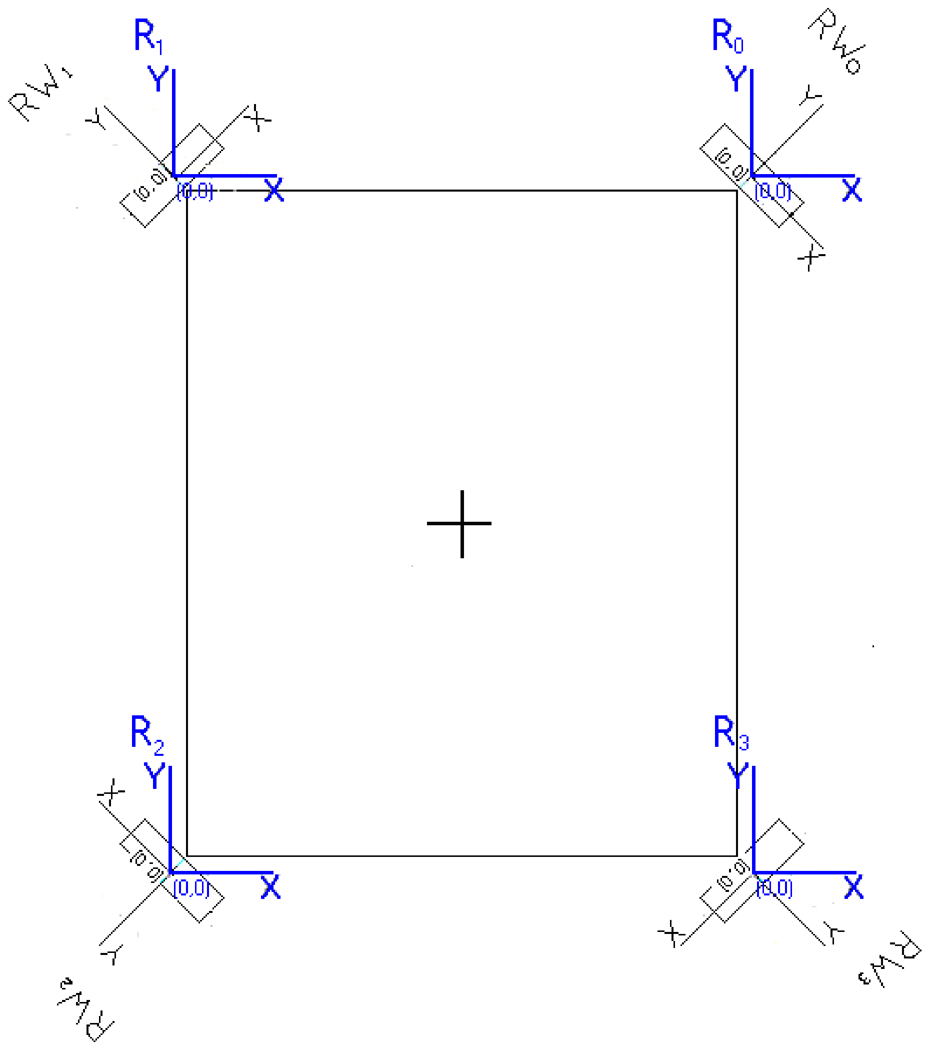
(2)

3. Position Tracking:($\vec{S}_F$)

To track the movement of the robot on the floor, we need to introduce several Cartesian coordinate systems [ref 8]. This is a means to keep the formulas of motion simple. It is easy to translate the resulting vectors from one coordinate system to another with linear transforms [ref 9] and vector arithmetic [ref 10]. These coordinate systems rotate with the robot. That is, as "θ" from formula (2) above changes, there is no effect on the coordinate systems "R_0", "R_1", "R_2", "R_3", "RW_0", "RW_1", "RW_2", or "RW_3".

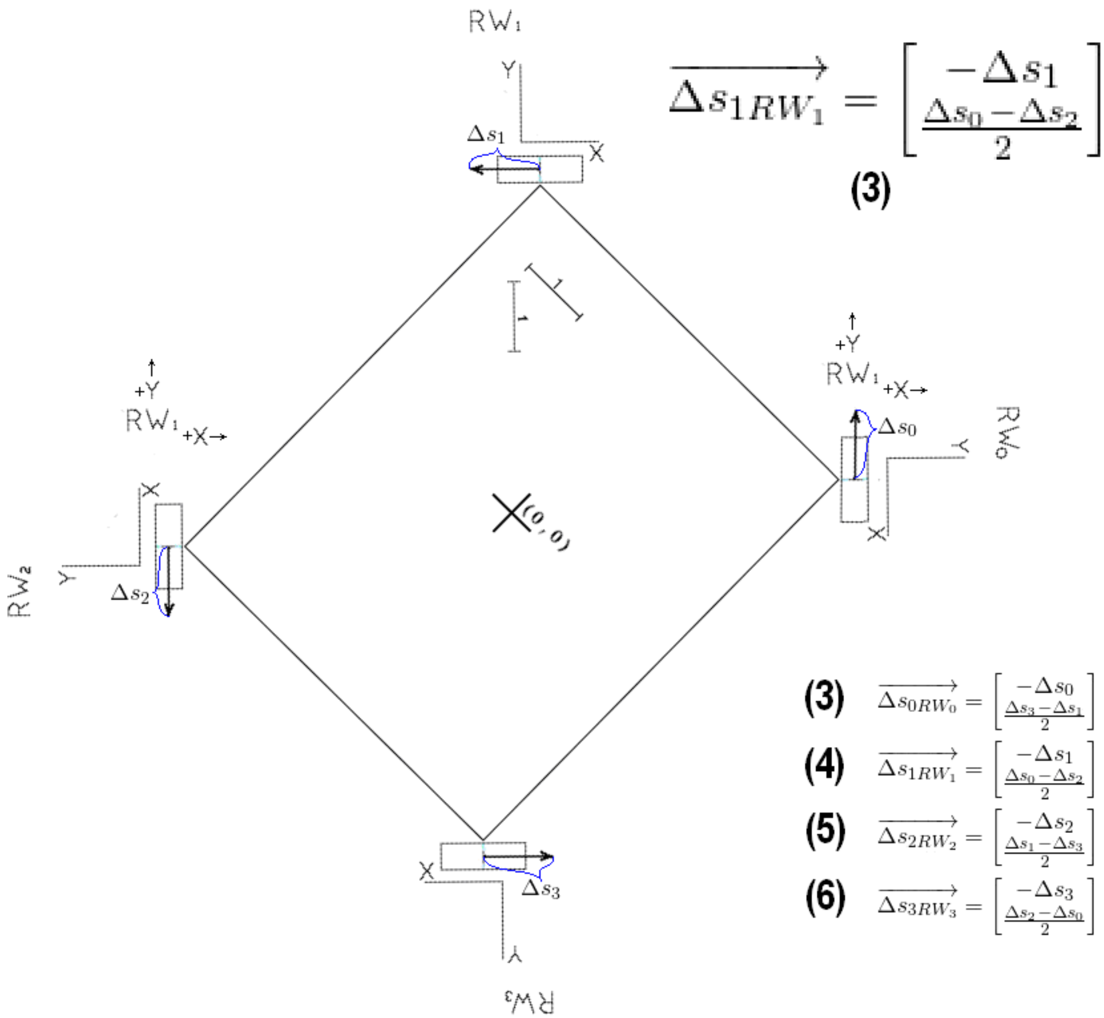
We use the symbols "R_0", "R_1", "R_2", and "R_3" to indicate the robot chassis coordinate system with +Y in the direction of the claw. "R_0", "R_1", "R_2", and "R_3" each have their origin at the center of their respective wheel.

In addition to the "$R_{wheel\#}$" coordinate systems that align with the robot chassis, each wheel has a coordinate system "$RW_{wheel\#}$". Each "$RW_{wheel\#}$" has (0,0) at the center of where the wheel touches the floor and +Y perpendicular to the wheel and away from the robot chassis. These coordinate systems are shown on the diagram below.



We use vectors to track the relative motion of the wheels. The vectors used are column vectors. The top element is the X value or the vector and the bottom element is the Y value. Column vectors are preferred because they are compatible with **matrix multiplication** [ref 11]. The Delta "Δ" in the formulas below represents "change in", and is used to indicate relative motion of the wheel during the 25 millisecond (mS) program loop. A millisecond is 1/1000 of a second.

Let's examine the motion vector for wheel 1. Omni-wheels are able to drive in the wheels direction or rotation but roll passively in the direction perpendicular to rotation. The X axis in "RW_1" below is aligned with "Δs_1" but the opposite direction. The Y axis is in the passive direction of wheel 1 so Y is not dependent on "Δs_1". However, Y is dependent on "Δs_0" and "Δs_2". By inspection "Δs_0" is in the direction of Y and "Δs_2" is in the opposite direction of Y. The X component motion vector for wheel 1 "Δs_{1RW_1}" is "$-\Delta s_1$" and Y of "Δs_{1RW_1}" is the average of "$-\Delta s_2$" and "Δs_0". See formula (3) below and similarly formulas (4), (5) and (6) below.



$$\overrightarrow{\Delta s_{1RW_1}} = \begin{bmatrix} -\Delta s_1 \\ \frac{\Delta s_0 - \Delta s_2}{2} \end{bmatrix} \quad (3)$$

$$\overrightarrow{\Delta s_{0RW_0}} = \begin{bmatrix} -\Delta s_0 \\ \frac{\Delta s_3 - \Delta s_1}{2} \end{bmatrix} \quad (3)$$

$$\overrightarrow{\Delta s_{1RW_1}} = \begin{bmatrix} -\Delta s_1 \\ \frac{\Delta s_0 - \Delta s_2}{2} \end{bmatrix} \quad (4)$$

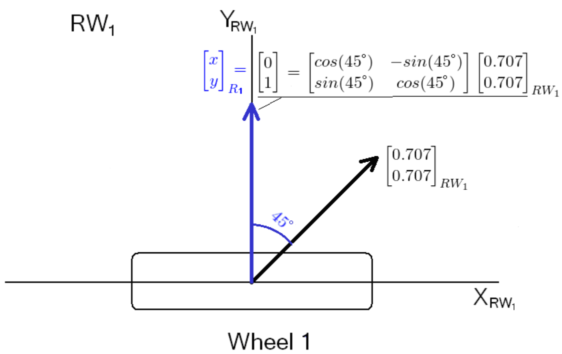
$$\overrightarrow{\Delta s_{2RW_2}} = \begin{bmatrix} -\Delta s_2 \\ \frac{\Delta s_1 - \Delta s_3}{2} \end{bmatrix} \quad (5)$$

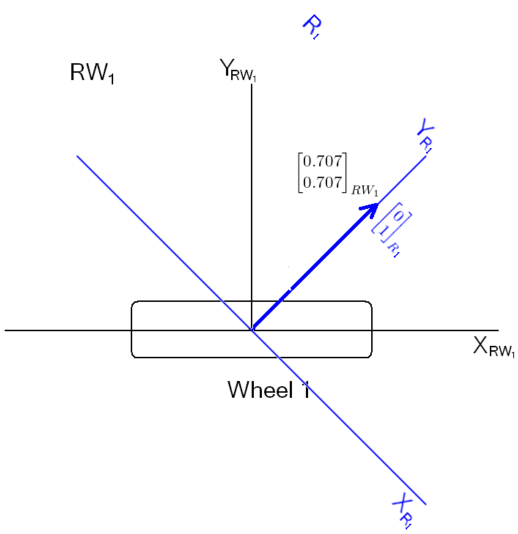
$$\overrightarrow{\Delta s_{3RW_3}} = \begin{bmatrix} -\Delta s_3 \\ \frac{\Delta s_2 - \Delta s_0}{2} \end{bmatrix} \quad (6)$$

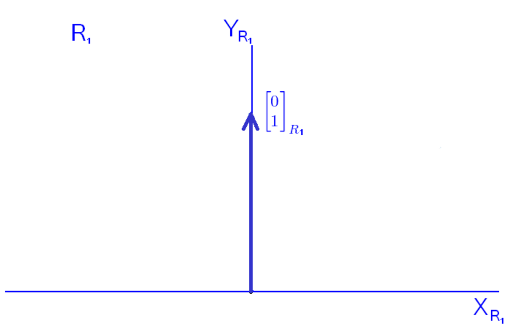
Now that we can find vectors for the motion of each wheel during the 25mS program loop we need to rotate these vectors so they become the vector of motion of their respective corner of the robot chassis. Formula (7) below is a rotation linear transform. It is used to rotate a vector about the origin. Knowing the angle between coordinate systems, this rotation is used to find a given vector's representation in the rotated system. In the diagrams below, formula (7) is shown transforming a vector from "RW_1" to "R_1". For the transform from "RW_0" to "R_0" the angle is -45º, from "RW_1" to "R_1" the angle is 45º, from "RW_2" to "R_2" the angle is 135º, and from "RW_3" to "R_3" the angle is -135º.

$$\textbf{(7)}\quad \begin{bmatrix} x \\ y \end{bmatrix}_{R_\#} = \begin{bmatrix} cos(\phi) & -sin(\phi) \\ sin(\phi) & cos(\phi) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{RW_\#}$$

See formulas (8),(9),(10), and (11) below and note that we use **matrix multiplication** [ref 11] in these formulas.







$$\begin{bmatrix} x \\ y \end{bmatrix}_{R_1} = \begin{bmatrix} cos(45^\circ) & -sin(45^\circ) \\ sin(45^\circ) & cos(45^\circ) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}_{RW_1}$$

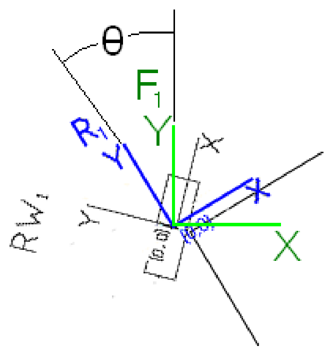
$$\textbf{(8)}\quad \overrightarrow{\Delta s0_{R0}} = \begin{bmatrix} cos(-45^\circ) & -sin(-45^\circ) \\ sin(-45^\circ) & cos(-45^\circ) \end{bmatrix} \overrightarrow{\Delta s0_{RW0}}$$

$$\textbf{(9)}\quad \overrightarrow{\Delta s1_{R1}} = \begin{bmatrix} cos(45^\circ) & -sin(45^\circ) \\ sin(45^\circ) & cos(45^\circ) \end{bmatrix} \overrightarrow{\Delta s1_{RW1}}$$

$$\textbf{(10)}\quad \overrightarrow{\Delta s2_{R2}} = \begin{bmatrix} cos(135^\circ) & -sin(135^\circ) \\ sin(135^\circ) & cos(135^\circ) \end{bmatrix} \overrightarrow{\Delta s2_{RW2}}$$

$$\textbf{(11)}\quad \overrightarrow{\Delta s3_{R3}} = \begin{bmatrix} cos(-135^\circ) & -sin(-135^\circ) \\ sin(-135^\circ) & cos(-135^\circ) \end{bmatrix} \overrightarrow{\Delta s3_{RW3}}$$

Now that the motion vectors are all aligned in the same direction lets convert them to motion vectors on the floor's coordinate system "F". We use "F_0", "F_1", "F_2", and "F_3" to indicate the floor's coordinate system for each associated corner of the robot. The only difference between these is the location of the origin. Since we are just finding relative motion vectors we just pretend that "R_1" and "F_1" coordinate systems share the same origin. Formula (12) below should be familiar. T_{RF} is a rotation transformation matrix that rotates the given motion vector by the latest value of the robot's angle on the floor "θ". (12) T_{RF} transforms the vector from coordinate system R to coordinate system F. (13) T_{FR} transforms the vector from coordinate system F back to coordinate system R. This rotation transformation matrix is used in Formulas (14), (15), (16), and (17) finding the relative motion each corner moved in the associated 25mS program loop.



$$\text{(12)}\quad T_{RF} = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

$$\text{(13)}\quad T_{FR} = \begin{bmatrix} cos(-\theta) & -sin(-\theta) \\ sin(-\theta) & cos(-\theta) \end{bmatrix}$$

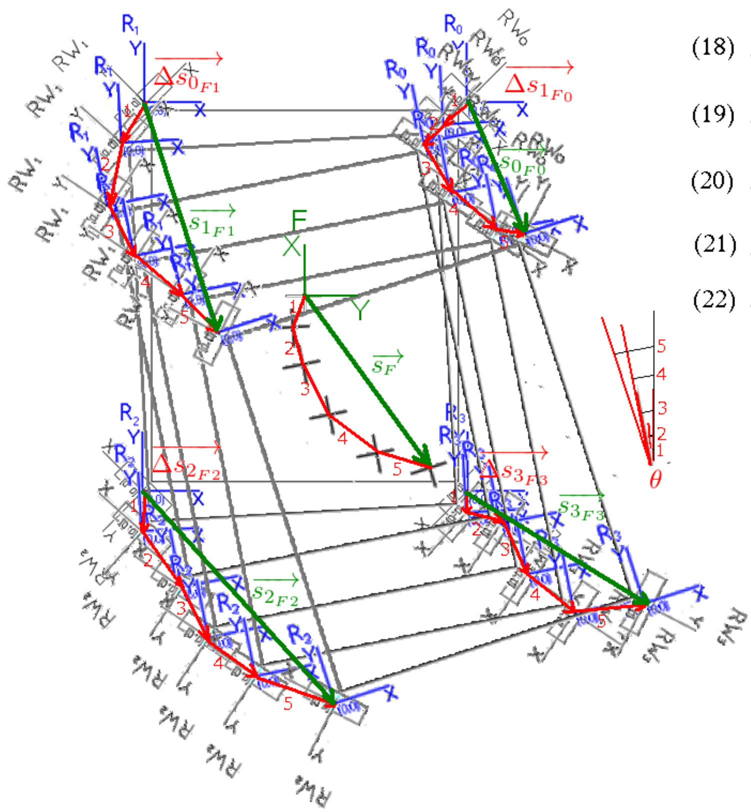
$$\text{(14)}\quad \overrightarrow{\Delta s0_{F0}} = T_{RF}\overrightarrow{\Delta s0_{R0}}$$

$$\text{(15)}\quad \overrightarrow{\Delta s1_{F1}} = T_{RF}\overrightarrow{\Delta s1_{R1}}$$

$$\text{(16)}\quad \overrightarrow{\Delta s2_{F2}} = T_{RF}\overrightarrow{\Delta s2_{R2}}$$

$$\text{(17)}\quad \overrightarrow{\Delta s3_{F3}} = T_{RF}\overrightarrow{\Delta s3_{R3}}$$

The delta vectors obtained from (14),(15),(16), and (17) above are used to find the absolute position on the floor. The absolute position is found for a given loop of the program by summing together [ref 12, 13] all of the delta values known until the given loop. See formulas (18),(19),(20), and (21) below. Note that the (i) indicates the loop number of the its delta vector like a function [ref 13] and is not a product term. The resulting absolute position vectors, for each wheel, indicate the position on the floor of the wheel since the first loop. Finally in equation (22) below, the average position of the four absolute position vectors for the wheels is taken. This is the absolute position of the center of the robot chassis on the floor.



(18) $\overrightarrow{s_{0_{F0}}} = \sum_{i=1}^{n} \overrightarrow{\Delta s_{0_{F0}}}(i)$

(19) $\overrightarrow{s_{1_{F1}}} = \sum_{i=1}^{n} \overrightarrow{\Delta s_{1_{F1}}}(i)$

(20) $\overrightarrow{s_{2_{F2}}} = \sum_{i=1}^{n} \overrightarrow{\Delta s_{2_{F2}}}(i)$

(21) $\overrightarrow{s_{3_{F3}}} = \sum_{i=1}^{n} \overrightarrow{\Delta s_{3_{F3}}}(i)$

(22) $\overrightarrow{s_F} = (\overrightarrow{s_{0_{F0}}} + \overrightarrow{s_{1_{F1}}} + \overrightarrow{s_{2_{F2}}} + \overrightarrow{s_{3_{F3}}})/4$

Basic Control:

The motors are gear motors that appear to reach equilibrium quickly for a given applied power. We have not bothered to implement a PID control for the wheels. We assume applied power equals speed set point. To prevent wheel slip we have limited the slew rate at which we apply the power to the wheels. Preventing wheel slip greatly improves position tracking.

The symbol omega "ω" is used for rotational rate it applied directly to all wheel motors. In formulas (23),(24),(25), and (26) below, "p" represents the power applied to the wheel indicate by its subscript. The "$\vec{e}$" symbol represents a "unit vector" in the direction of the motor drive indicated by its subscript. "$\vec{v}$" indicates the desired velocity vector of the robot.

We take the "dot product" [ref 14] of the two vectors $\vec{e}$ and $\vec{v}$. Since $\vec{e}$ is a unit vector, its length is 1 the resulting value will be the amount of $\vec{v}$ in the direction of $\vec{e}$.

(23) $p_0 = \omega + \overrightarrow{e_0} \cdot \overrightarrow{v}$

$$p_0 = \omega + \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix} \qquad let: \ \omega = 100 \quad v = \begin{bmatrix} 70 \\ 70 \end{bmatrix}$$

$$p_0 = \omega + (e_{0x}v_x + e_{0y}v_y) = 100 + \left(\frac{-1}{\sqrt{2}}\right)(70) + \left(\frac{1}{\sqrt{2}}\right)(70) = 100$$

(24) $p_1 = \omega + \overrightarrow{e_1} \cdot \overrightarrow{v}$

$$p_1 = \omega + \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix} \qquad let: \ \omega = 100 \quad v = \begin{bmatrix} 70 \\ 70 \end{bmatrix}$$

$$p_1 = \omega + (e_{0x}v_x + e_{0y}v_y) = 100 + \left(\frac{-1}{\sqrt{2}}\right)(70) + \left(\frac{-1}{\sqrt{2}}\right)(70) = 1$$

(25) $p_2 = \omega + \overrightarrow{e_2} \cdot \overrightarrow{v}$

$$p_2 = \omega + \begin{bmatrix} \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix} \qquad let: \ \omega = 100 \quad v = \begin{bmatrix} 70 \\ 70 \end{bmatrix}$$

$$p_2 = \omega + (e_{0x}v_x + e_{0y}v_y) = 100 + \left(\frac{1}{\sqrt{2}}\right)(70) + \left(\frac{-1}{\sqrt{2}}\right)(70) = 100$$

(26) $p_3 = \omega + \overrightarrow{e_3} \cdot \overrightarrow{v}$

$$p_3 = \omega + \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix} \qquad let: \ \omega = 100 \quad v = \begin{bmatrix} 70 \\ 70 \end{bmatrix}$$

$$p_3 = \omega + (e_{0x}v_x + e_{0y}v_y) = 100 + \left(\frac{1}{\sqrt{2}}\right)(70) + \left(\frac{1}{\sqrt{2}}\right)(70) = 199$$

Motor Scaling:

Now, since the maximum power that can be applied to the motor is 127, p must be scaled proportionally to each other.

$$u = Top\ speed\ of\ motor$$

$$if\ max(||p_0||, ||p_1||, ||p_2||, ||p_3||) > u\ Then$$

$$h = max(||p_0||, ||p_1||, ||p_2||, ||p_3||)$$

$$m_0 = \frac{up_0}{h}$$

$$m_1 = \frac{up_1}{h}$$

$$m_2 = \frac{up_2}{h}$$

$$m_3 = \frac{up_3}{h}$$

$$else$$

$$m_0 = p_0$$

$$m_1 = p_1$$

$$m_2 = p_2$$

$$m_3 = p_3$$

References:

[ref 1] http://www.vexrobotics.com/vex/products/accessories/motion/276-2177.html
[ref 2] http://www.vexrobotics.com/vex/products/accessories/motion/276-2185.html
[ref 3] http://www.vexrobotics.com/vex/products/accessories/sensors/276-1321.html
[ref 4] https://www.khanacademy.org/math/trigonometry/unit-circle-trig-func/radians_tutorial/v/radian-measure-and-arc-length
[ref 5] https://www.khanacademy.org/math/linear-algebra
[ref 6] https://www.khanacademy.org/math/cc-eighth-grade-math/cc-8th-systems-topic/cc-8th-systems-with-substitution/v/the-substitution-method
[ref 7] https://www.khanacademy.org/math/algebra/multiplying-factoring-expression
[ref 8] https://www.khanacademy.org/math/algebra-basics/core-algebra-graphing-lines-slope/core-algebra-graphing-coordinate-plan/v/the-coordinate-plane
[ref 9] https://www.khanacademy.org/math/linear-algebra/matrix_transformations/lin_trans_examples/v/linear-transformation-examples-rotations-in-r2
[ref 10] https://www.khanacademy.org/math/linear-algebra/vectors_and_spaces
[ref 11] https://www.khanacademy.org/math/algebra2/alg2-matrices/matrix-multiplication-alg2/v/matrix-multiplication-intro
[ref 12] https://www.khanacademy.org/math/precalculus/seq_induction/geometric-sequence-series/v/sigma-notation-sum
[ref 13] https://www.khanacademy.org/math/algebra/algebra-functions
[ref 14] https://www.khanacademy.org/math/linear-algebra/vectors_and_spaces/dot_cross_products/v/vector-dot-product-and-vector-length
[ref 5] http://www.quicklatex.com/

License: