

**Left or Right Follow Bug 2:** Talk about finding angle between vectors.  $V1.V2 = |V1| \times |V2| \times \cos(\theta)$

```
hitVectorx = (Target_x - hitPoint_x);
hitVectory = (Target_y - hitPoint_y);
currVectorx = (Target_x - ROBOTps.x);
currVectory = (Target_y - ROBOTps.y);
```

```
tempa = hitVectorx*currVectorx + hitVectory*currVectory;
tempb = sqrtf((hitVectorx*hitVectorx + hitVectory*hitVectory) * (currVectorx*currVectorx + currVectory*currVectory));
ThetaBetweenVectors = acosf(tempa/tempb);
```

**Left or Right Follow Bug 0:** Talk about checking if something on Left or right Where is the target in Robot coordinates?

T from Robot to World frame

$${}^W_R T = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & \text{Robot pose } X \\ \sin\theta & \cos\theta & 0 & \text{Robot pose } Y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rx is RobotposeX and Ry is RobotposeY

Then take the inverse

$${}^R_W T = \begin{bmatrix} R^T & -R^T d \\ 0 & 1 \end{bmatrix}$$

Gives

$${}^R_W T = \begin{bmatrix} \cos\theta & \sin\theta & 0 & -R_x \cos\theta - R_y \sin\theta \\ -\sin\theta & \cos\theta & 0 & R_x \sin\theta - R_y \cos\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
tempcos = cosf(ROBOTps.theta);
```

```
tempsin = sinf(ROBOTps.theta);
```

```
XinRobot = robotdest[statePos].x*tempcos + robotdest[statePos].y*tempsin - ROBOTps.x*tempcos - ROBOTps.y*tempsin;
```

```
YinRobot = -robotdest[statePos].x*tempsin + robotdest[statePos].y*tempcos + ROBOTps.x*tempsin - ROBOTps.y*tempcos;
```

So to find Targetx (Tx) and Targety (Ty) in Robot coordinates Tx and Ty are in World coordinates

$$[{}^R_W T] \begin{bmatrix} T_x \\ T_y \\ 0 \\ 1 \end{bmatrix} = \text{x and y in Robot coordinates}$$

$$\text{Target in Robot coordinates} = \begin{bmatrix} T_x \cos\theta + T_y \sin\theta - R_x \cos\theta - R_y \sin\theta \\ -T_x \sin\theta + T_y \cos\theta + R_x \sin\theta - R_y \cos\theta \end{bmatrix}$$