University Of Toronto: Unicycle PID Control Using Euler-Forward Numerical Method Simulation

Yoshiki Shoji

October 25, 2016

Acknowledgements

The idea for this project was mediated through the supervision of Professor G. M. T. D'Eleuterio, at University of Toronto Institute for Robotics Studies. This is not to be used for academic uses unless otherwise approved, nor plagiarized.

The Objective

Modeling a robot as a 'unicycle', we desire to create an iterative algorithm giving a feedback control in the form of PID, specifically using proportional gain, in order to regulate the unicycle to the origin pointing upwards starting from any pose restricted within a radius of 2m.

The Model

We first consider a 'unicycle' model for a robot. At any point in time, its rate of change in the translational and angular motion is given by the form:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta & 0 \\ sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

We then apply the transformation of the form $(x, y, z) \rightarrow (\rho, \alpha, \beta)$ where,

$$\rho = \sqrt{(x_d - x)^2 + (y_d - y)^2}$$
$$\beta = -tan^{-1} \frac{y_d - y}{x_d - x}$$
$$\alpha = -\beta - \theta$$

where ρ is the distance to the desired location, (x_d, y_d), β is the desired angle (measured counter-clockwise) which will give the robot its proper orientation referenced from its initial pose, and hence leaves α to be the difference between θ and β, which indicates how far away we are to the desired pose angle wise.

Under proper substitution, one can then find the model becomes:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -cos\alpha & 0 \\ \rho^{-1} sin\alpha & -1 \\ -\rho^{-1} sin\alpha & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}$$

However, as the system is non-linear, we will then linearize the model using $(\rho, \alpha, \beta) = (0, 0, 0)$ as we desire the robot to be regulated to the origin with pointing upwards.

Now, given the feedback law:

$$v = -k_\rho \rho, \quad \omega = -k_\alpha \alpha - k_\beta \beta$$

we find that through using the Eigen equation, in order to ensure stability,

$$k_\rho < 0$$

$$k_\alpha - k_\rho < 0$$

$$k_\beta k_\rho < 0$$

Algorithm

To create the algorithm, the Euler-Forward method was utilized to approximate the values through an iterative method. So here, we use:

$$x_{k+1} = x_k + h\dot{x}_k$$

Using the model, we further have:

$$\dot{x} = -k_\rho \rho cos\theta$$
$$\dot{y} = -k_\rho \rho sin\theta$$
$$\dot{\theta} = -k_\alpha \alpha - k_\beta \beta$$

Using all the relationships, we then find:

$$\dot{x} = -k_\rho \sqrt{x^2 + y^2} cos\theta$$
$$\dot{y} = -k_\rho \sqrt{x^2 + y^2} sin\theta$$
$$\dot{\theta} = -k_\alpha \Big(tan^{-1}\frac{y}{x} - \theta\Big) - k_\beta \Big(tan^{-1}\frac{y}{x} - \beta_d\Big)$$

From all of the above, we substitute in order to formulate the final numerical equations that will regulate the unicycle to the origin pointing upwards:

$$x_{k+1} = x_k - k_\rho h \sqrt{x_k^2 + y_k^2}\ cos\theta_k$$
$$y_{k+1} = y_k - k_\rho h \sqrt{x_k^2 + y_k^2}\ sin\theta_k$$
$$\theta_{k+1} = \theta_k - h\Big[k_\alpha \Big(tan^{-1}\frac{y_k}{x_k} - \theta_k\Big) - k_\beta \Big(tan^{-1}\frac{y_k}{x_k} - \beta_d\Big)\Big]$$

where,

$$k_\rho < 0$$
$$k_\alpha - k_\rho < 0$$
$$k_\beta k_\rho < 0$$
$$\beta_d = -\frac{\pi}{2}$$

Simulation

The figure below shows the unicycle starting in different locations and orientation. The problem at hand required us to restrain the motion within a circle of radius 2m.

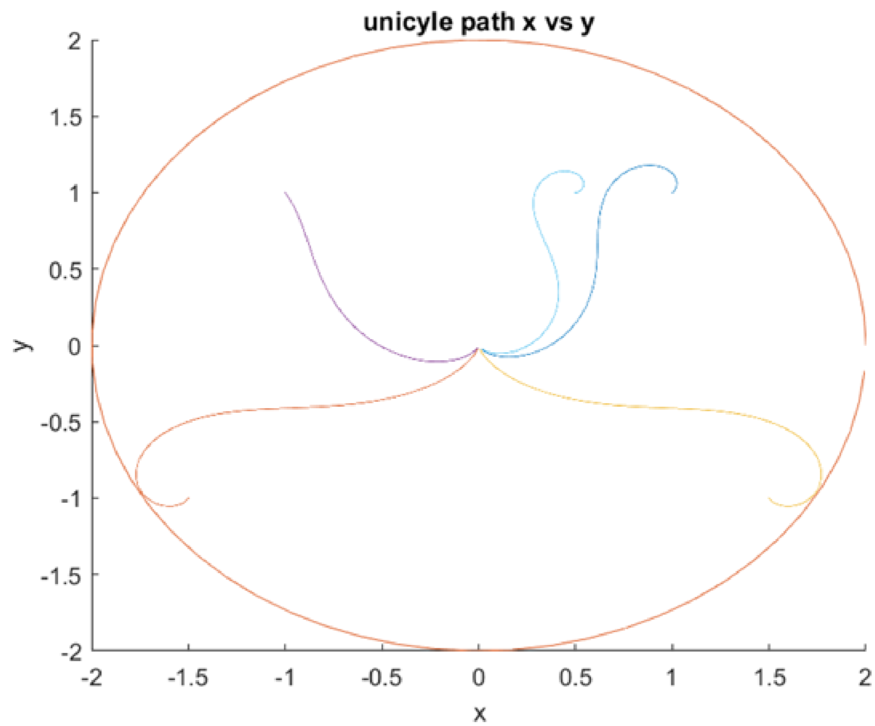

Figure 1: Unicycle starting in different poses converging to the origin pointing upwards

Modification: Following A Sinusoidal Path

In order to test the validity of the algorithm from the first part, the motivation is now to follow a sinusoidal path, the sine curve.

We first recognize the algorithm was established with static values, (x_d, y_d, β_d). In order to make this a dynamic problem which will enable us to follow a curve, we will use Matlab's linspace function retrieving values from the function, $y = 0.5sinx$. The model will now become:

$$x_{k+1} = x_k - k_\rho h\sqrt{x_k^2 + (0.5sinx_k)^2}\ cos\theta_k$$

$$y_{k+1} = 0.5sinx_k - k_\rho h\sqrt{x_k^2 + (0.5sinx_k)^2}\ sin\theta_k$$

$$\theta_{k+1} = \theta_k - h\Big[k_\alpha\Big(tan^{-1}\frac{0.5sinx_k}{x_k} - \theta_k\Big) - k_\beta\Big(tan^{-1}\frac{0.5sinx_k}{x_k}\Big)\Big]$$

where in order to ensure convergence, we have the following restrictions for the feedback values again as:

$$k_\rho < 0$$

$$k_\alpha - k_\rho < 0$$

$$k_\beta k_\rho < 0$$

$$\beta_d = 0$$

Simulation

The figure below shows the unicycle with different step-size values. The problem at hand required us to follow the sinusoidal path, $y = 0.5 sin x$, which is red curve starting at point, $(0, 0)$.

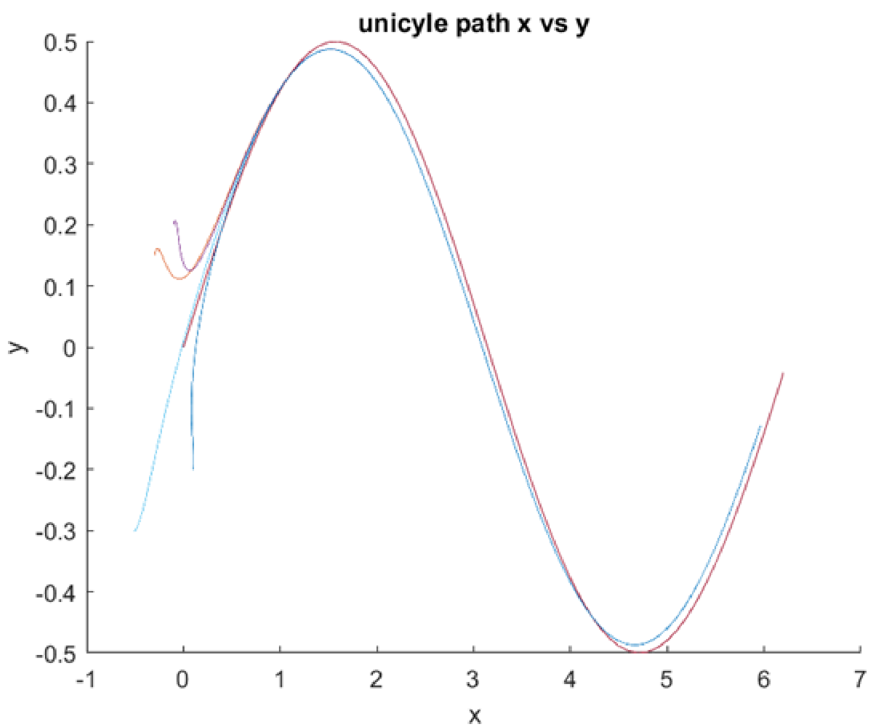

Figure 2: Unicycle following sinusoidal path

Appendix (a): The Variable Transformation Derivation

We first recognize the problem at hand, wherein:

$$\rho = \rho(x(t), y(t))$$

$$\beta = \beta(x(t), y(t))$$

$$\alpha = \alpha(x(t), y(t))$$

Under such circumstances, owing to multivariate calculus, differentiating with respect to time yields the results:

$$\dot{\rho} = \frac{\partial \rho}{\partial x}\frac{dx}{dt} + \frac{\partial \rho}{\partial y}\frac{dy}{dt} \qquad eq.1$$

$$\dot{\beta} = \frac{\partial \beta}{\partial x}\frac{dx}{dt} + \frac{\partial \beta}{\partial y}\frac{dy}{dt} \qquad eq.2$$

$$\dot{\alpha} = -\dot{\beta} - \dot{\theta} \qquad eq.3$$

We then go through differentiating one by one. For $eq.1$, we obtain:

$$-\rho\dot{\rho} = (x_d - x)\dot{x} + (y_d - y)\dot{y}$$

Manipulating the equation and using the relationship, $-tan(\beta) = \frac{y_d - y}{x_d - x}, \dot{x} = vcos(\theta), \dot{y} = vsin(\theta)$,the equation boils down to:

$$\dot{\rho}\Big(\frac{\rho}{x_d - x}\Big) = -vcos(\alpha)sec(\alpha + \theta)$$

Owing to trigonometry, we can further deduce:

$$sec(\alpha + \theta) = sec(\beta) = \frac{\rho}{x_d - x}$$

Hence, the final refined equation becomes:

$$\dot{\rho} = -vcos\alpha$$

Similar procedures follow for obtaining $\dot{\alpha}$\$. We first give the terms associated with the differential terms:

$$\frac{\partial \beta}{\partial x}\frac{dx}{dt} = \frac{-(y_d - y)\dot{x}}{(x_d - x)^2 + (y_d - y)^2} = -\rho^{-2}(y_d - y)\dot{x}$$

$$\frac{\partial \beta}{\partial y}\frac{dy}{dt} = \frac{(x_d - x)\dot{y}}{(x_d - x)^2 + (y_d - y)^2} = \rho^{-2}(x_d - x)\dot{y}$$

We then have the equation, using $\dot{x} = vcos\theta, \dot{y} = vsin\theta, \beta = -(\alpha + \theta), -tan\beta = \frac{y_d - y}{x_d - x}$,

$$\rho\dot{\beta}\Big(\frac{\rho}{x_d - x}\Big) = v\Big(sin\theta - tan(\alpha + \theta)cos\theta\Big)$$

$$\rho\dot{\beta}\Big(\frac{\rho}{x_d - x}\Big) = -v\Big(sin\alpha(sec\beta)\Big)$$

Owing to trigonometry, we again have:

$$\rho\dot{\beta}sec\beta = -vsin\alpha(sec\beta)$$

Hence, the final relationship then becomes the desired equation without neglecting the angular term:

$$\dot{\beta} = -\rho^{-1}vsin\alpha$$

Using the derived relationships above, we then have for α:

$$\dot{\alpha} = \rho^{-1}vsin\alpha - \omega$$

Putting all the above into matrix form, we finally obtain:

$$\begin{bmatrix}\dot{\rho}\\ \dot{\alpha}\\ \dot{\beta}\end{bmatrix} = \begin{bmatrix}-cos\alpha & 0\\ \rho^{-1}sin\alpha & -1\\ -\rho^{-1}sin\alpha & 0\end{bmatrix}\begin{bmatrix}v\\ w\end{bmatrix}$$

Appendix (b): Feedback Stability Derivation

The feedback we give to the system is:

$$v = -k_\rho \rho, \quad \omega = -k_\alpha \alpha - k_\beta \beta$$

Upon substitution using the transformed variables, we obtain:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} -cos\alpha & 0 \\ \rho^{-1} sin\alpha & -1 \\ -\rho^{-1} sin\alpha & 0 \end{bmatrix} \begin{bmatrix} -k_\rho \rho \\ -k_\alpha \alpha - k_\beta \beta \end{bmatrix}$$

To figure out stability, we will linearize the equations above and use the Eigen-value problem. The linearization equation is given as follows:

$$\dot{\rho} \approx f(\rho_o, \alpha_o) + \rho \frac{\partial f}{\partial \rho}(\rho_o, \beta_o) + \alpha \frac{\partial f}{\partial \alpha}(\rho_o, \beta_o)$$

where, $f(\rho, \alpha) = k_\rho \rho cos\alpha, (\rho_o, \beta_o) = (0, 0)$

Hence we obtain the following approximation near the origin:

$$\dot{\rho} = k_\rho \rho$$

Under similar procedures, the other approximations become:

$$\dot{\beta} = (k_\alpha - k_\rho)\alpha + k_\beta \beta$$

$$\dot{\alpha} = k_\rho \alpha$$

Putting all the above into matrix form, we obtain:

$$\begin{bmatrix} \dot{\rho} \\ \dot{\alpha} \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} k_\rho & 0 & 0 \\ 0 & k_\alpha - k_\rho & k_\beta \\ 0 & k_p & 0 \end{bmatrix} \begin{bmatrix} \rho \\ \alpha \\ \beta \end{bmatrix}$$

Hence, this becomes an Eigen-value problem. As we have a 3x3 matrix, we use Cramer's rule in finding the determinant for the following matrix:

$$\begin{bmatrix} k_\rho - \lambda & 0 & 0 \\ 0 & k_\alpha - k_\rho - \lambda & k_\beta \\ 0 & k_p & -\lambda \end{bmatrix}$$

In order to ensure stability, all Eigen-values must be negative. Hence,

$$k_\rho < 0$$

$$k_\alpha - k_\rho < 0$$

$$k_\beta k_\rho < 0$$