FUZZY CONTROLLER FOR TWO WHEEL INDEPENDENT DRIVE PIVOT STEERING VEHICLE*

Stephen W. Soliday

North Carolina Agricultural and Technical State University
Department of Electrical Engineering / MIPAR†

Abstract

A description for a vehicle consisting of two independently driven drive wheels and one free wheeling castor is given. The equations of motion are derived using Lagrange's equations and a model is developed using Runge-Kutta numerical integration. A fuzzy controller is then developed to control both the speed and direction of the vehicle. The parameters of the discourse sets for the fuzzy controller were optimized using a primitive form of Genetic Algorithm. Results of the controller's performance are discussed.

KEYWORDS: Fuzzy–Logic, Lagrangian, Runge–Kutta, Genetic–Algorithm.

1 INTRODUCTION

In this paper a fuzzy-logic controller was used to correct the course of a vehicle consisting of two independently driven wheels. The arrangement of the wheels allows for axial pivoting. It was desired that the control inputs should be a change in forward velocity and a change in the direction of the vehicle. In the subsequent sections, descriptions will be given on the physical model of the system and on the construction and training of the fuzzy logic controller.

2 DESCRIPTION OF VEHICLE

The controller was designed to specifically control a visually navigated vehicle. The vehicle will consist of two independent direct drive motors, and a free castor wheel. This configuration will support a chassis containing the motors, power supply and batteries. Above this will be the CPU chassis, and above that will be the equipment rack. The vehicle will process filtered images of paths through a neural network. The network will produce an output that will instruct the drive controller to turn left or right in order to remain on the path. The system may also require changes in forward velocity. The change in the direction and the change in velocity constitute the first two of four control signals for the fuzzy controller. Since the change in velocity and the change in direction are relative to the current position of the vehicle and not to an external frame of reference. The only other required states are the current velocity of the left and right wheels. These make up the remaining two control inputs.

3 DERIVATION OF EQUATIONS OF MOTION

In order to train the controller it was necessary to construct a computational model of the vehicle. This required deriving a set of differential equations of motion. Langrangian dynamics were best suited to this task[Landau 1976]. The simplified model, Figure 1 of the vehicle consists of a square slab and two cylindrical slab wheels. The first step in developing the equations of motion is to describe a set constraints for the independent variables. Tangential velocity V, and rotational velocity $d\varphi/dt$, are dependent on $d\theta_1/dt$ and $d\theta_2/dt$.

*This paper appeared in the First Industry/Academy Symposium on Research for Future Supersonic and Hypersonic Vehicles, vol 1, pp 106-111, Dec 1994

†MIPAR: Machine Intelligence and Power Associated Research

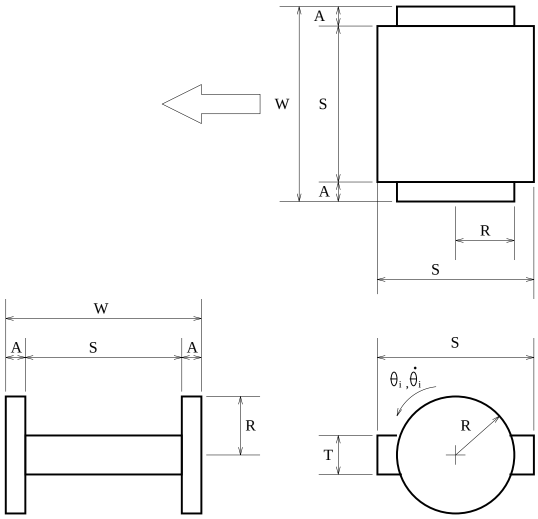

Figure 1: Vehicle Geometry

$$
\begin{aligned}
V &= \tfrac{R}{2}\left(\dot{\theta}_1 + \dot{\theta}_2\right) \\
\dot{\varphi} &= = \tfrac{R}{2}\left(\dot{\theta}_1 + \dot{\theta}_2\right)
\end{aligned} \tag{1}
$$

with these two constraints the kinetic energy terms may be built as functions of $T = T\left(d\theta_1/dt,\ d\theta_1/dt\right)$. The kinetic energy of the vehicle is built up from the translation and rotational kinetic energy of each piece. The kinetic energy of the slab is:

$$
T = \frac{1}{2}MV^2 + \frac{1}{12}MS^2\dot{\varphi}^2 \tag{2}
$$

The kinetic energy of a wheel is:

$$
\begin{aligned}
T &= \tfrac{1}{2}mV^2 + \tfrac{1}{4}mR^2\left(\dot{\theta}_1^2 + \dot{\theta}_2^2\right) + \\
&\quad \tfrac{1}{12}m\dot{\varphi}^2\left(4A^2 + 6AS + 3R^2 + 3S^2\right)
\end{aligned} \tag{3}
$$

Add Equation 2 and Equation 3 for both $d\theta_1/dt$ and $d\theta_2/dt$ and then substitute Equations 1 (a & b). After some tedious algebra the complete kinetic energy term may be found:

$$
\begin{aligned}
T &\equiv p_1\left(\dot{\theta}_1^2 + \dot{\theta}_2^2\right) + p_2\left(\dot{\theta}_1^2 - \dot{\theta}_2^2\right) + p_3\left(\dot{\theta}_1^2 + \dot{\theta}_2^2\right) \\
p_1 &\equiv \tfrac{R^2}{8}\left(M + 2m\right) \\
p_2 &\equiv \tfrac{R^2}{12S^2}\left(S^2M + m\left(4A^2 + 6AS + 3\left(R^2 + S^2\right)\right)\right) \\
p_3 &\equiv \tfrac{mR^2}{4}
\end{aligned} \tag{4}
$$

where T is the kinetic energy and p_n is a constant sub-expressions describing the geometry of the vehicle. Because the model of the vehicle will be on a level surface there is no potential energy term. Therefore $L \equiv T - V = T$. To find the accelerations of the wheels as functions of the torques applied the Lagrangian L was placed in a set of Lagrange's Equations:

$$
\begin{aligned}
\tau_1 &= \tfrac{d}{dt}\tfrac{\partial}{\partial\theta_1^2}L - \tfrac{\partial}{\partial\theta_1^2}L &= 2\left((p_1+p_2+p_3)\,\ddot{\theta}_1 + (p1-p2)\,\ddot{\theta}_2\right) \\
\tau_1 &= \tfrac{d}{dt}\tfrac{\partial}{\partial\theta_2^2}L - \tfrac{\partial}{\partial\theta_2^2}L &= 2\left((p1-p2)\,\ddot{\theta}_1 + (p_1+p_2+p_3)\,\ddot{\theta}_2\right)
\end{aligned} \tag{5}
$$

Solving for $d^2\theta_1/dt^2$ and $d^2\theta_2/dt^2$ independently and using new constant parameters, the final term in state space representation becomes:

$$
\begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} = \frac{1}{P_3}\begin{bmatrix} P_1 & P_2 \\ P_2 & P_1 \end{bmatrix} \cdot \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \tag{6}
$$

$$
\begin{aligned}
P_1 &\equiv \tfrac{24S^2}{R^2}(p_1+p_2+p_3) &= 5S^2M + 2m\left(4A^2+6AS+3\left(R^2+3S^2\right)\right) \\
P_2 &\equiv \tfrac{24S^2}{R^2}(p_2-p_1) &= S^2M - 2m\left(4A^2+6AS+3R^2\right) \\
P_3 &\equiv \tfrac{24S^2}{R^2}2\,(2p_1+p_3)\,(2p_2+p_3) &= (3m+M)\left(2s^2+m\left(8A^2+12AS+3\left(2R^2+3S^2\right)\right)\right)
\end{aligned} \tag{7}
$$

4 CONSTRUCTION OF MODEL

The computer model of the vehicle was constructed as a C++ class with knowledge of its position and wheel speed. The state vector for the system was:

$$
X = X\left(X, Y, \varphi, \dot{\theta}_1, \dot{\theta}_2, \tau_1, \tau_2\right) \tag{8}
$$

x, y, φ are position and direction of the vehicle relative to an inertial reference frame. θ_1 and θ_2 are the wheel velocities relative to the vehicle. τ_1 and τ_2 are the applied torque measured at the wheels. Member functions were used to alter the state variables and to move the vehicle. The function for moving the vehicle was built using a fourth order Runge-Kutta numerical integration[James 19xx].

4.1 Runge-Kutta Numerical Integration

The complete system of differential equations used to describe the motion of the vehicle is as follows:

$$
\begin{aligned}
\tfrac{d}{dt}\dot{\theta}_1 &= \tfrac{P_1\tau_1+P_2\tau_2}{P_3} &\equiv F_1\left(\tau_1,\tau_2,\varphi,\dot{\theta}_1,\dot{\theta}_2\right) \\
\tfrac{d}{dt}\dot{\theta}_2 &= \tfrac{P_1\tau_2+P_2\tau_1}{P_3} &\equiv F_2\left(\tau_1,\tau_2,\varphi,\dot{\theta}_1,\dot{\theta}_2\right) \\
\tfrac{d}{dt}\varphi &= \tfrac{R}{S}\left(\dot{\theta}_1-\dot{\theta}_2\right) &\equiv F_3\left(\tau_1,\tau_2,\varphi,\dot{\theta}_1,\dot{\theta}_2\right) \\
\tfrac{d}{dt}X &= \tfrac{R}{2}\left(\dot{\theta}_1+\dot{\theta}_2\right)\cos(\varphi) &\equiv F_4\left(\tau_1,\tau_2,\varphi,\dot{\theta}_1,\dot{\theta}_2\right) \\
\tfrac{d}{dt}Y &= \tfrac{R}{2}\left(\dot{\theta}_1+\dot{\theta}_2\right)\sin(\varphi) &\equiv F_5\left(\tau_1,\tau_2,\varphi,\dot{\theta}_1,\dot{\theta}_2\right)
\end{aligned} \tag{9}
$$

The five functions $k_{1,i}$ through $k_{4,i}$:

$$
\begin{aligned}
k_{1,i} &= h\cdot F_i(\ \tau_1,\ \tau_2,\ \varphi,\ \dot{\theta}_1,\ \dot{\theta}_2\ ) \\
k_{2,i} &= h\cdot F_i(\ \tau_1,\ \tau_2,\ \varphi+\tfrac{k_{1,3}}{2},\ \dot{\theta}_1+\tfrac{k_{1,1}}{2},\ \dot{\theta}_2+\tfrac{k_{1,2}}{2}\ ) \\
k_{3,i} &= h\cdot F_i(\ \tau_1,\ \tau_2,\ \varphi+\tfrac{k_{2,3}}{2},\ \dot{\theta}_1+\tfrac{k_{2,1}}{2},\ \dot{\theta}_2+\tfrac{k_{2,2}}{2}\ ) \\
k_{4,i} &= h\cdot F_i(\ \tau_1,\ \tau_2,\ \varphi+k_{3,3}\ \ \dot{\theta}_1+k_{3,1},\ \dot{\theta}_2+k_{3,2}\ )
\end{aligned} \tag{10}
$$

are used to compute a fourth-order Runge-Kutta for a series of simultaneous deferential equations

$$\begin{aligned}
\varphi &= \varphi + \tfrac{1}{6}\left(k_{1,1} + 2\left(k_{2,1} + k_{3,1}\right) + k_{4,1}\right) \\
\dot{\theta}_1 &= \dot{\theta}_1 + \tfrac{1}{6}\left(k_{1,2} + 2\left(k_{2,2} + k_{3,2}\right) + k_{4,2}\right) \\
\dot{\theta}_2 &= \dot{\theta}_2 + \tfrac{1}{6}\left(k_{1,3} + 2\left(k_{2,3} + k_{3,3}\right) + k_{4,3}\right) \\
X &= X + \tfrac{1}{6}\left(k_{1,4} + 2\left(k_{2,4} + k_{3,4}\right) + k_{4,4}\right) \\
Y &= Y + \tfrac{1}{6}\left(k_{1,5} + 2\left(k_{2,5} + k_{3,5}\right) + k_{4,5}\right)
\end{aligned} \tag{11}$$

5 FUZZY LOGIC

5.1 Introduction

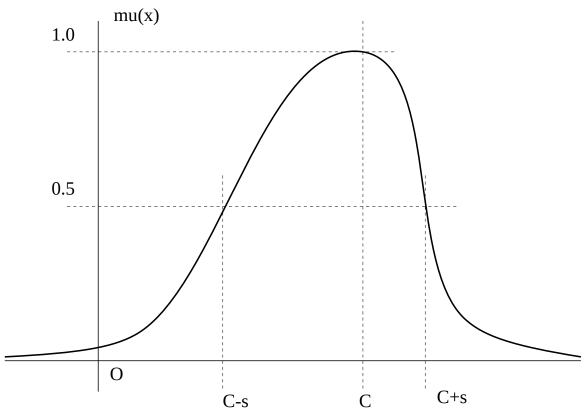

Figure 2: Fuzzy Membership

The fuzzy controller in this paper uses a table of rules to make fuzzy associations [1] between input and output fuzzy sets:

(if A and B then C)

A crisp scaler input is converted into a vector of fuzzy memberships, Equation 12, the elements of the input vectors are matched to the rules in the table. The minimum membership is applied to the output set and a center and an area are found, Equations 13 and 14. The crisp output is determined by locating the centroid produced by the centers and areas found by all of the rules in the table.

5.2 Fuzzy Set Membership

Fuzzy set membership is determined by a three parameter function, Figure 2. The first parameter is the center of the set. The other two parameters determine the left and right widths of the set at a membership value of one half. The membership function is defined to be the first derivative with respect to X of the arctangent function:

$$\mu_{i,\text{left}}(x) = \frac{\sigma^2_{i,\text{left}}}{(x - c_i)^2 + \sigma^2_{i,\text{left}}} \equiv \sigma_{i,\text{left}} \tan^{-1}\left[\frac{x - c_i}{\sigma_{i,\text{left}}}\right] \tag{12}$$

The area of a fuzzy set is important in determining the crisp output. The minimum of the input memberships is used as a level mark for the associated output fuzzy set. The area of the set under this level mark is:

$$\text{area}_{i,\text{left}} = \sigma_{i,\text{left}} \left[\frac{\pi}{2} + \sqrt{\text{level}^2 - \text{level}} - \tan^{-1} \sqrt{\frac{\text{level} - 1}{\text{level}}} \right] \tag{13}$$

Because of the difference in the right and left widths, the centroid of the set is offset from the center of the set. This centroid must be found by first determining the area and the centroid of the left and right halves:

$$\text{centroid}_{i,\text{left}} = c_i - \max \left[\frac{\sigma_{i,\text{left}}}{2} \sqrt{\frac{\text{level} - 1}{\text{level}}}, \sigma_{i,\text{left}} \right] \tag{14}$$

once the left and right centroids and areas are determined the centroid for the set is found:

$$\text{centroid}_i = \frac{\text{area}_{i,\text{left}} \cdot \text{centroid}_{i,\text{left}} + \text{area}_{i,\text{right}} \cdot \text{centroid}_{i,\text{right}}}{\text{area}_{i,\text{left}} + \text{area}_{i,\text{right}}} \tag{15}$$

The area of the set is just the sum of the left and right areas ($\text{area}_i = \text{area}_{i,\text{left}} + \text{area}_{i,\text{right}}$)

Finally the over all centroid of all of the rules outputs must be found. Since the areas of the individual sets represent the relative strength of the rule association they may be viewed as weights. This allows the overall centroid to be found analogous to the "center of mass" of a line of inhomogeneous particles. The equation for the center of mass is:

$$C.M. = \sum_{i-1}^{N} \text{area}_i \cdot \text{centroid}_i \ / \ \sum_{i-1}^{N} \text{area}_i \tag{16}$$

5.3 Training Events

Using the model developed in Section 4, a table of input/output pairs was developed. Vehicle states and control signals were chosen at random. These were then applied to the model to determine the next state. The current state of the wheel velocities and the difference in forward velocity and directional velocity were recorded as the input vector. The two torques that were randomly chosen, created the next state and were recorded as the output vector. An input/output vector pair constituted one training event. 1000 such events were produced and saved in a file. This file was then used to train the table of rules.

5.4 Rules Table

The table of rules consists a list of an association of input and output fuzzy-sets. As an example, the rule (**PS, NL, NS, PS; ZE, PL**) can be stated linguistically as (*if the change in forward velocity is positive-small and the change in rotational velocity is negative-large and the current angular velocity of the right wheel is negative-small and the angular velocity of the left wheel is positive-small then the torque applied to the right wheel is zero and the torque applied to the left wheel is positive-large.*) Each entry in the table of corresponds to the combination of input sets. The input to the controller is in 4-space and there are five fuzzy sets for each of the input discourses, that makes 625 rules or 5^4 entries in the rules table. The values at the entry position are the output combination and a relative strength. The strength is determined by taking the product of the four fuzzy input memberships and the two output fuzzy memberships. This strength is used to train, by identification, the table of rules' entries[Sayyarrodsari 1984]. As each input/output pair is presented, the output set with the largest membership is combined with every combination of input sets and a strength is calculated and matched with an existing rule entry. If the strength is greater the rule entry is replaced by the new combination. After all events have been applied the discourse sets must be optimized. This was done using a primitive genetic algorithm.

5.5 Optimization of Discourse Sets

A genetic algorithm was used to optimize the fuzzy sets. For fitness testing the model described in Section 4 was used rather than a fixed table. Each gene sequence consisted of the center to center separation and left and right widths of each of the fuzzy sets for all four inputs. Each parameter was given a realistic upper and lower bound and coded using gray code with eight bits. The total gene sequence consisted of 720 bits A desired trajectory was created using cosine functions for the two input torques. One (*the ideal vehicle*) vehicle was controlled by these time series functions for torque. Another vehicle (*the controlled vehicle*) was given the same initial conditions of the *ideal vehicle*. This vehicle was controlled using the fuzzy controller. The difference between the current position of the *controlled vehicle* and the next position of the *ideal vehicle* was used to create command inputs for the fuzzy controller. The error in the final positions of the two vehicles became the fitness criteria for the genetic algorithm. Simple two point crossover and bit wise mutation were used. The selection method was to sort the population by fitness. The fittest individual was bread with a random member of the population. Two offspring were produced with each mating. The two parents and the two offspring were saved. This was repeated until the count of the new population matched the count of the old population. The remaining members expired. A fixed percentage of the population was then mutated. This number was usually around two percent. After the mutation process was completed the cycle was repeated.

6 RESULTS

The final controller proved to be fairly robust. The vehicle could carry up to five percent of its mass as payload with out significant error in final destination. This error could be removed completely by adding a constant multiplier to the generated torque. A payload of ten percent of the mass of the car required a multiplier of 1.15 times the generated torque. This made the controller very robust. The controller could also function with small deviations in wheel radius on the order of two percent without significant error (*more than adequate for tire wear.*) Although the GA made significant improvement in the control of the vehicle. The optimized discourse sets differed from the original in a fairly linear fashion. This was not the case however for an earlier test of the controller on the classic inverted pendulum[Kosko 1992]. The optimized discourse sets for that controller differed in a very non-linear fashion. The vehicle described in this paper was designed to operate in conjunction with a neural network capable of associating road images with turning signals. The neural network will be executing on a separate processor. This would allow the next road image to be processed while the previous torque control was being determined. On every platform tested the controller was able to make control associations at a faster rate then the neural network could make navigation decisions. Therefor it can be stated that under the time constraints of the navigation system this fuzzy controller meets the requirements of "real time" controler.

7 ACKNOWLEDGEMENT

This research was supported in part by the Advanced Research Projects Agency under contract N00600–93–K–2051. I greatly appreciate their support, without which this paper could not have been successful.

References

[Kosko 1992] Kosko, B "Neural Networks and Fuzzy Systems", Prentice Hall, ISBN: 0-13-611435-0

[James 19xx] M.L. James, G.M. Smith, J.C. Wolford, "Applied Numerical Methods for Digital Computation", Harper

[Landau 1976] L.D.Landau, E.M.Lifshitz "Mechanics Course of Theoretical Physics", Pergamon Press, ISBN: 0-08-029141-4

[Sayyarrodsari 1984] B.Sayyarrodsari, "Hierarchy and its Implications in the Design of Fuzzy Logic Controllers", M.S.Thesis, Department of Electrical Engineering, North Carolina A