A Novel Motor Control Algorithm for Two-Wheeled and Caterpillar-Tracked Autonomous Vehicles Using a Fuzzy Navigation Abstraction

Antony P. Gerdelan
gerdelan@gmail.com

Abstract – Presented here is a new algorithm for mapping the new Fuzzy Logic robot navigation layer outputs into usable motor control instructions for vehicles with twin-drive actuator configuration. This paper is written as a practical guide for immediate implementation of a capable motor control layer into an intelligent vehicle, and includes complete derivation of relevant formulae.

Keywords: *Robot soccer, fuzzy logic, intelligent motor control, autonomous vehicle motion.*

1. Introduction

It is our goal to create a complete robot system. The robot soccer platform is being used as a development testbed as it combines a complete range of intelligent autonomous vehicle systems; including machine vision, multi-agent strategy, and fast robot navigation. The competitive nature of robot soccer stimulates development of increasingly capable autonomous systems.

We have used a layered, procedural approach to combine the different facets of the robot system. *Figure 1* illustrates the layers of our current system.

Our previous works have created a novel robot navigation system that is a hybrid of Fuzzy Logic and the A* algorithm [1,2]. The fast navigation system developed incorporates dynamic obstacle avoidance, target seeking and predictive opponent evasion behaviour. Referring to *Figure 1*, the third and fourth layers, respectively, represent the A* and Fuzzy Logic based layers of the navigation system.

Of interest to us in this paper is the bottom layer in *Figure 1* – the motor control system. As input, the motor control layer is passed a defuzzified *forward speed* and *rotational speed* from the Fuzzy control layer. The motor control layer maps the defuzzified speeds to specific motor velocities, expressed in corresponding voltages, which it outputs to each motor (the actuators).

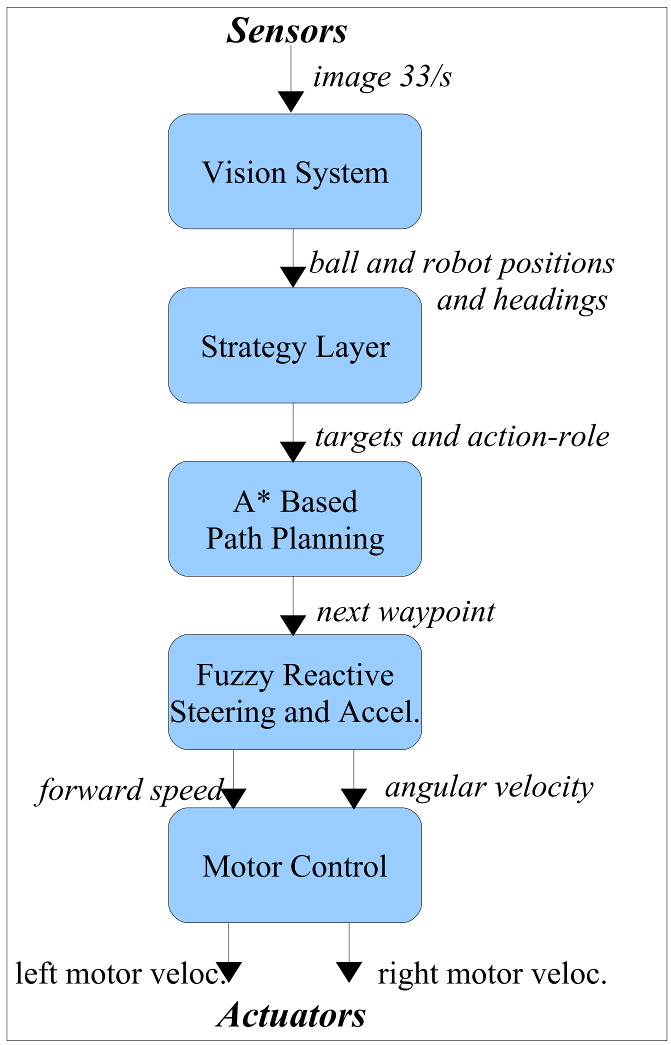

Figure 1 – Our multi-layered robot soccer system

2. Fuzzy Navigation Background

Robot Soccer teams have only a very small window of time in which to perform all layers of processing. In addition, calculation is not only performed for one robot, but for an entire team.

Because motion planning for each robot can only be allocated a very small sub-window of the time frame available for processing, teams are limited as to their algorithmic options for smoothing the paths of their robots. A team with a CPU-non-intensive motion-smoothing ability possesses a significant advantage over competitors.

We have used Fuzzy Logic for robot navigation; a novel approach that eliminates the need for the logically complex CPU-intensive motion calculations popular in modern robotics. This approach enables *smooth-path* motion calculation that not only supersedes the discontinuity typical during transition between curved and straight components of motion in the popular Reeds-Shepp method [3], and the Circle-Line method, reported by J-H Kim [4] as being the most popular method, but also requires none of the calculation overheads.

Furthermore, because the fuzzy logic system that we have developed performs path-smoothing *implicitly*, we do not run into the common problem encountered by methods that perform path-smoothing as a post-processing step, that is, we do not run the risk of our smoothed path colliding with obstacles.

3. Twin-Drive Steering Systems

Regular, four-wheeled vehicles are mechanically controlled by both steering and acceleration mechanisms, and so the *forward speed*, *rotational speed* abstraction that we are using can map directly to the actuators of the vehicle. *Forward speed* would be sent directly to the motor, and *rotational speed* would reflect the proportionate angle of the front wheels.

Vehicles without adjustable front wheels are subject to a special set of mechanics for controlling the steering of the vehicle. Vehicles of this type are most commonly caterpillar-tracked, but wheeled vehicles of this type, such as the Caterpillar Skid Steer Loader have also been developed with the same steering principles.

The robots that we have used in our robot soccer system have the peculiarity of having only two wheels, and therefore fall into the same category of motor control as tracked vehicles.

Movement of this type demands that the speeds of each track or wheels on each side of the vehicle can be set independently. A number of systems to facilitate this sort of independent drive exist; based on separate gear systems, or separate hydraulic pumps, however, the specific system we will investigate in this paper is the *dual motor* approach; that is, an independent motor is employed to control each wheel or track of the vehicle.

In order to change the direction of travel, vehicles of this type must slow down or stop the wheels or track on one side of the vehicle; the unequal speed ratio between wheels effectively creates torsion and subsequently alters the path of motion of the vehicle.

4. Aim

The aim of the motor control system outlined in this paper is to facilitate a smooth, arcing path of motion for two-wheeled, fixed-steering, and tracked autonomous vehicles – allowing these vehicles to steer whilst driving forwards by blending left and right motor speeds.

The advantage presented by the proposed system is that vehicles of this type are able to move more efficiently; in terms of robot soccer, this means that robots are able to move much more quickly, than robots that are constrained to turning on the spot before moving.

5. Requirements

In order to simultaneously drive and steer without blundering into obstacles, an intelligent navigation system is required as the parent layer to this motor control layer of the robot system. We have used a cascade of fuzzy systems as our parent layer (refer to *Figure 1*). The navigation layer must direct steering and forward speed, and pass this abstraction of movement to the motor control system. Intelligent and Fuzzy navigation systems are detailed in previous works [1,2].

6. Assumptions

Because the time period considered between motor control calculations is very small (approximately 0.0303 seconds in our system) we can disregard acceleration, and assume that the arc of motion for the vehicle within our time period will always be approximately circular and not parabolic. This allows us to greatly reduce the amount of calculation required and therefore also the latency between calculation and action.

We have also disregarded inclusion of friction, robot momentum, angular motor momentum, inertia and other forces in our motor-control calculations, as it is not feasible in our system to gather measurements for all of these external and internal forces.

These additional forces can, however, be accounted for implicitly to some degree by tuning the fuzzy sets of the fuzzy logic layer of the robot system – ideally for individual robots.

There is scope for motor control refinement for individual robots, to take into account variable or unmeasurable forces, by adjusting fuzzy sets through learning algorithms.

7. Method

The properties of the circular arc of motion of the robot; that is, the angular velocity ω and radius r of the arc, are determined by the magnitude and ratio between magnitudes of the inner and outer wheel and track velocities V_O and V_I, proportionate to the wheel separation distance d. Refer to *Figure 2* for illustration.

Rotational physics formulae can be used to determine the left and right motor velocities required to reproduce the forward velocity and angular velocity required by the fuzzy layer's abstraction of motion.

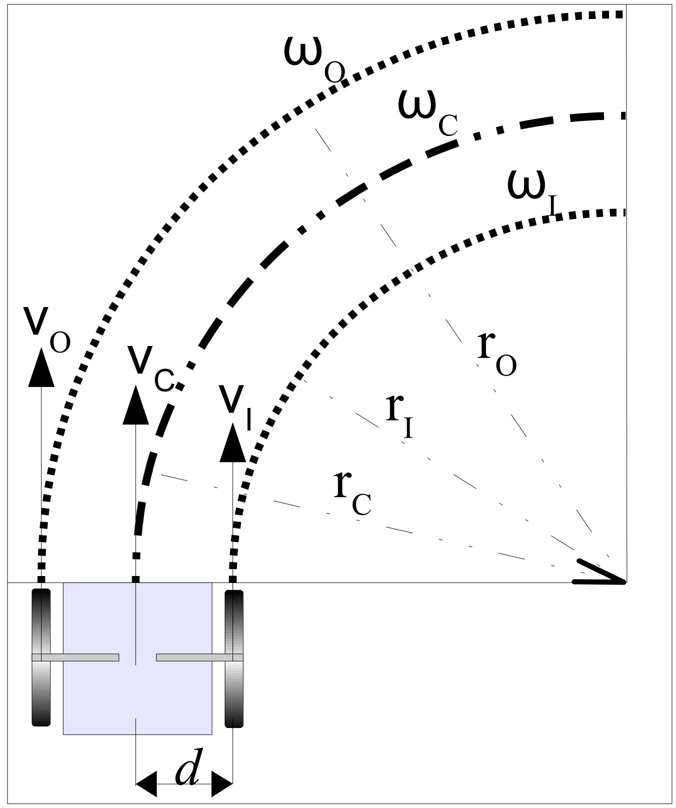

Figure 2 – Rotational variables for a twin drive vehicle

We can consider the following linear-angular relation - the definition of angular measure in radians:

$$s = \theta r \qquad \text{(equation 1)}$$

Where s represents the distance a point in moved along a circular arc of angle θ and with radian measure r, we can derive the equation with respect to time t, with r set constant:

$$\frac{ds}{dt} = \frac{d\theta}{dt} r$$

We can see that ds/dt is the linear speed – the magnitude of our linear velocity v, and $d\theta/dt$ is our angular speed. We can now express our equation in the form:

$$v = \omega r$$

Again, we express ω in terms of the radian measure. We can now derive our left and right motor instructions.

The Fuzzy Reactionary Layer of the system (*see Figure 1*) must feed information indicating the direction to turn the vehicle, either as a signed output angle value, or as the value of a flag variable. Knowing the direction to turn, we can simply apply the *outermost* and *innermost* motor instructions to the appropriate motor with a simple switching statement.

Taking the inputs passed from the Fuzzy layer; *forward speed* and *angular velocity*, we can say that:

$v_C =$ *forward speed*
and
$\omega_C =$ *angular velocity*

Where v_C and ω_C are the forward speed and angular velocity for the centre of the vehicle. We however, are interested in the variables at the *actuators* (wheels or tracks) of the vehicle.

We next establish the radius for the arc motion of from the centre of the vehicle:

$$r_C = \frac{v_C}{\omega_C}$$

With the radius for the arc of motion for the centre of the vehicle we can calculate the other radii using a known actuator distance *d* from the centre of the vehicle. We are assuming here that the actuators are equally separated from the centre of the vehicle.

We know that all three radial velocities (ω_C, ω_O, and ω_I) must be equal. Therefore we can now establish the required motor velocities using our derived *r* and ω values:

$$v_O = \omega_C \cdot \left(\frac{v_C}{\omega_C} + d\right)$$

and

$$v_I = \omega_C \cdot \left(\frac{v_C}{\omega_C} - d\right)$$

which can be further reduced:

$$v_O = v_C + d \cdot \omega_C \qquad \text{(equation 2)}$$

and

$$v_I = v_C - d \cdot \omega_C \qquad \text{(equation 3)}$$

For cases where the Fuzzy layer dictates that the required required linear velocity is very low (close to zero) we can take advantage of the special ability of some twin-drive vehicles to oppose the direction of the left and right motors.

The behaviour can not be smoothly blended with the *turn and drive* behaviour (*equations 2 and 3*) as the inertia and stress created on a motor during transition from forward to reverse motor direction is very difficult to manage.

The change in relative location for centre of rotation reverses direction as one of the motors is reversed, which requires another dimension of complexity to be accounted for by the Fuzzy navigation layer.

The number of potential mappings for motor instructions is greater than one for each forward speed and angular velocity pair. Because our system must be very fast, it benefits from the reduction of this sort of complexity.

For these reasons, we have completely separated our navigation logic into two cases where our vehicle must either:

1. Simultaneously steer and drive
2. Turn on the spot

This separation requires a high-level switching statement, based on the angular velocity and linear speed requirements, to decide which means of motivation to employ at each considered instance.

For the case where the vehicle must turn on the spot, we can say based on our derivation of *equation 1* that:

$$v_O = -v_I = d \cdot \omega \quad \text{(equation 4)}$$

Where we have substituted our radius *r* for *d*, because we are turning about the centre of our vehicle. We must then simply set the left motor equal to minimum of the two velocities (v_O *and* v_I from *equation 4)* if we are turning left, and vice versa for the case when we are turning right.

We can compound our working into a consolidated algorithm (*Algorithm 1*).

Algorithm 1 *Motor Control Algorithm for Two-Motor Autonomous Vehicles Using a Fuzzy Navigation Abstraction*

The input is a *forward speed* v_C and an *angular velocity* ω from the Fuzzy navigation layer. The Fuzzy layer must also appropriately set a flag *TURNLEFT*. We must also know the distance from the centre of the vehicle to each wheel *d*.

1. **if** the linear velocity value v_C is close to zero **then** jump to step 4

2. **if** the angular velocity value ω is close to zero **then**
   set both v_L and v_B equal to v_C
   and terminate algorithm
   else
   Proceed to step 3.

3. Using *equations 2 and 3* calculate linear velocities v_A and v_B*:*

   $v_A = \omega \cdot (v_C / \omega + d)$
   $v_B = \omega \cdot (v_C / \omega - d)$

   Jump to step 5.

4. If the vehicle is turning on the spot, we calculate opposing velocities for the motors using *equation 4*:

   $v_A = -v_B = d \cdot \omega$

   Proceed to step 5.

5. Determine the motor velocities for specific motors (v_L and v_B)

   if *TURNLEFT* **then**
   $v_L = \min(v_A, v_B)$
   $v_R = \max(v_A, v_B)$
   else
   $v_L = \max(v_A, v_B)$
   $v_R = \min(v_A, v_B)$

---

8. Considerations

The are several factors to consider in implementation of the algorithm presented in this paper:

- Disparity in performance between left and right motors
- Variations is motor response from vehicle to vehicle
- Variation in response of motors during operation
- Left and right actuators operating on different surfaces

Depending on the magnitude of error experienced by a particular vehicle, additional coping mechanisms may or may not be required to correct the motion of the vehicle.

Disparity between left and right motor response is inevitable, and may be correcting for by adding some additional offset to the velocity instruction sent to a particular motor.

Individual vehicles will exhibit different operating characteristics, re-enforcing the need for fuzzy set definitions to be adjustable for individual vehicles.

However, it is likely that the variation in response for a particular motor will be non-linear over a range of velocity instructions (Voltages) and indeed also over a period of operation as the motor temperature increases. This will result in some error being expressed in the path of motion of the vehicle.

This error is typically only minor in nature in a Fuzzy-controlled system, due to its dynamic nature, as the vehicle will continue to be sent course correcting instructions as long as it is not positioned with the desired orientation.

In environments more complex than the robot soccer field, it is certain that actuators will at some point be operating, without prior information, on different types of terrain, and will produce unintended motion. Although this scenario will not prevent the vehicle from operating, again due to the dynamic nature of the Fuzzy system, the effectiveness of the system could be improved with an intelligent actuator-monitoring and adjustment approach.

10. Results

We have implemented the new motor control layer in the robot soccer game and observe that the system is successful. The robots retain the ability to quickly turn on spot when advantageous, but otherwise produce more effective motion *en route* to a destination.

The motor control algorithm presented in this paper has driven the soccer robots to another evolutionary level; the time taken to complete a path of motion can be reduced significantly, which is all important to speed-critical scenarios such as robot soccer.

It must be noted that the specific motor response of the soccer robots requires significant tuning to not only the Fuzzy navigation layer but also to the path planning layer before the desired motion can be achieved.

The degree of tuning and optimisation required due to non-simulated real-world factors presents scope for a machine learning or Genetic Algorithm. Messom [5] discusses Genetic Algorithms for motion control, tuning the same twin-motor actuators that we are using in our system.

The motor control algorithm presented in this paper consumes minimal processing time, and has been proven to successfully map a fuzzy abstraction of motion to real left and right motor instructions for two-wheeled soccer robots.

This research presents considerable potential for application to other vehicles of this type, particularly tracked vehicles using two motors, and there is scope for several optimisation and adaptive extensions to this system.

References

[1] A. P. Gerdelan, N. H. Reyes "*Synthesizing Adaptive Navigational Robot Behaviours using a Hybrid Fuzzy A* Approach*", Presented at "9th Fuzzy Days 2006 International Conference on Computational Intelligence" in Dortmund Sep 2006. Published in the the Springer Series "Advances in Soft Computing".

[2] A. P. Gerdelan, N. H. Reyes "*A Novel Hybrid Fuzzy A* Robot Navigation System for Target Pursuit and Obstacle Avoidance*", from the proceedings of the First New Zealand Korean Joint Workshop on Advance of Computational Intelligence Methods and Applications, 2006, AUT University, Auckland, New Zealand.

[3] J. A. Reeds, L.A. Shepp, "*Optimal paths for a car that goes both forwards and backwards*". Pacific Journal of Mathematics, 145(2):367-393, 1990.

[4] J-H Kim, "*Robot Control Program III*", Intelligent Control Lab, Korea Advanced Institute of Science and Technology, 1999.

[5] C.H. Messom, "*Genetic Algorithms for Autotuning Mobile Robot Motion Control*", Res. Lett. Inf. Math. Sci, 2002, (3), pp 129-134, ISSN 1175-2777.