Modeling and Fuzzy Control of a Four-Wheeled Mobile Robot

István Kecskés*, Zsuzsanna Balogh**, Péter Odry**
*Appl-DSP, Palić, Serbia
**Subotica Tech, Subotica, Serbia
kecskes.istvan@gmail.com, balogh.zsuzsanna7@gmail.com, odry@vts.su.ac.rs

Abstract — While studying the movement of driven robots a four-wheeled car is one of the most commonly used and preferred models. In the modeling process it can be assumed that the motion occurs in a two-dimensional space and the object can be driven forward or backward. In the foreseen task, the robot must reach a target point by some predefined specific control requirements. The simplified kinematic bicycle model of a four-wheeled robot car, mentioned in [1] has been upgraded for the purposes of this research. It has been observed that the simplified model was not sufficiently adequate and accurate. Reference [1], besides the previous simplified model, also contains a P route controller. A new Fuzzy route control has been applied, because it was more customizable compared to the simple PID control. This article describes a comparison of the results between P and Fuzzy controllers. Based on the results obtained it has been concluded that it would be worthwhile to further develop this model and control system.

I. INTRODUCTION

Nowadays there are a large number of implemented land, water and air robots. Any robot structure may be in focus, one of the most important tasks is the appropriate moves of the robot in space. For this it is always important to consider the number of degrees of freedom, in which the object can move. The number of drives that can be assigned to a robot will determine its degrees of freedom. If the robot is equipped with as many drives, as its number of degrees of freedom, then it can achieve all the possible directions for movement. In many cases there is no need to assign a drive for every freedom, because the robot works equally well with less drives. For example, it is the case with this car which has been the subject of study in this paper. This vehicle has three degrees of freedom, but is only equipped with two drives. In many cases gravity is counted as an additional drive [1, 2].

As an example, the train travels along the rails, forward or backward. Each configuration corresponds to a situation that can be described by a scalar quantity $q$. The set of all possible configurations is the configuration space, or C space, denoted by $C$ and $q \in C$. The train has one degree of freedom since $q$ is a scalar.

The train is also equipped with one actuator that propels it forwards or backwards along the rail. It can be seen that the train is unable to move sideways. Similarly this can be observed in the case of cars, boats, and airplanes. However, it is not impossible that these vehicles enter into a lateral position. You can think of a car parking or mooring of a ship in a port. If the vehicle follows a suitable route and performs specific maneuvers, it can move sideways [1].

The configuration of a car in a specific plane can be described by three generalized coordinates $q=(x, y, \theta)$. The car has only two actuators, one less than its degrees of freedom, and it is therefore an under-actuated system. This imposes limitations on the way in which it can move. One disadvantage is that the vehicle cannot directly get to any point of the configuration space with the shortest direct route, but it needs to follow some other path. Despite the limitations the car is the simplest and most efficiently controlled vehicle.

In robotics such a car is described as a non-holonomic vehicle. It means that the motion of the car is subject to one or more non-holonomic constraints. A non-holonomic constraint can only be written in terms of the derivatives of the configuration variables and cannot be integrated to a constraint in terms of configuration variables [1].

The wheels one of the most important key elements of the car. The traditional, well known wheel can be excellently controlled. In case cornering friction occurs between the wheel and the road, that is very effectively counteracted by the centripetal acceleration which otherwise would require a separate drive.

II. THEORETICAL KINEMATIC MODEL OF BICYCLE

The scheme of the theoretical kinematic model of a bicycle can be seen on Fig. 1. The structure of the car was marked with light gray, while the structure of the bicycle with dark gray, the local coordinate system with red, and the global coordinate system with blue. The angle of the steering wheel is gamma ($\gamma$), the rear wheel speed $v$ is in direction $x$. The directions of the axis of the two wheels are marked with blue lines, and they intersect at point ICR (Instantaneous Centre of Rotation). The distance between the rear wheel from the ICR point is $R_1$, and from the front wheel, $R_2$. The bicycle’s rear wheel is attached to the body, while the front wheel can be steered [1].

The $x$, direction of the local coordinate system is the direction movement of the vehicle, while the starting point is at the middle of the rear shaft ($V'$ point). The
The speed of the vehicle in the direction is zero because the wheels cannot slip sideways.

\[ v_x = v, v_y = 0 \]  \hspace{1cm} (1)

The blue lines show the direction along which the wheels cannot move, i.e., the lines of no motion, and these intersect at the point known as ICR. The reference point of the vehicle thus follows a circular path and its angular velocity is

\[ \dot{\theta} = \frac{v}{R_1} \]  \hspace{1cm} (2)

From (2) the turning radius \( R_1 = L / \tan(\gamma) \), can be defined, where \( L \) is the shaft distance or the wheelbase. The mechanical steering angle \( \gamma \) is limited, and the maximum value of this dictates the minimum value of \( R_1 \). Note that \( R_2 > R_1 \) which means the front wheel must follow a longer path and therefore rotate more quickly than the back wheel.

When a four-wheeled vehicle goes around a corner on a corner the two steered wheels follow circular paths of different radii and therefore the angles of the steered wheels \( \gamma_L \) and \( \gamma_R \) should be slightly different. This problem is easy to solve with the frequently used Ackermann steering mechanism, which drives wheels with different speeds [1].

In the global coordinate system, the robot system can be written in the following equations (3, 4, 5), which describes the kinematic model [1, 4]:

\[ \dot{x} = v \cos \theta \]  \hspace{1cm} (3)

\[ \dot{y} = v \sin \theta \]  \hspace{1cm} (4)

\[ \dot{\theta} = \frac{v}{L} \tan \gamma \]  \hspace{1cm} (5)

The change of heading direction \( \dot{\theta} \) is referred to as the turn rate, the heading rate or yaw rate and can be measured by a gyroscope. It can also be deduced from the angular velocity of the wheels on the left- and right-hand sides of the vehicle.

Equations (3, 4, 5) capture some other important characteristics of a wheeled vehicle. When \( v = 0 \) then \( \dot{\theta} = 0 \), that is, it is not possible to change the vehicle's orientation when it is not moving. It is a generally accepted fact that something must move in order to turn.

If the steering angle is \( \frac{\pi}{2} \), then the front wheel is orthogonal to the back wheel, the vehicle cannot move forward and the model enters an undefined region [1].

III. BUILDING THE KINEMATIC MODEL

Fig. 2 shows a kinematic model of the bicycle, implemented in Simulink based on (3), (4) and (5). There are two inputs: the wheel speed (\( v \)) and steering angle (\( \gamma \)). There is defined lower and upper limit for each input. This is built into a Subsystem that is named “Bicycle”, and that block mask has also been used with the following parameters: \( X_{\text{star}}[\text{m}], Y_{\text{star}}[\text{m}], V_{\text{min}}[\text{m/s}], V_{\text{max}}[\text{m/s}], \text{ShaftDistance}[\text{m}], \text{GammaMin}[\text{rad}], \text{GammaMax}[\text{rad}] \).

Reference [1] contains a simplified kinematic model of the bicycle (Fig. 3), but the model used in this paper is a fully completed one. On Fig. 3 it can be seen that the model does not correspond to (5). The tangent of the angle calculation and the division with shaft distance (\( L \)) are missing.
Fig. 4 shows an example where the simulation runs with the same parameters on the simplified and upgraded model. The above demonstration illustrates that bicycle has turned on different routes since the shaft distance is \( L = 0.5 \) m.

IV. DRIVING OF THE MODEL

Depending on requirements the robot can achieve a number of regulation options. The goal can be to reach a certain point, trajectory tracking, tracking a moving target, tracking a line, turning into a predetermined direction (angle), etc. In addition, you can choose one from the number of controllers. Following the p controller in [1], the authors kept the regulation option to reach a target point, but applied a fuzzy controller (fig. 5).

This is a feedback system to ensure that the vehicle heads for the specified target, according to the conditions specified in the Fuzzy controller. It compares \( x_B \) and \( y_B \) outputs of the bicycle kinematic model with the \( x_D \) and \( y_D \) (desired) target coordinates. The results of subtraction are \( x_E \) and \( y_E \) (error) which show how far away the robot is from the goal:

\[
\begin{align*}
    x_E &= x_D - x_B \\ 
    y_E &= y_D - y_B
\end{align*}
\] (6) (7)

Since the controller needs to know how far the currently located vehicle is from the target, the distance needs to be calculated (10). It is necessary to know the deviation between the angle of the robot’s direction (\( \theta \)) as well as the proper angle leading to the goal (\( \alpha \)). Equations (8) and (9) can be written with the help of the diagram on Fig. 6.

\[
\begin{align*}
    \alpha &= \tan^{-1}\left(\frac{y_E}{x_E}\right) \\ 
    \varphi &= \theta - \alpha
\end{align*}
\] (8) (9)
\[
    d = \sqrt{x_E^2 + y_E^2}
\] (10)

The block CompareToConstant in the model checks whether the distance is greater than the specified tolerance value (goaltol). If this distance is smaller, then its output will be zero and thus immediately reset speed to
If this lock is disabled, the robot will also reach the finish line, but will go further.

V. BUILDING FUZZY ROUTE CONTROLLER

There are two inputs as illustrated on Fig. 7: \(d\) as the distance and \(\phi\) as the angle of direction error. The outputs are \(v\) as velocity and \(\gamma\) as the angle of the steering wheel, with which the model is controlled. These will be the inputs of the kinematic model. \(\text{Trimf}\) was chosen as a membership function for its easy use and implementation [3]. Other parameters were left as default values, because there was no purpose in their detailed study and optimization.

Two Membership Functions were created for the two types of ratings determined for the distance (Fig. 8): either close to the goal (\(\text{small}\)) or away from the goal (\(\text{big}\)). The two membership functions have been set up in the form as shown, so that the \(\text{small}\) value can influence the output with a higher percentage. It is also evident that this function is activated from a 5m distance. This serves the purpose to gradually decrease the speed of the vehicle. It has been determined with the help of Fuzzy rules that if the distance is small, the speed will be reduced.

The second input is the \(\phi\) angle having the following value domain: \([-\pi, +\pi]\). This is the angle, which shows how much the deviation is between the target and the orientation of the car. There are three values: right, left, or straight. Thus indicating that relative to the target the car is to the right, left or straight. The following rules have been set up:

- If the robot-car stands in the left direction, the speed is reduced and it turns to the right.
- If the robot-car stands in the right direction, the speed is reduced and it turns to the left.
- If the robot-car stands in the target’s direction, then it moves forward with high-speed.

The output speed (\(v\)) can be slow or fast (Fig. 9). Based on the rules laid down the car's speed will take values in the given [0 30] m/s range. The range of values in this case is just one example.

The second output is the \(\gamma\), which is adjusted to the second input (\(\phi\)) described with three membership functions: go left (\(\text{goLeft}\)), go straight (\(\text{straight}\)) or go to the right (\(\text{goRight}\)). Its limit is consistent with the boundaries of the steering wheel defined in the model with variables \(\text{GammaMin}\) and \(\text{GammaMax}\). In this example it is \([-1 +1]\) radian.

Fig. 10 demonstrates the rules set up for the Fuzzy controller.

Fig. 11 shows the Fuzzy’s surface of the output in terms of speed. In the three-dimensional coordinate system the surface displays all the possible values of speed which depend on two inputs: \(\phi\) and \(d\). The speed viewed from the \(\phi\) angle parameter is the maximum value, when the car is moving towards the goal line. The speed is less when the vehicle turns more to the left or right. The rule of speed in terms of distance is: the greater the distance the higher the speed.
VI. COMPARISON OF P AND FUZZY CONTROLLER

In order to compare the P and Fuzzy controller both of them had to be built. Fig. 12 shows the P controller implementation from [1]. The values of the P controller (0.5 for speed and 4 for the angle) are not derived from reference [1], it has been assumed that the author defined these two values empirically.

In the example shown, the initial parameters are as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_{\text{time}}$</td>
<td>5</td>
</tr>
<tr>
<td>$x_{\text{des}}$</td>
<td>-30</td>
</tr>
<tr>
<td>$y_{\text{des}}$</td>
<td>15</td>
</tr>
<tr>
<td>$x_{\text{start}}$</td>
<td>0</td>
</tr>
<tr>
<td>$y_{\text{start}}$</td>
<td>0</td>
</tr>
<tr>
<td>$v_{\text{min}}$</td>
<td>0</td>
</tr>
<tr>
<td>$v_{\text{max}}$</td>
<td>40</td>
</tr>
<tr>
<td>$\text{shaft}_{d}$</td>
<td>0.5</td>
</tr>
<tr>
<td>$\text{gammamin}$</td>
<td>-0.2</td>
</tr>
<tr>
<td>$\text{gammamax}$</td>
<td>0.2</td>
</tr>
<tr>
<td>$\text{goal}_{tol}$</td>
<td>0.3</td>
</tr>
</tbody>
</table>

Fig. 13 shows the paths taken by the vehicle to be almost identical with the difference that the P controller is not driven to the finish during the given time (5 sec). This is because the P controller always slows down the vehicle as it approaches the goal.

The angle changes very abruptly at P solution (Fig. 14.). The Fuzzy controller changes the steering angle more precisely, so it turns the vehicle more smoothly toward the target. Fig. 14 also shows that the P controller gives higher speed at the beginning, and then it slows the vehicle down as it rolls towards the goal. However the vehicle never reaches the target. The Fuzzy keeps a slower speed while the object turns towards the target, then as it finds the right angle, it increases the speed of the car. At the end of the road it will slow down as it is pre-determined with rules.
VII. CONCLUSION

The simplified kinematic bicycle model of a four-wheeled robot car, mentioned in [1] has been upgraded for the purposes of this research. It has been observed that the simplified model was not sufficiently adequate and accurate. In addition we have developed a Fuzzy controller which operates in accordance with our requirements. The described example illustrates the differences between the two controllers, their advantages and disadvantages. The Fuzzy controller shows a better solution in terms of speed, because its behavior is closer to reality. The advantage of Fuzzy is that it can be customized by rules to the desired behavior. Of course, Fuzzy required more calculations than the P controller.

In this paper a fitness analysis for the current example has been performed, which can be calculated from the simulation results (Table I.). Based on the fitness evaluation the Fuzzy also demonstrates a better value.

<table>
<thead>
<tr>
<th>Property of driving</th>
<th>Description</th>
<th>Formula</th>
<th>P controller</th>
<th>Fuzzy controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>completion time</td>
<td>Driving is better if completion time is shorter</td>
<td>$t_c = t \mid d \leq 0.5 m$</td>
<td>8.76</td>
<td>2.48</td>
</tr>
<tr>
<td>average speed</td>
<td>Magnitude of speed normalizes the other properties</td>
<td>$\bar{v} = \frac{1}{t_c} \int_{t=0}^{t_c} v(t)dt$</td>
<td>4.51</td>
<td>15.52</td>
</tr>
<tr>
<td>mean square of acceleration</td>
<td>Driving is better if acceleration is smaller and without peaks</td>
<td>$e = \frac{1}{t_c} \left( \int \frac{(dv)}{dt} \right)^2 dt$</td>
<td>2095</td>
<td>779.6</td>
</tr>
<tr>
<td>fitness function</td>
<td>The value which indicates effectiveness of driving or controlling.</td>
<td>$F = \frac{10^3}{t_c \cdot \bar{v} \cdot e}$</td>
<td>1.207</td>
<td>3.324</td>
</tr>
</tbody>
</table>

REFERENCES


