ABSTRACT
The paper is focused on the planned fuzzy-interpolative controllers. Such controllers are pushing away the main disadvantage of the soft computing approach: the lack of accuracy and of optimizing criteria. The study case refers to the car following algorithm. A new distance gap planner is proposed. The planner is designed on the constant time to collision criteria. The constant time to collision method perform almost similarly to the existing methods but can be implemented much easier, by interpolative networks, and can coordinate in a safe manner the whole traffic flow.

KEYWORDS: Fuzzy-interpolative distance control, Distance-velocity mapping, Distance planner, Time to collision, Constant time to collision.

1. INTRODUCTION
The Soft Computing (SC) emerged as result of the necessity to enable computers with human-like reasoning. That means that SC is able to operate with symbolic represented knowledge, analyzed in a qualitative manner. Thanks to SC, systems presenting different kind of uncertainties that are barely accessible to conventional approach become controllable. The main SC methods (fuzzy logic, neural networks, genetic algorithms, etc.) are usually used in a flexible way, often merged in different hybrid sub-sequential methods. SC is complementary to the classic control, and therefore their merging offers great perspectives.

A basic assumption is that simple SC controllers are able to operate with excellent results if they are disposing of significant knowledge about the controlled plant. This knowledge becomes compatible with computers if it can be modeled. Models can be used either directly (internal models included into the control algorithm) or indirectly, by assisting the design of the Planning Systems (PS) [1]. The planner used by PS is embedding relevant knowledge about the plant and about the way we want the system to perform.

In a previous paper [5] a railway coach model assisted the building of a distance-velocity mapping, used as a planner for a special brake controller. This controller is able to stop the vehicle at an imposed position, maintaining all the time an almost constant braking effort. The distance-velocity mapping is representing the evolution of the vehicle in the case of a constant braking force. The control of the brake becomes extremely simple because the system is asked to perform closely to its natural behavior. Undesirable over or under-brakings are such way avoided. The aim of this paper is to bring new arguments underlining the matching between SC controllers and PS.

A specific SC item is used: The Fuzzy-Interpolative Controller (FIC) [4, 5, 6], etc. As part of SC, FICs can cope with an extremely wide range of applications. Their main advantages are: versatility, fast and easy development, simple and cheap implementations, etc. FICs are building a bridge between theory people and the microcontroller programmers.

The case study is focused on the optimization of the distance-gap between cars, by means of a nonlinear distance-velocity mapping, developing a previous paper [5] where a linear mapping was used.
2. THE PLANNED FUZZY-INTERPOLATIVE DISTANCE CONTROLLER

We will consider the simplest version of the car following problem: two cars, Car1 and Car2 that is following Car 1. The PD planned fuzzy-interpolative distance controller was introduced in [5]. It consists into a McVicar-Whelan rule base. The input variables are the distance $d$ between the cars and its derivative $cd$. The output variable is the force $F$, positive for traction and negative for braking. All the fuzzyfications are done using fuzzy partitions with triangular shape membership functions. Prod-sum inference and COG defuzzification are applied.

<table>
<thead>
<tr>
<th>$d$</th>
<th>very small</th>
<th>small</th>
<th>medium</th>
<th>great</th>
</tr>
</thead>
<tbody>
<tr>
<td>positive</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>great</td>
<td>zero</td>
<td>positive small</td>
<td>positive great</td>
<td>positive great</td>
</tr>
<tr>
<td>positive</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>small</td>
<td>negative small</td>
<td>zero</td>
<td>positive small</td>
<td>positive great</td>
</tr>
<tr>
<td>zero</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>negative</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>small</td>
<td>negative great</td>
<td>negative small</td>
<td>zero</td>
<td>positive small</td>
</tr>
<tr>
<td>negative</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
</tr>
<tr>
<td>great</td>
<td>negative great</td>
<td>negative small</td>
<td>negative small</td>
<td>zero</td>
</tr>
</tbody>
</table>

Table 1. The control rules of the distance controller

According to the fuzzy-interpolative technique, this fuzzy controller can be implemented by the following look-up-table with linear interpolations (written in SIMULINK-MATLAB syntax):

\[
\begin{align*}
\text{row } [d \text{ (m)}]: & \quad [5 \ 10 \ 25 \ 50] \\
\text{column } [cd \text{ (m/s)}]: & \quad [-1.5 \ -0.5 \ 0 \ 0.5 \ 1.5] \\
\text{output } [F \text{ (kN)}]: & \quad [-4 \ -4 \ -1 \ -1 \ 0; \ -4 \ -1 \ 0 \ 0 \ 1; \ -1 \ 0 \ 0 \ 1 \ 4; \ 0 \ 1 \ 1 \ 4 \ 4]
\end{align*}
\]

The controller is nonlinear. In emergency cases its action is strong ($F_{\text{max}} = -4 \text{ kN}$), while in usual cases $\text{abs}(F)$ is not exceeding 1 kN. Due to the fuzzy nature (implemented by a linear interpolation network) the control surface $F(d, cd)$ is continuous. In this primary version, installed on Car2, the controller is able to avoid collisions, but it has an obvious tendency to produce oscillations. The oscillations are caused by the fact that $d$ is not adapted to $v_2$, the velocity of Car2, as one can see in Eq. (1). The very small, small, medium, and great distances are centered on 5, 10, 25 and 50 meters, with no adaptation to the velocity.

One answer to this quest was verified in [5] by planning the distance between the cars according to $v_2$. A linear imposed distance-velocity mapping $d_i(v_2)$ was used:

\[
\begin{align*}
\text{input } [v_2 \text{ (km/h)}]: & \quad [0 \ 5 \ 200] \\
\text{output } [d_i \text{ (m)}]: & \quad [5 \ 5 \ 15]
\end{align*}
\]

The same controller can be used, replacing the input variables $d$ and $cd$ by the distance error $e$ and its derivate $ce$. This way the performance of the controller is quite satisfactory because the planning is significantly easing the task of any possible controller.

Extrapolating this conclusion one can anticipate that the association with planners could catalyze the economic effect of the simple, robust and versatile SC controllers.
3. SAFETY MEASURES – THE TIME TO COLLISION INDICATOR

Besides the improving of the driver’s comfort and safety the in-vehicle systems have to integrate themselves into hierarchical systems designed to improve the traffic flow as well as the traffic safety. Recent technological developments are emerging in the field of automate driving such as the Autonomous Intelligent Cruise Control (AICC) or the Collision Avoidance Systems (CAS). We can distinguish direct safety benefits (enhanced driving performance and minimization of crash consequences, etc.) and indirect safety benefits (reduced driver stress and fatigue, reduced conflicts and variance in behavior, etc.). On the other hand, any approach in this domain must be carefully balanced because of the safety risks (driver distraction, reduced situation awareness, loss of skill, etc.). A set of facilities with different degrees of implication in the driving action are clustered into the so-called Advance Driver Assistance Systems (ADAS) [2, 3].

Several indicators measure the characteristics of the traffic flow: the time-to-collision (TTC), the time-to-accident (TTA), the post-encroachment-time (PET), the deceleration-to-safety-time (DTS), the number of shockwaves, etc. TTC is the time that it takes before Car2 collide with Car1, assuming unchanged speeds of both vehicles during this approach. Negative TTC imply that the vehicle in front drives faster, i.e. there is no danger. Positive TTC express a certain unsafe approach. TTC is linked to the longitudinal driving task and supports a set of safety techniques as the Autonomous Intelligent Cruise Control (AICC) assisting the driver to keep a safe distance to his predecessor and the Intelligent Speed Adapter (ISA) limiting the speed to the prevailing speed limit of the current road section under all circumstances [3].

Among other specifications referring to speed, acceleration and deceleration limitations, AICC is introducing a particular \( d_i(v_2) \) law:

\[
d_i(v_2) = z_0 + z_1 \cdot v_2 + z_2 \cdot v_2^2 = 3 + z_1 \cdot v_2 + 0.01 \cdot v_2^2
\]  

(3)

Several settings are used, for example \( z_1 = 0.8 \) s or \( z_1 = 0.6 \) s. The acceleration and deceleration stem from a car-following law, aiming to control the gap towards the desired values.

Starting from TTC one can introduce global indicators such as the Time-to-Collision distribution or the Cumulative TTC Exposure Times. By assessing TTC values at regular time steps or in continuous time, a TTC trajectory of a vehicle can be determined. Doing this for all vehicles present on a road segment one can determine the frequency (or absolutely, the exposure time) of the occurrence of certain TTC values, and by comparing these distributions or its cumulative counterpart for different scenarios, one can appreciate the traffic safety [3].

4. THE CONSTANT TIME TO COLLISION

4.1 Stabilizing TTC

A step forward in the use of TCC is to control Car2 with the purpose to stabilize TTC. This regime can be called Constant Time To Collision (CTTC) and it has obvious advantages:

- the best possible time evolution of the collision risk for a single vehicle (constant);
- the best possible distribution of the collision risk for all the running cars (constant);
- the possibility to control in a simple yet effective manner the traffic on extended road sections: long TTCs means high traffic safety while short TTCs means high traffic density;
- CTTC can be realized by each vehicle by itself, free of external conditionings.

There are two ways to obtain CTTC: a) the on-line estimation of TTC as \( d_i(v_2-v_1) \) and its control by a TTC controller and b) the off-line stabilization of TTC by \( d_i(v_2) \) mappings.

The on-line version principle seems extremely simple at the first glance, but it can be applied only when \( d \) is changing, because TTC is infinite when \( v_2 - v_1 = 0 \). The implementation is asking highly accurate and fast procedures because of the time delays of the distance sensor cumulated with the time demanded by the arithmetic operations and the control algorithm.

Although the off-line version seems a bit more complicate, this time the greatest part of the operations, included the control algorithm, consists only in picking-up numbers from look-up tables and linear interpolations.
4.2 CTTC by distance gap planners

The assumption supporting the CTTC by distance gap planner’s method (CTTCDP) is the following: a distance gap planner build by preserving constant TTC will produce CTTC.

The method can be investigated by computer simulations, using a simple model of the tandem Car1-Car2, used in other previous papers [vvvv], as shown in fig. 1.

Figure 1. A SIMULINK-MATLAB model of the tandem Car1-Car2

Accurate knowledge about the specific parameters of Car2 (traction, brakes, weight, aerodynamic coefficient, etc.) can be such way taken into account, comparing to the simplified and leveling analytic representation (3). The disadvantage raised by the complexity of the model is avoided by using it only for the design of the \(d_i(v_2)\) mapping that is replacing equation (3).

Obtaining a CTTC \(d_i(v_2)\) mapping can be done by a back propagation procedure, replacing the distance controller by a TTC controller (essentially by replacing the output \(d\) of the TTC block by TTC). In order to avoid the \(v_2 - v_1 = 0\) case the simulation scenario will include only constant force accelerations/brakings of Car1. The recorded \(d\) represents the searched planner \(d_i\) for a certain TTC. This curve can be easily approximate by 1D look-up tables, as shown in figure 2.

Similar curves result for any desired TTC. One can this way obtain 2D look-up tables for \(d_i(v_2, TTC)\) mappings, opening the perspective of the adaptation of the traffic flow by means of different TTCs imposed by the highway’s administrations, according to the traffic conditions.

5. SIMULATION RESULTS

In figure 3 a simulation is illustrating CTCC. One can compare \(v_1\) with two different speeds of Car2: one obtained using a linear \(d_i(v_2)\) mapping, the other using a TTC \(d_i(v_2)\) mapping. The higher accuracy of CTTC is obvious. In figure 4 one observes the \(d\) evolution in the two cases, CTTC’s optimizing effect being again evident (much smaller distance gap in overall). The simulation was performed for identical cars, for a high traffic flow, at a very high top speed in order to verify the safety of the method.

The differences between CTTC and AICC – equation (3) - are almost unobservable, especially at normal speeds. Other simulations showed that the differences between the fuzzy-interpolative \(d\) controller and a linear PID are also almost unobservable, for correct adjustments.
Figure 2. A CTTC $d_i(v_2)$ mapping

Figure 3. A CTTC simulation
Figure 4. A comparison of the distances issued from the linear programmer and from CTTC

3. CONCLUSIONS

The fuzzy-interpolative controllers, as a soft computing issue, are very simple, cheap and easy to use, but they are not optimal. Their disadvantage can be put away if they are associated with planners, able to optimize their operation on the base of previous knowledge about the system.

This approach is exemplified for the car following case, by introducing a distance gap planner that is designed such way that the time to collision is controllable. Comparing to the existing methods, the constant time to collision is able to optimize the traffic flow taking in account the specific parameters of each individual car, in an optimal yet simple manner.

Comparing to other controllers, the planned fuzzy-interpolative ones can offer optimal performances at extremely low costs in every aspect (computing resources, time, etc.)

5. REFERENCES


The references will be detailed and developed for the final version