Autonomous RC Cars for Control Research and Education
Implementation of Virtual Potential Based Navigation and Platooning

De Jager, T. R.; Meinders, N. K.; Van Vugt, T. A.; Zomerdijk, W.; Ferrari, R. M.G.

DOI
10.1109/CCTA49430.2022.9966019

Publication date
2022

Document Version
Final published version

Published in
Proceedings 2022 IEEE Conference on Control Technology and Applications, CCTA 2022

2022 IEEE Conference on Control Technology and Applications (CCTA)
August 22-25, 2022. Trieste, Italy

Autonomous RC Cars for Control Research and Education: Implementation of Virtual Potential Based Navigation and Platooning

T.R. de Jager*† N.K. Meinders*† T.A. van Vugt*† W. Zomerdijk*†
R.M.G. Ferrari*

*Dept. of Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands.

Abstract— **Autonomous and collaborative vehicles not only are seen as a possible solution to reducing congestion and traffic related fatalities. They also provide an excellent multi-domain test bench for engineering education at undergraduate and graduate level. Yet, the use of real scale platforms for experimental educational activities bears prohibitive costs and complexity. While several small scale autonomous platforms have been developed in recent years to address this issue, still they require a significant investment of time and money, which is not always ideal for undergraduate education. Furthermore, none of the available platforms are specifically developed for platooning experiments. In this paper, we detail the results of an undergraduate student's project where a pair of relatively low-cost, off-the-shelf small scale RC cars have been used to implement and test a well known platooning algorithm from the literature. Furthermore, a Virtual Potential Field (VPF) based lateral controller has been included in order to allow the cars to navigate a prescribed closed-circuit track. Self-location of each car has been obtained via a ceiling-mounted motion capture system. Results have shown that, even using a relatively low sampling rate of 10 Hz, accuracies in the order of 1 cm can be obtained when platooning at 0.5 m/s along a circuit of 4 by 3 m. As further improvements to the platform, apart from higher sampling rates for the control law, the inclusion of onboard perception is being explored, in order to eliminate the need for an external motion capture system.**

I. Introduction

In populous urban areas across the world, road congestion, pollution, and traffic accidents caused by human error are of increasing concern. In order to ease such negative situations, solutions based on autonomous and collaborative vehicles are being broadly researched at academic and industrial laboratories [1], [2]. Furthermore, according to a recent article by The New York Times, 108 companies in the U.S. alone are using public roads as a "lab for self driving experiments" [3]. These considerations support the need for including the design of control and perception algorithms of autonomous vehicles in the curricula of engineering education [4], [5], [6], [7].

Furthermore, as recognised by [7], robots and, by extension, autonomous vehicles constitute an excellent multi-domain test bench for developing in undergraduate students skills and topics such as design thinking, analysis, problem solving, perception, control and planning. Yet the cost and complexity of real scale platforms for testing and developing autonomous vehicle solutions are prohibitive for educational settings. This led in recent years to the development of several, mostly open-source small scale autonomous platforms. Notable examples include, ordered by increasing cost and complexity: the Donkey Car[1]; the Berkeley Autonomous Race Car (BARC) [8]; the MIT Racecar [4]; the F1/10 project [9], [6]; the AutoRally [10]. While such small scale autonomous vehicles include increasingly sophisticated arrays of sensors and powerful onboard processing, their cost and required investment of time can still be high, especially for educational activities. Furthermore, none of these platforms are specifically designed for cooperative driving functionalities, such as platooning.

In order to address these issues, in this paper we explored the use of a relatively inexpensive, off-the-shelf autonomous RC car for undergraduate education in autonomous, cooperative navigation. In particular, we will present the results obtained by a team of four undergraduate students in Mechanical Engineering, which were tasked with implementing a proof of concept platooning experiments using such small scale autonomous cars. In particular, the students' project involved the following tasks:

1) to develop a simple navigation algorithm such that the platoon can drive along a predefined circuit;
2) to design a lateral control law, based on VPF, that the cars in the platoon can use to drive along a predefined circuit;
3) to obtain a kinematic model of the cars that can account for their nonlinear, non-holonomic behaviour, which can be used for computing steering actions;
4) to implement a longitudinal control law that allows the cars to form and maintain a platoon;
5) to validate in an experimental setting the lateral and longitudinal control laws developed.

In section II background information on the modelling and control approaches used in this research are presented. Then, section III describes the experimental setting and the methodology used to evaluate the students' design. Furthermore, section IV presents and discusses the results, and possible origins of nonidealities within the research. Lastly, the conclusions and future recommendations are drawn in section V.

II. Theoretical Background

This section presents the theoretical foundations on which the car's individual and cooperative control laws have been developed.

† These authors contributed equally.

[1] `https://www.donkeycar.com`

2022 IEEE Conference on Control Technology and Applications (CCTA) | 978-1-6654-7338-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/CCTA49430.2022.9966019

A. Virtual Potential Field

The concept of VPF has been successfully used for navigation and track following of autonomous vehicles or robots [11], [12], [13]. The intuition behind VPF is to use an artificial potential to generate forces that will keep an autonomous vehicle on the intended path, similar to how magnetic fields are used inside particle accelerators. In particular, a VPF can be computed as the combination of both repulsive and attractive components. The former are employed to steer a vehicle away from static and dynamic obstacles, such as the road boundaries or other vehicles. The latter are used to drive the vehicle towards a desired position, such as a point along the intended track to follow, or the intended position of the vehicle as part of a platoon. Once such an artificial VPF is defined, the resulting forces can be computed as the negative gradient of the VPF and translated into actual steering and acceleration commands.

In this work, a combination of Gaussian potentials are used, as in [14], to generate lateral forces for steering the car towards a desired path along a given track. Assuming a straight track in the (x, y) plane, aligned with the x direction, the VPF value, $U_{r,sr}$, in Equation 1 will cause the car to steer towards the centre of the track, with lateral coordinate denoted by y_c.

$$U_{r,sr} = A_r(1 - e^{-b_r(y-y_c)})^2 \tag{1}$$

A_r is a scaling factor, with larger values giving a higher amplitude of the VPF. b_r corresponds to the steepness of the Gaussian curve with a larger value increasing the steepness. By combining several of such potentials, each one aligned with a different portion of a given circuit, it is possible to obtain an overall VPF that enforces tracking of a path along that circuit (see Figure 2). The intuition is that Equation 1 attains its minimum at the centre line of the circuit, while it increases as a car moves laterally towards the lane boundaries.

B. Bicycle model

In order to translate the virtual lateral force into an actual force steering the car, a simplified kinematic model, called bicycle model, has been used [15]. Equation 2 and 3 are prescribed by the bicycle model, assuming no lateral slip.

$$\dot{x}_f sin(\theta + \delta) - \dot{y}_f cos(\theta + \delta) = 0 \tag{2}$$

$$\dot{x} sin(\theta) - \dot{y} cos(\theta) = 0 \tag{3}$$

Where (x, y) and (x_f, y_f) are the rear and front wheel coordinates, respectively, and θ is the orientation of the car, all of these described in the global frame. δ is the steering angle of the front wheel. The distance between the front and rear wheel is L, thus the orientation of the car (x_f, y_f) can be expressed as $x_f = x + Lcos(\theta)$ and $y_f = y + Lsin(\theta)$. Substituting this into Equation 2 gives Equation 4.

$$\dot{x} sin(\theta + \delta) - \dot{y} cos(\theta + \delta) - \dot{\theta} L cos(\delta) = 0 \tag{4}$$

Also provided that $\dot{x} = vcos(\theta)$ and $\dot{y} = vsin(\theta)$, where v is the velocity of the car, results in Equation 5.

$$\dot{\theta} = \frac{vtan(\delta)}{L} \tag{5}$$

Besides, it is known that $\dot{\theta} = \frac{v}{R}$, where R is the radius of the curve that the car would make with a certain steering angle. Combining this with Equation 5 gives $tan(\delta) = \frac{L}{R}$.

The potential field gives a virtual force, F_{pf}, that is acting on the car in the lateral direction. This virtual force leads to a physical force, the centripetal force, given by $F_c = \frac{mv^2}{R}$, where m is the mass of the car. As explained in subsection II-A, the virtual force must be equal to the centripetal force. As the car mass is known and its longitudinal velocity can be measured, the turning radius can be obtained from the VPF and Equation 6 is derived for the steering angle.

$$\delta = tan^{-1}\left(\frac{LF_{pf}}{mv^2}\right) \tag{6}$$

By designing a control law that tracks this desired δ, the car can track a given circuit, as detailed in subsection III-E.

C. Longitudinal control and Platooning

Cooperative Adaptive Cruise Control (CACC) [1], [16] will be used to design the longitudinal velocity control law. As a proof-of-concept only two cars will be considered: a leader car, whose velocity, v_L, will follow a prescribed profile, and a follower car, which will employ CACC to track the reference distance from the leader given by Equation 7.

$$d_r(t) = r + hv_F(t) \tag{7}$$

Where d_r is the desired distance between the cars, r is the standstill distance, h is the time gap and $v_F(t)$ is the velocity of the follower car. The distance error $e(t)$ is computed using Equation 8, where d is the actual distance between the cars.

$$e(t) = d - d_r(t) \tag{8}$$

Combining Equation 7 and 8 leads to Equation 9, where $p_L(t)$ and $p_F(t)$ are the positions of the leader and follower cars and L their length. The time derivative of the error is given by Equation 10.

$$e(t) = p_L(t) - p_F(t) - L - r - hv_F(t) \tag{9}$$

$$\dot{e}(t) = v_L(t) - v_F(t) - ha_F(t) \tag{10}$$

In [16] the desired acceleration, u_F, of the follower car is computed from the control law given by Equation 11.

$$\dot{u}_F(t) = \frac{1}{h}\left[-u_F(t) + (k_p e(t) + k_d \dot{e}(t)) + u_L(t)\right] \tag{11}$$

Where u_L is the known desired acceleration of the leader, assumed to be communicated to the follower, and k_p and k_d are control gains. Finally, the relation between the follower's desired acceleration and its actual one, a_F, is given by Equation 12.

$$\dot{a}_F = \frac{1}{\tau}(u_F(t) - a_F(t)) \tag{12}$$

Where τ is a time constant accounting for the drive-train dynamics.

III. Method

The method and experimental setup used in this paper are explained in this section.

A. Erle-Rover and -Brain

The Erle-Rover is a Linux-based unmanned ground vehicle with an artificial robotic brain (see bottom-right of Figure 3). It uses the Robot Operating System (ROS) and Ardupilot (APM).

B. Motion capture system

In order to accurately track the position of the Rover, an Optitrack Motion Capture System (Mocap), in combination with the software Motive, was used. The system consists of a series of IR cameras, which are used to track reflective markers attached to the Rover.]

C. Circuit

The circuit consists of curved and straight segments. The grid for the circuit is made up of multiple sample grids of one meter segments, each consisting of multiple connected grid points. The amount of grid points in x and y direction and the width of the grid are adjustable. Different sub-grids are curved, translated, rotated and/or a scaled in length and connected to each other. This results in the sample grid illustrated in Figure 1. A more complex circuit with multiple turns was also created. The grid is defined in the global frame.

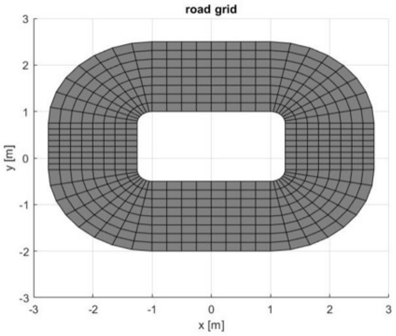


Fig. 1. Road grid made out of a combination of grid points that are connected.

D. Feedforward

A reference steering angle and longitudinal speed have been computed as a function of the position along the track. These are used as feedforward action in the cars' control law. This allows for using small angle approximations in the feedback action, as, during normal driving conditions, the cars will need to correct only small deviations from the track.

E. Feedback

A feedback action was implemented for making the lateral control robust to unmodelled dynamics and other nonidealities of the rover. As illustrated in Figure 3, the VPF gradient is computed as a function of actual longitudinal and lateral position along the circuit, and converted to a centripetal force that must be applied to the car. By means of the bycicle model, presented in subsection II-B, this is translated into a steering angle, which is fed through a PID transfer function and then added on the top of the nominal steering angle provided by the feedforward action.

For the sample grid explained in subsection III-C, the potential field is calculated using Equation 1. This sample potential field is plotted over each sub-grid, which gives the total potential field shown in Figure 2. The smoothness of the gradient of the potential field for a straight part depends only on the amount of grid points in the y-direction, as the potential field is constant in the x direction. However, for curved parts the potential field is not constant in the x direction.

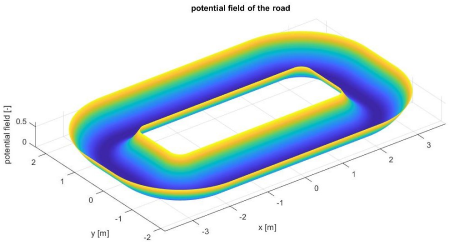


Fig. 2. Static potential field.

With the position measurement of the Mocap system, the grid point closest to the car can be found. At this grid point, the gradient of the potential field will be calculated. Based on the position of the rover relative to the grid point, the gradient is calculated using a forward, backward or central difference method.

F. Simulink model

A simplified version of the Simulink model used to control the leader car is shown in Figure 3. The feedforward has a velocity and a steering angle as output. If the car is not in the middle of the road, a virtual force from the potential field will act on it. The bicycle model translates this force to the desired steering angle. After the bicycle model, the PID-controller is placed to prevent oscillations, decrease the steady state error, and decrease the response time of the Rover. Subsequently, the calculated steering angle will be added to the feedforward and is translated to a so-called RC-override command sent to the rover via ROS.

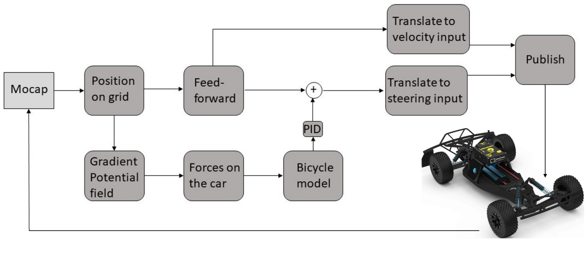


Fig. 3. Control scheme of the leader car (Erle-Rover with Erle-Brain [17]).

G. Follower car

A simplified version of the Simulink model with the most important control flows, based on formulas for CACC given in subsection II-C, can be seen in Figure 4.

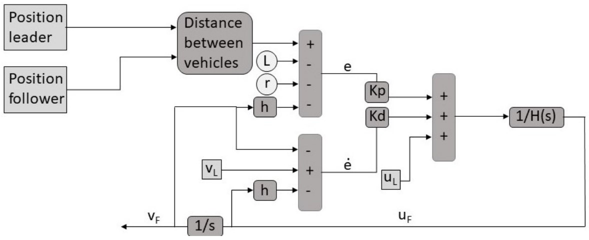


Fig. 4. Implementation of a CACC.

Authors in [16] present the output of the CACC as the desired acceleration, u_F. However, it is not possible to send an acceleration command to the Rover, only RC-override commands for the speed and steering angle can be sent. Therefore, the follower's desired acceleration, u_F, is integrated to obtain a desired velocity. The desired velocity is sent to the rover as an RC-override command.

The steering input for the follower car is calculated in the same way as described in subsection III-D and subsection III-E. The control scheme of the follower car is similar to Figure 3. The only difference is that the CACC controller directly uses the position on the grid, provided by the Mocap data, and translates it to a velocity input.

H. Software architecture

ROS, Matlab, and Matlab Simulink were used for controlling both the Rovers. Matlab and Matlab Simulink were used for implementing the control laws. ROS was used as a middleware between the computer running the control laws, the Mocap system, and the two Rovers.

I. Experiments

Before real-life testing, the Simulink model was verified in Gazebo and the potential field parameters were tuned using multiple virtual simulations. The accuracy of the system in real life was tested for different velocities. The optimum potential field parameters obtained from Gazebo were used in the experiments. Before the start of each test, a calibration was performed to accurately correlate the RC-override value with the velocity of the car. The performance of the follower car was evaluated based on the distance to the centre of the road of the car and the difference in distance between the leader and follower car and their desired distance.

IV. Results and discussion

The obtained results of the experiments are presented and discussed in this section. The parameters presented in Table I were used for performing simulation.

A. Accuracy

This section is divided into three subsections. In the first subsection the accuracy with different VPF parameters is given. The accuracy with different velocities and the accuracy for the follower car are given in the second and third subsection, respectively. The first two subsections provide the accuracy for following a predefined trajectory and the last subsection provides the accuracy for keeping a formation.

TABLE I
Parameters used for the experiments.

Parameter	Value
# points in X direction	141 [-]
# points in Y direction	151 [-]
width of the field for standard circuit	0.75 [m]
width of the field for complex circuit	0.30 [m]
dt Simulink model	0.1 [s]
stand still distance (r)	0.25 [m]
time gap (h)	0.50 [s]

1) Accuracy for different VPF parameters: The potential field is tuned by varying the values for the amplitude A_r and the steepness b_r. The tuning is done via experimental iteration, with a constant velocity of 0.5 m/s. The results for the circuit presented in Figure 1, are given in Table II. The goal is that the Mean Absolute Deviation (MAD) from the circuit is minimum. The MAD is determined to be the difference in the real and intended position of the car. The absolute difference is taken, such that the deviations do not cancel each other out. It can be concluded from Table II, that the minimum MAD is achieved when using $A_r = 20$ and $b_r = 0.3$. This combination also has the lowest peak value, defined as the maximum deviation from the circuit. The path of the Rover plotted on the circuit can be seen in Figure 5.

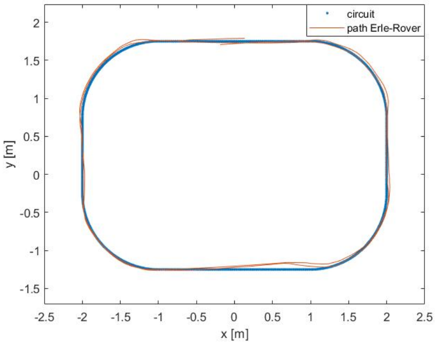


Fig. 5. Standard circuit, single Rover experiment ($Ar = 20$, $b_r = 0.30$).

When a combination of A_r and b_r leads to a field that is too strong, the absolute gradient of the potential field, defined as $\left|\frac{\Delta U}{\Delta S}\right|$ where U is the potential field value and S the position of a point, will rise substantially. As a result, even a small deviation from the circuit result in the maximum steering angle of the Rover, which causes the system to behave unstable. When the field is too weak, the steering angle of the Rover will be insufficient to return to its desired location and can result in a deviation larger than the width of the circuit. Both situations can be observed in Table II.

2) Accuracy with different velocities: Apart from a change in VPF parameters, the accuracy also depends on the velocity and the acceleration of the Rover. In Table III the accuracy is given for various velocities. It can be concluded that the accuracy decreases when the velocity increases. A possible reason for this is that signals were sent for fixed time

TABLE II

FOR DIFFERENT VALUES FOR A_r AND b_r, THE MAD AND THE PEAK DEVIATION FROM THE SIMPLE CIRCUIT ARE MEASURED.

A_r [-]	b_r [-]	MAD [mm]	Peak [mm]
10	0.30	$(7 \pm 7) \cdot 10^1$	$3.5 \cdot 10^2$
10	0.40	$(4 \pm 4) \cdot 10^1$	$1.9 \cdot 10^2$
15	0.30	$(7 \pm 7) \cdot 10^1$	$3.3 \cdot 10^2$
20	0.20	$(7 \pm 7) \cdot 10^1$	$2.9 \cdot 10^2$
20	0.25	$(5 \pm 5) \cdot 10^1$	$2.4 \cdot 10^2$
20	0.30	$(2 \pm 2) \cdot 10^1$	$9.2 \cdot 10^1$
20	0.35	$(6 \pm 7) \cdot 10^1$	$4.2 \cdot 10^2$
20	0.40	$(8 \pm 8) \cdot 10^1$	$3.2 \cdot 10^2$
25	0.30	$(4 \pm 3) \cdot 10^1$	$1.6 \cdot 10^2$
30	0.25	$(3 \pm 2) \cdot 10^1$	$1.4 \cdot 10^2$
30	0.30	$(5 \pm 4) \cdot 10^1$	$1.8 \cdot 10^2$
30	0.40	$(5 \pm 3) \cdot 10^1$	$1.5 \cdot 10^2$
40	0.25	$(4 \pm 3) \cdot 10^1$	$1.6 \cdot 10^2$
40	0.30	$(9 \pm 6) \cdot 10^1$	$3.2 \cdot 10^2$
80	0.25	$(8 \pm 7) \cdot 10^1$	$3.0 \cdot 10^2$

periods, resulting in bigger distances travelled i the same time period for higher velocities. This leads to a bigger deviation from the circuit. Also, the standard deviation increases with higher velocities, meaning that the results show oscillations. Another reason can be, that there is more slip at higher velocities.

TABLE III

FOR DIFFERENT VELOCITIES OF THE ROVER, THE MAD AND THE PEAK DEVIATION FROM THE SIMPLE CIRCUIT ARE MEASURED.

Velocity [m/s]	MAD [mm]	Peak [mm]
0.25	$(8 \pm 7) \cdot 10^0$	$6.2 \cdot 10^1$
0.50	$(2 \pm 1) \cdot 10^1$	$4.7 \cdot 10^1$
0.75	$(5 \pm 4) \cdot 10^1$	$1.7 \cdot 10^2$
1.00	$(1 \pm 1) \cdot 10^2$	$5.4 \cdot 10^2$
1.25	$(2 \pm 2) \cdot 10^2$	$9.7 \cdot 10^2$
1.50	$(4 \pm 2) \cdot 10^2$	$8.2 \cdot 10^2$

3) Accuracy while following a leader car: For determining the accuracy of the leader and follower car, a more complex circuit was used Figure 6. The velocity used in this experiment is a constant velocity of 0.5 m/s. For the leader car, the MAD is $(3 \pm 2) \cdot 10$ mm, the peak is 84 mm. The follower car had a MAD of $(2 \pm 2) \cdot 10^1$ mm, with a peak of 96 mm.

The second accuracy for following a leader car is defined as the actual distance between the cars on the circuit and the desired distance between the cars. Again the accuracy is defined by the MAD, which is $(4 \pm 2) \cdot 10^2$ mm. This is a rather high value in comparison with the MAD of following a predefined circuit. A first explanation for this result can be that only a RC-override command message, indirectly representing the velocity, can be sent to the Rover and not an acceleration value. A calibrated look up table was used, which cannot be perfectly smooth. As a consequence, the follower car cannot drive at the exact desired velocity. A second reason for the relative high MAD, is the velocity profile which is measured using Mocap. This causes some errors even though filtering using a moving average is used—otherwise the velocity profiles of the Rovers are very noisy. More about these errors caused by Mocap can be read in subsubsection IV-B.3. Thirdly, the control sampling time dt that we managed to steadily maintain in Simulink was 0.1 seconds. Considering the maximum speed at which the rover was tested was 1.5 m/s, it is easy to see that the sampling time used was too high.

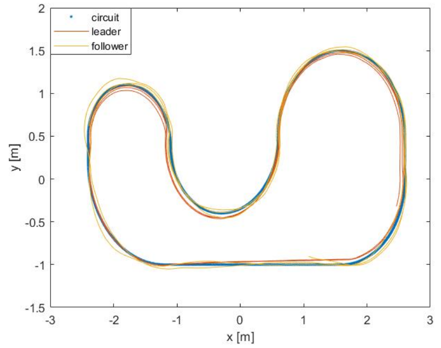


Fig. 6. Complex circuit, platooning experiment ($Ar = 20$, $b_r = 0.30$).

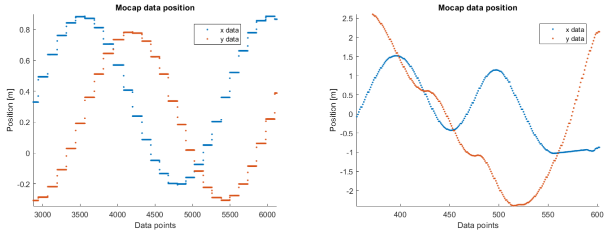
Furthermore, there is a delay between sending information to the Rover and the Rover responding to that information. This results in an error which causes a lower accuracy, as explained in subsubsection IV-B.1. Lastly, there will always be a calculation error when determining the accuracy.

B. Nonidealities that limit the accuracy

In this section, the nonidealities in the Rover, the model, and the Mocap system will be explained.

1) Rover nonidealities: The RC-input for the velocity changes over time as the battery drains or when using different ground surfaces. Therefore, small deviations between real and intended speed can be observed. Furthermore, the Rover was not designed for indoor use. The suspension, and chassis of the Rover were vibrating, and the tires were lacking grip, which negatively affected the Mocap data and accuracy. Another nonideality is the steering of the Rover. During testing, it was observed that the same steering input not always results in the same steering angle. An explanation for this behaviour is that the servo motor is too weak or that the Rover has slack, leading to an inconsistent steering angle. Finally, delays can be observed in the Rover's robotic brain when too many messages were sent. While running the Simulink model without limitations, the delay could even reach four seconds. This issue was initially solved using Simulink Real Time, but the 'Paced Run' option can also be used. This is arguably a better solution.

2) Model nonidealities: The potential field method was created for a point mass, which is free to move in all directions. However, the car was modelled as a bicycle, which restrains lateral movement. This means that a lateral force will not directly result in a lateral movement, but it will



(a) Mocap data with dt = 1/120 s. (b) Mocap data with dt = 1/10 s.

Fig. 7. Mocap data illustrating the staircase effect.

result in a steering angle. To simplify the bicycle model, it is assumed that there is no lateral slip, since only low velocities were used. However, when the car moves at higher velocities, this assumption is no longer valid. As a consequence, the bicycle model needs to be expanded to include lateral slip.

3) Mocap nonidealities: The Rover velocity is calculated by taking the time derivative of its position, obtained by the Mocap system. The x and y position (of a test circuit) of the Mocap system can be seen in Figure 7. The data is shown as a staircase. As the Mocap system is an optical system and computes the position of the markers at a pixel resolution, this could introduce the quantisation effect. This causes the derivative to give either zero or high peak values. The frame rate of the Mocap system was changed from 120 Hz to 10 Hz, solving the quantisation effect. The mean squared error will be approximately $\Delta^2/12$, where Δ the quantisation step size [18]. According to [19], the method of oversampling and low-pass filtering can be applied to reduce the signal to noise ratio by a factor described with $\text{SNR}_{A/D-gain} = 10\log_{10}(f_{s,new}/f_{s,old})$. Increasing the sampling rate by a factor of four, 6 dB of gain can be achieved.

V. Conclusion

The first research question was, whether it is possible to implement virtual potential based navigation on the available autonomous RC car platform. When looking at the results, it can be concluded that it is indeed possible to implement this to control the steering of a RC car.

The second research question was, which level of accuracy could be obtained in following a given trajectory or keeping a given formation. From the results, it can be concluded that a high level of accuracy is achieved for the scenario that one Rover is driving at 0.50 m/s. This accuracy corresponds to a MAD of $(2 \pm 1) \cdot 10$ mm. However, when driving faster, the level of accuracy is low. The MAD for keeping a given formation is $(4 \pm 2) \cdot 10^2$ mm. This corresponds to a low level of accuracy. Also the follower car obtained a MAD of $(2 \pm 2) \cdot 10^1$ mm for following a given trajectory. The reason for the high standard deviation is that no exact velocities can be sent to the Rover.

The third research question was, which nonidealities limit in practice this accuracy. The main nonideality is the fact, that the RC car is not a point mass. Other nonidealities are due to the Rover, the models used—i.e. the Simulink and bicycle model—and the Mocap system.

To improve the accuracy and the use of potential fields for navigation, there are some future recommendations. Firstly, a more advanced bicycle model can be used to better predict the vehicle behaviour. Secondly, if more computational power can be made available a smaller time step can be used. This will allow to update the steering input of the RC car at a higher frequency. Thirdly, different PID controllers can be used for different 'cases'. For example, use different controllers for different speeds and for different curvatures or straight paths. Lastly, a Kalman filter can be used to filter the Mocap data, as a raw noisy signal cannot be numerically differentiated.

References

[1] J. Ploeg, B. T. Scheepers, E. Van Nunen, N. Van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2011, pp. 260–265.

[2] T. van der Sande and H. Nijmeijer, *From Cooperative to Autonomous Vehicles*. Cham: Springer International Publishing, 2017, pp. 435–452.

[3] R. Furchgott, "Public streets are the lab for self-driving experiments," *The New York Times*, 2021. [Online]. Available: https://www.nytimes.com/2021/12/23/business/tesla-self-driving-regulations.html

[4] S. Karaman et al., "Project-based, collaborative, algorithmic robotics for high school students: Programming self-driving race cars at mit," in *2017 IEEE integrated STEM education conference (ISEC)*. IEEE, 2017, pp. 195–203.

[5] S. S. Srinivasa et al., "Mushr: A low-cost, open-source robotic racecar for education and research," *arXiv preprint arXiv:1908.08031*, 2019.

[6] A. Agnihotri, M. O'Kelly, R. Mangharam, and H. Abbas, "Teaching autonomous systems at 1/10th-scale: Design of the f1/10 racecar, simulators and curriculum," in *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, 2020, pp. 657–663.

[7] R. Bautista-Montesano, R. Galluzzi, V. Gomez-Aladro, R. Bustamante-Bello, and R. Ramirez-Mendoza, "Autonomous vehicles as a development platform: From high school to faculty," in *2021 IEEE Global Engineering Education Conference (EDUCON)*. IEEE, 2021, pp. 43–49.

[8] J. Gonzales, F. Zhang, K. Li, and F. Borrelli, "Autonomous drifting with onboard sensors," in *Advanced Vehicle Control: Proceedings of the 13th International Symposium on Advanced Vehicle Control (AVEC16)*, 2016, p. 133.

[9] M. O'Kelly et al., "F1/10: An open-source autonomous cyber-physical platform," *arXiv preprint arXiv:1901.08567*, 2019.

[10] B. Goldfain et al., "Autorally: An open platform for aggressive autonomous driving," *IEEE Control Systems Magazine*, vol. 39, no. 1, pp. 26–55, 2019.

[11] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[12] E. Rimon and D. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.

[13] J.-C. Latombe, *Robot motion planning*. Springer Science & Business Media, 2012, vol. 124.

[14] E. Snapper, "Model-based path planning and control for autonomous vehicles using artificial potential fields," Master's thesis, Delft University of Technology, 2018.

[15] J. M. Snider, "Automatic steering methods for autonomous automobile path tracking," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-09-08, February 2009.

[16] E. Semsar-Kazerooni, K. Elferink, J. Ploeg, and H. Nijmeijer, "Multi-objective platoon maneuvering using artificial potential fields," *IFAC-PapersOnLine*, 2017.

[17] "Erle-rover," accessed via https://robots.ros.org/erle-rover/ at 22-02-2022.

[18] R. Gray and D. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, 1998.

[19] R. G. Lyons, *Understanding Digital Signal Processing*, 1st ed. USA: Addison-Wesley Longman Publishing Co., Inc., 1996.