IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received June 16, 2019, accepted July 7, 2019, date of publication July 15, 2019, date of current version August 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2929022

A Simplified Optimal Path Following Controller for an Agricultural Skid-Steering Robot

BENJAMIN FERNANDEZ[1,2], (Member, IEEE), PEDRO JAVIER HERRERA[2], AND JOSE ANTONIO CERRADA[2]

[1]Research and Advanced Engineering, AGCO GmbH, 87616 Marktoberdorf, Germany
[2]Departamento de Ingeniería de Software y Sistemas Informáticos, ETSI Informática, National University of Distance Education, 28040 Madrid, Spain

Corresponding author: Benjamin Fernandez (mail@benjaminfernandez.info)

This work was supported in part by the Spanish Ministry of Economy and Competitiveness under Grant DPI2016-77677-P and Grant AGL2017-83325-C4-3-R, and in part by the Community of Madrid and Structural Funds of the EU under Grant S2018/NMT-4331.

ABSTRACT The dynamics of a skid-steering robot present intrinsic non-linearities that make the design and implementation of a controller a very complex task, time-consuming, and difficult to implement into an embedded system with limited resources. This paper presents a simplified first order digital model approximation and an optimal observer-based control approach for the tracking of the lateral position of such robots. In order to verify the validity of this proposal, 3D real-time interactive simulations and real validations with an agricultural skid-steering robot were performed with satisfactory results.

INDEX TERMS Agricultural robotics, autonomous off-road vehicles, digital control, guidance control, optimal control, path following, skid-steering robot, robotics, UGV.

I. INTRODUCTION

The accelerated population growth and the continuous shortage of labor in the area of agriculture, are two of the main motivations for the growingly interest in the area of robotics and precision farming. Here, agricultural vehicles play a very important role, and a lot of research activities related to navigation, path planning and control have been increasingly taking place in the past recent years. For instance, [1] presents a new concept with a fleet of small robots providing a solution for soil compaction in a scalable and energy-efficiently manner. In the same line of small vehicles, here we present a controller for a skid-steered robot used for corn seeding tasks.

Skid-steering robots are very popular in different off-road applications such as agriculture [2], mining [3] and foraging [4] due to its simple construction and flexibility. Its maneuver is a result of a difference between the angular velocities of the right and left wheels. This implies that the speed and the yaw rate of the robot depend on each other making the modeling and control of the robot a complex problem. Generally speaking, there are different modeling and control strategies for autonomous ground vehicles depending on the steering type of the vehicle, and its application [5]. But before diving into the modeling and control solutions, one should decide which kind of motion will be appropriate: Point-to-point, Path following or Trajectory tracking [6]. For the application presented in this work where the robot has to drive very accurately along the seeding lane, path following will be the best approach to adopt, since the robot will be getting a precise path between the navigation points. Furthermore, in contrast to the trajectory tracking, the timing between the navigation points does not have to be considered.

Regarding the modeling part, there are three types of models that can be used for the design and implementation of the control strategy: Geometric, Kinematic and Dynamic. Where among those, the geometric and kinematic models are the most simple and popular ones [5]. For the case of Skid-steering, one popular kinematic approach was first presented in [7], where the system is based on a ICR (Instantaneous Center of Rotation). A similar approach was presented in [8], where also sliding velocity was considered. In [9], the same kinematic model was used and transformed into path-following kinematics. Also based on a ICR, [10] presents an experimental method for reconstructing the kinematic model based on a laser scanner sensor. Alternatively, [11] uses the experimental data from an inertial measurement unit (IMU), to construct the kinematic equation of motion based on an Extended Kalman Filter (EKF).

The associate editor coordinating the review of this manuscript and approving it for publication was Zhen Li.

Regarding the vehicle dynamics, pioneer work can be found in [12] introducing an operative nonholonomic constraint into the equation of motion, based on the lateral speed and the angular velocity. This approach has been further used in different works such as those presented in [13]–[16]. Less popular methods for modeling skid-steering vehicles can also be found in the literature. For instance, a method for modeling the system dynamics by applying switched system optimization is presented in [17]. A basic lateral dynamics analysis can be found in [18] and to calculate performance and power consumption, [19] and [20] present simplified models based on torque and terrain analysis with validated results. Also, [21] uses a so-called Magic Formula to predict steering performance and power consumption based on the force/slip characteristics.

As aforementioned, the control strategy depends to some extent on the model to be used. The simplest methods are related to the geometry of the vehicle and some examples are Follow the Carrot [22], Pure Pursuit [23] and Stanley method, being the latest one of the most widely used and its name comes from the robot that won the 2005 DARPA Grand Challenge [24]. For kinematic and dynamic models, the complexity increases and one way to deal with this, is to separate the yaw rate control and the tracking control to deal with them separately and apply adaptive control to approximate the yaw rate dynamics as presented in [25]–[29]. Nevertheless, none of these methods have been applied to a skid-steering robot. Specifically for Skid-steering, [9] presents a controller based on Lyapunov and the vehicle kinematics from [8]. Based on the vehicle dynamics, [13] and [14] present a controller considering the problem of practical stabilization for skid-steering mobile robots also known as SSMR and a unified tracking and regulation control law. Based on a backstepping procedure that guarantees the Lyapunov stability, [15] presents also a controller design based on the skid-steering dynamics. Using nonlinear model predictive control, [16] presents a controller design by means of a Jacobian motion planning based on the Endogenous Configuration Space. Although the skid-steering methods just mentioned present good results, they are used for trajectory tracking and not for path following. Finally, in order to cover for uncertainties and changes in the terrain, robust and adaptive methods can be found in [30]–[33].

The majority of the solutions mentioned above are either used for trajectory tracking or require complex operations. This means that using embedded control units (ECU) with limited resources may not provide enough computing power. Therefore, the motivation for the solution presented here was to find a more practical approach for path following control in order to provide a way to simplify the design and implementation by using less complex modeling and operations. This comes with the cost of not considering non-linearities and uncertainties in the model which might lead to instability in terrains where the slippage is very high, or in situations where one wheel loses contact with the terrain. Nevertheless, the simplified optimal problem presented here, allows for the use of only one ECU with limited resources and might cover the big majority of terrain situations. Here, we present a linear solution with an optimal controller, applied to a first order digital model approximation based on the kinematics presented in [7], that calculates the parameters online based on the speed changes.

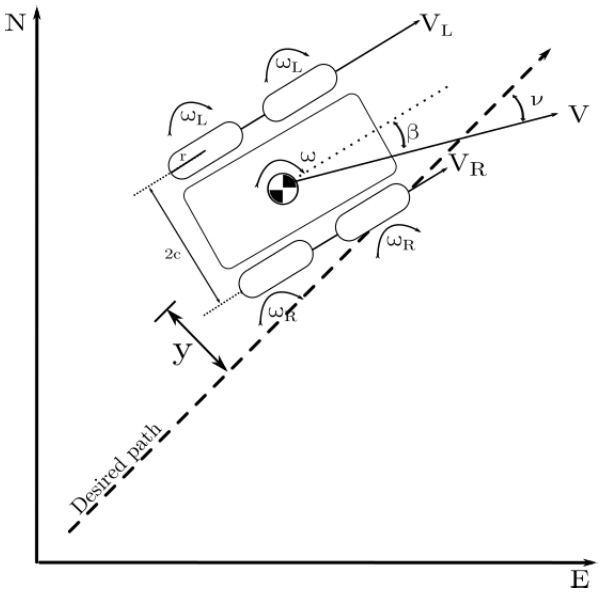

FIGURE 1. Lateral position with respect to a desired path.

This paper is divided into four sections. Section I describes the purpose of this work and summarizes the related work. Section II presents the approximated kinematics for the lateral position and its solution to the optimal control tracking problem. A 3D real-time interactive simulation and its validation with a real robot are presented in Section III. Finally, the conclusions can be found in Section IV.

II. METHODS

The design and implementation for the tracking control problem of a skid-steering robot is divided into a lateral position model approximation and an optimal, observer-based, tracking control solution.

A. MODEL APPROXIMATION

Figure 1 shows a representation of a skid-steering robot where y is the lateral position to a desired path and it is also the variable to be controlled. The yaw rate of the robot (ω) and its linear velocity (V) are produced by the difference between the velocities of the right and left axis (V_R and V_L respectively), where β represents the angle of the velocity vector to the center of gravity. The angular velocities of the right and left wheels are represented by ω_R and ω_L respectively and the distance between axes is $2c$. The angle with respect to the desired path is represented by ν and r is the wheel radius.

From the kinematic point of view, we can define the yaw rate at the center of gravity and the linear velocity by the following equations:

$$V = r * \frac{\omega_L + \omega_R}{2}, \tag{1}$$

$$\omega = r * \frac{-\omega_L + \omega_R}{2c}. \tag{2}$$

By rewriting (2) into right and left velocities we obtain (3). And by defining a time delay τ we find a first order yaw-rate transfer function presented in (4).

$$\omega = \frac{V_R - V_L}{2c} = \frac{\Delta V}{2c}. \tag{3}$$

$$G_\omega(s) = \frac{\omega(s)}{\Delta V(s)} = \frac{1/2c}{\tau \cdot s + 1}. \tag{4}$$

Finally using the z-transform with a sampling time T_s we obtain the following digital transfer function:

$$G_\omega(z^{-1}) = \frac{b_{r_1} z^{-1}}{1 + a_{r_1} z^{-1}} \tag{5}$$

where

$$\begin{aligned} b_{r_1} &= (1/2c) * (1 - e^{-T_s/\tau}) \\ a_{r_1} &= -e^{-T_s/\tau}. \end{aligned} \tag{6}$$

Generally speaking, the lateral position y with respect to a *desired path* is related to the yaw rate of the vehicle in the form of (7), where β is the side slip angle and ν the course angle with respect to the *desired path* (Fig. 1).

$$\begin{aligned} \dot{y} &= V \ sin(\nu) \\ \dot{\nu} &= \omega + \dot{\beta}. \end{aligned} \tag{7}$$

Neglecting side slip, linearizing and using the small angle approximation we find the transfer function of the lateral position with respect to the yaw rate in (8).

$$G_y(s) = \frac{y(s)}{\omega(s)} = \frac{V}{s^2}. \tag{8}$$

Again, using the z-transform we obtain the following digital transfer function of the lateral position with respect to the yaw rate:

$$G_y(z^{-1}) = \frac{b_{l_0} + b_{l_1} z^{-1}}{1 + a_{l_1} z^{-1} + a_{l_2} z^{-2}}, \tag{9}$$

where

$$\begin{aligned} b_{l_0} &= V * T_s^2/2 \\ b_{l_1} &= V * T_s^2/2 \\ a_{l_1} &= -2 \\ a_{l_2} &= 1. \end{aligned} \tag{10}$$

Multiplying (5) and (9) we obtain the following digital model of the lateral position with respect to a *desired path* to be used for the controller design:

$$G_{y/\Delta V} = \frac{b_2 z^{-2} + b_3 z^{-3}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3}}, \tag{11}$$

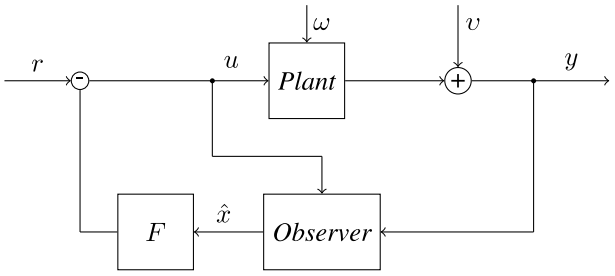

FIGURE 2. Closed-loop system with state feedback and an observer.

where

$$\begin{aligned} b_2 &= b_{r_1} * b_{l_0} \\ b_3 &= b_{r_1} * b_{l_1} \\ a_1 &= a_{l_1} + a_{r_1} \\ a_2 &= 1 + a_{l_1} * a_{r_1} \\ a_3 &= a_{r_1}. \end{aligned} \tag{12}$$

Finally, the following digital state space of third order can be used to represent our plant model:

$$\begin{aligned} x(k+1) &= \Phi \cdot x(k) + \Gamma \cdot \Delta V(k) \\ y(k) &= C \cdot x(k) \end{aligned} \tag{13}$$

where

$$\begin{aligned} \Phi &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix} \\ \Gamma &= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ C &= \begin{bmatrix} b_3 & b_2 & 0 \end{bmatrix}. \end{aligned} \tag{14}$$

B. OPTIMAL CONTROL

A general controller with observer and state feedback F can be found in Figure 2 and it is also known as the Linear Quadratic Gaussian (LQG) problem. Here ω is the process noise and υ the measurement noise.

Figure 3 depicts the detailed observer-based optimal controller of the system presented in Figure 2. As it can be seen, the observer is constructed with a copy of the plant plus the estimator gain L for correcting the error between the measured output and the output produced by the observer. As already explained, the signal to be controlled is the lateral position y, which is not stable in open loop since a step in the steering input u (which for the skid-steering robot corresponds to ΔV) will keep the vehicle turning in circles. The system is closed with an *Observer* and a state feedback F for the regulation dynamics. For the tracking dynamics, the gain K is added to be able to follow a reference lateral position r, i.e. zero when the vehicle drives along the desired path and one for driving 1 m to the left, parallel to the desired path (see Fig. 1). Negative values used for the reference have the meaning of driving on the right side of the desired path.

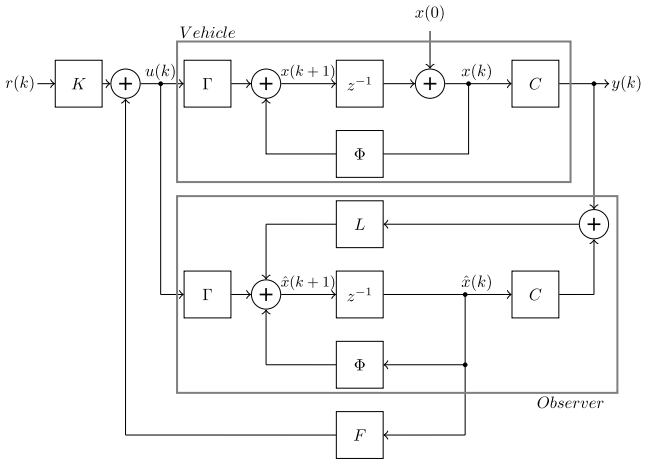

FIGURE 3. Block diagram of an observer-based state space controller.

1) REGULATION

Generally speaking, the conditions for the design of the observer-based controller are that the system (the *Vehicle*) is observable and controllable. This means that the observer should be able to estimate all the state variables of the system and that the feedback should be able to bring all the poles to a desired place. To ensure this, the conditions presented here without proof are that the observability and the controllability matrices have full rank (Eqs. (15) and (16) respectively).

$$rank\begin{pmatrix} C \\ C\Phi \\ C\Phi^2 \\ \vdots \\ C\Phi^{(n-1)} \end{pmatrix} = n. \tag{15}$$

$$rank(\Phi \quad \Phi\Gamma \quad \Phi^2\Gamma \quad \dots \quad \Phi^n\Gamma) = n. \tag{16}$$

where n is the number of state variables. Then, using the linear regulation problem we can find the solution to the system in Figure 3 by first finding the state feedback $u = F \cdot x$ that minimizes the performance index

$$J_{LQG} = \frac{1}{2}\sum_{k=0}^{\infty}[x^T(k)Qx(k) + u^T(k)Ru(k)], \tag{17}$$

where the optimal solution is

$$F = -(R + \Gamma^T \cdot P_f \cdot \Gamma)^{-1} \cdot \Gamma^T \cdot P_f \cdot \Phi \tag{18}$$

and P_f is the solution to the Riccati equation

$$P_f = Q + \Phi^T \cdot P_f \cdot \Phi - \Phi^T \cdot P_f \cdot \Gamma \cdot (R+\Gamma^T \cdot P_f \cdot \Gamma)^{-1} \cdot \Gamma^T \cdot P_f \cdot \Phi. \tag{19}$$

For the *Observer*, the state estimate L is equivalent to the state feedback F by making the following replacements:

$$Q \to Q_e; \quad R \to R_e; \quad \Phi \to \Phi^T; \quad \Gamma \to C^T; \quad F \to L; \quad P_f \to P_l. \tag{20}$$

Therefore, by duality the optimal estimator gain is

$$L = -((R_e + C \cdot P_l \cdot C^T)^{-1} \cdot C \cdot P_l \cdot \Phi^T)^T, \tag{21}$$

where P_l is the solution to the Riccati equation

$$P_l = Q_e + \Phi \cdot P_l \cdot \Phi^T - \Phi \cdot P_l \cdot C^T \cdot (R_e+C \cdot P_l \cdot C^T)^{-1} \cdot C \cdot P_l \cdot \Phi^T. \tag{22}$$

Here Q, Q_e, R and R_e can be used as tuning parameters to deal with uncertainties in the system. The details about how to find the solution to this problem are described in [34]and [35].

2) TRACKING

To achieve zero steady-state error, i.e. the system output y equals the reference input r in steady state, the static gain should be 1. This can be done by finding the static gain of the system and multiplying it by its inverse. For instance, the static gain from r to y for the system in Figure 3 is $K \cdot C(\Phi + \Gamma F - I)^{-1}\Gamma$. Therefore, we can choose

$$K = \frac{1}{C(\Phi + \Gamma F - I)^{-1}\Gamma}. \tag{23}$$

This requires calculating the inverse of the matrix $(\Phi + \Gamma F - I)$ which is assumed to be non-singular since we expect a constant static gain for a stable system. Nevertheless, calculating the inverse of a matrix could consume a lot of computing resources. An alternative to solve this problem is by assuming that in steady state, the state vector x as well as the control input u will take the constant values x_{ss} and u_{ss} respectively and that $y_{ss} = r_{ss}$. Therefore, in steady state the system will take the form

$$\begin{aligned} x_{ss} &= \Phi x_{ss} + \Gamma u_{ss} \\ r_{ss} &= C x_{ss} \end{aligned} \tag{24}$$

and for the system of Figure 3 the control law will take the form

$$u_{ss} = F x_{ss} + K r_{ss}, \tag{25}$$

since for an observable system we can assume that $\hat{x}_{ss} \approx x_{ss}$. We can relate the steady-state vector to the constant reference input as $x_{ss} = K_x r_{ss}$ and the steady-state control input that keeps x at x_{ss} as $u_{ss} = K_u r_{ss}$. Rewriting then the control law of (25) we have that $K_u r_{ss} = F K_x r_{ss} + K r_{ss}$ and solving for K we find that

$$K = F \cdot K_x + K_u. \tag{26}$$

To find K_x and K_u we use the same relation as for the control law and rewriting (24) we have that

$$\begin{aligned} K_x r_{ss} &= \Phi K_x r_{ss} + \Gamma K_u r_{ss} \\ r_{ss} &= C K_x r_{ss}. \end{aligned} \tag{27}$$

Solving for K_x and K_u we then find the solution to the following equation without using the inverse by applying the Gauss elimination or the Gauss-Jordan method:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \Phi - I & \Gamma \\ C & 0 \end{bmatrix} \begin{bmatrix} K_x \\ K_u \end{bmatrix}. \tag{28}$$

TABLE 1. Technical data of a skid-steering robot.

Robot data		
T_s	0.100	ms
τ	0.100	
$2c$	0.455	m
V_x	0.100-1.500	m/s
Q	0.100	
R	0.100	

III. RESULTS

Data from Table 1 provide sufficient information to design and implement the model and the controller introduced in the previous section. The parameters Q and R can be used as tuning parameters to cover for other factors not included such as weight and other uncertainties due to the use of a simplified first order system. The cycle time used is 100 ms and the reason for that, is that even though a much shorter cycle time is ideally desired, e.g. 20 ms, the industrial ECU used for the robot has limited resources and is also in charge of all other tasks inside the robot such as calculations, filtering, documentation, path planning, communications, etc. Therefore, the cycle time used is the biggest possible to save resources to other task with higher priority such as position filtering and correction. Here again, the simplified first order model, allowed us to perform quicker and simplified calculations.

Using $V_x = V = 0.5$ m/s and substituting the rest of the data from Table 1 into (9), (10), (11) and (12) we obtain the representation of the vehicle (Eq. (14)), expressed in (29) which is used in the observer as well (see Fig. 3). For the range of working speeds, these calculations take place in real-time in the ECU using the measured forward speed.

$$\Phi = \begin{bmatrix} 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 1.000 \\ 0.367 & -1.735 & 2.367 \end{bmatrix}$$
$$\Gamma = \begin{bmatrix} 0.000 \\ 0.000 \\ 1.000 \end{bmatrix}$$
$$C = \begin{bmatrix} 0.003 & 0.003 & 0.000 \end{bmatrix}. \quad (29)$$

The controllability and the observability matrices have full rank (Eqs. (30) and (31) respectively). Therefore, our system is controllable and observable and we can proceed to find the optimal solution for the state feedback and state estimate.

$$rank \begin{pmatrix} 0.000 & 0.000 & 1.000 \\ 0.000 & 1.000 & 2.367 \\ 1.000 & 2.367 & 3.871 \end{pmatrix} = 3, \quad (30)$$

$$rank \begin{pmatrix} 0.003 & 0.003 & 0.000 \\ 0.000 & 0.003 & 0.003 \\ 0.001 & -0.006 & 0.011 \end{pmatrix} = 3. \quad (31)$$

It was shown in the previous section that the state feedback F and the state estimate L can be found with the Riccati equations (19) and (22). Nevertheless, as it can be seen, the solution to the differential nonlinear Riccati equation depends on itself. There are different methods to solve the equation and one would be in an iterative manner. Therefore, for a digital system (19) and (22) can be rewritten as following:

$$P_f(k) = Q + \Phi' \cdot P_f(k-1) \cdot \Phi - \Phi' \cdot P_f(k-1) \cdot \Gamma \cdot (R + \Gamma' \cdot P_f(k-1) \cdot \Gamma)^{-1} \cdot \Gamma' \cdot P_f(k-1) \cdot \Phi, \quad (32)$$

$$P_l(k) = Q_e + \Phi \cdot P_l(k-1) \cdot \Phi' - \Phi \cdot P_l(k-1) \cdot C' \cdot (R_e + C \cdot P_l(k-1) \cdot C')^{-1} \cdot C \cdot P_l(k-1) \cdot \Phi', \quad (33)$$

where the solution can be found by iterating at each cycle time until P_f and P_l converge [35]. For our problem, the solution converges after around 100 cycles with the initial values of $P_f = P_l = 0$ (Eqs. (34) and (35)). By performing one iteration every cycle time of 100 ms, these parameters will be found after 10 s which might be too slow for our vehicle. For example driving at 1.5 m/s, the vehicle will not be able to follow the path at least for the first 10 m. Nevertheless, it was possible to perform up to around 50 iterations inside the cycle time of 100 ms. Another way to solve that, is to first calculate the parameter in a simulation. Then, this parameters can be used as the initial conditions inside the vehicle and since there might be discrepancies between the simulation and the real vehicle, some calculations still have to be done. Nevertheless, it will converge only after a couple of iterations. This will be the case as well for the different range of working speeds, since for each speed a different solution has to be calculated in real-time. Also in this case, the discrepancies of the solutions between different speeds are small enough to be calculated in real-time.

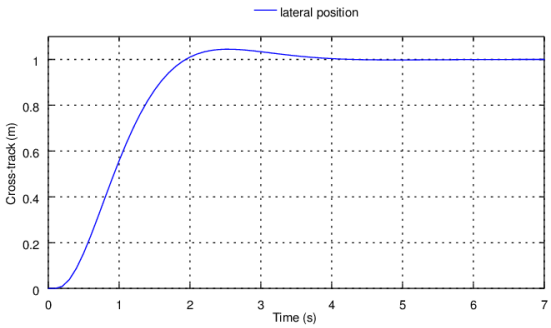

FIGURE 4. Step input of 1 m to the left of the desired path.

$$P_f = \begin{bmatrix} 0.003 & -0.011 & 0.009 \\ -0.011 & 0.046 & -0.037 \\ 0.009 & -0.037 & 0.029 \end{bmatrix}. \quad (34)$$

$$P_l = \begin{bmatrix} 544.010 & 615.560 & 687.410 \\ 615.560 & 706.210 & 797.820 \\ 687.410 & 797.820 & 911.010 \end{bmatrix}. \quad (35)$$

From the solution to the Riccati equation, the optimal state feedback and state estimate as well as the tracking gain are expressed in (36) and the controller is ready to be implemented. Figure 4 illustrates a step response of 1 m using these results (Eqs. (29) and (36)) to validate

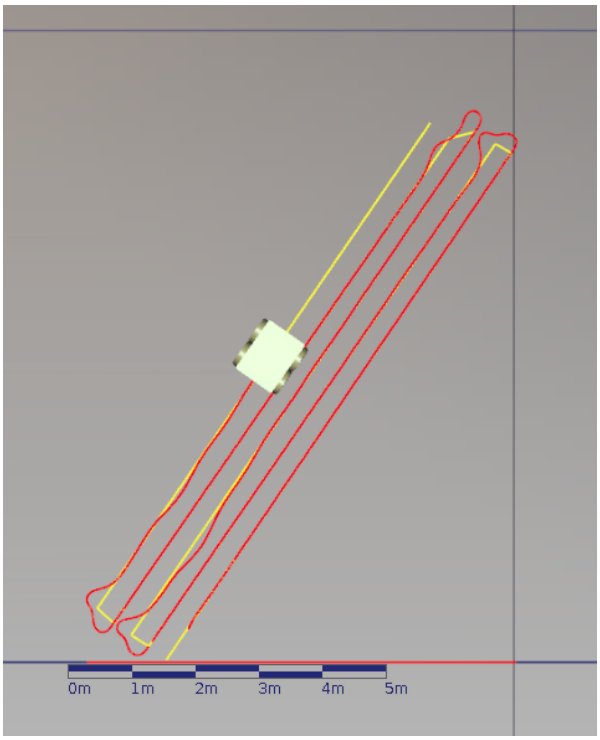

FIGURE 5. 3D real-time simulation with openframeworks and OpenGL (path following).

the controller.

$$F = \begin{bmatrix} -0.084 & 0.326 & -0.260 \end{bmatrix}$$
$$L = \begin{bmatrix} -35.332 \\ -39.701 \\ -44.083 \end{bmatrix}$$
$$K = 2.774. \tag{36}$$

A. SIMULATION

A 3D real-time simulation was programmed with C++ using Openframeworks and some representative results can be seen in Figures 6 and 6 [36]. The yellow line represents the way-path to be followed by the robot and contains real geographic coordinates. The red line is the actual driven path. It can be seen in Figure 6 that the error on the lateral position increases in the turning points since those correspond to an input step applied by changing the course angle to the desired path (ν in Figure 1).

For the simulation, the forward speed of the robot was changing from 1.5 m/s on the lane to 0.1 m/s at the headland. After each turn, we can appreciate some small oscillations before the robot is completely on the lane and this is due to two factors. The first one is because Q and R were tuned to provide the quickest response. Bigger values will avoid the oscillations, but the robot will take longer to arrive at the lane and the turning curve will be larger. The second one is because of the change in the forward speed, which requires new real-time calculations of the control parameters. When slowing down from 1.5 to 0.5 m/s before the turning points,

FIGURE 6. 3D real-time simulation with Openframeworks and OpenGL (path following).

the adaptation cannot be appreciated, which confirms that the solution converges fast enough. When speeding up again to 1.5 m/s after the headland, some small effect is appreciated since at higher speeds the vehicle needs higher response dynamics that have to be calculated, and at those speeds, if the vehicle needs 5 cycles (500 ms) to calculate the control parameters it will take around 75 cm.

From Figure 6 one could expect that the smoothness of the robot decreases when following smooth curves, since it will have a continues input of steps by updating small straight ''desired paths'' with their corresponding course angles ν. Nevertheless, Figure 6 illustrates how the robot smoothly follows a lane change of 2 m. At the bottom of the way-line, the step input is very clear with a change in the course angle of 90^o. Then at the lane change, in every cycle time the course angle is updated to the corresponding change and the steps do not affect the driving smoothness, having an error of around 7 cm at the curve. Similar errors were found in the different simulations performed.

B. VALIDATION

The simulation results were validated with a real agricultural skid-steering robot (Fig. 7) electrically powered with 4 wheels able to reach a maximal speed of 1.5 m/s. It has a weight of 40 kg and a state-of-the-art GNSS receiver with RTK-correction which provides global geographic position with a standard deviation of 2 cm. The ECU with a

FIGURE 7. Skid-steered robot for corn seeding provided by AGCO GmbH | Fendt.

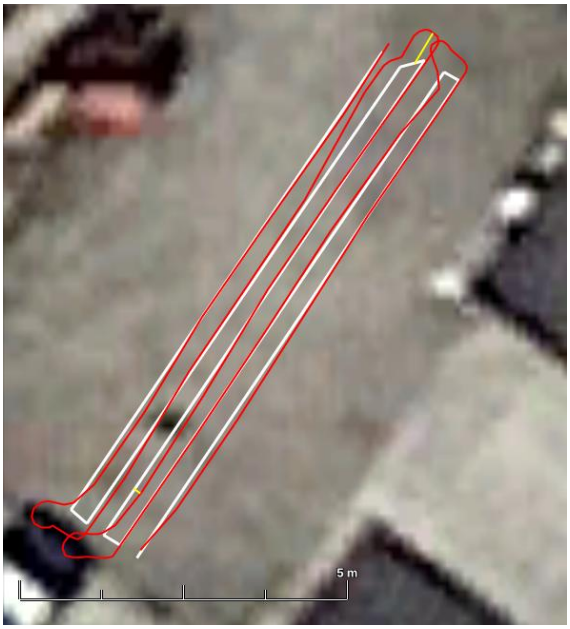

FIGURE 8. Real skid-steering robot. Lane-tracking control on the field. Visualisation with google-earth.

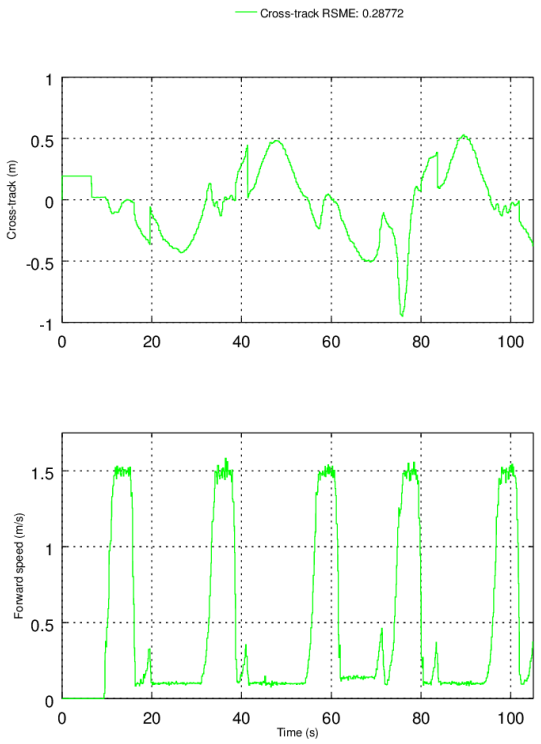

FIGURE 9. (a) Lateral position error throughout the driven path. (b) Change of forward speed throughout the driven path. Minimum $V_x = 0.2$ m/s and maximum $V_x = 1.5$ m/s.

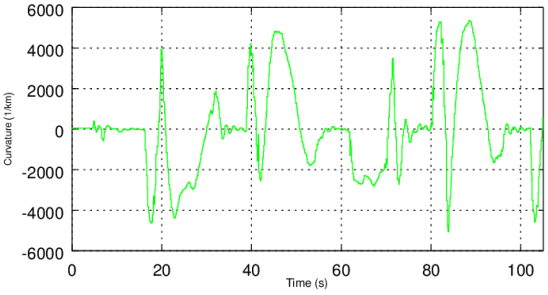

FIGURE 10. Curvature applied in km^{-1}.

400 MHz processor, contains a gyroscope, WLAN module and a CAN interfaces for the different internal communications. The angular velocity of the motors is measured with hall sensors.

Once on the field, the reduction of the cycle time from 100 ms to 50 ms was tested, and although the system was performing mainly good, we were experiencing from time to time some heartbeat lost due to the delay of some tasks. Therefore, we changed back to 100 ms cycle time since the accuracy delivered was enough for following the path.

Figure 8 shows the results using the real robot. Here the white line is the same as the yellow one used in the simulation to be able to compare directly with the simulation results. Each lane is around 10 m long and the separation between lanes is 33 cm, which corresponds to the separation used for the type of corn and the seeding task. As expected, it can be seen that the real robot presents a bigger error. The yellow line at the top presents a separation of around 90 cm at the time of changing the lane(see Fig. 9). On the other hand, the largest error when entering the lane is around 13 cm (yellow line at the third lane from right to left, almost at the bottom of Figure 8). The results are satisfactory if the cross-track at the headland is not considered very important by the user, since the error measured once on the lane is less than 5 cm and can be related to measurement noise and disturbances due to irregularities on the terrain. This error is also comparable to results obtained with more complex solutions [14], [16]. The same test was performed in different terrains and the robot produced similar results.

In Figure 10, the signal produced by the controller is presented in the form of a curvature, which is the inverse of the turning radius. It can be recognized where the turning points took place through the peaks (e.g. at 20 and 40 s). One of the main advantages of using an optimal controller can be seen here, since the signal does not contain high frequency members and the changes are very smooth without stressing

the motors. Since the observer acts as a filter, the control signal does not react to unnecessary disturbances and measurement noise without losing much responsiveness. On the other hand, it could take longer to arrive to the lane at the turning points. Nevertheless, the trade-off between actuator stress and system accuracy of an optimal controller can be appreciated here.

IV. CONCLUSIONS AND FUTURE WORK

This paper presented a first order model approximation for a skid-steering robot. The model was used to find a linear solution of an observer-based optimal control with a tracking gain. The methods presented for the model simplification and the optimal control provide a way to speed up the design and implementation by eliminating intrinsic non-linear complexity. This process also allowed an easy code implementation not only for the simulation, but also into the embedded ECU. The results were presented and the system was able to follow successfully the desired path, being the results between the simulation and the real robot very similar.

As future work, this method will take the form of a Multiple Input Multiple Output (MIMO) system to include speed and course angle control to be able to improve the error around the turning points.

ACKNOWLEDGMENT

The authors would like to thank AGCO for the provision of the robot and the technical support of this research.

REFERENCES

[1] T. Blender, T. Buchner, B. Fernandez, B. Pichlmaier, and C. Schlegel, "Managing a mobile agricultural robot swarm for a seeding task," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2016, pp. 6879–6886. [Online]. Available: http://ieeexplore.ieee.org/document/7793638/

[2] P. M. Blok, K. van Boheemen, F. K. van Evert, J. IJsselmuiden, and G.-H. Kim, "Robot navigation in orchards with localization based on particle filter and Kalman filter," *Comput. Electron. Agricult.*, vol. 157, pp. 261–269, Feb. 2019.

[3] P. Petrov, J. de Lafontaine, P. Bigras, and M. Tetreault, "Lateral control of a skid-steering mining vehicle," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, vol. 3, Oct./Nov. 2000, pp. 1804–1809.

[4] Y. Gu, J. Strader, N. Ohi, S. Harper, K. Lassak, C. Yang, L. Kogan, B. Hu, M. Gramlich, R. Kavi, and J. Gross, "Robot foraging: Autonomous sample return in a large outdoor environment," *IEEE Robot. Autom. Mag.*, vol. 25, no. 3, pp. 93–101, Sep. 2018.

[5] N. H. Amer, H. Zamzuri, K. Hudha, and Z. A. Kadir, "Modelling and control strategies in path tracking control for autonomous ground vehicles: A review of state of the art and challenges," *J. Intell. Robot. Syst.*, vol. 86, no. 2, pp. 225–254, May 2017. doi: 10.1007/s10846-016-0442-0.

[6] A. De Luca, G. Oriolo, and C. Samson, *Feedback Control of a Nonholonomic Car-Like Robot*. Berlin, Germany: Springer, 1998, pp. 171–253. doi: 10.1007/BFb0036073.

[7] J. L. Martínez, A. Mandow, J. Morales, S. Pedraza, and A. García-Cerezo, "Approximating kinematics for tracked mobile robots," *Int. J. Robot. Res.*, vol. 24, no. 10, pp. 867–878, 2005.

[8] A. Mandow, J. L. Martínez, J. Morales, J. L. Blanco, A. García-Cerezo, and J. González, "Experimental kinematics for wheeled skid-steer mobile robots," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, Oct./Nov. 2007, pp. 1222–1227.

[9] G. Huskic, S. Buck, and A. Zell, "Path following control of skid-steered wheeled mobile robots at higher speeds on different Terrain types," in *Proc. IEEE Int. Conf. Robot. Automat.*, May/Jun. 2017, pp. 3734–3739.

[10] T. Wang, Y. Wu, J. Liang, C. Han, J. Chen, and Q. Zhao, "Analysis and experimental kinematics of a skid-steering wheeled robot based on a laser scanner sensor," *Sensors*, vol. 15, no. 5, pp. 9681–9702, 2015.

[11] J. Yi, H. Wang, J. Zhang, D. Song, S. Jayasuriya, and J. Liu, "Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation," *IEEE Trans. Robot.*, vol. 25, no. 5, pp. 1087–1097, Oct. 2009.

[12] L. Caracciolo, A. de Luca, and S. Iannitti, "Trajectory tracking control of a four-wheel differentially driven mobile robot," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 4, May 1999, pp. 2632–2638.

[13] D. Pazderski, K. Kozlowski, and W. E. Dixon, "Tracking and regulation control of a skid steering vehicle," in *Proc. Amer. Nucl. Soc. 10th Int. Top. Meeting Robot. Remote Syst.*, Mar. 2004, pp. 369–376. [Online]. Available: http://www.scopus.com/inward/record.url?eid=2-s2.0-3042718125&partnerID=tZOtx3y1

[14] D. Pazderski and K. Kozłowski, "Trajectory tracking control of skid-steering robot—Experimental validation," *IFAC Proc. Volumes*, vol. 41, no. 2, pp. 5377–5382, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1474667016397993

[15] J.-Y. Jun, M.-D. Hua, and F. Benamar, "A trajectory tracking control design for a skid-steering mobile robot by adapting its desired instantaneous center of rotation," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 4554–4559. [Online]. Available: http://ieeexplore.ieee.org/document/7040100/

[16] K. Tchoń, K. Zadarnowska, Ł. Juszkiewicz, and K. Arent, "Modeling and control of a skid-steering mobile platform with coupled side wheels," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 63, no. 3, pp. 807–818, 2015.

[17] T. M. Caldwell and T. D. Murphey, "Switching mode generation and optimal estimation with application to skid-steering," *Automatica*, vol. 47, no. 1, pp. 50–64, 2011. doi: 10.1016/j.automatica.2010.10.010.

[18] L. Xueyuan, Z. Yu, H. Jibin, J. Chongbo, and G. Jing, "Lateral dynamic simulation of skid steered wheeled vehicle," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 1095–1100. [Online]. Available: http://ieeexplore.ieee.org/document/6629612/

[19] T. Guo and H. Peng, "A simplified skid-steering model for torque and power analysis of tracked small unmanned ground vehicles," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 1106–1111.

[20] S. Al-Milli, L. D. Seneviratne, and K. Althoefer, "Track–Terrain modelling and traversability prediction for tracked vehicles on soft Terrain," *J. Terramech.*, vol. 47, no. 3, pp. 151–160, Jun. 2010. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0022489810000042

[21] B. Maclaurin, "A skid steering model using the magic formula," *J. Terramech.*, vol. 48, no. 4, pp. 247–263, 2011.

[22] A. L. Rankin, C. D. Crane, III, D. G. Armstrong, A. D. Nease, and H. E. Brown, "Autonomous path-planning navigation system for site characterization," in *Proc. SPIE*, vol. 2738, May 1996, pp. 176–186.

[23] M. J. Barton, "Controller development and implementation for path planning and following in an autonomous urban vehicle," M.S. thesis, School Aerosp., Mech. Mechtron. Eng., Univ. Sydney, Sydney, NSW, Australia, Nov. 2001.

[24] S. Thrun *et al.*, "Stanley: The robot that won the DARPA grand challenge," *J. Field Robot.*, vol. 23, no. 9, pp. 661–692, 2006. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.20147

[25] J. B. Derrick and D. M. Bevly, "Adaptive control of a farm tractor with varying yaw dynamics accounting for actuator dynamics and saturations," in *Proc. IEEE Int. Conf. Control Appl. (CCA)*, Sep. 2008, pp. 547–552.

[26] J. B. Derrick and D. M. Bevly, "Adaptive steering control of a farm tractor with varying yaw rate properties," *J. Field Robot.*, vol. 26, nos. 6–7, pp. 519–536, 2009. doi: 10.1002/rob.20291.

[27] J. B. Derrick, D. M. Bevly, and A. K. Rekow, "Model-reference adaptive steering control of a farm tractor with varying hitch forces," in *Proc. Amer. Control Conf.*, Jun. 2008, pp. 3677–3682.

[28] E. Gartley and D. M. Bevly, "Online estimation of implement dynamics for adaptive steering control of farm tractors," *IEEE/ASME Trans. Mechatronics*, vol. 13, no. 4, pp. 429–440, Aug. 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4598861/

[29] B. Fernandez, P. J. Herrera, and J. A. Cerrada, "Self-tuning regulator for a tractor with varying speed and hitch forces," *Comput. Electron. Agricult.*, vol. 145, pp. 282–288, Feb. 2018.

[30] S. Arslan and H. Temeltaş, "Robust motion control of a four wheel drive skid-steered mobile robot," in *Proc. 7th Int. Conf. Elect. Electron. Eng. (ELECO)*, Dec. 2011, pp. II-415–II-419.

[31] R. S. Inoue, J. P. Cerri, M. H. Terra, and A. A. G. Siqueira, "Robust recursive control of a skid-steering mobile robot," in *Proc. 16th Int. Conf. Adv. Robot. (ICAR)*, Nov. 2013, pp. 1–6.
[32] J. Taheri-Kalani and M. J. Khosrowjerdi, "Adaptive trajectory tracking control of wheeled mobile robots with disturbance observer," *Int. J. Adapt. Control Signal Process.*, vol. 28, no. 1, pp. 14–27, 2014. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/acs.2382
[33] B. Fernandez, P. J. Herrera, and J. A. Cerrada, "Robust digital control for autonomous skid-steered agricultural robots," *Comput. Electron. Agricult.*, vol. 153, pp. 94–101, Oct. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016816991830783X
[34] K. J. Åström and B. Wittenmark, *Computer-controlled Systems: Theory and Design* (Information and System Sciences). Upper Saddle River, NJ, USA: Prentice-Hall, 1997.
[35] K. Ogata, *Discrete-time Control Systems*, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1995.
[36] B. Fernandez, J. A. Cerrada, and J. Gross, "Tractor-implement real time interactive 3D simulation based on openframeworks and OpenGL," in *Proc. Int. Conf. Mach. Control Guid.*, Oct. 2016, pp. 3677–3682. [Online]. Available: https://mcg2016.irstea.fr/wp-content/uploads/2017/05/MCG2016_paper_23.pdf

BENJAMIN FERNANDEZ received the B.Sc. degree in electromechanical engineering from the Moterrey Institute of Technology, Mexico, the M.Sc. degree in mechatronics from the Technische Universität Hamburg (TUHH), Germany, the M.B.A. degree in technology management from the Northern Institute of Technology Management, Hamburg, Germany, and the Ph.D. degree in systems and control engineering from the National University of Distance Education (UNED), Spain.

He is currently involved in the areas of agricultural robotics and automation with the Department of Research and Advanced Engineering, AGCO GmbH, Germany. His research activities include navigation, path planning and following, collision avoidance, unmanned vehicles, and swarm robotics.

PEDRO JAVIER HERRERA received the M.Sc. and Ph.D. degrees in computer science from the Complutense University of Madrid, Spain.

He is currently an Associate Professor with the Software Engineering and Computer Systems Department, National University of Distance Education (UNED), Spain. His research activities include computer vision, pattern recognition, artificial intelligence, and robotics.

Dr. Herrera is a member of the Spanish Society on Automation and Control (CEA-IFAC).

JOSE ANTONIO CERRADA received the M.Sc. and Ph.D. degrees in industrial engineering from the Polytechnic University of Madrid, Spain.

He is currently a Full Professor with the Software Engineering and Computer Systems Department, National University of Distance Education (UNED), Spain. He has authored ten books and over 100 articles in journals and conference proceedings. His research interests include robotics, RFID technologies, and software engineering.