On the Comparison of Model-Based and Model-Free Controllers in Guidance, Navigation and Control of Agricultural Vehicles

Erkan Kayacan, Erdal Kayacan, I-Ming Chen, Herman Ramon and Wouter Saeys

Abstract In a typical agricultural field operation, an agricultural vehicle must be accurately navigated to achieve an optimal result by covering with minimal overlap during tillage, fertilizing and spraying. To this end, a small scale tractor-trailer system is equipped by using off the shelf sensors and actuators to design a fully autonomous agricultural vehicle. To alleviate the task of the operator and allow him to concentrate on the quality of work performed, various systems were developed for driver assistance and semi-autonomous control. Real-time experiments show that a controller, which gives a satisfactory trajectory tracking performance for a straight line, gives a large steady-state error for a curved line trajectory. On the other hand, if the controller is aggressively tuned to decrease the tracking error for the curved lines, the controller gives oscillatory response for the straight lines. Although existing autonomous agricultural vehicles use conventional controllers, learning control algorithms are required to handle different trajectory types, environmental uncertainties, such as variable crop and soil conditions. Therefore, adaptability is a must rather than a choice in agricultural operations. In terms of complex mechatronics systems, e.g. an agricultural tractor-trailer system, the performance of model-based

E. Kayacan
Coordinated Science Lab, Distributed Autonomous Systems Lab,
University of Illinois at Urbana -Champaign, Urbana, IL 61801, USA
e-mail: erkank@illinois.edu

E. Kayacan (✉) · I.-M. Chen
School of Mechanical & Aerospace Engineering, Nanyang Technological University,
Singapore 639798, Singapore
e-mail: erdal@ntu.edu.sg

I.-M. Chen
e-mail: michen@ntu.edu.sg

H. Ramon · W. Saeys
Department of Biosystems, Division of Mechatronics, Biostatistics and Sensors,
KU Leuven, 3001 Leuven, Belgium
e-mail: herman.ramon@kuleuven.be

W. Saeys
e-mail: wouter.saeys@kuleuven.be

R. John et al. (eds.), *Type-2 Fuzzy Logic and Systems*,
Studies in Fuzziness and Soft Computing 362,
https://doi.org/10.1007/978-3-319-72892-6_3

and model-free control, i.e. nonlinear model predictive control and type-2 neuro-fuzzy control, is compared and contrasted, and eventually some design guidelines are also suggested.

1 Introduction

Agriculture is the oldest, and also still the most important, economic activity of the modern humankind society. Archaeological excavations show that we, as humankind, started this thrilling adventure approximately 11,500 years ago. This decision was not only to stop being hunters and gatherers but also to start adapting the nature to our needs instead of only adapting ourselves to the facts of the wild life. This adventure started with wild barley, wheat and lentils in the South Asia (Fertile crescent and Chogha Golan) [1–3]. Our new skill, the skill of dealing with the soil and growing domestic plants instead of eating only wild ones, was the first step of our civilization which caused a domino effect such as paving the way for living as clans in villages and even the rise of complex religions.

Whereas average life expectancy was around 25 years in the Paleolithic and Neolithic eras, thanks to modern medicine, in particular Alexander Fleming who discovered penicillin, it has reached to 80 years in the last century [4]. In other words, our world is constantly being overcrowded. According to the United Nations Food and Agriculture Organization (FAO), our world has to double food production by 2050 to meet rising demand. Since it is an obvious fact that we can no longer clear more forest, one of the possible solutions is to increase the overall agricultural production efficiency among which the application of intelligent agricultural vehicles.

Considering the high demand for increased efficiency, productivity and safety in farming operations, a precise trajectory tracking is needed for agricultural vehicles to improve quality meanwhile reducing cost. When the motivations are carefully examined, the following requirements can be identified for an autonomous production machine, such as a tractor-trailer system: smart (intelligent) and productive (automation). In light of these aforementioned conditions, theoretical and practical control and design methods, i.e. model-free and model-based methods, are proposed throughout this article.

1.1 Role of Robots in Agriculture

Agriculture is not only a vital economic activity of a civilized society but also a necessity for our survival. Therefore, technological developments have always been playing an important role to make the most of our land even in challenging geographical locations. Our aim has always been to use our land in a more efficient way under significant climate and pre-assumed meteorological conditions.

The use of production machines and intelligent vehicles in agriculture is always promising as it allows us to make simultaneous operations that cannot be performed by a human operator. For instance, when working with an agricultural machine (*e.g.* combine harvester), apart from navigating the machine, the operator must also supervise the work performed by the machine. To be a skilled operator, even for a particular agricultural production machine, is not sufficient since the operator must always adapt the machine settings due to time-varying crop and soil characteristics as well as environmental conditions. Switching paying attention to between the steering and the machine control results in an increase in the deviation from the optimal path in practice. To alleviate the task of the operator and allow him to concentrate on the quality of work performed, provision of some autonomous functions to an agricultural vehicle is the main task of the robotic system. In this respect, a driver assistance and semi-autonomous control system for an agricultural robot will be developed in this article. To dispose a fully autonomous system, a tractor is equipped with off the shelf actuators and sensors to achieve the aforementioned goals. On behalf of an operator, the developed advanced learning control algorithms are implemented in real-time to deal with changing soil conditions as well as longitudinal speed. All the aforementioned challenges tell us the same thing: *adaptability is a must rather than a choice*.

1.2 Why Do We Need Agricultural Robots?

There are at least four reasons that ensure the necessity of using autonomous agricultural vehicles in the future:

1. Constantly rising energy and labor costs (need for efficient machines)
2. Continuously adapt the machine settings (multitasking)
3. Maintain the fixed performance and accuracy (a human operator may get bored or tired after some time especially under challenging working conditions, *e.g.* under hot and sunny conditions)
4. Not possible to increase the size of the machines (limited road capacity)

1.3 What Are the Requirements of the Agricultural Vehicles?

Without exception, all agricultural operations have a strict requirement: accurate navigation. For instance, throughout tillage, fertilizing and spraying, the production machine must be operated with a high accuracy to avoid overlapping field operations. The field rows must be nicely parallel and evenly distributed so that for example the weed rows can be easily driven between them. In fact, this requirement is challenging as it can be observed that there is always considerable overlap and variation in plant distances in the field even in manual operation. The reason is that these vehicles have to operate in hilly, bumpy and sometimes muddy off-road conditions as well as they generally have to deal with thew dynamics of a trailer.

1.4 Literature Review

The first harvesting robot was introduced in The United States of America to harvest citrus [5]. After this successful implementation, it was also used to harvest apples in France in 1985 [6]. 2009, a robotic arm, which is capable of harvesting asparagus, was developed by the Industrial Technology Center of Nagasaki in Japan [7]. Afterwards, in 2011, a prototype robotic platform, which has the ability to detect spherical fruits by benefiting from image processing, was developed in [8]. It is concluded that the proposed platform can increase the overall efficiency by reducing the spent time for harvesting. As a vision-based method, in another study, detection of red and bicoloured apples on tree with an RGB-D camera has been reported [9]. Furthermore, an agribot has been developed by Birla Institute of Technology and Science to minimize the labor of farmers and increase the accuracy of the work [10]. As can be seen from the previous implementations, there have existed significant research and development in agricultural robotics. One of the most challenging tasks is to guide the mobile robotic platforms accurately on different soil conditions.

The main goal of guidance of agricultural vehicles is to drive the vehicle on an agricultural field for specific purposes by keeping it as close as to the target trajectory. There are numerous implementations of multitasking path planning for multi-vehicle cases [11, 12]. In one of them, the path planning is carried out just for one vehicle, leading vehicle, the rest of the vehicles follow it by ensuring the desired relative distances. A master-slave navigation system has been proposed in [13] where the automated slave vehicle always follows the master vehicle whether the master vehicle is autonomous or not. Another stable controller for a four-wheel mobile robot to track between rows on a field has been designed in [14] while a nonlinear model predictive controller has been proposed for a tractor-trailer system in [15]. Moreover, online learning algorithms have been integrated into control algorithms. A fuzzy controller has been designed where its membership functions (MFs) have been adapted to changing working conditions [16]. However, this paper was lack of analyzing the robustness of the proposed learning algorithm considering different environment conditions. A guidance method based on a grid map of the agricultural field has been proposed in [12] in which the grid information is used to make a feasible path from the starting point of the vehicle to the desired destination in the field. Moreover, an autonomous orchard vehicle has been developed to help fruit production in which the perception system is based on global positioning system and a two-dimensional laser scanner [17]. It can be concluded from all previous studies that learning algorithms must be used to design a controller for the purpose of guidance regarding different working conditions to obtain accurate trajectory tracking performance. After giving the design details of the autonomous agricultural vehicle, whereas we will elaborate different control algorithms to accurately navigate the agricultural vehicle under certain uncertainties in the working environment, interested readers may refer to [18] for a detailed analysis about the role of global navigation satellite systems (GNSSs) in the navigation strategies of agricultural robots.

Amongst the two well-known inference methods, as a learning model-free controller, Takagi-Sugeno-Kang (TSK) fuzzy structure has significant advantages over its Mamdani counterpart as it has tunable weights on the consequent part of the rules which allows us to update them using appropriate optimization algorithms [19]. Consequently, they are preferred in real-time application where the working conditions vary over the operation. What is more, TSK models are computationally more efficient. Considering the recent advances and proved capabilities of type-2 fuzzy logic controllers (T2FLCs) over their type-1 counterparts [20–26], we prefer to use a TSK T2FLC to handle uncertainties in the autonomous tractor-trailer system in this paper. On the other hand, as a model-based approach, a nonlinear model predictive controller (NMPC) is preferred as an advanced control algorithm. Some parameters are estimated using a nonlinear moving horizon estimator (NMHE), and fed to the model which is being used by the NMPC. The overall scheme is a learning model-based controller.

Model-based and model-free control approaches are compared and contrasted for wet clutch control problem [27]. However, the parameter update strategy in the model-free approaches considered in [27], genetic-based machine learning and reinforcement learning, are different than the method used in this paper. For instance, whereas agents take actions in an environment to maximize a cumulative reward in reinforcement learning, Lyapunov stability-based learning rules are used in the type-2 fuzzy structure in this paper which are shown to be stable using a candidate Lyapunov function.

1.5 Motivation

In terms of complex mechatronics systems, the performance of model-based and model-free control, i.e. nonlinear model predictive control and type-2 neuro-fuzzy control, is compared and contrasted by means of their design and implementation simplicity and efficiency. Moreover, some design guidelines are also suggested for the control complex mechatronic systems where there exist more than one subsystem.

1.6 Organization of the Paper

Section 2 gives the system description of the tractor-trailer system. Section 3 explains the self learning model-free and model-based algorithms; some guidelines for a controller design selection are also suggested. Finally, some conclusions are drawn from this study in Sect. 4.

2 Prototyped Autonomous Agricultural Vehicle

In order to be used during tillage, fertilizing or spraying, a small scale tractor with a trailer shown in Fig. 1 is equipped with relatively cheap sensors resulting in a fully autonomous agricultural ground robotic system. The main expectation from the designed vehicle is to follow a predetermined trajectory in outdoor environment with a high accuracy to decrease overlap during agricultural operations.

2.1 Localization

Localization and positioning systems are broadly categorized into two groups: local and global. Whereas image processing, lazer, etc. belong to local positioning systems, global positioning systems make use of satellite systems. Thanks to the recent developments in the field of GNSSs we have up to cm accuracy in real-time kinematic (RTK) GNSSs to navigate our tractor-trailer system precisely.

The requirements are to model the system, identify its parameters and design learning controllers for the system shown in Fig. 1. As this tractor has hydraulic wheel and steering, and is four-wheel-drive, it is representative for many modern agricultural vehicles. The most suitable places for mounting GNSS antennas for the tractor and trailer are the tractor rear axle center and the trailer rear axle center, respectively. Since the horizontal accuracy for civilian GPS is still around 4 meter, we have decided to use RTK differential GPS (DGPS) in our system. The resulting accuracy is 0.03 m according to the specifications of the manufacturer. In order to receive the correction signals via internet, we have preferred Flepos network by using a *Digi Connect WAN 3G* modem.

As the real-time controller, PXI platform (National Instruments Corporation, Austin, TX, USA) is selected. The GNSS and the modem are connected to the real-time controller via serial connection. The main responsibility of the real-time controller is to receive and process all the necessary sensory data, such as steering angles, GNSS measurements, etc., and to generate the control signals for the tractor and

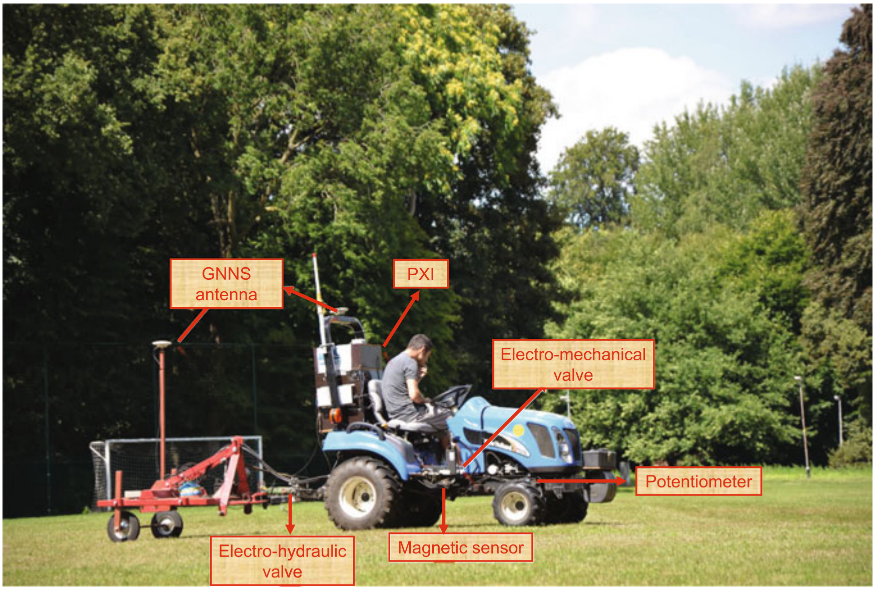

Fig. 1 The tractor-trailer system

trailer actuators separately. The control algorithms are implemented in *LabVIEW*^{TM} version 2011 (National Instruments, Austin, TX, USA). The working frequency of the overall control system is chosen as 5-Hz.

2.2 Steering Mechanisms

We have preferred to use a potentiometer, which is mounted on the front axle, to measure the tractor front wheel angle. An inductive sensor is used to measure the angle between the trailer and its drawbar. Both sensors have 1° precision. The rpm of the diesel engine has been measured by using a hall effect sensor (Hamlin, USA) which is connected to the shaft between the diesel engine and oil pump. Figure 2 shows the potentiometers and the hydrostat spindle actuator.

Low level controllers, proportional-integral (PI) controllers, generate the voltage for the electro-hydraulic valves based on the difference between the reference and measured steering angles. The longitudinal velocity of the tractor is measured by encoders mounted on the rear wheels of the tractor. A low level controller (PID) generates the voltage for the spindle actuator (LINAK A/S, Silkeborg, Denmark) taking into account the difference between the reference and measured pedal positions. The pedal position is measured by a magnetic sensor.

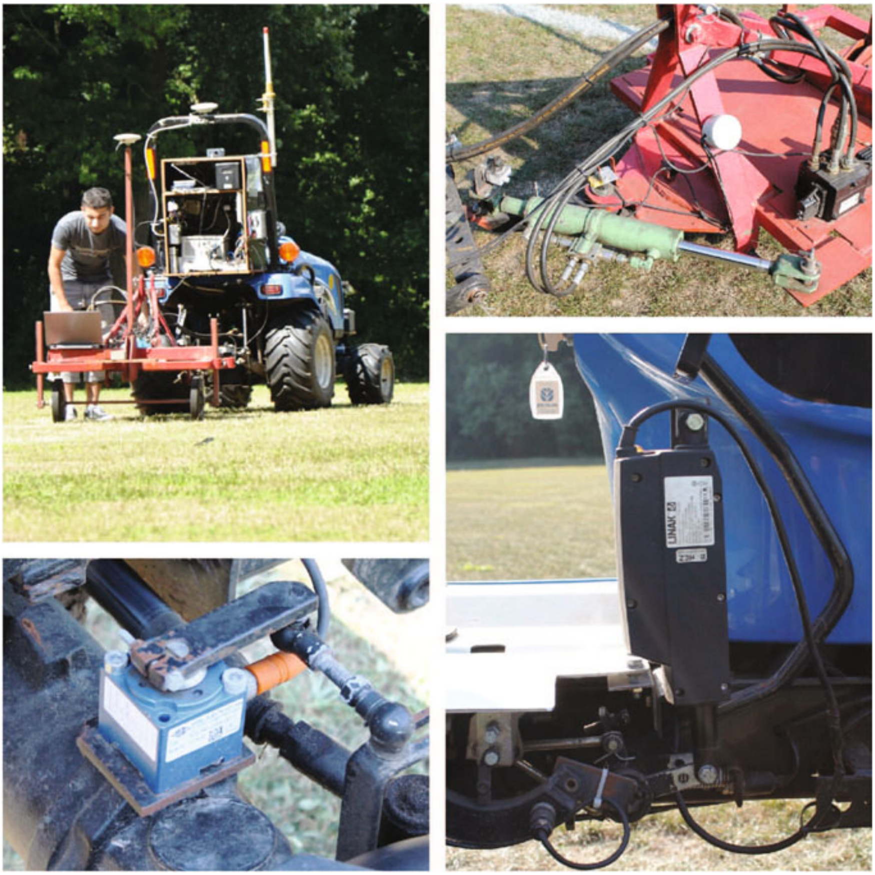

Fig. 2 Trailer actuator (top right), potentiometer (bottom left) and hydrostat spindle actuator (bottom right)

3 Self-learning Control Algorithms

The current commercial systems use simple controllers to minimize the deviation from the target path by adjusting the steering angle. These systems work well for the following straight lines under uniform soil conditions with a constant speed. However, when the soil conditions or speed change, the controllers must be tuned again. Furthermore, they use independent controllers for the absolute steering of the tractor and the relative steering of the trailer. Since both controllers will exhibit selfish behavior, this often leads to a sub-optimal result, especially for curved target paths in which the steering action of the tractor works against that of the trailer.

As a solution to the selfish behavior of the decentralized and static control algorithms, self-learning controllers have been designed in this investigation. A learning

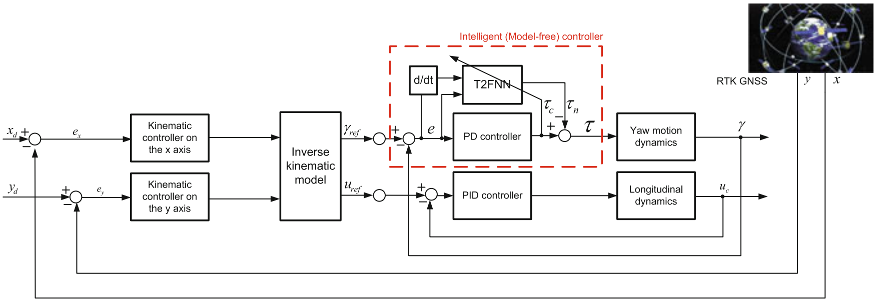

Fig. 3 Block diagram of the model-free controller

control algorithm, no matter it is model-based or model-free, is more than welcome as it will adapt itself against the parameter, crop and soil condition variations.

3.1 Model-Free Learning: Type-2 Fuzzy Neural Network Control

The proposed control scheme used in this part of the study is illustrated in Fig. 3. Since we have realized that the performance progress comes more from the yaw dynamics control accuracy of the overall control system, we have preferred to use only a conventional proportional-integral-derivative (PID) controller for the longitudinal dynamics, and design the intelligent model-free controller for the yaw dynamics. In the yaw dynamics control, a PD controller is used to guarantee the stability of the system during the initial learning. After a finite time, a type-2 fuzzy neural network (T2FNN) takes the control responsibility of the system, and the output of the PD controller goes to zero. Such a control scheme is called feedback error learning [28]. Thanks to the model-free structure of the controller, the dynamics and interactions between the subsystems are learnt online, and the optimal control signal is applied to the system. An outer loop for both the x and the y axes is also designed to correct the trajectory following errors on the relevant axes.

In the designed T2FLC, a triangular MF is preferred. There are two different approaches to construct type-2 triangular MFs. One is to blur the width of the MF Fig. 4a while the other is to blur the center of the MF Fig. 4a. In Fig. 4, the red line represents the upper MF, and the blue line shows the lower MF. Their corresponding membership values are $\overline{\mu}(x)$ and $\underline{\mu}(x)$, respectively.

The strength of the rule R_{ij} is calculated as a T-norm of the MFs in the premise part by using a multiplication operator:

$$\underline{W_{ij}} = \underline{\mu_{1i}(x_1)}\ \underline{\mu_{2j}(x_2)} \text{ and } \overline{W_{ij}} = \overline{\mu_{1i}(x_1)}\ \overline{\mu_{2j}(x_2)} \tag{1}$$

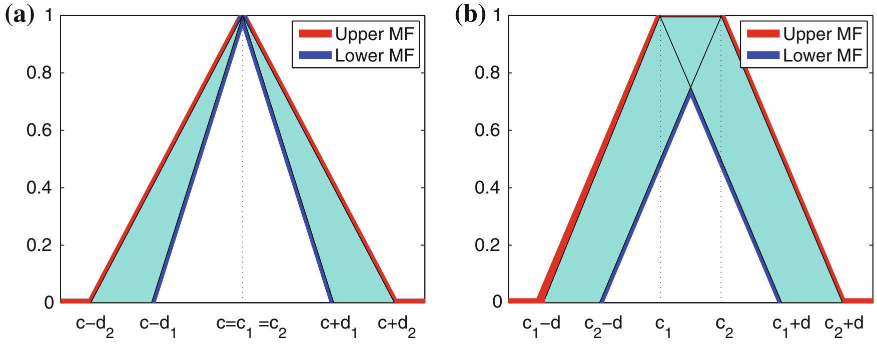

Fig. 4 A type-2 fuzzy triangular MF with uncertain width (**a**) and uncertain center (**b**)

The type-2 fuzzy triangular membership values $\underline{\mu_{1i}(x_1)}$, $\overline{\mu_{1i}(x_1)}$, $\underline{\mu_{2j}(x_2)}$, and $\overline{\mu_{2j}(x_2)}$ of the inputs x_1 and x_2 in the above expression have the following appearance:

$$\underline{\mu_{1i}(x_1)} = \begin{cases} 1 - \left|\frac{x_1 - c_{1i}}{\underline{d_{1i}}}\right| & |x_1 - c_{1i}| < \underline{d_{1i}} \\ 0 & otherwise \end{cases} \tag{2}$$

$$\overline{\mu_{1i}(x_1)} = \begin{cases} 1 - \left|\frac{x_1 - c_{1i}}{\overline{d_{1i}}}\right| & |x_1 - c_{1i}| < \overline{d_{1i}} \\ 0 & otherwise \end{cases}$$

$$\underline{\mu_{2j}(x_2)} = \begin{cases} 1 - \left|\frac{x_2 - c_{2j}}{\underline{d_{2j}}}\right| & |x_2 - c_{2j}| < \underline{d_{2j}} \\ 0 & otherwise \end{cases}$$

$$\overline{\mu_{2j}(x_2)} = \begin{cases} 1 - \left|\frac{x_2 - c_{2j}}{\overline{d_{2j}}}\right| & |x_2 - c_{2j}| < \overline{d_{2j}} \\ 0 & otherwise \end{cases}$$

Since we do not prefer to use an iterative type-reduction method in this paper, we prefer to use an approximated model of a type-2 fuzzy logic system which is denoted as A2-C0 fuzzy system. The rationale is to be able to use an optimization algorithm, which is a sliding mode control theory-based one in this paper, to tune the antecedent and consequent parameters. The fuzzy *If-Then* rule is defined as follows:

$$R_{ij}: \quad \text{If } x_1 \text{ is } \tilde{A}_{1i} \text{ and } x_2 \text{ is } \tilde{A}_{2j}, \text{ then } f_{ij} = d_{ij} \tag{3}$$

The output of the network is calculated as follows:

$$\tau_n = \int_{W_{11}\epsilon[\underline{W}_{11},\overline{W}_{11}]} \cdots \int_{W_{IJ}\epsilon[\underline{W}_{IJ},\overline{W}_{IJ}]} 1 \bigg/ \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij}(x) f_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} W_{ij}(x)} \tag{4}$$

wheref_{ij} is given by the *If-Then* rule. The inference engine used in this paper replaces the type-reduction which is given as:

$$\tau_n = \frac{q(t)\sum_{i=1}^{I}\sum_{j=1}^{J} \underline{W}_{ij} f_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} \underline{W}_{ij}} + \frac{\left(1-q(t)\right)\sum_{i=1}^{I}\sum_{j=1}^{J} \overline{W}_{ij} f_{ij}}{\sum_{i=1}^{I}\sum_{j=1}^{J} \overline{W}_{ij}} \tag{5}$$

The design parameter q, weights the sharing of the lower and the upper firing levels of each fired rule. After the normalization of (5), the output signal of the T2FNN will obtain the following form:

$$\tau_n = q(t)\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\underline{\widetilde{W_{ij}}} + \left(1-q(t)\right)\sum_{i=1}^{I}\sum_{j=1}^{J} f_{ij}\widetilde{\overline{W_{ij}}} \tag{6}$$

where $\underline{\widetilde{W_{ij}}}$ and $\widetilde{\overline{W_{ij}}}$ are the normalized values of the lower and the upper output signals of the neuron ij::

$$\underline{\widetilde{W_{ij}}} = \frac{\underline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J} \underline{W_{ij}}} \text{ and } \widetilde{\overline{W_{ij}}} = \frac{\overline{W_{ij}}}{\sum_{i=1}^{I}\sum_{j=1}^{J} \overline{W_{ij}}}$$

The following vectors can be specified:

$$\begin{aligned}
\underline{\widetilde{W}}(t) &= \left[\underline{\widetilde{W_{11}}}(t)\ \underline{\widetilde{W_{12}}}(t)\ldots\ \underline{\widetilde{W_{21}}}(t)\ \ldots\ \underline{\widetilde{W_{ij}}}(t)\ \ldots\ \underline{\widetilde{W_{IJ}}}(t)\right]^T \\
\widetilde{\overline{W}}(t) &= \left[\widetilde{\overline{W_{11}}}(t)\ \widetilde{\overline{W_{12}}}(t)\ldots\ \widetilde{\overline{W_{21}}}(t)\ \ldots\ \widetilde{\overline{W_{ij}}}(t)\ \ldots\ \widetilde{\overline{W_{IJ}}}(t)\right]^T \\
F &= [f_{11}\, f_{12}\ \ldots\ f_{21}\ \ldots\ f_{ij}\ \ldots\ f_{IJ}]
\end{aligned}$$

The following assumptions have been used in this investigation: Both the input signals $x_1(t)$ and $x_2(t)$, and their time derivatives can be considered bounded:

$$|x_1(t)| \le \widetilde{B}_x,\quad |x_2(t)| \le \widetilde{B}_x \qquad \forall t \tag{7}$$

$$|\dot{x}_1(t)| \le \widetilde{B}_{\dot{x}},\quad |\dot{x}_2(t)| \le \widetilde{B}_{\dot{x}} \qquad \forall t \tag{8}$$

where $\widetilde{B}_x$ and $\widetilde{B}_{\dot{x}}$ are assumed to be some known positive constants. It is obvious that $0 < \underline{\widetilde{W}_{ij}} \leq 1$ and $0 < \overline{\widetilde{W}_{ij}} \leq 1$. In addition, it can be easily seen that $\sum_{i=1}^{I} \sum_{j=1}^{J} \underline{\widetilde{W}_{ij}} = 1$ and $\sum_{i=1}^{I} \sum_{j=1}^{J} \overline{\widetilde{W}_{ij}} = 1$. It is also considered that, τ and $\dot{\tau}$ will be bounded signals too, *i.e.*

$$|\tau(t)| < B_{\tau}, \quad |\dot{\tau}(t)| < B_{\dot{\tau}} \quad \forall t \tag{9}$$

where B_{τ} and $B_{\dot{\tau}}$ are some known positive constants.

3.1.1 Proposed Sliding Mode Control (SMC) Theory-Based Learning Algorithm

The zero value of the learning error coordinate $\tau_c(t)$ can be defined as a time-varying sliding surface, *i.e.*,

$$S_c\left(\tau_n, \tau\right) = \tau_c(t) = \tau_n(t) + \tau(t) = 0 \tag{10}$$

The sliding surface is defined as follows:

$$S_p(e, \dot{e}) = \dot{e} + \chi e \tag{11}$$

where χ is a positive constant which defines the slope of the sliding surface.

Definition A sliding motion will appear on the sliding manifold $S_c\left(\tau_n, \tau\right) = \tau_c(t) = 0$ after a time t_h, if the condition $S_c(t)\dot{S}_c(t) = \tau_c(t)\,\dot{\tau}_c(t) < 0$ is satisfied for all t in some nontrivial semi-open subinterval of time of the form $\left[t, t_h\right) \subset \left(0, t_h\right)$.

3.1.2 Proposed Parameter Update Rules for the T2FNN

Theorem 1 *If the adaptation laws for the parameters of the considered T2FNN are chosen as [28]:*

$$\dot{\underline{c}}_{1i} = \dot{\overline{c}}_{1i} = \dot{c}_{1i} = \dot{x}_1 \tag{12}$$

$$\dot{\underline{c}}_{2j} = \dot{\overline{c}}_{2j} = \dot{c}_{2j} = \dot{x}_2 \tag{13}$$

$$\dot{\underline{d}}_{1i} = \underline{\mu}_{1i} \frac{-\alpha \underline{d}_{1i}^{2}}{x_1 - c_{1i}} sgn(\tau_c) sgn\left(\frac{x_1 - c_{1i}}{\underline{d}_{1i}}\right) \tag{14}$$

$$\dot{\overline{d}}_{1i} = \overline{\mu}_{1i} \frac{-\alpha \overline{d}_{1i}^{2}}{x_1 - c_{1i}} sgn(\tau_c) sgn\left(\frac{x_1 - c_{1i}}{\overline{d}_{1i}}\right) \tag{15}$$

$$\dot{\underline{d}}_{2j} = \underline{\mu}_{2j} \frac{-\alpha \underline{d}_{2j}^{\,2}}{x_2 - c_{2j}} sgn(\tau_c) sgn\left(\frac{x_2 - c_{2j}}{\underline{d}_{2j}}\right) \tag{16}$$

$$\dot{\overline{d}}_{2j} = \overline{\mu}_{2j} \frac{-\alpha \overline{d}_{2j}^{\,2}}{x_2 - c_{2j}} sgn(\tau_c) sgn\left(\frac{x_2 - c_{2j}}{\overline{d}_{2j}}\right) \tag{17}$$

$$\dot{f}_{ij} = -\frac{\left(q(t)\underline{\widetilde{W}}_{ij} + \left(1 - q(t)\right)\overline{\widetilde{W}}_{ij}\right)\alpha sgn(\tau_c)}{\left(q(t)\underline{\widetilde{W}} + \left(1 - q(t)\right)\overline{\widetilde{W}}\right)^T \left(q(t)\underline{\widetilde{W}} + \left(1 - q(t)\right)\overline{\widetilde{W}}\right)} \tag{18}$$

$$\dot{q}(t) = -\frac{\alpha sgn(\tau_c)}{F(\underline{\widetilde{W}} - \overline{\widetilde{W}})^T} \tag{19}$$

where α is a sufficiently large positive design constant satisfying the inequality below:

$$\alpha > B_{\dot{\tau}} \tag{20}$$

Then, given an arbitrary initial condition $\tau_c(0)$, the learning error $\tau_c(t)$ will converge to zero within a finite time t_h.

Proof The reader is referred to [28].

The relation between the sliding line S_p and the zero adaptive learning error level S_c is determined by the following equation:

$$S_c = \tau_c = k_D \dot{e} + k_P e = k_D \left(\dot{e} + \frac{k_p}{k_D} e\right) = k_D S_p \tag{21}$$

The tracking performance of the feedback control system can be analyzed by introducing the following Lyapunov function candidate:

$$V_p = \frac{1}{2} S_p^2 \tag{22}$$

Theorem 2 *If the adaptation strategy for the adjustable parameters of the T2FNN is chosen as in (12)–(19), then the negative definiteness of the time derivative of the Lyapunov function in (22) is ensured.*

Proof The reader is referred to [28].

Remark The obtained result means that, assuming that the SMC task is achievable, using τ_c as a learning error for the T2FNN together with the adaptation laws (12)–(19) enforces the desired reaching mode followed by a sliding regime for the system under control.

3.1.3 Experimental Results for the Model-Free Controller

Even if such a trajectory is not common in a typical agricultural operation, an 8-shaped time-based trajectory is preferred in this investigation. The simple reason behind such a trajectory is to be able to elaborate the performance of the intelligent model-free controller both for straight and curved lines.

As can be seen from Figs. 5a–c, the proposed model-free control scheme consisting of a T2FNN working in parallel with a conventional controller gives a better trajectory following accuracy than the one where only a PD controller acts alone. One can claim that the same performance can be obtained by further tuning the conventional controller. However, when there exists more than one subsystem as well as the changing parameters of the system model and the variations in working conditions (soil and crop variability), this task is not straightforward.

In order to show the adaptability capability of the proposed scheme, we show the difference between the first, second and thirds turns for different controllers in Fig. 5a. As it is expected, when the PD controller acts alone, its performance does not change from the first turn to the consequent turns. Thanks to the learning capability, the T2FNN working in parallel with a PD controller gives a better performance in its second and third turns. The results in Fig. 5b show performance improvement of approximately 30% in the case of having a PD controller working in parallel with the T2FNN.

The controller signals coming from the PD controller and the T2FNN can be seen in Fig. 5c. In its first turn, the dominating control signal is coming from the PD controller. In its second turn (starting from 120th s), the T2FNN is able to take over the control, thus becoming the leading controller. Every time there is a change in the reference signal, after a finite time, the output of the PD controller again tends to go to zero. As can be seen from Fig. 5d, the T2FNN significantly increases the control accuracy of the yaw dynamics of the system.

3.1.4 Discussions for Model-Free Control

The real-time test results are promising in a way that when the system is controlled by using a conventional controller in parallel with a T2FNN, the accuracy of the overall controller increases. In this method, the conventional controller is responsible for the stability of the system in the beginning of the learning process. After the learning process starts, the T2FNN controller learns the system dynamics and takes the responsibility of controlling the system gradually. In other words, there is no need to well-tune the conventional controller. It is to be noted that in complex mechatronic systems where there exist more than one subsystem, well-tuning of different controllers on different subsystems is a tedious work.

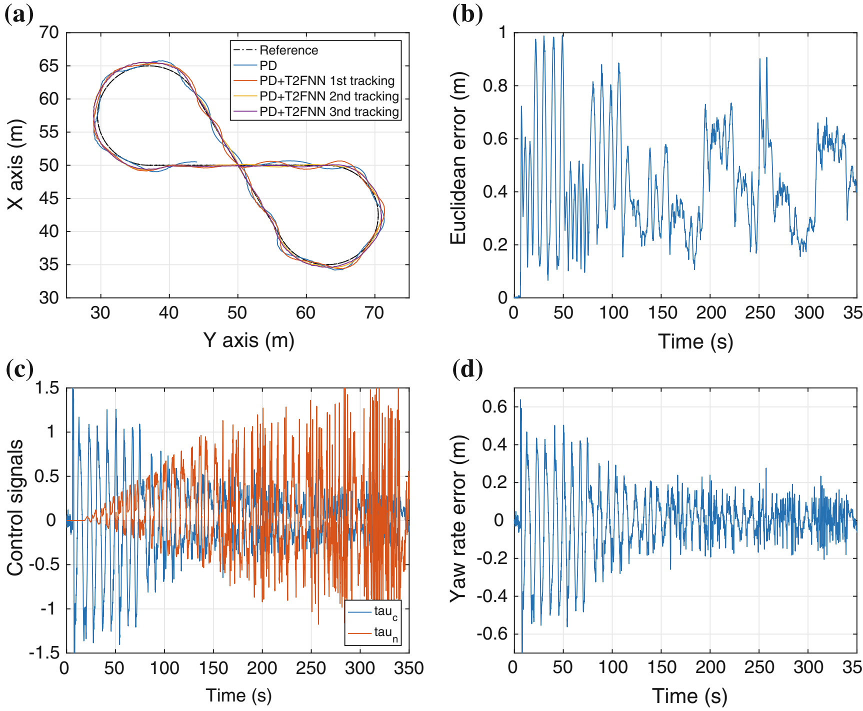

Fig. 5 **a** Reference and actual trajectories **b** Euclidean error to the reference trajectory **c** Control inputs **d** Yaw rate error

3.2 Model-Based Learning: Model Predictive Control-Moving Horizon Estimation Framework

Nonlinear model predictive control and nonlinear moving horizon estimation framework is illustrated in Fig. 6 and a system model is required to design this framework. The equations for the tractor are written as follows:

$$\begin{aligned}\dot{x} &= \mu v \cos(\psi)\\ \dot{y} &= \mu v \sin(\psi)\\ \dot{\psi} &= \frac{\mu v \tan(\kappa\delta)}{L}\\ \dot{v} &= -\frac{v}{\tau} + \frac{K}{\tau}HP\end{aligned} \tag{23}$$

where x, y and ψ denote respectively the positions and yaw angle of the tractor while v denotes the speed. The steering angle and hydrostat position are respectively

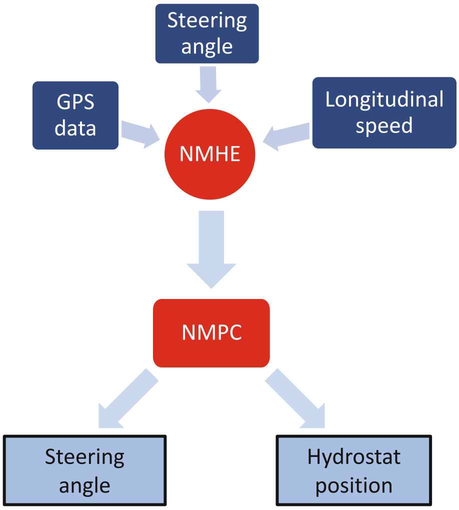

Fig. 6 Block diagram of the NMHE-NMPC framework

denoted by δ and *HP*. Additionally, μ and η denote the traction coefficients for the wheel and side slips.

The equations in (23) are formulated in the following form;

$$\dot{\xi} = f\left(\xi, u, p\right) \tag{24}$$

$$y = h\left(\xi, u, p\right) \tag{25}$$

with

$$\xi = \left[x\ y\ \psi\ v\right]^T \tag{26}$$

$$u = \left[\delta\ HP\right]^T \tag{27}$$

$$p = \left[\mu\ \eta\right]^T \tag{28}$$

$$y = \left[x\ y\ v\ \delta\ HP\right]^T \tag{29}$$

where ξ, u, p and y denote respectively the vectors of state, input, parameter and output of the system. The measured physical parameter is: $L = (1.4\,\mathrm{m})$, and the identified parameters are [38]: $\tau = 2.05$ and $K = 0.016$ for the speed model while the engine speed is at 2500 RPM.

3.2.1 Nonlinear Moving Horizon Estimation

In advanced model-based control structures, learning phenomena are required and realized through online parameter estimation as they make use of the system model

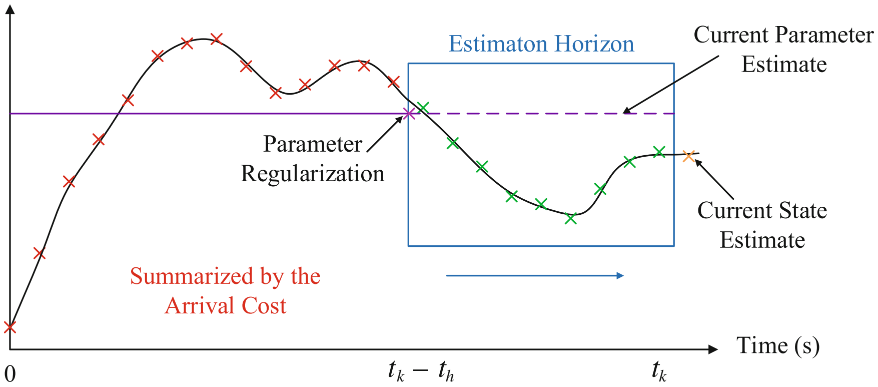

Fig. 7 Illustration for the concept of NMHE

to generate control signals, and have to deal with uncertain and varying process conditions. Therefore, it is inevitable to use adaptive models. In this study, nonlinear moving horizon estimation method has been chosen as a state and parameter estimation algorithm because it considers the state and parameter estimation within the same problem and allows to incorporate constraints both on states and parameters. NMHE is illustrated in Fig. 7 and formulated in (30).

$$\begin{aligned} \min_{\xi(.),dp,u(.)} \quad & \left\| \begin{matrix} \hat{\xi} - \xi(t_k - t_h) \\ \hat{p} - p \end{matrix} \right\|^2_{V_s} + \int_{t_k - t_h}^{t_k} \left(\|y_m - y(t)\|^2_{V_y} \right) dt \\ \text{subject to} \quad & \dot{\xi}(t) = f\left(\xi(t), u(t), p\right) \\ & y(t) = h\left(\xi(t), u(t), p\right) \\ & \xi_{min} < \xi < \xi_{max} \\ & p_{min} < p < p_{max} \qquad \forall t \in [t_k, t_{k+1}] \end{aligned} \tag{30}$$

In practice, only state estimation is not enough to know the system behaviour when uncertain systems are considered. Hence, parameter estimation is required to determine unmeasurable parameters. A parametric least square estimation subject to the system model and/or boundary conditions has been studied. There are many software packages to solve optimization problems for offline parameter estimation [29].

Two approximations have been proposed for online parameter estimation which is necessary simultaneously with state estimation to find system behavior accurately. In the first choice, model parameters are assumed as so-called "random constants" represented by a differential equation $\dot{\xi}_p = 0$ with initial value $\xi_p(t_k) = p_k$. This approach results in time-invariant parameters over the estimation horizon. If jumps or drifts for parameters are expected, which is the case in practice under varying working conditions, a model bias would occur. As a solution to the jump and drifts problem, the model parameters must be assumed as time-varying. Model parameters are assumed as so-called "random walk" by a differential equation $\dot{\xi}_p = \frac{d^p}{\Delta t}$ with sampling time Δt and initial value $\xi_p(t_k) = p_k$. It is assumed that the parameters are time-varying Gaussian random variables in the arrival cost.

The reference estimated values $\hat{\xi}(t_k - t_h)$ and $\hat{p}$ are taken from the solution of NMHE at the previous estimation instant. The arrival cost matrix V_s has been chosen as a so-called smoothed EKF-update based on sensitivity information obtained while solving the previous NMHE problem [30]. The contributions of the past measurements to the inverted Kalman covariance V_s are downweighted by a process noise covariance matrix D_{update} in (32) [31–35].

The adaptive kinematic model presented in (23) is used in the NMHE design for the state and parameter estimation. The NMHE problem is solved at each sampling time with the following constraints on the parameters:

$$\begin{aligned} 0.25 &\leq \mu \leq 1 \\ 0.25 &\leq \eta \leq 1 \end{aligned} \tag{31}$$

The standard deviations of the measurements have been set to $\sigma_x = \sigma_y = 0.03$ m, $\sigma_v = 0.1$ m/s, and $\sigma_\delta = 0.0175$ rad based on the information obtained from the real-time experiments. Thus, the following weighting matrices V_y, and D_{update} have been used in the NMHE implementation:

$$\begin{aligned} V_y &= diag(\sigma_x, \sigma_y, \sigma_v, \sigma_\delta)^{-1} \\ &= diag(0.03, 0.03, 0.1, 0.0175)^{-1} \end{aligned} \tag{32}$$

$$\begin{aligned} D_{update} &= diag(x, y, \psi, \mu, \eta, v) \\ &= diag(10.0, 10.0, 0.1, 0.25, 0.25, 0.1)^{-1} \end{aligned} \tag{33}$$

The estimation horizon t_h is set to 3 s.

3.2.2 Nonlinear Model Predictive Control

In this study, an NMPC formulation at each sampling time t is considered in the following form:

$$
\begin{aligned}
\min_{\xi(.),u(.)} \quad & \int_{t_k}^{t_k+t_h} \Big(\|\xi_r(t) - \xi(t)\|_Q^2 + \|u_r(t) - u(t)\|_R^2 \Big) dt \\
& + \|\xi_r(t_k + t_h) - \xi(t_k + t_h)\|_S^2 \\
\text{s.t.} \quad & \xi(t_k) = \hat{\xi}(t_k) \\
& \dot{\xi}(t) = f\big(\xi(t), u(t), p\big) \\
& \xi_{min} \le \xi(t) \le \xi_{max} \\
& u_{min} \le u(t) \le u_{max} \quad \forall t \in [t_k, t_k + t_h]
\end{aligned}
\tag{34}
$$

where the first and last parts are called the stage cost and the terminal penalty enforced the stability of NMPC in [36] in which $Q \in \mathbb{R}^{n_\xi \times n_\xi}$, $R \in \mathbb{R}^{n_u \times n_u}$ and $S \in \mathbb{R}^{n_\xi \times n_\xi}$ are symmetric positive definite weighting matrices, ξ_r and u_r denote respectively the references for the states and inputs, ξ and u denote respectively the states and inputs, t_k denotes the current time, t_h denotes the prediction horizon. $\hat{\xi}(t_k)$ denotes the estimated state vector by the NMHE, ξ_{min}, ξ_{max}, u_{min} and u_{max} denote respectively the upper and lower constraints on the state and input. The terminal constraints are denoted by $\xi(t_k + t_h)_{min}$ and $\xi(t_k + t_h)_{max}$. The first sample of $u(t)$, $u(t, \xi(t)) = u^*(t_k)$, is applied to the system and the NMPC problem is solved again over a moved horizon for the subsequent sampling time.

The constraints on the inputs, the steering angle and hydrostat position references, are written:

$$
\begin{aligned}
-35\,\text{deg} \; & \le \delta(t) \le \; 35 \text{ deg} \\
0\% \quad & \le HP(t) \le \; 100\%
\end{aligned}
\tag{35}
$$

The references for the state and inputs are written:

$$
\begin{aligned}
\xi_r &= (x_r, y_r, \psi_r, v_r)^T \\
u_r &= (\delta_r, HP_r)^T
\end{aligned}
\tag{36}
$$

The inputs references are the last measured steering angle and hydrostat position while the states references are relied on the reference trajectory to be tracked.

The weighting matrices Q, R and S have been written:

$$
\begin{aligned}
Q &= diag(1, 1, 0, 0) \\
R &= diag(5, 5) \\
S &= diag(10, 10, 0, 0)
\end{aligned}
\tag{37}
$$

The prediction horizon t_h is set to 3 s.

3.2.3 Experimental Results for the Model-Based Controller

Throughout the experiments, the tractor has faced with uneven terrain, and been succeed in staying on-track for the NMHE-NMPC framework as shown in Fig. 8a. The sampling time of the frameworks is 200 ms in real-time. The Euclidean error for the tractor is shown in Fig. 8b. The mean values of the Euclidean error of the tractor have been obtained 18.16 cm for the straight lines while 52.02 cm for the curved lines. It is observed that the trajectory tracking error for the system for straight lines has been less than the one for the curved lines as shown in Fig. 8b.

The outputs of the controller, which are the steering angle reference for the tractor (δ^t), and the hydrostat position (HP) reference as illustrated in Fig. 8c. As seen in this figure, the control signals stay within the bounds. Moreover, the estimated traction parameters by the NMHE are shown in Fig. 8d. The estimates stay within the bounds.

For auto generation of the C codes, an open source software is preferred: the ACADO [37] code generation tool. This too can be used to solve the constrained nonlinear optimization problems in the NMPC and NMHE. First, we have created the

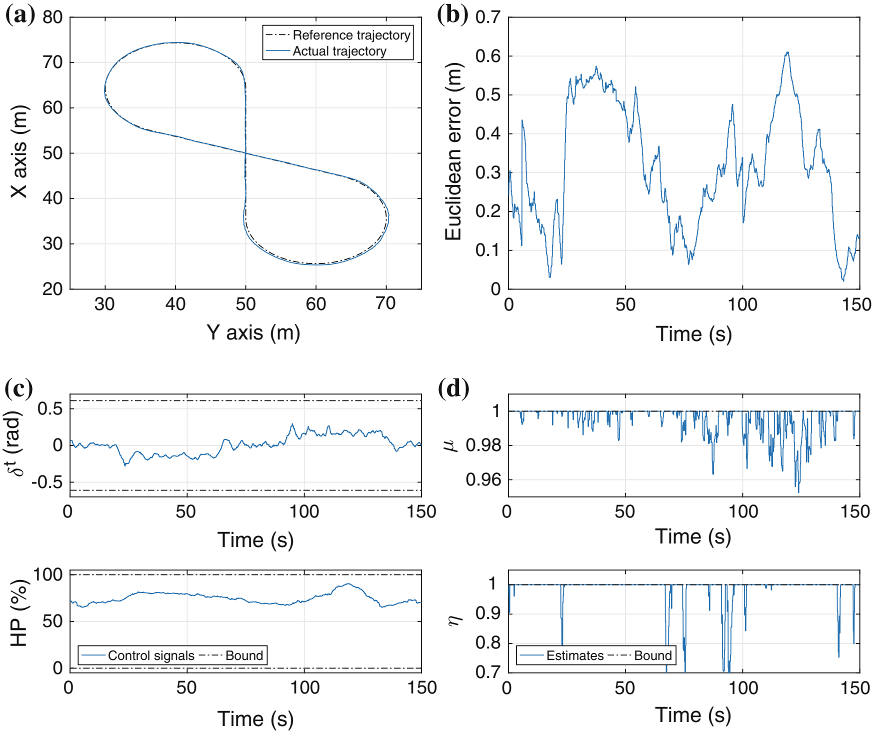

Fig. 8 **a** Reference and actual trajectories **b** Euclidean error to the reference trajectory **c** Control inputs **d** Traction parameters

C codes by using the ACADO, which is then converted into a .dll file to be used in LabVIEW. Detailed information on the ACADO code generation tool can be found in [29, 37].

3.3 Overall Comparison of the Model-Based and Model-Free Learning Control

Table 1 is a candidate guideline to choose an appropriate control algorithm for the control of agricultural robotic systems which are generally complex mechatronic systems. According to observations, when the model of the system as well as the interactions between the subsystems are precisely modeled, a model-based controller (MPC-MHE framework) is preferable. This advanced control framework is not only very accurate but also robust. Moreover, there are some open course fast solvers, such as Acado toolkit, which generates C/C++ codes for a real-time implementation. What is more, although early MPC applications were restricted only to slow systems that long computation times could be tolerated, recent progress in microprocessor technology has motivated applications of MPC for fast dynamic systems, such autonomous vehicles.

However, if the modeling of the system is challenging or unfeasible, model-free control algorithms can be used even if it might be difficult to prove their stability. In addition to their instability problems, pure model-free methods may be unstable in the beginning of the experiment depending on the initial weights. If the system dynamics are fast, this may cause serious problems, such as fast vehicles or unmanned aerial vehicles. In order to make sure that the system is stable in the beginning of the learning process, an alternative method, which is the combination of a conventional controller and an intelligent structure. This fusion is called feedback error learning, which is also promising in real time if there is no precise model at hand. These controllers have the ability of learning throughout the operation if an appropriate optimization algorithm is used.

No matter the model-free controller is a pure model-free or a feedback error learning-based controller, another prominent feature of them is that the time spent for modeling does not exist for model-free controllers. It is to be noted that the modeling stage may take more time than designing of a controller in the case of having a model-based controller. In particular, in addition to its nonlinearities, if the system has dead-zones and hysteresis, modeling of the system is a very tedious work [38]. These challenging systems include, but are not limited to, electro-hydraulic actuators and valves, diesel engines and pneumatic actuators.

Table 1 Guidelines for the selection of the best modeling and control approach for the complex mechatronic systems

Model-based techniques	
When?	Interactions are known accurately
Why?	Allows to design the controller analytically and to prove the stability of the overall system
Why not?	In practical applications, the interactions are not so easy to be modelled
Pure model-free techniques	
When?	Interactions are difficult to be known
Why?	No need the mathematical model of the system to be controlled
Why not?	Impossible to prove the stability of the overall system and impossible to calculate the parameters of the controller analytically
Feedback error learning	
When?	A conventional controller to guarantee the stability of the plant
Why?	After the intelligent controller has learned the system dynamics, it takes the responsibility of controlling the system
Why not?	The stability of the overall system may be challenging to be shown

4 Conclusions

A fully autonomous tractor-trailer system is designed and prototyped by using off the shelf components. The system is able to follow both straight line and curved line trajectories with a satisfactory accuracy. Both model-based and model-free controllers are designed to navigate the system, and their performances are compared and contrasted. According to the real-time results, when the model of the system as well as the interactions between the subsystems are precisely modeled, a model-based controller is preferable. On the other hand, a model-free controller is preferable if the mathematical model of the system is challenging or unfeasible. As a model-free control algorithm, type-2 fuzzy logic controllers are able to learn the systems dynamics online, and have the ability to control the system with a limited information about the system. As a parameter update algorithm, a sliding mode control theory-based learning algorithm is preferred which need neither partial derivatives nor matrix inversions. These features make the learning algorithm not only robust but also computationally efficient which is a big advantage in real-time implementations where the computation power is limited.

References

1. S. Riehl, M. Zeidi, N.J. Conard, Emergence of agriculture in the foothills of the Zagros mountains of Iran. Science **341**(6141), 65–67 (2013). https://doi.org/10.1126/science.1236743, http://www.sciencemag.org/content/341/6141/65.abstract

2. S. Riehl, M. Benz, N. Conard, H. Darabi, K. Deckers, H. Nashli, M. Zeidi-Kulehparcheh, Plant use in three Pre-Pottery Neolithic sites of the northern and eastern Fertile Crescent: a preliminary report. Veg. Hist. Archaeobotany **21**(2), 95–106 (2012). https://doi.org/10.1007/s00334-011-0318-y
3. G. Willcox, The roots of cultivation in Southwestern Asia. Science **341**(6141), 39–40 (2013). https://doi.org/10.1126/science.1240496, http://www.sciencemag.org/content/341/6141/39.short
4. J. Oeppen, J.W. Vaupel, Broken limits to life expectancy. Science **296**(5570), 1029–1031 (2002). https://doi.org/10.1126/science.1069675, http://www.sciencemag.org/content/296/5570/1029.short
5. F. Sistler, Robotics and intelligent machines in agriculture. IEEE J. Robot. Autom. **3**(1), 3–6 (1987). https://doi.org/10.1109/JRA.1987.1087074
6. R. Harrell, Economic analysis of robotic citrus harvesting in Florida. Trans. ASAE **30**(2), 298–304 (1987)
7. N. Irie, N. Taguchi, T. Horie, T. Ishimatsu, Asparagus harvesting robot coordinated with 3-D vision sensor, in *Industrial Technology, ICIT 2009. IEEE International Conference on*, pp. 1–6 (2009). https://doi.org/10.1109/ICIT.2009.4939556
8. H. Shen, D. Zhao, W. Ji, Y. Chen, J. Lv, Research on the Strategy of Advancing Harvest Efficiency of Fruit Harvest Robot in the Oscillation Conditions, in *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2011 International Conference on*, vol. 1 (2011), pp. 215–218. https://doi.org/10.1109/IHMSC.2011.58
9. T.T. Nguyen, K. Vandevoorde, N. Wouters, E. Kayacan, J.G.D. Baerdemaeker, W. Saeys, Detection of red and bicoloured apples on tree with an RGB-D camera. Biosyst. Eng. (2016). https://doi.org/10.1016/j.biosystemseng.2016.01.007, http://www.sciencedirect.com/science/article/pii/S1537511016000088
10. A. Gollakota, M.B. Srinivas, Agribot A multipurpose agricultural robot, in *2011 Annual IEEE India Conference* (2011), pp. 1–4. https://doi.org/10.1109/INDCON.2011.6139624
11. D.A. Johnson, D.J. Naffin, J.S. Puhalla, J. Sanchez, C.K. Wellington, Development and implementation of a team of robotic tractors for autonomous peat moss harvesting. J. Field Robot. **26**(6–7), 549–571 (2009), https://doi.org/10.1002/rob.20297
12. D. Bochtis, C. Sorensen, S. Vougioukas, Path planning for in-field navigation-aiding of service units. Comput. Electron. Agric. **74**(1), 80–90 (2010). https://doi.org/10.1016/j.compag.2010.06.008, http://www.sciencedirect.com/science/article/pii/S0168169910001250
13. N. Noguchi, J. Will, J. Reid, Q. Zhang, Development of a master–slave robot system for farm operations. Comput. Electron. Agric. **44**(1), 1–19 (2004). https://doi.org/10.1016/j.compag.2004.01.006, http://www.sciencedirect.com/science/article/pii/S0168169904000316
14. C. Cariou, R. Lenain, B. Thuilot, M. Berducat, Automatic guidance of a four-wheel-steering mobile robot for accurate field operations. J. Field Robot. **26**(6–7), 504–518 (2009). https://doi.org/10.1002/rob.20282
15. J. Backman, T. Oksanen, A. Visala, Navigation system for agricultural machines: Nonlinear Model Predictive path tracking. Comput. Electron. Agric. **82**, 32–43 (2012). https://doi.org/10.1016/j.compag.2011.12.009, http://www.sciencedirect.com/science/article/pii/S0168169911003218
16. H. Hagras, M. Colley, V. Callaghan, M. Carr-West, Online learning and adaptation of autonomous mobile robots for sustainable agriculture. Auton. Robots **13**(1), 37–52 (2002). https://doi.org/10.1023/A:1015626121039
17. M. Bergerman, S.M. Maeta, J. Zhang, G.M. Freitas, B. Hamner, S. Singh, G. Kantor, Robot farmers: autonomous orchard vehicles help tree fruit production. IEEE Robot. Autom. Mag. **22**(1), 54–63 (2015). https://doi.org/10.1109/MRA.2014.2369292
18. F. Rovira-Ms, I. Chatterjee, V. Siz-Rubio, The role of GNSS in the navigation strategies of cost-effective agricultural robots. Comput. Electron. Agric **112**, 172–183 (2015). https://doi.org/10.1016/j.compag.2014.12.017, http://www.sciencedirect.com/science/article/pii/S0168169914003275. Precision Agriculture

19. A.V. Topalov, E. Kayacan, Y. Oniz, O. Kaynak, Adaptive neuro-fuzzy control with sliding mode learning algorithm: Application to Antilock Braking System, in *2009 7th Asian Control Conference* (2009), pp. 784–789
20. H. Li, C. Wu, P. Shi, Y. Gao, Control of nonlinear networked systems with packet dropouts: interval type-2 fuzzy model-based approach. IEEE Trans. Cybern. **45**(11), 2378–2389 (2015). https://doi.org/10.1109/TCYB.2014.2371814
21. H.K. Lam, H. Li, C. Deters, E.L. Secco, H.A. Wurdemann, K. Althoefer, Control design for interval type-2 fuzzy systems under imperfect premise matching. IEEE Trans. Ind. Electron. **61**(2), 956–968 (2014). https://doi.org/10.1109/TIE.2013.2253064
22. O. Castillo, P. Melin, A review on interval type-2 fuzzy logic applications in intelligent control. Inf. Sci. **279**, 615–631 (2014). https://doi.org/10.1016/j.ins.2014.04.015, http://www.sciencedirect.com/science/article/pii/S0020025514004629
23. J. Mendel, H. Hagras, W.W. Tan, W.W. Melek, H. Ying, *Introduction To Type-2 Fuzzy Logic Control: Theory and Applications*, 1st edn. (Wiley-IEEE Press, 2014)
24. E. Kayacan, O. Kaynak, R. Abiyev, J. Trresen, M. Hvin, K. Glette, Design of an adaptive interval type-2 fuzzy logic controller for the position control of a servo system with an intelligent sensor, in *International Conference on Fuzzy Systems* (2010), pp. 1–8. https://doi.org/10.1109/FUZZY.2010.5584629
25. E. Kayacan, E. Kayacan, M.A. Khanesar, Identification of nonlinear dynamic systems using type-2 fuzzy neural networks—A novel learning algorithm and a comparative study. IEEE Trans. Industr. Electron. **62**(3), 1716–1724 (2015). https://doi.org/10.1109/TIE.2014.2345353
26. E. Kayacan, W. Saeys, E. Kayacan, H. Ramon, O. Kaynak, Intelligent control of a tractor-implement system using type-2 fuzzy neural networks. *2012 IEEE International Conference on Fuzzy Systems* (2012), pp. 1–8. https://doi.org/10.1109/FUZZ-IEEE.2012.6250790
27. A. Dutta, Y. Zhong, B. Depraetere, K.V. Vaerenbergh, C. Ionescu, B. Wyns, G. Pinte, A. Nowe, J. Swevers, R.D. Keyser, Model-based and model-free learning strategies for wet clutch control. Mechatronics **24**(8), 1008–1020 (2014). https://doi.org/10.1016/j.mechatronics.2014.03.006, http://www.sciencedirect.com/science/article/pii/S0957415814000622
28. E. Kayacan, E. Kayacan, H. Ramon, O. Kaynak, W. Saeys, Towards agrobots: trajectory control of an autonomous tractor using type-2 fuzzy logic controllers. Mechatron. IEEE/ASME Trans. **20**(1), 287–298 (2015). https://doi.org/10.1109/TMECH.2013.2291874
29. B. Houska, H.J. Ferreau, M. Diehl, ACADO toolkit - An open-source framework for automatic control and dynamic optimization. Optimal Control Appl. Meth. **32**(3), 298–312 (2011)
30. D. Robertson, *Development and Statistical Interpretation of Tools for Nonlinear Estimation* (Auburn University, 1996)
31. T. Kraus, H. Ferreau, E. Kayacan, H. Ramon, J.D. Baerdemaeker, M. Diehl, W. Saeys, Moving horizon estimation and nonlinear model predictive control for autonomous agricultural vehicles. Comput. Electron. Agric. **98**, 25–33 (2013)
32. E. Kayacan, J.M. Peschel, E. Kayacan, Centralized, decentralized and distributed nonlinear model predictive control of a tractor-trailer system: A comparative study. *2016 American Control Conference (ACC)*, (2016), pp. 4403–4408. https://doi.org/10.1109/ACC.2016.7525615
33. E. Kayacan, E. Kayacan, H. Ramon, W. Saeys, Robust tube-based decentralized nonlinear model predictive control of an autonomous tractor-trailer system. IEEE/ASME Trans. Mechatron. **20**(1), 447–456 (2015). https://doi.org/10.1109/TMECH.2014.2334612
34. E. Kayacan, E. Kayacan, H. Ramon, W. Saeys, Learning in centralized nonlinear model predictive control: application to an autonomous tractor-trailer system. IEEE Trans. Control Syst. Technol. **23**(1), 197–205 (2015). https://doi.org/10.1109/TCST.2014.2321514
35. E. Kayacan, E. Kayacan, H. Ramon, W. Saeys, Distributed nonlinear model predictive control of an autonomous tractor–trailer system. Mechatronics **24**(8), 926–933 (2014). https://doi.org/10.1016/j.mechatronics.2014.03.007, http://www.sciencedirect.com/science/article/pii/S0957415814000634
36. D. Mayne, J. Rawlings, C. Rao, P. Scokaert, Constrained model predictive control: stability and optimality. Automatica **36**(6), 789–814 (2000)

37. B. Houska, H.J. Ferreau, M. Diehl, An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. Automatica **47**(10), 2279–2285 (2011)
38. E. Kayacan, E. Kayacan, H. Ramon, W. Saeys, Nonlinear modeling and identification of an autonomous tractor–trailer system. Comput. Electron. Agric. **106**, 1–10 (2014). https://doi.org/10.1016/j.compag.2014.05.002, http://www.sciencedirect.com/science/article/pii/S0168169914001252