High Precision Control of Tracked Field Robots in the Presence of Unknown Traction Coefficients

Erkan Kayacan
Distributed Autonomous Systems Laboratory
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
Urbana, IL 61801
erkank@mit.edu

Sierra N. Young
Department of Civil and Environmental Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801
snyoung2@illinois.edu

Joshua M. Peschel
Dept. of Agricultural and Biosystems Engineering
Iowa State University
Ames, Iowa 50011-3270
peschel@iastate.edu

Girish Chowdhary
Distributed Autonomous Systems Laboratory
Coordinated Science Laboratory
Dept. of Agricultural and Biological Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801
girishc@illinois.edu

Abstract

Accurate steering through crop rows that avoids crop damage is one of the most important tasks for agricultural robots utilized in various field operations, such as monitoring, mechanical weeding, or spraying. In practice, varying soil conditions can result in off-track navigation due to unknown traction coefficients so that it can cause crop damage. To address this problem, this paper presents the development, application, and experimental results of a real-time receding horizon estimation and control (RHEC) framework applied to a fully autonomous mobile robotic platform to increase its steering accuracy. Recent advances in cheap and fast microprocessors, as well as advances in solution methods for nonlinear optimization problems, have made nonlinear receding horizon control (RHC) and receding horizon estimation (RHE) methods suitable for field robots that require high frequency (milliseconds) updates. A real-time RHEC framework is developed and applied to a fully autonomous mobile robotic platform designed by the authors for in-field phenotyping applications in Sorghum fields. Nonlinear RHE is used to estimate constrained states and parameters, and nonlinear RHC is designed based on an adaptive system model which contains time-varying parameters. The capabilities of the real-time RHEC framework are verified experimentally, and the results show an accurate tracking performance on a bumpy and wet soil field. The mean values of the Euclidean error and required computation time of the RHEC framework are respectively equal to 0.0423 m and 0.88 milliseconds.

1 Introduction

Due to the petroleum crisis and growing demands for renewable energy sources [Tverberg, 2012], there have been increased research efforts to improve bioenergy crop breeding for biofuel production [Wang et al., 2016, Buckeridge et al., 2014]. The success and sustainability of biofuels are dependent on increasing plant yields [Vega-Sanchez and Ronald, 2010, Yano and Tuberosa, 2009]. Top-yielding plant traits (phenotypes) are linked to their corresponding

genes (genotypes). Identifying useful phenotypes can help in identifying which genes will increase the productivity of biofuel crops. However, the process of phenotyping is cost and labor intensive. On the other hand, stationary high throughput phenotyping platforms, or greenhouse based phenotyping systems are bulky, costly, and require elaborate infrastructure [Fiorani and Tuberosa, 2013]. Therefore, there is a high need for an automated, mobile, sensory platform that could significantly increase the throughput and accuracy of screening crops in the field.

One of the major challenges in developing robotic phenotyping platforms is high-precision guidance and control in muddy and uneven agricultural fields [Kayacan et al., 2014a, Kayacan et al., 2015a]. A high-precision control scheme is necessary to avoid crop disturbance and damage. For example, Figure 1 depicts the TERRA-MEPP phenotyping robot under development at the University of Illinois at Urbana Champaign. It is a typical tracked robot that carries hyperspectral, infrared, visual spectrum, and lidar sensors for phenotyping. Due to the necessity to carry large sensing payloads, the width a phenotyping robot can be significant, for example, it is 0.48 m for the robot shown in this study. However, the row spacing of sorghum cultivated for biofuel production may vary from 0.5 m to 1 m, with a typical row spacing measured centerline to centerline; of 0.78 m as used in this study. Furthermore, the stem width of a fully mature sorghum plant and the accuracy of the global navigation satellite system used in this study can be up to 0.03 m so that the effective row spacing is reduced to 0.72 m, which leaves only 0.12 m (4.5 in) of space on either side of the robot. Uncertain traction coefficients and uneven fields can make it difficult for the robot to ensure that it stays within the 0.12 m tolerance so as to not damage the crop. It is the goal of this paper to describe the development of high precision guidance and motion control system that can ensure a navigational tolerance of strictly less than 0.12 m in real-world field environments.

Figure 1: Field robot in sorhgum plants in Energy Farm, Urbana, IL, USA.

Tracked robots are preferred to wheeled robots for field-based phenotyping applications due to their improved traction and large contact area with the ground, which minimizes adverse impacts on the soil [Bekker, 1956]. Tracked robots, however, have complex track-ground interactions and slippage due to differential velocities between treads, which can prompt difficulties in control. The presence of these time-varying parameters can degrade the path-tracking performance when traditional control approaches (e.g., proportional-integral-derivative control) are employed [Fukao et al., 2000]. The reason is that traditional controllers are not aware of soil parameters of unknown terrain which play a vital role in determining field robot's speed and steering, which are in turn utilized for developing traction control

algorithms [Kayacan et al., 2012b]. Therefore, there is a need to develop advanced control algorithms with online parameter estimation to update key terrain and slip parameters, which can be used to improve robot mobility and localization.

Field robots are multi-input-multi-output systems and must be aware of actuator limitations [Kayacan et al., 2012a, Kayacan et al., 2015b, Kayacan and Peschel, 2016, Kayacan, 2017]. Therefore, real-time applications of linear model predictive controllers (MPCs), i.e., RHCs, have been used for path tracking of mobile robots, but there are significant limitations of these linear control techniques [Kayacan et al., 2015c]. Early linear MPCs have been designed based on a tracking-error based linear model, and the total control input consists of the feedback and feedforward control actions [Lee et al., 2001, Gu and Hu, 2006, Lins Barreto et al., 2014]. In this scheme, the linear MPC is the feedback controller, and the feedforward control action is derived by taking the reference trajectory and system model into consideration [Klancar and Skrjanc, 2007]. However, these linear control techniques are unable to ensure accurate tracking performance if the systems start off-track or there exist uncertainties, e.g., unknown terrain parameters. Linear control techniques perform well for systems that work at fixed operating-points around which the system is linearized. However, mobile robots work at varying operating points (e.g., tracking of a curvilinear trajectory) and are subjected to several uncertainties (e.g., bumpy fields, variable soil conditions) so that locally linearized models are infeasible. Additionally, linear MPCs cause large system errors in the prediction horizon with a possible instability of the system because the mismatch between the linearized model and the mobile robot increases when mobile robots move away from the target path [Falcone et al., 2007]. To overcome these limitations, a robust tube-based MPC has been developed and tested in real-time [Kayacan et al., 2016b]. Although the system could stay-on track, a highly accurate trajectory tracking performance has not been obtained.

To achieve highly accurate tracking performance for constrained navigation in row crops, nonlinear RHC approaches have been widely used, particularly in agricultural applications as they are capable of taking constraints on states and inputs into account and can be designed for nonlinear models [Backman et al., 2012, Utstumo et al., 2015, Kayacan et al., 2018]. Additionally, an accurate online estimation of traction parameters in a system model is crucial as the tire-soil interactions change throughout operations in varying, slippery conditions. Thus, the system model can represent all the system dynamics and interactions in the real-time system. Although nonlinear RHC approach is computationally intensive, availability of compact and powerful computational packages and advances in numerical methods for solving optimization problems have allowed for real-time applications of constrained nonlinear optimization problems for complex systems [Naveau et al., 2017, Oh et al., 2015]. Motivated by the previous studies, this study employs a nonlinear receding horizon estimation and control (RHEC) framework for path tracking with a mobile, phenotyping robot.

The main contributions of this study are as follows.

- In this work, we augment the kinematic model with traction parameters to capture soil characteristics of agricultural fields, while the system model is specified a priori and remains unchanged during operation in traditional implementations.

- We provide robust tracking performance when uncertainty is high as uncertainty is reduced through learning whereas traditional RHC approaches do not typically account for model uncertainty.

- Highly tracking accuracy, e.g., less than 0.12 m, for a tracked field robot under unknown and variable soil conditions, has been obtained in real-time by developing and implementing a nonlinear RHEC framework with a sampling time in the range of milliseconds.

- Computation times for the nonlinear RHEC have been decreased by restricting the number of Gauss-Newton iterations to 1 while solving the nonlinear optimization problems.

This paper is organized as follows: Section 2 provides the system model with traction parameters and the description of the tracked mobile robot used in this study. Section 3 is devoted to describing the nonlinear RHEC approaches. Implementations and solution methods are presented in Section 4. Validation and field tests for the real-time nonlinear RHEC algorithms are presented in Section 5. Finally, conclusions are summarized in Section 6.

2 Field Robot

2.1 System Description

The field robot as shown in Fig. 2 has been built utilizing practical, hands-on experience with modeling, control and estimation techniques, and various sensors and actuators. It is 0.48 m wide, 1.02 m long and around 45 kgs, allowing it to navigate 0.78 m wide crop rows without damaging the plants and maintaining vertical stability. The mast on the robot can be adjusted to match the sorghum canopy height and is used as a mounting platform for both the plant imaging sensors and a Real-Time Kinematic (RTK) differential Global Navigation Satellite System (GNSS) to acquire positional information. A Septentrio Altus APS-NR2 GNSS receiver (Septentrio Satellite Navigation NV, Belgium) is used to obtain highly accurate positional information, which has a specified position accuracy of 0.03 m at a 5-Hz measurements rate. The Trimble network supplies the RTK correction signals via the 4G internet. Two DG-158 A DC motors (SuperDroid Robots Inc., U.S.) capable of 200 W power output with 16.95 Newton-meter rated torque are used to drive the tracks. E2 optical encoders (US Digital, USA) on DC motors provide speed information with an accuracy of 0.035 m/s. Two 12 V, 40 Ah 4s 20c lipo batteries power the field robot and enable over 3 hours of operation.
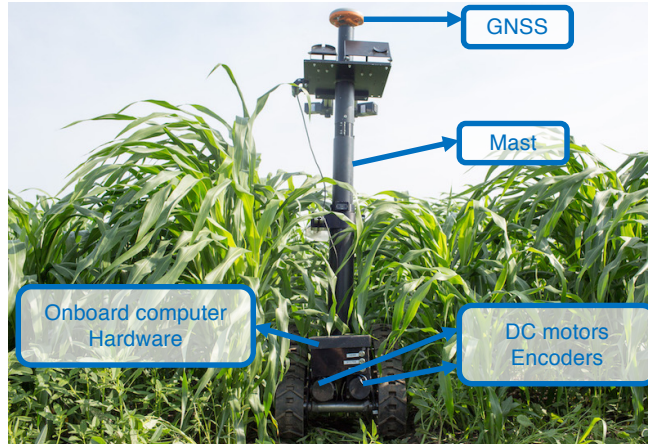

Figure 2: Field robot in sorghum breeding plots in Energy farm, IL, USA.

The real-time estimation and control framework is implemented and executed on an on-board computer (Raspberry Pi 3 Model B), which is equipped with 1.2 GHz Quad Core Cortex-A53 64-Bit CPU, 1 GB RAM. It acquires the GNSS data, the speed of the wheels, and controls the tracked robot by sending signals to the Kangaroo x2 motion controller (Dimension Engineering, USA). The generated control input is sent to the motion controller via the on-board computer, and speed measurements of the wheels are received with the same way. Positional measurements are obtained via Bluetooth. The sampling frequency is set to 5-Hz due to the fact that the maximum update rate of the GNSS is equal to 5-Hz.

The block diagram of the hardware is illustrated in Fig. 3. The nonlinear receding horizon estimation (RHE) receives the measurements (8) from onboard sensors, and estimates the full states and parameters (5)-(6). The states and parameters estimated by the nonlinear RHE and a reference trajectory are used as inputs into the nonlinear RHC, which generates the control signal, i.e., the desired yaw rate. The desired yaw rate and speed as reference command signals are then sent to the Kangaroo x2 Motion Controller that adds self-tuning motion control to a Sabertooth dual 12A motor driver (Dimension Engineering, USA), which allows us to control the speed of the DC motors. The Kangaroo x2 Motion Controller, which is a two channel self-tuning PID controller, functions as the field robot's low-level controller by using feedback from the encoders attached to the DC motors to determine the required control signals. The Kangaroo x2 motion controller outputs the modified command signals to Sabertooth dual 12A motor driver, which correlates the given control signals to the necessary output voltages needed by the DC motors. In other words, the inner loop, the speed of the DC motors, is controlled by the Kangaroo x2 motion controller, which eliminates the need of manually tune the low-level controller, and executed at a rate of 50-Hz.
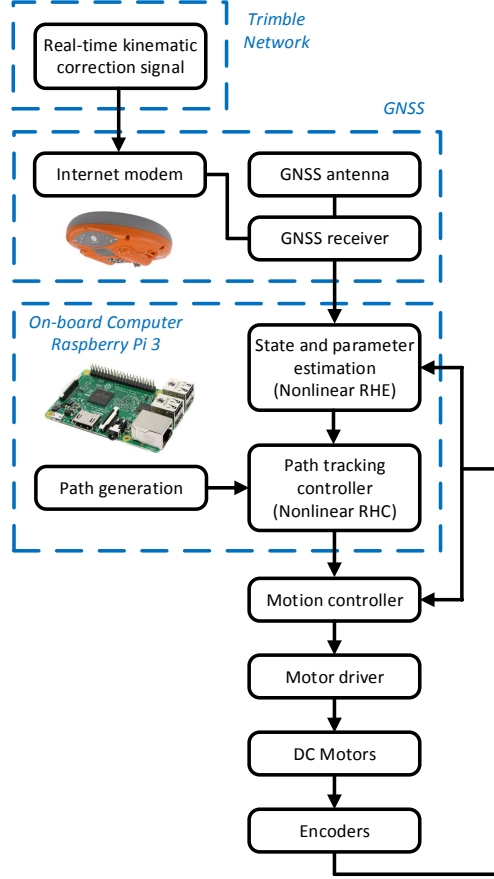
Figure 3: Block diagram of hardware

2.2 System Model with Traction Coefficients

The schematic diagram of the field robot, which is a typical tracked mobile robot, is illustrated in Fig. 4. The velocities of two driven tracks result in linear velocity $v = (v_l + v_r)/2$ and angular velocity $\omega = (v_r - v_l)/b$ with the distance between tracks b. The traditional kinematic model is formulated as follows [Kanayama et al., 1990]:

$$\begin{aligned} \dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{1}$$

where x and y denote the position of the field robot, θ denotes the yaw angle, v denotes the speed, ω denotes the yaw rate.

The effect of terrain characteristics on robot performance and track should be well covered because the tracks are the unique connections between the ground and robot, and nearly all forces and moments applied to the robot are transmitted through the tracks [Iagnemma et al., 2004, Ray, 2009]. From track-terrain interaction dynamics, soil characteristic plays a vital role in determining vehicle speed and steering, which in turn are utilized for developing traction control algorithms. The knowledge of soil parameters of unknown terrain is then advantageous for improving vehicle performance [Lee and Iagnemma, 2016, Lee et al., 2016]. Therefore, instead of using the traditional kinematic model of a mobile robot (1), an adaptive nonlinear kinematic model that is an extension of the traditional model is derived by adding two traction parameters (μ, κ) to minimize deviations between the real-time system and system
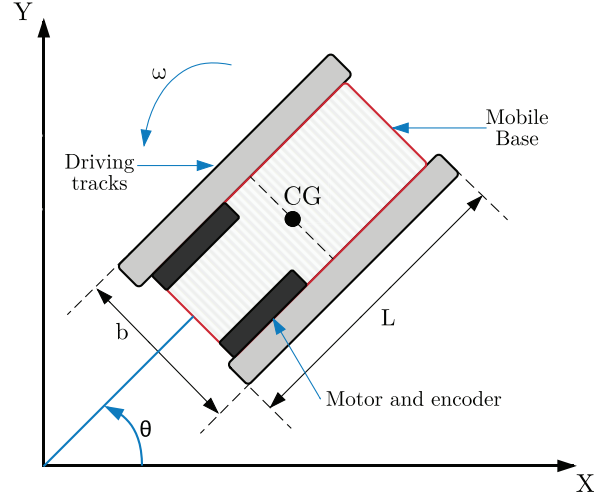
Figure 4: Schematic illustration of the field robot

model of the field robot in this study. The field robot can be formulated with the following equations:

$$
\begin{aligned}
\dot{x} &= \mu v \cos \theta \\
\dot{y} &= \mu v \sin \theta \\
\dot{\theta} &= \kappa \omega
\end{aligned}
\tag{2}
$$

where μ and κ denote the traction parameters. It is noted that they must be between zero and one, and it is inherently arduous to measure them. If the traction parameters are equal to 1, i.e., $\mu = \kappa = 1$, there exist no slip. The percentages of longitudinal and side slips are respectively found as $1 - \mu$ and $1 - \kappa$. These parameters show the effective speed μv and steering $\kappa \omega$ of the field robot. To avoid bias, these traction parameters must be estimated along with the full system state in each iteration based on a number of past measurements.

In the rest of the paper, we denote a nonlinear system model as

$$
\dot{\xi}(t) = f\left(\xi(t), u(t), p(t)\right)
\tag{3}
$$

where $\xi \in \mathbb{R}^{n_\xi}$ is the state vector, $u \in \mathbb{R}^{n_u}$ is the control input, $p \in \mathbb{R}^{n_p}$ is the system parameter vector, $f(\cdot, \cdot, \cdot) : \mathbb{R}^{n_\xi + n_u + n_p} \longrightarrow \mathbb{R}^{n_\xi}$ is the continuously differentiable state update function and $f(0, 0, p) = 0 \ \forall t$. The derivative of ξ with respect to t is denoted by $\dot{\xi} \in \mathbb{R}^{n_\xi}$.

Similarly, a nonlinear measurement model denoted $z(t)$ can be described with the following equation:

$$
z(t) = h\left(\xi(t), u(t), p(t)\right)
\tag{4}
$$

where $h : R^{n_\xi + n_u + n_p} \longrightarrow R^{n_z}$ is the measurement function which describes the relation between the variables of the system model and the measured outputs of the real-time system.

The state, parameters, input and output vectors are respectively denoted as follows:

$$
\begin{aligned}
\xi &= \begin{bmatrix} x & y & \theta \end{bmatrix}^T & (5) \\
p &= \begin{bmatrix} v & \mu & \kappa \end{bmatrix}^T & (6) \\
u &= \omega & (7) \\
z &= \begin{bmatrix} x & y & v & \omega \end{bmatrix}^T & (8)
\end{aligned}
$$

3 Nonlinear Receding Horizon Estimation and Control

3.1 Nonlinear Receding Horizon Estimation

The least squares estimation problem uses all available past measurements to estimate the current values of the system states. In this estimation scheme, the number of measurements increases and the optimization problem is enlarged since a new measurement is acquired at every time instant. Therefore, solving the relevant optimization problem becomes infeasible in real-time as time approaches infinity. However, in practice, the computational burden of the estimation problem must remain constant so that the number of past measurements considered in the estimation scheme must also be constant. Furthermore, the relationship between the system model with current estimated parameters and measurements in the past can be inconsistent. To deal with this problem, past measurements are deemed less influential or left out. The idea of the RHE approach is to use only a constant number of the past measurements so that the oldest data are removed in the moving window when new data is acquired [Robertson et al., 1996]. Thus, a fixed size window will be moving over the increasing measurement data so that the optimization problem will be fixed size. Then, the required method to incorporate the pre-information from the old measurements, which are not inside the moving window, and to prevent the heavy influence of the old measurements is discussed in subsection 3.1.1.

State estimation alone is not enough to know the system behavior in real-time when uncertain systems are considered; therefore, parameter estimation is required to determine uncertain parameters. A parametric least square estimation subject to the system model an/or boundary conditions was studied extensively in the literature, and there are many software packages to solve optimization problems for offline parameter estimation [Houska et al., 2011]. Moreover, two approximations have been proposed for online parameter estimation, which is necessary to determine system behavior in conjunction with state estimation accurately. In the first choice, model parameters are assumed as so-called random constant 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 will occur. To solve parameter jumps 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 $\dot{\xi}_p = \frac{dp}{\triangle t}$ with sampling time $\triangle t$ and initial value $\xi_p(t_k) = p_k$ [Hughes, 1995].

In the nonlinear RHE method, the parameters are assumed to be time-invariant in estimation horizon and not subject to process noise in the estimation horizon. However, it is also assumed that the parameters are time-varying Gaussian random variables in the arrival cost. For this reason, additional weighting factors must be added as the variance of the parameters noise and the parameters will appear only in the arrival cost. The extended weighting matrix $W \in \mathbb{R}^{(n_{xi}+n_p)\times(n_\xi+n_p)}$ is written as follows:

$$W = diag(W_\xi, W_p) \tag{9}$$

where $W_\xi \in \mathbb{R}^{n_\xi \times n_\xi}$ and $W_p \in \mathbb{R}^{n_p \times n_p}$ denote the weighting matrices for the state noise and parameter pseudo-variance. Large values in W result in large variations in the corresponding state and parameter estimates, and vice-versa.

The nonlinear RHE is formulated as follows:

$$
\begin{aligned}
\min_{\xi(t),p,u(t)} \quad & \frac{1}{2}\left\{ \left\| \begin{array}{c} \hat{\xi} - \xi(t_{k-N+1}) \\ \hat{p} - p \end{array} \right\|_{H_N}^2 + \sum_{i=k-N+1}^{k} \|z_m(t_i) - z(t_i)\|_{H_k}^2 \right\} \\
\text{s. t.} \quad & \dot{\xi}(t) = f\big(\xi(t), u(t), p\big) \\
& z(t) = h\big(\xi(t), u(t), p\big) \\
& \xi_{min} \leq \xi(t) \leq \xi_{max} \\
& p_{min} \leq p \leq p_{max} \qquad \forall t \in [t_{k-N}, t_k]
\end{aligned}
\tag{10}
$$

where the deviations of the oldest states in the estimation horizon and the parameters from the prior estimates $\hat{\xi}$ and $\hat{p}$ are minimized by a symmetric positive semi-definite weighting matrix H_N. Deviations of the measured and system outputs in the estimation horizon are minimized by a symmetric positive semi-definite weighting matrix H_k, which is

equal to the inverse of the measurement noise covariance matrix V_k [Ferreau et al., 2012]. As seen from the formulation, the objective function consists of two parts: the arrival and quadratic costs. The arrival cost stands for the early measurements $t = [t_{0,k-N+1}]$, and summarizes the deviations of the system outputs with respect to the measured outputs before the beginning of the estimation horizon. The quadratic cost stands for the recent measurements $t = [t_{t-N+1,k}]$, and corresponds to deviations of the system outputs with respect to the measured outputs within the estimation window. The estimation horizon is denoted by N, and lower and upper constraints on the model states and parameters are respectively denoted by ξ_{min}, ξ_{max}, p_{min} and p_{max}.

3.1.1 Arrival Cost

The arrival cost in (10) is necessary for the stability of the nonlinear RHE. The estimation horizon for the arrival cost is equal to t_{k-N} and approaches to infinity as time goes to infinity. Therefore, a real-time solution for the RHE formulation will be infeasible after a finite time. In order to forestall this problem, an iterative approach, a smoothed EKF update, is proposed for the approximation of the arrival cost. It is run with measurements at time t_{k-N} and the inverted Kalman covariance matrix H_N in every iteration to update Kalman estimates $\hat{\xi}, \hat{p}$. The main advantage of using the inverted Kalman covariance matrix is to have an adaptive weighting matrix. In case of using a constant weighting matrix, if the deviations in the arrival cost are small and the weighting matrix is large, then the chattering effect could be observed on the estimates. If the deviations in the arrival cost are large and the weighting matrix is small, then the estimates could not reach the true values in a reasonable time.

The reference estimated values for the states and parameters $\hat{\xi}$ and $\hat{p}$ are acquired from the previous solution of RHE. Bounded arrival cost is a requirement because if the inverted Kalman covariance is too heavy, the arrival cost may approach infinity. Therefore, the contributions of the past measurements to the inverted Kalman covariance H_N are down-weighted by a process noise covariance matrix W in (9). It is to be noted that H_N is upper bounded by the inverse of W since $W^{-1} - H_N$ is positive semi-definite. The calculation of H_N can be found in [Robertson, 1996].

3.2 Nonlinear Receding Horizon Control

RHC approach has the ability to anticipate the system behavior on a finite horizon by minimizing a cost function consisting of the references, states, and inputs and deal with hard constraints on states and inputs. A nonlinear system model is described in (3). At every time step, the states and inputs have to fulfill the following:

$$\xi \in \nless, \quad u \in \mathbb{U} \tag{11}$$

where $\mathbb{U}$ is compact subset of $\mathbb{R}^{n_u}$, i.e., $\mathbb{U} \subseteq \mathbb{R}^{n_u}$, $\nless$ is closed subset of $\mathbb{R}^{n_\xi}$, i.e., $\nless \subseteq \mathbb{R}^{n_\xi}$, and each set contains the origin in its interior. The constraints on each input are uncoupled in the sense that the feasible regions of the inputs do not affect each other.

The following formulation is solved at each sampling time t:

$$
\begin{aligned}
\min_{\xi(t),u(t)} \quad & \frac{1}{2}\left\{\left\{\sum_{i=k+1}^{k+N-1} \|\xi_r(t_i) - \xi(t_i)\|_{Q_k}^2 + \|u_r(t_i) - u(t_i)\|_R^2\right\} \right. \\
& \left. + \|\xi_r(t_{k+N}) - \xi(t_{k+N})\|_{Q_N}^2\right\} \\
\text{s. t.} \quad & \xi(t_k) = \hat{\xi}(t_k) \\
& \dot{\xi}(t) = f\big(\xi(t), u(t), p\big) \\
& \xi_{min} \leq \xi(t) \leq \xi_{max} \quad t \in [t_{k+1}, t_{k+N}] \\
& u_{min} \leq u(t) \leq u_{max} \quad t \in [t_{k+1}, t_{k+N-1}]
\end{aligned}
\tag{12}
$$

where $Q_k \in \mathbb{R}^{n_\xi \times n_\xi}$, $R \in \mathbb{R}^{n_u \times n_u}$ and $Q_N \in \mathbb{R}^{n_\xi \times n_\xi}$ are symmetric and positive semi-definite weighting matrices, ξ_r and u_r are the state and input references, ξ and u are the states and inputs, t_k is the current time, N is the prediction

horizon, $\hat{\xi}(t_k)$ is the estimated state vector by the RHE, ξ_{min}, ξ_{max}, u_{min} and u_{max} denote respectively the upper and lower constraints on the state and input, $i = k+1, \ldots, k+N-1$, $u(t) = [u(t_{k+1}), \ldots, u(t_{k+N-1})]$ is the input sequence over prediction horizon N, and $\xi(t)$ is the state trajectory obtained by applying the control sequence $u(t)$ to the system. The first element of the input sequence $u(t)$ is applied to the systems:

$$u(t_{k+1}, \xi_{k+1}) = u^*(t_{k+1}) \tag{13}$$

and the aforementioned nonlinear RHC formulation is solved again over a shifted prediction horizon for the subsequent time instant. It is important to point out that the control input $u^*(t_{k+1})$ is precisely the same as it would be if all immeasurable states and parameters acquire values equal to their estimates based on the estimation up to current time t_k due to the certainty equivalence principle.

The first cost is the stage cost, which is the cost throughout the prediction horizon, and the second cost is the terminal penalty, which is the cost at the end of the prediction horizon. The terminal penalty is stated for stability reasons [Mayne et al., 2000, Rawlings and Mayne, 2009].

The formulation in (12) is a nonlinear and nonconvex optimization problem, and the computational complexity depends on the order of the system, the nonlinearity of the system, the horizon length and the nonlinear optimization solver.

4   Implementations

4.1   Implementation of Nonlinear RHE

The inputs of the nonlinear RHE are the variables in the output vector (8), which includes the position, yaw rate and speed of the field robot. The outputs of the nonlinear RHE, the position, yaw angle, speed and traction parameters, are the full state and parameter vectors (5)-(6). The nonlinear RHC requires full state and parameter as an input; therefore, the estimated values by the nonlinear RHE are fed to the nonlinear RHC to generate the desired yaw rate applied to the field robot.

The nonlinear RHE formulation is solved at every sampling instant with the following constraints on the traction parameters:

$$
\begin{aligned}
0 \leq \quad \mu \quad \leq 1 \\
0 \leq \quad \kappa \quad \leq 1
\end{aligned}
\tag{14}
$$

The weighting matrix H_k represents noise characteristics of sensors and is determined using standard deviation of measurements. The measurements have been perturbed by Gaussian noise with standard deviation of $\sigma_x = \sigma_y = 0.03$ m, $\sigma_\omega = 0.0175$ rad/s, $\sigma_v = 0.05$ m/s based on experimental analysis. Additionally, the weighting matrix W represents process noise, which is chosen based on the objective. Low gain in the process noise results in better estimation accuracy; however, it causes time-lag between true and estimated values. Therefore, in this paper, the weighting coefficients for the measured states and parameters (e.g., x an y positions, speed) are selected large while the weighting coefficients for the immeasurable states and parameters (e.g., yaw angle and traction parameters) are selected small. The following weighting matrices H_k and W are used in nonlinear RHE:

$$
\begin{aligned}
H_k = V_k^{-1} \quad &= \quad diag(\sigma_x^2, \sigma_y^2, \sigma_v^2, \sigma_\omega^2)^{-1} \\
&= \quad diag(0.03^2, 0.03^2, 0.05^2, 0.0175^2)^{-1} \\
W \quad &= \quad diag(x^2, y^2, \theta^2, v^2, \mu^2, \kappa^2) \\
&= \quad diag(10.0^2, 10.0^2, 0.1^2, 1.0^2, 0.25^2, 0.25^2)
\end{aligned}
\tag{15}
$$

4.2 Implementation of Nonlinear RHC

The nonlinear RHC formulation is solved at every sampling instant with the following constraint on the input:

$$-5.73°\,\mathrm{s}^{-1} \leq \omega(t) \leq 5.73°\,\mathrm{s}^{-1} \tag{16}$$

The state and input references for the field robot are changed online and defined as follows:

$$\xi_r = [x_r, y_r, \theta_r]^T \quad \text{and} \quad u_r = \omega_r \tag{17}$$

where x_r and y_r are the position references, ω_r is the yaw rate reference, and the yaw angle reference is calculated from the position references as follows:

$$\theta_r = \mathrm{atan2}\,(\dot{y}_r, \dot{x}_r) + \lambda\,\pi \tag{18}$$

where λ describes the desired direction of the field robot ($\lambda = 0$ for forward and $\lambda = 1$ for backward). If the yaw rate reference calculated from the reference trajectory is used as the input reference, steady state error might occur in case of a mismatch between the system model and real system. Therefore, the measured yaw rate is used as the input reference to penalize the input rate in the objective function.

Three different weighting matrices Q_k are selected to evaluate the yaw angle effect on the trajectory tracking in Section 5.1.

$$
\begin{aligned}
Q_k^1 &= diag(1,1,10) & (19) \\
Q_k^2 &= diag(1,1,0) & (20) \\
Q_k^3 &= diag(1,1,1) & (21)
\end{aligned}
$$

and the third one Q_k^3 is used in Section 5.2. The weighting matrices R and Q_N for all experimental studies are selected as follows:

$$R = 10 \quad \text{and} \quad Q_N = 10 \times Q_k \tag{22}$$

The weighting matrix for the input R is set larger than the one for the states Q_k to ensure well damped closed-loop system behavior. The weighting matrix for the terminal penalty Q_N is set to 10 times larger than the one for the states Q_k. Therefore, the last deviations between the predicted states and their references in the prediction horizon are minimized in the objective function 10 times more than the previous points in the prediction horizon. The reason is that the error at the end of the prediction horizon plays a very critical role for the stability issue of the control algorithm.

4.3 Extended Kalman Filter

An extended Kalman filter (EKF) is designed based on the traditional kinematic model in (1). The discrete-time model used by the EKF is written with a sampling interval T_s as follows:

$$
\begin{aligned}
x_{k+1} &= x_k + T_s v_k \cos\theta_k \\
y_{k+1} &= y_k + T_s v_k \sin\theta_k \\
\theta_{k+1} &= \theta_k + T_s \omega_k
\end{aligned}
\tag{23}
$$

The general form of the estimated system model is:

$$
\begin{aligned}
\widehat{\xi}_{k+1} &= f(\widehat{\xi}_k, u_k) + w_k \\
\widehat{z}_{k+1} &= h(\widehat{\xi}_k, u_k) + v_k
\end{aligned}
\tag{24}
$$

The differences between the kinematic model and the estimation model are the process noise w_k and the observation noise v_k both in the state and the measurement equations. They are both assumed to be independent with zero mean multivariate Gaussian noises with covariance matrices W_k and V_k, respectively:

$$w_k \frown N(0, W_k)$$
$$v_k \frown N(0, V_k) \tag{25}$$

The covariance matrix V_k is formulated in (15), and the covariance matrix W_k is formulated as follows:

$$\begin{aligned} W_k &= diag(x^2, y^2, \theta^2, v^2) \\ &= diag(10.0^2, 10.0^2, 0.1^2, 1.0^2) \end{aligned} \tag{26}$$

The convariance matrix W_k for the EKF is different than the covariance matrix for the nonlinear RHE (15) because the traction parameters are not estimated by the EKF. The EKF is designed based on the traditional kinematic model (1) because it cannot deal with constraints on parameters. If the traction parameters are estimated, their estimates might be lower than zero [Haseltine and Rawlings, 2005], which results in instability of the control algorithm.

4.4 Solution Methods

RHEC methods for systems require online solutions of nonlinear least square optimization problems at each sampling time. In this study, single solution method, consisting of a fusion between multiple shooting and generalized Gauss-Newton methods, has been used to solve both nonlinear RHEC optimization problems. This approach is valid because the formulation of nonlinear RHC problem is akin to that of nonlinear RHE problem. The generalized Gauss-Newton method, derived from the classical Newton method, was developed for least-squared problems. This method is advantageous because it does not require difficult computations of the second derivatives; however, it is challenging to foreknow the number of iterations to reach a solution of the desired accuracy. To overcome this challenge, the solution proposed in [Diehl et al., 2002, KRA,] has been used where the number of Gauss-Newton iterations is restricted to 1, and the initial value of each optimization problem takes on the value of the previous one intelligently. Hence this improves the convergence of the Gauss-Newton method.
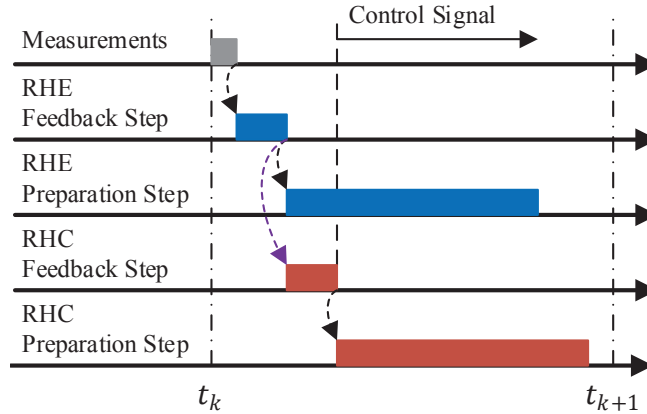

Figure 5: Illustration of the solution method

Unlike [Vukov et al., 2015], the Gauss-Newton iteration is divided into two parts: preparation and feedback parts. As illustrated in Fig. 5, the preparation part is executed prior to the feedback part, and the feedback part is executed after measurements for RHE and estimates for RHC are available. In the preparation part, the system dynamics are integrated with the previous solution, and objectives, constraints, and corresponding sensitives are evaluated. In the feedback part, a single quadratic programming is solved with the current measurements for the RHE and the current estimates for the RHC. Thus, the new estimates for the RHE and the new control signal for the RHC are obtained. Compared to the classical method, this method minimizes feedback delay and produces similar results with higher computational efficiency [Diehl et al., 2002].

To solve the constrained nonlinear optimization problems in the nonlinear RHEC, the *ACADO* code generation tool was used. *ACADO* is an open source software package for optimization problems that generates C-code [Houska et al., 2011]. The obtained dense quadratic problem sub-problems are solved by the online quadratic problem solver [Ferreau et al., 2008].

5 Experimental Results

The aim of the real-time experiments is to track a predefined trajectory with a field robot exposed to varying soil conditions characterized by a wet and bumpy field. Highly accurate guidance is required to avoid crop, e.g., sorghum, damage during planting, weeding, and growth; however, slippage makes this guidance problem difficult. In this study, a space-based trajectory approach is employed rather than a time-based trajectory approach. The reason is that the speed of the field robot is featured with large uncertainty arising from the variation of terrain, load, turning resistance and surrounding obstacles. These uncertainties can impose large time deviations, which challenge the tracking of the reference time-based trajectory. However, it is to be pointed out that the field robot is not forced to be at a particular point on the trajectory at a certain sampling instant in space-based trajectory approach.

The reference generation method in this paper is as follows. If the field robot starts off-track, the closest point on the target trajectory is computed first, and then the desired points in the horizon are determined. A trajectory consisting of straight and curved lines is tracked so that the performance of the designed framework for both scenarios is investigated. The prediction and estimation horizons for the nonlinear RHEC are set to 3 seconds.

5.1 Scenario A: Effect of Yaw Angle Weight on RHC

In this subsection, the coefficient of the yaw angle in the weighting matrix for nonlinear RHC is evaluated. Trajectory tracking performance of nonlinear RHCs based on nonlinear RHE is shown in Fig. 6. It is evident that if the coefficient of the yaw angle is larger than the coefficients for the position as in (19), the field robot cannot reach the desired trajectory after a reasonable time. However, if the coefficient of the yaw angle is set to zero as in (20), in other words, the yaw angle error between the reference yaw angle and actual yaw angle is not minimized, the field robot oscillates around the reference trajectory. To overcome this problem, the beginning of the desired points is defined as a fixed forward distance from the closest point on the reference trajectory at every sampling instant, and the optimum distance was selected regarding the speed of vehicles [Kayacan et al., 2016a].

In this paper, the minimization of the yaw angle error in the nonlinear RHC design is implemented to solve the aforementioned limitations. The coefficient of the yaw angle is set to equal to the coefficients for the position as in (21). It can be seen from Fig. 6 that the field robot reaches the reference trajectory after a reasonable time and stay on-track, which highlights the importance of correctly weighting of the yaw angle in the cost function in the space-based trajectory approach.

5.2 Scenario B: Comparison of RHCs based on RHE and EKF

Motivated by results from the previous section, the yaw angle error is minimized in the nonlinear RHC formulations based on the nonlinear RHE and the extended Kalman filter (EKF). As can be seen from Fig. 7, the field robot reaches the reference trajectory after it is started off-track and stays-on track for nonlinear RHCs based on both the nonlinear RHE and the EKF.

The Euclidean errors calculated using raw GNSS data to the space-based reference trajectory for nonlinear RHCs based on both the nonlinear RHE and EKF are shown in Fig. 8. The mean values of the Euclidian error for the nonlinear RHCs based on the nonlinear RHE and the EKF are 0.0423 m and 0.0514 m, respectively. The nonlinear RHEC framework benefits from traction parameter estimates and results in less error compared to the RHC based on the EKF. As previously mentioned in Section 1, the available space on either side of the robot is limited to only 0.12
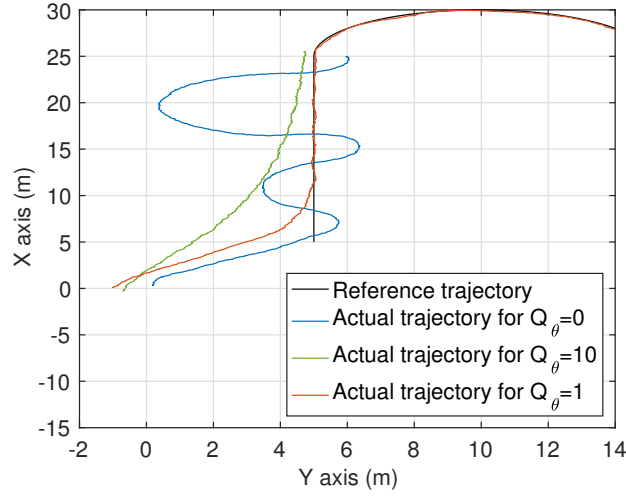
Figure 6: Reference and actual trajectories for different weighting of the yaw angle. These results clearly demonstrate that the penalization of yaw angle error is extremely important in RHC formulation. The presented approach in the paper should be contrasted with the typical approach of choosing small coefficients for R matrix, which essentially leads to low-gain control. By minimizing yaw angle error in the cost function in RHC formulation explicitly, large coefficients for R matrix can be employed, which leads to better tracking performance.
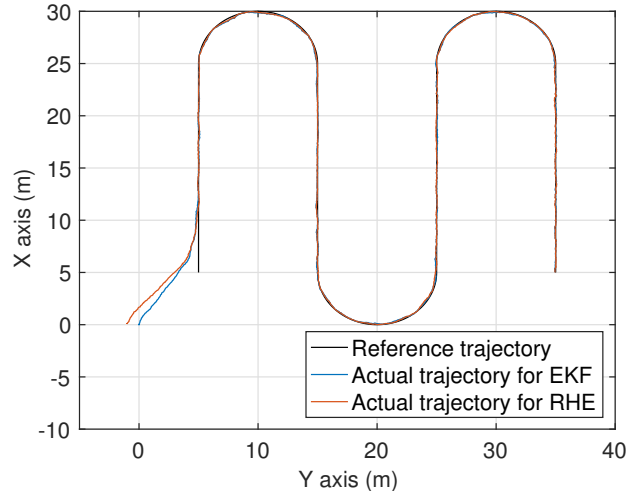
Figure 7: Reference and actual trajectories. The Euclidian errors are plotted in Fig. 8.

m; therefore, the error must be smaller than this limit to avoid crop damage and keep the robot-centered in the row. The number of violations is shown in Fig. 9. The results of multiple experiments indicate that the RHEC framework does not violate this error constraint, while the RHC based on the EKF violates it 17 times during the path tracking of straight lines. This demonstrates the capability of the RHEC framework.

The nonlinear RHE performance for the yaw angle and traction coefficients is shown in Fig. 10. It is difficult to measure the yaw angle in practice, but it plays a vital role in the trajectory tracking problem. If it is not estimated, then the system model is deteriorated so that model-based controllers cannot be employed. As can be seen, the estimated values for the traction parameters are within the constraints specified in (14) and consistently stay close to the upper bound. It is observed that model parameter estimates stabilize at certain values which secure stable trajectory-tracking. Therefore, these traction parameters' estimates play a significant role in the nonlinear RHC performance,
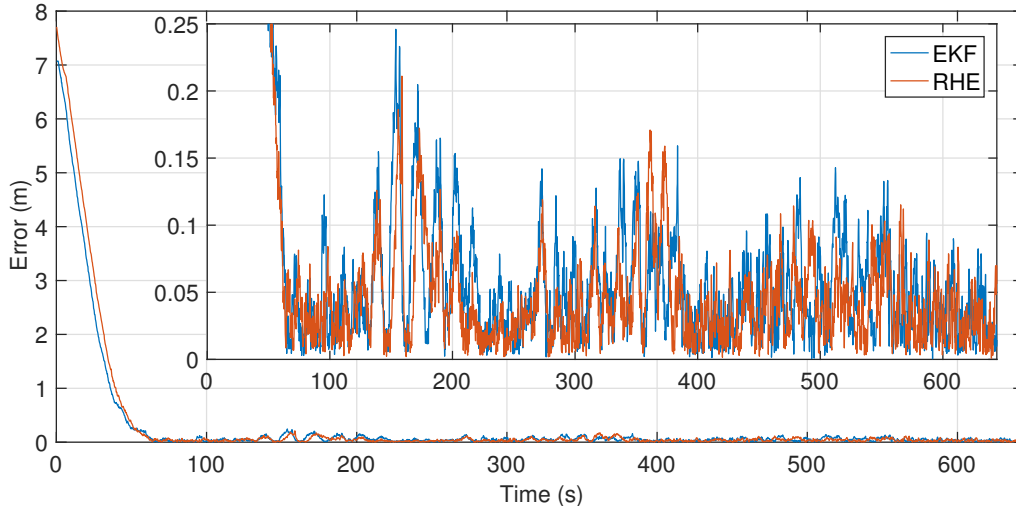
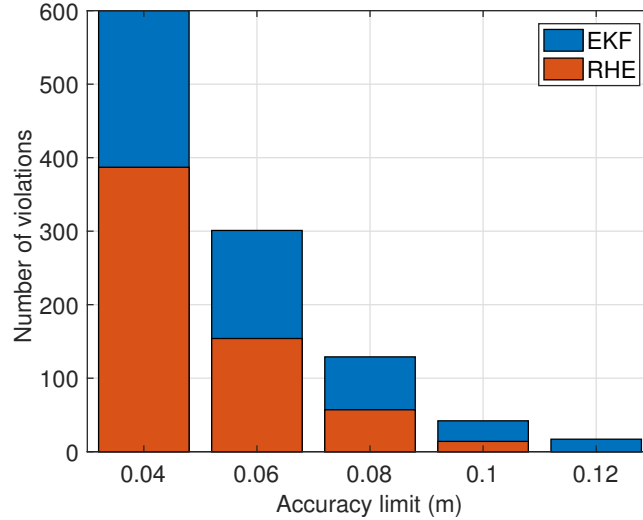Figure 8: Euclidian error calculated using raw GNSS data

Figure 9: Number of violations for straight lines tracking. The available space on either side of the field robot is restricted to 0.12 m. The field robot controlled by the RHEC framework does not violate this limit; therefore, it does not cause crop damage.

and consequently the trajectory tracking performance. It is to be noted that hard-coding can result in more accurate trajectory tracking performance. However, the soil conditions must be in a steady-state, and the identification of parameters is required, which is a time-consuming task [Kayacan et al., 2014b, Kayacan et al., 2013]. For this reason, the nonlinear RHE is beneficial because it provides the estimates of the traction parameters, and the framework ensures an accurate trajectory tracking.

The measured and estimated speeds are shown in Fig. 11. The speed of the field robot is constant because a space-based trajectory approach is employed in the paper, and the nonlinear RHE can cope with noise on the speed measurements.

The output of the nonlinear RHC, the reference yaw rate, is shown in Fig. 12. The output of the nonlinear RHC reaches the bounds, but never violates them. This demonstrates the capability of dealing with constraints on inputs.
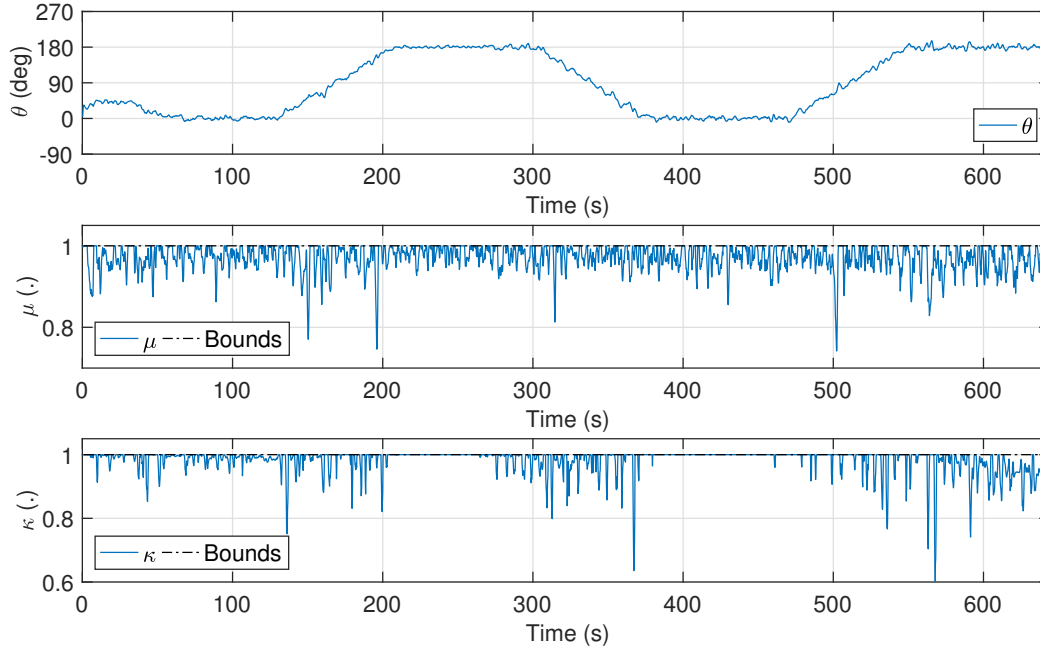
Figure 10: Nonlinear RHE output: The yaw angle and traction parameters estimates
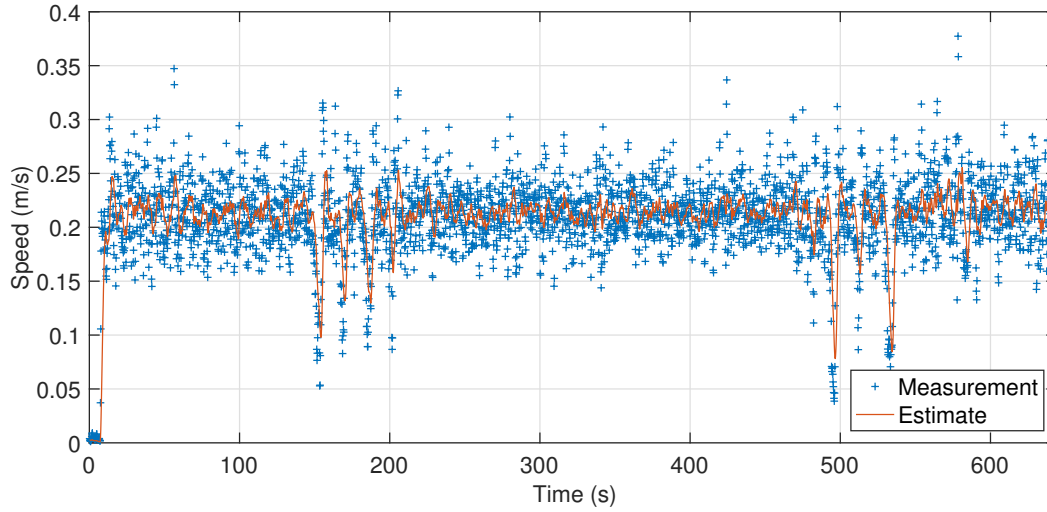

Figure 11: Actual and estimated speeds

Moreover, the performance of the low-level controller for the speed of the DC motors is sufficient.

Real-time algorithms for RHE and RHC comprise respectively three states, two parameters, one control input and 15 prediction intervals, and three states, one control input, and 15 prediction intervals. The computations times for the nonlinear RHEC are summarized in Table 1. As seen from the table, the average computation times for the nonlinear RHE and RHC are equal to 0.48 ms and 0.40 ms, respectively. Thus, the overall computation time for the nonlinear RHEC framework is equal to 0.88 ms with a reasonable maximum computation time of 2.8534 ms for real-time applications.
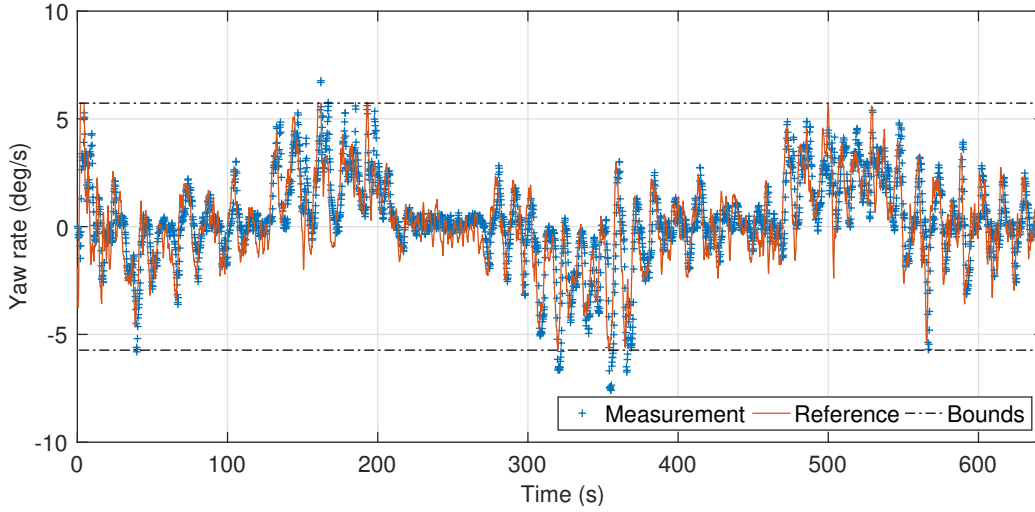
Figure 12: Reference and actual yaw rates

Table 1: Computation times of the nonlinear RHEC.

	Minimum (ms)	Average (ms)	Maximum (ms)
Nonlinear RHE			
Preparation	0.2921	0.3523	1.0075
Feedback	0.0766	0.1309	0.4925
Overall	0.3687	0.4832	1.5000
Nonlinear RHC			
Preparation	0.2725	0.3751	1.2077
Feedback	0.0149	0.0215	0.1457
Overall	0.2874	0.3966	1.3534

A single quadratic programming iteration at each time instant may result in a sub-optimal solution; therefore, it is necessary to observe the Karush-Kuhn-Tucker (KTT) tolerance, which measures the performance of RHEC methods in terms of optimality. The KKT tolerances for the RHEC are shown in Fig. 13. It is to be noted that the KKT tolerance is equal to zero for linear systems due to the fact that a quadratic programming at each iteration is solved precisely. Therefore, non-zero values demonstrate the non-linearity of the RHEC problems. The largest KKT tolerances for the RHC arise at the beginning of the experiments because the field robot initially started off-track in the sense that error values are large. The reason is that since a single iteration is completed, the obtained solution is only a rough approximation to the optimal solution. It can be seen that the KKT tolerance decreases with time until the field robot stays on-track. The trajectory references continuously change throughout the experiment; therefore, the KKT tolerance cannot further decrease. Furthermore, although the KKT tolerance for the RHE is large at the beginning due to poor initialization, it decreases after several time periods.

6 Conclusion

A computationally efficient, nonlinear RHEC framework based on an adaptive nonlinear model has been investigated for the guidance of the field robot. The experimental results in the field trials have shown that the nonlinear RHE can perform precise online estimation of the immeasurable states and parameters. Also, the nonlinear RHC has the capability of accurately controlling the field, phenotyping robot with reasonable accuracy. The importance of correctly weighting of the yaw angle in the cost function is highlighted for the space-based trajectory approach. The mean value
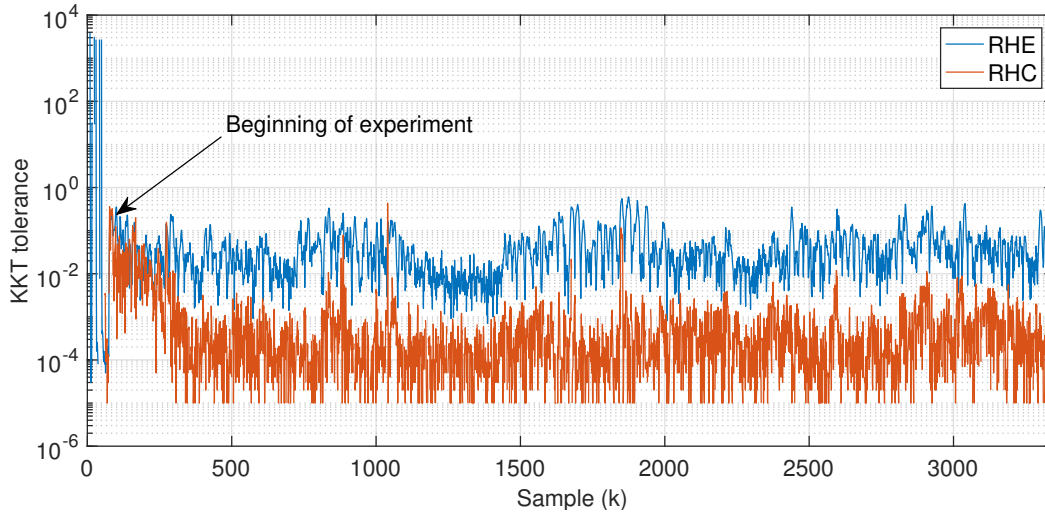
Figure 13: KKT tolerances

of the Euclidean error to the reference trajectory is 0.0423 m. Additionally, the RHEC framework does not violate the upper bound of the required tracking accuracy, i.e., 0.12 m, while the RHC based on the EKF violates it 17 times for the tracking of straight lines. Although RHEC framework is computationally intensive, the proposed solution method is computationally efficient, with a mean total required computation time of 0.88 ms and the worst-case computation time of 2.85 ms.

The high accuracy of this nonlinear RHEC framework is of importance to plant breeders and farmers because there is a growing need for rapid, mobile, field-based phenotyping technologies that can navigate safely and autonomously through row based crops. By developing a framework that can handle unknown and immeasurable slip parameters, field robots will be able to navigate a variety of agricultural field conditions without damaging the crop.

Recent developments in microprocessors technology and fast solution tools for RHEC framework have changed the well-known paradigm in a way that the belief of using RHEC for only relatively slow dynamic systems is no longer true. The comparative results presented in this paper also show that RHEC implementations for agile robotic systems are eligible to obtain highly accurate tracking performance.

Acknowledgments

Dr. Erkan Kayacan is a postdoctoral researcher at Massachusetts Institute of Technology since February 2018.

References

Backman, J., Oksanen, T., and Visala, A. (2012). Navigation system for agricultural machines: Nonlinear model predictive path tracking. *Computers and Electronics in Agriculture*, 82:32 – 43.

Bekker, M. (1956). *Theory of Land Locomotion*. University of Michigan Press, Ann Arbor, MI.

Buckeridge, M. S., Carpita, N. C., McCann, M. C., et al. (2014). *Plants and bioenergy*. Springer,.

Diehl, M., Bock, H., Schlder, J. P., Findeisen, R., Nagy, Z., and Allgöwer, F. (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4):577 – 585.

Falcone, P., Borrelli, F., Asgari, J., Tseng, H. E., and Hrovat, D. (2007). Predictive active steering control for autonomous vehicle systems. *Control Systems Technology, IEEE Transactions on*, 15(3):566–580.

Ferreau, H., Kraus, T., Vukov, M., Saeys, W., and Diehl, M. (2012). High-speed moving horizon estimation based on automatic code generation. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 687–692.

Ferreau, H. J., Bock, H. G., and Diehl, M. (2008). An online active set strategy to overcome the limitations of explicit mpc. *International Journal of Robust and Nonlinear Control*, 18(8):816–830.

Fiorani, F. and Tuberosa, R. (2013). Future scenarios for plant phenotyping. *Annu Rev Plant Biol*, 64:267–291.

Fukao, T., Nakagawa, H., and Adachi, N. (2000). Adaptive tracking control of a nonholonomic mobile robot. *IEEE Transactions on Robotics and Automation*, 16(5):609–615.

Gu, D. and Hu, H. (2006). Receding horizon tracking control of wheeled mobile robots. *Control Systems Technology, IEEE Transactions on*, 14(4):743–749.

Haseltine, E. L. and Rawlings, J. B. (2005). Critical evaluation of extended kalman filtering and moving-horizon estimation. *Industrial & Engineering Chemistry Research*, 44(8):2451–2460.

Houska, B., Ferreau, H. J., and Diehl, M. (2011). Acado toolkitan open-source framework for automatic control and dynamic optimization. *Optimal Control Applications and Methods*, 32(3):298 – 312.

Hughes, B. (1995). Random walks and random environments. *Oxford*, 1.

Iagnemma, K., Kang, S., Shibly, H., and Dubowsky, S. (2004). Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers. *IEEE Transactions on Robotics*, 20(5):921–927.

Kanayama, Y., Kimura, Y., Miyazaki, F., and Noguchi, T. (1990). A stable tracking control method for an autonomous mobile robot. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 384–389.

Kayacan, E. (2017). Multiobjective h_∞ control for string stability of cooperative adaptive cruise control systems. *IEEE Transactions on Intelligent Vehicles*, 2(1):52–61.

Kayacan, E., Bayraktaroglu, Z. Y., and Saeys, W. (2012a). Modeling and control of a spherical rolling robot: a decoupled dynamics approach. *Robotica*, 30(4):671680.

Kayacan, E., Kayacan, E., Chen, I.-M., Ramon, H., and Saeys, W. (2018). *On the Comparison of Model-Based and Model-Free Controllers in Guidance, Navigation and Control of Agricultural Vehicles*, pages 49–73. Springer International Publishing, Cham.

Kayacan, E., Kayacan, E., Ramon, H., Kaynak, O., and Saeys, W. (2015a). Towards agrobots: Trajectory control of an autonomous tractor using type-2 fuzzy logic controllers. *IEEE/ASME Transactions on Mechatronics*, 20(1):287–298.

Kayacan, E., Kayacan, E., Ramon, H., and Saeys, W. (2012b). Velocity control of a spherical rolling robot using a grey-pid type fuzzy controller with an adaptive step size. *IFAC Proceedings Volumes*, 45(22):863 – 868. 10th IFAC Symposium on Robot Control.

Kayacan, E., Kayacan, E., Ramon, H., and Saeys, W. (2013). Modeling and identification of the yaw dynamics of an autonomous tractor. In *2013 9th Asian Control Conference (ASCC)*, pages 1–6.

Kayacan, E., Kayacan, E., Ramon, H., and Saeys, W. (2014a). Distributed nonlinear model predictive control of an autonomous tractor-trailer system. *Mechatronics*, 24(8):926 – 933.

Kayacan, E., Kayacan, E., Ramon, H., and Saeys, W. (2014b). Nonlinear modeling and identification of an autonomous tractor-trailer system. *Computers and Electronics in Agriculture*, 106(Supplement C):1 – 10.

Kayacan, E., Kayacan, E., Ramon, H., and Saeys, W. (2015b). Robust tube-based decentralized nonlinear model predictive control of an autonomous tractor-trailer system. *IEEE/ASME Transactions on Mechatronics*, 20(1):447–456.

Kayacan, E., Kayacan, E., Ramon, H., and Saeys, W. (2015c). Towards agrobots: Identification of the yaw dynamics and trajectory tracking of an autonomous tractor. *Computers and Electronics in Agriculture*, 115:78 – 87.

Kayacan, E. and Peschel, J. (2016). Robust model predictive control of systems by modeling mismatched uncertainty. *IFAC-PapersOnLine*, 49(18):265 – 269. 10th IFAC Symposium on Nonlinear Control Systems NOLCOS 2016.

Kayacan, E., Peschel, J. M., and Kayacan, E. (2016a). Centralized, decentralized and distributed nonlinear model predictive control of a tractor-trailer system: A comparative study. In *2016 American Control Conference (ACC)*, pages 4403–4408.

Kayacan, E., Ramon, H., and Saeys, W. (2016b). Robust trajectory tracking error model-based predictive control for unmanned ground vehicles. *IEEE/ASME Transactions on Mechatronics*, 21(2):806–814.

Klancar, G. and Skrjanc, I. (2007). Tracking-error model-based predictive control for mobile robots in real time. *Robotics and Autonomous Systems*, 55(6):460 – 469.

Lee, S. U., Gonzalez, R., and Iagnemma, K. (2016). Robust sampling-based motion planning for autonomous tracked vehicles in deformable high slip terrain. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2569–2574.

Lee, S. U. and Iagnemma, K. (2016). Robust motion planning methodology for autonomous tracked vehicles in rough environment using online slip estimation. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3589–3594.

Lee, T.-C., Song, K.-T., Lee, C.-H., and Teng, C.-C. (2001). Tracking control of unicycle-modeled mobile robots using a saturation feedback controller. *Control Systems Technology, IEEE Transactions on*, 9(2):305–318.

Lins Barreto, J. C., Scolari Conceicao, A. G., Dorea, C. E. T., Martinez, L., and De Pieri, E. R. (2014). Design and implementation of model-predictive control with friction compensation on an omnidirectional mobile robot. *IEEE/ASME Transactions on Mechatronics*, 19(2):467–476.

Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814.

Naveau, M., Kudruss, M., Stasse, O., Kirches, C., Mombaur, K., and Soures, P. (2017). A reactive walking pattern generator based on nonlinear model predictive control. *IEEE Robotics and Automation Letters*, 2(1):10–17.

Oh, H., Kim, S., and Tsourdos, A. (2015). Road-map assisted standoff tracking of moving ground vehicle using nonlinear model predictive control. *IEEE Transactions on Aerospace and Electronic Systems*, 51(2):975–986.

Rawlings, J. B. and Mayne, D. Q. (2009). *Model predictive control: Theory and design*. Nob Hill Pub.

Ray, L. E. (2009). Estimation of terrain forces and parameters for rigid-wheeled vehicles. *IEEE Transactions on Robotics*, 25(3):717–726.

Robertson, D. (1996). *Development and Statistical Interpretation of Tools for Nonlinear Estimation*. Auburn University.

Robertson, D. G., Lee, J. H., and Rawlings, J. B. (1996). A moving horizon-based approach for least-squares estimation. *AIChE Journal*, 42(8):2209–2224.

Tverberg, G. (2012). Oil supply limits and the continuing financial crisis. *Energy*, 37(1):27–34.

Utstumo, T., Berge, T. W., and Gravdahl, J. T. (2015). Non-linear model predictive control for constrained robot navigation in row crops. In *2015 IEEE International Conference on Industrial Technology (ICIT)*, pages 357–362.

Vega-Sanchez, M. and Ronald, P. (2010). Genetic and biotechnological approaches for biofuel crop improvement. *Curr Opin Plant Biol*, 21(2):218–224.

Vukov, M., Gros, S., Horn, G., Frison, G., Geebelen, K., Jørgensen, J., Swevers, J., and Diehl, M. (2015). Real-time nonlinear mpc and mhe for a large-scale mechatronic application. *Control Engineering Practice*, 45:64 – 78.

Wang, Y., Fan, C., Hu, H., Li, Y., Sun, D., Wang, Y., and Peng, L. (2016). Genetic modification of plant cell walls to enhance biomass yield and biofuel production in bioenergy crops. *Biotechnology Advances*, 34(5):997 – 1017.

Yano, M. and Tuberosa, R. (2009). Genome studies and molecular genetics-from sequence to crops: genomics comes of age. *Curr Opin Plant Biol*, 12(2):103–106.