EasyChair Preprint

№ 1458

Experimental validation of structured receding horizon estimation and control for mobile ground robot slip compensation

Nathan Wallace, He Kong, Andrew Hill and Salah Sukkarieh

September 4, 2019

Experimental validation of structured receding horizon estimation and control for mobile ground robot slip compensation

Nathan D. Wallace, He Kong, Andrew J. Hill and Salah Sukkarieh

Abstract To achieve high accuracy tracking performance for wheeled mobile robots in spatially varying terrain conditions, it is necessary to estimate both the robot's state and the slip conditions of the environment to a high degree of precision. The receding horizon estimation and control (RHEC) framework presents a systematic, adaptive optimisation approach to this problem, to which our prior work proposed a structured blocking (SB) extension to address performance limitations for motion both at high speeds and over varying terrain. In this work, we validate these results in a series of preliminary field experiments with the Swagbot platform, demonstrating performance improvements in position tracking of up to 7%, and up to 13% for speed tracking at speeds of 1.5 and 2.5 m/s.

1 Introduction

Accurate navigation in difficult off-road conditions is one of the major challenges faced in development and deployment of autonomous agents for applications such as farming and environmental monitoring [1]. This is primarily due to the presence of arbitrary unknown disturbances, which are hard to model. In fact, estimation of dynamic systems with unknown inputs has attracted a lot of attention in the systems and control literature. We refer the reader to [2] for a brief overview of the existing literature and more recent progress on this subject. Especially in the context of field robot motion control, the presence of unknown and spatio-temporally varying traction conditions can degrade the performance of traditional control approaches such as PID control [3], which do not account for induced wheel slippage on vehicle handling. It is therefore necessary to estimate the traction parameters online, so that the control algorithm can compensate for wheel slippage and maintain good tracking

N. D. Wallace (✉) · H. Kong · A. J. Hill · S. Sukkarieh
Australian Centre for Field Robotics, The University of Sydney,
NSW 2050, Australia, e-mail: n.wallace@acfr.usyd.edu.au

Fig. 1 ACFR's Swagbot platform pictured during a field trial at a cattle station near Nevertire, NSW. Swagbot is a lightweight, electric robot designed to collect data on pasture and livestock, and perform tasks such as autonomous weeding. The platform's rocker chassis design enables it to clamber over uneven terrain and low-lying obstacles.

performance. This is important in situations where the robot is operating near crops or livestock, to ensure sufficient clearance is maintained.

Nonlinear receding horizon estimation and control (RHEC) is one strategy that has been successfully applied to this problem, demonstrated on a variety of platforms, and shown to perform well at speeds up to $\sim$2 m/s [4]–[7]. The focus of our prior work is on extending the RHEC framework to improve performance at higher speeds and over more rapidly varying terrains. Building upon insights in [2], [8]–[10]—and especially the technique in [11]—this is achieved by employing a structured blocking (SB) strategy, which enforces a blocking structure on the estimated slip parameter sequence. The performance of the proposed strategy was demonstrated in simulation to achieve improvements in tracking performance of up to $\sim$9% [12, 13] when compared to the previous RHEC approaches [4]–[7], which fix the slip parameters to a single value over the estimation horizon—henceforth termed full-horizon blocking (FHB).

With the aim of validating the above method experimentally, in this paper, we implement the nonlinear RHEC approach with structured blocking on the Australian Centre for Field Robotics' (ACFR) Swagbot electric ground vehicle—shown in Figure 1—for trajectory tracking in unknown and variable slip environments.

We compare the results of our method against the FHB strategy adopted in [4]–[7], and show that the RHEC with SB achieves improved path tracking performance compared to the FHB method, with a reduction of up to 7% in RMSE. Speed tracking at 2.5 m/s was significantly improved also, with a 13% reduction in RMSE, although a marginal reduction in speed tracking performance was observed for the 1.5 m/s test. Better tracking performance is important for time and energy efficient operation of the mobile platform, in conjunction with data-driven motion cost models [14]. Further tests are planned to better characterise and validate control performance in more challenging environments.

2 Related Work

There is an extensive body of literature on slip estimation for wheeled mobile robots (WMR), covering approaches such as: bounded uncertainty models, extended kinematic models (KM) which account for slip effects, machine learning models and dynamic models. We focus primarily on the extended KM approach here, as it is the most relevant to the presented work. A more thorough review of the other methods can be found in our previous works [12, 13].

Extended Kalman filters (EKF) with kinematic constraints have been used for slip estimation in WMRs, with slip modelled as a white-noise velocity perturbation [15] or as a multiplicative factor within the KM for a skid-steered WMR [16]. In [17] a tractor-trailer system with steering slip is modelled by generalized bicycle kinematics with an additional multiplicative factor, using an EKF to estimate state and slip parameters, and a nonlinear RHC for path tracking. Lateral tracking errors of $<$15 cm on straight and gently curved paths are achieved at 3.3 m/s.

A similar method for modelling slip is adopted in [4] for a tractor, augmenting a bicycle KM with two multiplicative factors to capture longitudinal and side slip (rather, slip in the steering angle)—which are assumed to be constant over the estimation horizon. Due to the existance of state and parameter constraints, and a nonlinear measurement function, a more sophisticated receding horizon estimation (RHE) strategy was required. A receding horizon control (RHC) strategy was used to control the wheel velocity and steering rate, and this RHEC framework is tested experimentally on wet, uneven grass, with average tracking errors of 0.26 m on gentle curves, and 0.6 m in headland turns at $\sim$2 m/s.

Similar approaches have been applied to tracked and skid-steered WMRs in [5] and [6], using a unicycle KM with factors for longitudinal and lateral slip, achieving tracking errors of $<$12 cm at $\sim$0.2 m/s for the tracked, and $<$5 cm at $\sim$0.3 m/s for the skid-steered robot. A tractor-trailer system was modelled in [7] by a generalised tricycle KM with three factors for longitudinal and slide slip. Centralised RHE was used, and the distributed RHC framework is compared against a centralised strategy, achieving comparable tracking errors of $<$10 cm with faster solution times.

Recently, the above RHEC framework was compared against a linear RHC approach in [18], using input-state linearisation with an EKF estimator to compensate for system state nonlinearity. It is shown that RHEC produces improved trajectory tracking performance over the input-state linearisation approach, although at a slight increase in computation cost.

A tractor is modelled in [19] by an extended bicycle KM with additive side slip effects on both wheels—ignoring longitudinal slip. A nonlinear adaptive control law embedded with an RHC algorithm was designed to avoid tracking overshoot, achieving $<$ 15 cm tracking errors with velocity manually controlled, though subsequent observations in [20] revealed decreased performance as vehicle speed increases. A mixed backstepping kinematic and dynamic observer was developed to improve slip observation, improving tracking performance at speeds of up to 5 m/s.

3 Preliminaries

Notation: Throughout this paper, $[a_1,\ldots,a_n]$ denotes $\left[a_1^T,\ldots,a_n^T\right]^T$, where $a_1,\ldots,a_n$ are appropriately dimensioned scalars/vectors/matrices. The weighted Euclidean norm is denoted by $\|a\|_R^2 = a^T R a$. Subscripts i,j,k denote the value of that variable at time t_k, ie $a_k = a(t_k)$. $\mathscr{I}_i^j$ denotes the set of integers between i and j.

3.1 System Model

The vehicle model used in this work is an adaptation of the bicycle KM, which is known to approximate car-like vehicle behaviour quite accurately [21]–[23]. Slip is incorporated in the model via two multiplicative slip parameters; one for longitudinal slip, and another for steering slip.

This steering slip formulation was adopted in [17] to capture the effect of vehicle inertia on the steering angle, yielding a slipping angle proportional to the steering angle. This allows the slip parameter to remain somewhat constant regardless of the steering motion, whereas an additive slip parameter would change value constantly—especially in curves. This model assumes that there are no external forces acting in the sideways direction on the vehicle, which is typically the case, save for sharp turns at high speeds and steep inclines.

The equations describing the robot's motion are as follows:

$$\begin{bmatrix} \dot{x}_{pos} \\ \dot{y}_{pos} \\ \dot{\beta} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} \kappa u_1 \cos\beta \\ \kappa u_1 \sin\beta \\ \frac{\kappa u_1}{l}\tan(\mu\cdot\delta) \\ u_2 \end{bmatrix}, \tag{1}$$

where x_{pos} [m] and y_{pos} [m] are the position of the virtual 'rear wheel'—or centre of the rear axle in the case of a 4-wheeled vehicle—in global cartesian coordinates, β [rad] is the yaw angle or heading, and δ [rad] is the steering angle; shown in Figure 2. The control variables are the wheel speed with respect to the ground u_1 [m/s], and the steering rate u_2 [rad/s]. The parameter l [m] represents the wheel base.

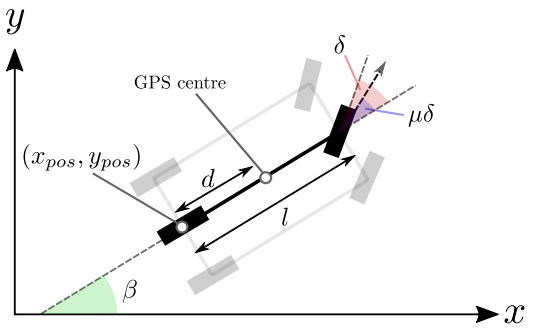

Fig. 2 Schematic illustration of adapted bicycle model with steering slip affecting the front wheel. The steering angle δ is shown in red, the effective steering angle $\mu\delta$ is shown in blue and the vehicle heading β is shown in green. The GPS centre is offset by a distance d from the rear-axle centre on the Swagbot platform.

The parameters κ, μ represent the longitudinal and steering slip respectively. κ relates commanded wheel speed to actual ground speed, capturing the influence of slip, and any error or uncertainty in the vehicle odometry. μ results from vehicle inertia's impact on steering, which is approximately relative to the steering angle. The parameter value range is conservatively chosen to be $[0,1]$—assuming no over-steering and no negative slip. Though circumstances do exist under which this assumption does not hold, for the environment and speeds tested, this assumption closely matched observations, and enforcing these bounds ensured the problem solution was both well-behaved, and quickly solvable. The choice of domain also allows the percentage slip experienced to be expressed as $1-\kappa$ and $1-\mu$.

It is our goal to eventually accommodate the effects of lateral slip in our model, either by introducing a side-slip term, or adopting a more sophisticated 4-wheel independent steer/drive model. Both approaches increase the difficulty of the optimisation problem, and in adding additional unobservable parameters, could risk compromising the estimates of each. That said, for the conditions and speeds featured in our experiments, the robot's momentum was insufficient for significant lateral slippage to occur, and thus the primary slip contributions were due to longitudinal and steering slip. For this reason, and for ease of comparison with the existing motion models used in related literature, we chose to ignore the lateral slip component.

4 Method

4.1 Receding Horizon Estimation and Control

An RHEC strategy is adopted in this work, which consists of solving two least squares optimisation problems—one for estimation, and one for predictive control—resulting in a closed-loop control strategy.

It should be noted that while we detail the standard RHC case in this section, a *packeted RHC* approach was implemented on the platform, which publishes the entire control horizon, rather than just the next action. This improves robustness to time overrun by ensuring suitable actions are available even when a control iteration is missed. The packeted approach is otherwise identical to standard RHC. In our testing, time overrun was only observed during the first 1–2s of operation; a consequence of the initialisation overhead of internal MATLAB functions.

The nonlinear RHEC problem is formulated as follows. Let $x_k \in \mathbb{R}^{n_x}$ represent the state of the system, and $y_k \in \mathbb{R}^{n_y}$ be an observation of this system state—here, the cartesian position and heading of the robot in world co-ordinates, and the steering angle. Let $u_k \in \mathbb{R}^{n_u}$ represent the control actions taken—the commanded forward speed and steering angle—and $p_k \in \mathbb{R}^{n_p}$ the system parameters to be estimated, which in our case are the longitudinal and steering slip experienced by the robot. We assume the system evolves in accordance to the given dynamic model $\dot{x}_k = f(x_k, u_k, p_k)$, and observations are taken in accordance with the measurement model function $y_k = h(x_k, u_k, p_k)$. Let variables (x_k, y_k, u_k, p_k) refer to the *real* pro-

cess. These each have associated *decision* and *optimal decision* variables in the optimisation, which we denote $(\chi_k, \eta_k, \nu_k, \rho_k)$ and $(\hat{x}_k, \hat{y}_k, \hat{u}_k, \hat{p}_k)$ respectively. The values of these decision variables are iteratively varied, subject to the constraints, by the optimiser, eventually yielding the optimal decision once the objective is minimised.

At each sampling time t_k, let the real system state be x_k. A sensor measurement y_k is taken, and the RHE uses the past N_e measurements $y_j,\ j \in \mathscr{I}_{k-N_e+1}^{k}$ to estimate (x_k, p_k). This estimate $(\hat{x}_k,\ \hat{p}_k)$ is used by the RHC module along with a time-based reference trajectory $\lambda_j = \left[\lambda_j^x, \lambda_j^u\right],\ \lambda_j^x \subseteq \mathbb{R}^{n_x}, \lambda_k^u \subseteq \mathbb{R}^{n_u}, j \in \mathscr{I}_{k+1}^{k+N_c}$. The RHC module computes the optimal control actions over a finite forward horizon of N_c timesteps, and the next action $\hat{u}_k$ is executed by the robot. This process repeats each sampling time (Δt seconds apart) until the end of the trajectory is reached.

4.2 Nonlinear Receding Horizon Estimation

RHE is a powerful optimisation-based estimation technique that provides a systematic framework for handling constraints and nonlinearities. In contrast to full information estimation (FIE)—which utilises the entire history of available information—RHE uses only measurements taken within a finite time-frame, and captures the information of prior measurements in an arrival cost term.

Given N_e past measurements $y_j,\ j \in \mathscr{I}_{k-N_e+1}^{k}$ and a measurement model function h, the constrained estimation problem to be solved at time t_k is:

$$\min_{\chi(\cdot),\nu(\cdot),\rho(\cdot)} \quad \left\{ \Gamma_{k-N_e+1} + \sum_{i=k-N_e+1}^{k} \left\| \eta_i - y_i \right\|_{R_k}^2 \right\} \tag{2}$$

$$s.t. \quad \left.\begin{aligned} \dot{\chi}_j &= f(\chi_j, \nu_j, \rho_j) \\ \eta_j &= h(\chi_j, \nu_j, \rho_j) \\ \chi_{min} &\leq \chi_j \leq \chi_{max} \\ \rho_{min,j} &\leq \rho_j \leq \rho_{max,j} \end{aligned}\right\} \ j \in \mathscr{I}_{k-N_e+1}^{k}$$

$$\Gamma_{k-N_e+1} = \left\| \begin{matrix} \chi_{k-N_e+1} - \hat{x}_{k-N_e+1} \\ \rho_{k-N_e+1} - \hat{p}_{k-N_e+1} \end{matrix} \right\|_{\Pi_{k-N_e+1}^{-1}}^2 \tag{3}$$

where Γ_{k-N_e+1} is the arrival cost function, $\hat{x}_{k-N_e+1} = \hat{x}_{k-N_e+1|k-N_e}$ is the optimal state prediction at t_{k-N_e+1}, R_k is the symmetric positive semi-definite weighting matrix equal to the inverse of the measurement noise covariance matrix [24], and Π^{-1} is the inverted covariance matrix of a smoothed EKF—as considered in [4, 25].

In the context of localisation and slip estimation, this RHE problem chooses the sequence of positions, control actions and slip values which minimise the least-squares distance between the extended KM and the measured robot state over the horizon. Consideration of the motion constraints allows the optimiser to estimate the extent of the slippage the robot's motion was subject to.

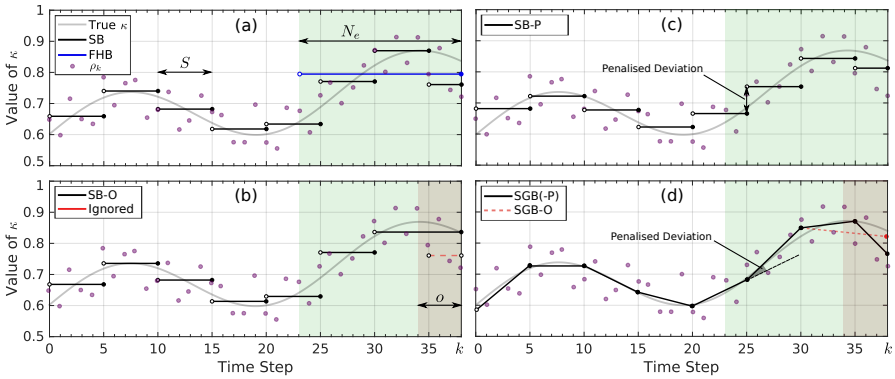

Fig. 3 Illustration of the SB extensions. The true κ value—not typically observable, shown here purely for illustrative purposes—is shown in light grey, and the best fit parameter estimate ρ_k based on noisy measurement data is shown in purple. (a) The FIE SB structure for block size $S = 5$; the RHE observes only the last N_e time steps. The equivalent FHB block is shown in blue. (b) SB with an overlap $o = 4$. The terminal block in the sequence (red) has size $3 < o$ and thus is suppressed. (c) SB with block value deviations penalised. (d) SGB variants.

4.3 Structured noise blocking in estimation

One of the main limitations inherent in constrained estimation methods such as RHE—and especially FIE—is that an optimisation problem must be solved at each sampling time. This burden can be reduced either via reformulation as a multi-parametric quadratic program [26], exploiting the optimisation structure [27], or via the structured blocking strategy introduced in our previous work [11, 12, 13].

The SB strategy has recently been adopted in FIE and RHE for estimating the process noise sequence [11], and more importantly, it was shown that the segment structure of the noise sequence from SB FIE must also be enforced over the finite estimation horizon of RHE (i.e., steps between $T - N$ and $T - 1$). The resulting SB RHE strategy becomes a dynamic estimator with periodically varying computational complexity. An illustrative example of this concept is shown in Figure 3 (a).

Our recent work applies this strategy to nonlinear RHEC. In [13], it was observed that while SB improved state estimation quality, the parameter estimates in the terminal block—used for the following RHC step—fluctuated in quality, due to the changing size of the terminal block in the receding horizon. A SB with overlapping blocks (SB-O) strategy was proposed to ameliorate this issue, enabling a trade-off between state and parameter estimate quality via an overlap extent parameter o. An example of this strategy is shown in Figure 3 (b).

A number of complementary strategies for SB RHEC were also introduced in [12], including the addition of a value-shift penalty between consecutive blocks (SB-P)—shown in Figure 3 (c)—and a strategy combining this with SB-O, termed SB-OP. Furthermore, gradient blocking methods were also proposed, which, in contrast to the aforementioned value blocking methods, instead perform blocking on the first-derivative of the parameter sequence. An example is shown in Figure 3 (d).

4.4 Nonlinear Receding Horizon Control

The RHC algorithm seeks to predict the system behaviour over a finite time horizon via minimisation of a cost function composed of states, inputs and references, and similarly to RHE, the RHC framework also supports handling of state and input constraints.

For RHC, at the current time t_k we wish to predict the control sequence for the next N_c time steps; thus the constrained optimisation problem to be solved is:

$$\min_{\nu(\cdot)} \quad \left\{ \sum_{i=k+1}^{k+N_c-1} \left\| \begin{matrix} \lambda_i^x - \chi_i^\lambda \\ \lambda_i^u - \nu_i^\lambda \end{matrix} \right\|_{V_k}^2 + \Omega_{k+N_c} \right\} \tag{4}$$

$$s.t. \quad \left. \begin{matrix} \dot{\chi}_j = f(\chi_j, \nu_j, \rho_j) \\ \chi_{min} \leq \chi_j \leq \chi_{max} \\ \nu_{min} \leq \nu_j \leq \nu_{max} \end{matrix} \right\} \; j \in \mathscr{I}_{k+1}^{k+N_c}$$

$$\Omega_{k+N_c} = \left\| \lambda_{k+N_c} - \chi_{k+N_c}^\lambda \right\|_{V_N}^2 \tag{5}$$

where Ω_{k+N_c} is the terminal cost function, $\chi_k^\lambda \in \mathbb{R}^{n_\lambda}$ is the subset of the state variables χ_k with corresponding entries in λ_k, and V_k, V_N are the symmetric positive-semidefinite weighting matrices.

In the context of path tracking, the RHC problem chooses the sequence of control actions which, given the current position and slip parameter values, minimises the least-squares distance between the model predicted motion and the target reference path over the horizon.

4.5 Solution methods

Both RHE (2) and RHC (4) are nonlinear least squares optimisation problems, so similar methods can be used to solve both. In this work, the generalised Gauss-Newton multiple-shooting method—derived from the classical Newton method—is used. This method was developed specifically for solving least-squares optimisation problems quickly, by utilising the real-time iteration scheme proposed in [28], which improves convergence and reduces computation cost.

To solve the constrained nonlinear optimisation problems involved in RHEC, the *ACADO* code-generation tool is used, which is an open-source software package which generates customised real-time RHE and RHC algorithms as efficient, self-contained C-code modules [29]. The generated dense quadratic sub-problems are then solved using the *qpOASES* online quadratic program solver [30].

5 Experimental Setup

The objective in these real-time experiments is to demonstrate the performance of the proposed RHEC with SB strategy for accurately following a user-defined trajectory (waypoints and speed) over a grass field on a slight incline—an example of the scenario described can be seen in Figure 4. We note that while the chosen test site is homogeneous in terms of terrain type, it does contain a steady incline, which we expect to impact slip in a more systematic and controlled manner as opposed to changing the type of terrain—an important property when attempting to characterise performance during preliminary testing.

The platform used in these experiments is the ACFR Swagbot (Figure 1). The measurements available from this platform include the GPS position and orientation, the angles of the steering joints, and the angular rate of the drive joints. The joint angles are used to calculate the virtual centre steering angle, steering rate and speed. The pose is provided by the Trimble BD982 GNSS, and the joint measurements by encoders and Hall-effect sensors for the steering and drive joints respectively.

The system runs using a combination of MATLAB, Python and ACFR internal software modules, with inter-communication handled using ROS. The RHEC algorithm is implemented in MATLAB, receiving sensor readings via custom ROS buffer messages, and publishing the sequence of control commands over the forward horizon to an Ackermann module, which emulates Ackermann steering behaviour on the Swagbot platform. The system diagram describing the sensing-estimation-control-actuation pipeline as implemented on the robot is shown in Figure 5.

The GPS centre is offset to correspond to the robot body frame centre, whereas the bicycle model position is defined to be the centre of the rear axle—which lies at a distance of $d = 0.64$m directly behind the robot centre. Thus the position measurements at the rear axle vary nonlinearly with the platform heading β.

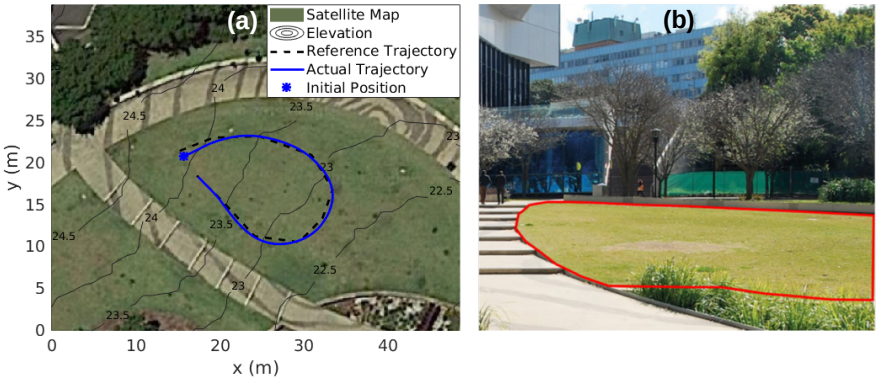

Fig. 4 (a) Reference (black) and actual (blue) trajectory followed by Swagbot using the SB RHEC strategy. Contour map displays the elevation in metres. (b) A photograph of the testing area - the region used for the experiments is highlighted with a red border.

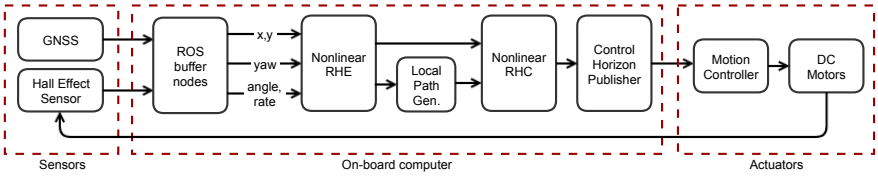

Fig. 5 System diagram of the RHEC implementation on the Swagbot platform.

Conservative estimates of variance for the sensor measurements were used: $\sigma_x = \sigma_y = 0.12$m for the GPS position measurement and $\sigma_\beta = 0.035$ for the heading (based on the GPS accuracy on the day), $\sigma_\delta = 0.035$ for the steering angle, and $\sigma_{u_1} = 0.1$ m/s and $\sigma_{u_2} = 0.17$ rad/s for the actuation measurements.

The RHE and RHC modules are implemented in ACADO and MATLAB. For simplicity, we opt to use similar weightings as employed in [4]. The horizon length and step size used are $N_e = N_c = 15$ and $\Delta t = 0.2$ s respectively, to ensure sufficient spatial distance between consecutive sampling points, and a long enough horizon to allow reliable convergence to the reference path. The steering angle is constrained to $-45^\circ \leq \delta \leq 45^\circ$, and the initial guesses for the estimated slip parameter values are $\kappa_0 = \mu_0 = 0.5$.

5.1 RHE Configuration

The measurement function used in these experiments is $h(x,u) = [x_{pos} + d\cos(\beta), y_{pos} + d\sin(\beta), \beta, \delta, u_1, u_2]^T$, with d being the offset defined in Section 5. The measurement variances are used in the calculation of the RHE weighting matrix, $R_k = \text{diag}\left(\sigma_x^2, \sigma_y^2, \sigma_\beta^2, \sigma_\delta^2, \sigma_{u_1}^2, \sigma_{u_2}^2\right)^{-1} \in \mathbb{R}^{6\times 6}$.

To enforce the desired blocking structure within the ACADO solver—as ACADO does not yet explicitly support parameter estimation—the state and system model (1) is augmented with the slip parameter terms and two corresponding virtual controls—u_κ, u_μ—as follows: $\left[\ldots, \dot{\kappa}, \dot{\mu}\right]^T = \left[\ldots, u_\kappa, u_\mu\right]^T$. The virtual controls are introduced to control the rate of change of the slip parameters κ and μ, which are constrained to be zero everywhere except at the start of a blocking interval.

In the FHB case, u_κ, u_μ are set to zero, fixing the parameter value over the horizon. To implement block overlapping, two blocking parameters S and o are introduced, specifying block size and overlap extent respectively. Each sampling time the bounds for u_κ, u_μ are set to be zero everywhere except at the start of a new block, both defining and enforcing the blocking structure. The overlap constraint is enforced by suppressing the formation of new blocks after time $t_k - o\Delta t$.

To penalise the parameter value shift between blocks, the virtual control terms u_κ, u_μ are added to the measurement function, $h(x,u) = \left[\cdots, u_\kappa, u_\mu\right]^T$, and corresponding weighting terms w_{u_κ}, w_{u_μ} are added to R_k. The value of these weighting

terms dictates the extent to which block value deviations are penalised relative to the other measured values. The value of the virtual u_κ, u_μ measurements are always set to zero, thus penalising the magnitude of the value shift. Additionally, the shift in parameter values between blocks is constrained to be $-1/\Delta t \leq \dot{\kappa}, \dot{\mu} \leq 1/\Delta t$—the maximum shift in value possible within the bounds.

For the experiments conducted, the SB-OP RHEC variant is used, with a block-size and overlap $S = o = 3$, and penalty weightings of $w_{u_\kappa} = w_{u_\mu} = 10$. We have considered only the value-blocking variant in this paper, due to its more consistent performance across all speeds.

It should be noted that persistent excitation is required to accurately estimate the traction parameters, since the mechanism by which these parameters are observed is proprioceptive in nature. As the robot's speed and steering angle approach zero, these parameters will become unobservable.

5.2 RHC Configuration

A 'global' space-based reference trajectory $\Lambda_j = [x_{ref}^{(j)}, y_{ref}^{(j)}, u_{1,ref}^{(j)}, u_{2,ref}^{(j)}]^T$ consists of a sequence of waypoints augmented with a desired control reference. At each t_k, a new time-based 'local' trajectory λ is generated from Λ, as defined in Section 4.1.

This is achieved by taking the current position estimate and finding the closest point $(\tilde{x}_{ref}, \tilde{y}_{ref})$ on the global trajectory. The next N_c points along the trajectory, spaced equally $\Delta t \cdot \tilde{u}_{ref}^k$ metres along the path, are then returned as the local trajectory λ_k. Interpolation is performed to obtain the associated control values $(\tilde{u}_{1,ref}, \tilde{u}_{2,ref})$. The weighting matrices for the RHC step are $V_k = \text{diag}(1.0, 1.0, 5.0, 5.0)$ and $V_T = 10 \cdot V_k$, and the steering rate constraints are $-45^\circ/s \leq u_2 \leq 45^\circ/s$.

6 Results and Discussion

Despite being initialised in a slightly off-track position, Swagbot was observed to quickly converge to the reference trajectory and remain on-track thereafter. For comparison of the reference and actual trajectories of the robot, see Figure 4 (a). This trajectory was tracked one time each for the FHB and SB-OP methods at 1.5 m/s, and again at 2.5 m/s. The trajectory was manually specified by selecting waypoints, and as such is composed of straight line segments with abupt turns, which are impossible to track perfectly with an Ackermann vehicle. The controller instead produces a smoother trajectory, managing to remain close to the reference while also minimising steering effort. It is worth noting that while the infeasible reference path will result in a larger tracking error compared to an equivalent clothoid path, it does present a more challenging tracking problem for the controller.

The packeted RHC method allows the system to continue tracking even if a sampling time deadline is missed, though this behaviour was only observed in the first

1–2s of the tracking, due to initialisation overhead. As a safeguard, if more than a second passes without an update, or if the estimator or controller fails, the robot is immediately commanded to stop.

An example of the slip parameter estimates for the 2.5 m/s runs can be seen in Figure 6. We observe that the SB-OP strategy is able to recover far more quickly from the (intentionally) poor initial slip estimate than the FHB strategy, rapidly converging to the expected slip values, albeit with some initial oscillation in μ. Overall, the values for both parameters tended to fall within the 0.8–1 range. While slip bounds of 0–1 were chosen as the parameter constraints, these were fairly conservative for the conditions, and these bounds may be relaxed or tightened to suit the environment. Tighter ranges would serve to increase speed of convergence, and reduce risk of instability due to noise and outlier measurements, though the model would then have difficulty accounting for motions resulting from slippage outside the described range.

The value of κ gradually decreased over the course of the run, indicating an increase in slip correllating with the gradual increase in slope along the trajectory (from roughly -5° to 5°). The large initial oscillations in the μ estimate were expected, as the steering slip is unobservable at the initial steering angle and rate of zero. Once turn commands are given, the value quickly converges to a more reasonable estimate. We do see an increase in steering slip occurring at the 9–15s interval, which corresponds to when the sharpest turn is negotiated.

In terms of path tracking, the impact of SB-OP's improved slip estimation performance can best be seen in the 0–5s interval of Figure 7 (a), where the strategy recovers quickly from a larger initial position error, and displays better position tracking performance up until the point that the two strategies' estimates converge. This clearly demonstrates SB-OP's improved ability to compensate for sudden and significant changes in slip conditions. This same trend is also reflected in both the speed tracking and steering effort minimisation, as shown in Figure 7 (b) and (c) respectively.

There is no significant drift in the magnitude of the position error over the run (Figure 7 (a)), as would be expected in the presence of nonuniform slip conditions, indicating that the controller has successfully compensated for the increased slip-

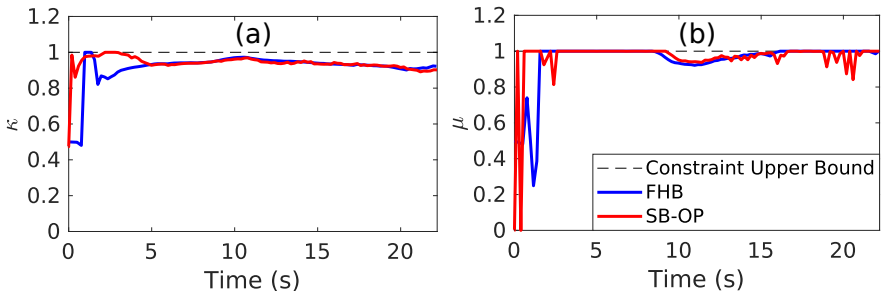

Fig. 6 Estimated slip parameter values for RHE at 2.5 m/s. (a) Estimated κ value. (b) Estimated μ value. Black dashed line denotes parameter upper bound of 1.

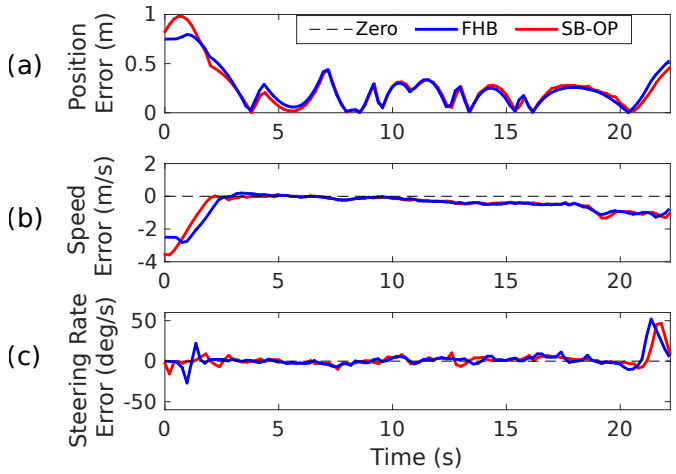

Fig. 7 Tracking error comparison between FHB and SB-OP at 2.5 m/s. (a) Position error, measured as distance from closest point on reference path. (b) Speed (u_1) error. (c) Steering rate (u_2) error. Note that the purpose of the steering rate tracking is to minimise steering effort, rather than perfectly track a given profile (ie. $u_{2,ref} = 0$), thus zero error only occurs when not actively turning.

page in the uphill section; this is also mirrored by the decrease in estimated κ in Figure 6 (a). We do however observe an increase in speed tracking error (Figure 7 (b)) for this segment, indicating that the longitudinal slip experienced is not wholly being compensated for. We suspect the reason for this lies in our overly conservative initial estimate for the wheel speed measurement covariance, which lowers the weighting of this term in the optimisation problem, and thus limits how aggressively the speed error is minimised. Further characterisation of the speed sensing pipeline is planned in order to provide a more realistic covariance estimate.

The tracking performance of the methods at each tested speed is summarised in Figure 8, displaying RMSE for the position and speed; adjusting for the discrepancy in starting displacement. The improved slip estimation afforded by the SB-OP strategy is reflected by a 4–7% reduction in root-mean-squared error (RMSE); an overall improvement in tracking performance comparable with our simulation results.

In terms of speed tracking, a significant improvement was observed at 2.5 m/s, with $\sim$13% reduction in RMSE. For 1.5 m/s, the speed tracking of SB-OP was slightly worse than FHB, and we suspect that this may be due to the increased susceptability to measurement noise at these lower speeds, where the slip variation over the horizon is less significant, and thus better approximated by a larger block-size.

Optimal selection of $N_e, N_c, S, o, w_{u_\kappa}, w_{u_\mu}$ is guaranteed to yield tracking performance at least as good as FHB in all cases, as FHB is a special case of SB-OP (where $o = N_e$)—the reduced performance seen here is a result of a suboptimal manual selection of these parameters. At lower speeds, the spatial interval between successive samples—and consequently, the spatial extent of the horizon—is reduced substan-

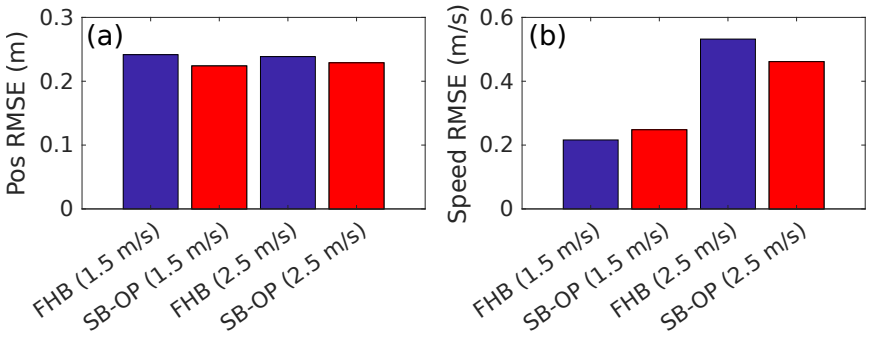

Fig. 8 Tracking RMSE for each run, for (a) the position, and (b) the speed.

tially, and thus measurement noise becomes a larger contributor to variance between the samples, compromising parameter estimation accuracy. One way to resolve this issue involves increasing the sampling interval, and thereby the spatial span of the horizon, however, sparse sampling is undesirable in a control context, as it limits the controller's ability to adapt quickly to changes in conditions. With SB, it is possible to increase the horizon length without adjusting the sampling interval, whilst modulating the block size S to curtail the number of parameters to estimate, and thus the complexity of the problem, emulating the benefits of larger sampling intervals without sacrificing controllability.

Similarly, o encodes the minimum spatial extent necessary for a good current slip estimate, and w_{u_κ}, w_{u_μ} encode the expected rate of spatial variance of surface slip in the environment. All these quantities can reasonably be selected as a function of platform speed, requiring only determination of a sufficient spatial interval for accurate parameter estimation. Thus only one design parameter would be required to determine a near optimal configuration for the SB-OP across the full range of operating speeds. Optimal selection of $o, w_{u_\kappa}, w_{u_\mu}$, however, would also be dependent on the roughness (spatial slip variance) of the terrain.

7 Conclusion

An RHEC algorithm for tracking trajectories in slippery environments was implemented on the Swagbot platform and preliminary testing of the strategy was performed. Both FHB and SB methods were tested, and their performance compared. Results demonstrated SB improved tracking performance by 4–7% over FHB. Future work will involve further testing and validation on more varied and challenging terrain conditions, optimal blocking parameter and horizon selection as a function of speed, and application of the RHEC SB strategy with a more complex 4 wheel independent steer/drive model.

References

1. M. Li, K. Imou, K. Wakabayashi and S. Yokoyama, Review of research on agricultural vehicle autonomous guidance, *Int J Agric and Biol Eng*, Vol. 2, No. 3, pp. 1–16, 2009.
2. H. Kong and S. Sukkarieh, An internal model approach to estimation of systems with arbitrary unknown inputs, *Automatica*, accepted and to appear, 2019.
3. T. Fukao, H. Nakagawa and N. Adachi, Adaptive tracking control of a nonholonomic mobile robot, *IEEE Trans. on Robotics and Automation*, Vol. 16, No. 5, pp. 609–615, 2000.
4. T. Kraus, H. J. Ferreau, E. Kayacan, H. Ramon, J. De Baerdemaeker, M. Diehl, and W. Saeys, Moving horizon estimation and nonlinear model predictive control for autonomous agricultural vehicles, *Computers and Electronics in Agriculture*, Vol. 98, pp. 25–33, 2013.
5. E. Kayacan, S. N. Young, J. M. Peschel, and G. Chowdhary, High-precision control of tracked field robots in the presence of unknown traction coefficients, *J. Field Robot.*, pp. 1–13, 2018.
6. E. Kayacan, Z. Zhang and G. Chowdhary, Embedded high precision control and corn stand counting algorithms for an ultra-compact 3D printed field robot, *In Proceedings of Robotics: Science and Systems*, pp. 1–8, Pittsburgh, Pennsylvania, 2018.
7. E. Kayacan, E. Kayacan, H. Ramon, and W. Saeys, Distributed nonlinear model predictive control of an autonomous tractor–trailer system, *Mechatronics*, Vol. 24, No. 8, pp. 926–933, 2014.
8. H. Kong, G. Goodwin and M. M. Seron, Predictive metamorphic control, *Automatica*, Vol. 49, pp. 3670–3676, 2013.
9. G. C. Goodwin, H. Kong, G. Mirzaeva, and M. Seron, Robust model predictive control: reflections and opportunities, *Journal of Control and Decision*, Vol. 1, No. 2, pp. 115–148, 2014.
10. H. Kong and S. Sukkarieh, Metamorphic moving horizon estimation *Automatica*, Vol. 97, pp. 167–171, 2018.
11. H. Kong and S. Sukkarieh, Suboptimal receding horizon estimation via noise blocking, *Automatica*, Vol. 98, pp. 66–75, 2018.
12. N. D. Wallace, H. Kong, A. J. Hill and S. Sukkarieh, Structured noise blocking strategies for receding horizon estimation and control of mobile robots with slip, *In Proceedings of Australasian Conference on Robotics and Automation*, pp. 1–10, Lincoln, New Zealand, 2018.
13. N. D. Wallace, H. Kong, A. J. Hill and S. Sukkarieh, Receding horizon estimation and control with structured noise blocking for mobile robot slip compensation, *2019 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–7, Montreal, Canada, 2019.
14. N. D. Wallace, H. Kong, A. J. Hill and S. Sukkarieh, Motion Cost Characterisation of an Omnidirectional WMR on Uneven Terrains, *Proc. of Joint 12th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles and 1st IFAC workshop on Robot Control*, accepted and to appear, Daejeon, Korea, 2019.
15. G. Dissanayake, S. Sukkarieh, E. Nebot, and H. Durrant-Whyte, The aiding of a low-cost strapdown inertial measurement unit using vehicle model constraints for land vehicle applications, *IEEE Trans. on Robotics and Automation*, Vol. 17, No. 5, pp. 731–747, 2001.
16. J. Yi, H. Wang, J. Zhang, D. Song, S. Jayasuriya and J. Liu, Kinematic modeling and analysis of skid-steered mobile robots with applications to low-cost inertial-measurement-unit-based motion estimation, *IEEE Trans. on Robotics*, Vol. 25, No. 5, pp. 1087–1097, 2009.
17. J. Backman, T. Oksanen, A. Visala, Navigation system for agricultural machines: nonlinear model predictive path tracking, *Computers and Electronics in Agriculture*, Vol. 82, pp. 32–43, 2012.
18. E. Kayacan, W. Saeys, H. Ramon, C. Belta and J. M. Peschel, Experimental validation of linear and nonlinear MPC on an articulated unmanned ground vehicle, *IEEE/ASME Transactions on Mechatronics*, Vol. 23, No. 5, pp. 2023–2030, 2018.
19. R. Lenain, B. Thuilot, C. Cariou, and P. Martinet, High accuracy path tracking for vehicles in presence of sliding: application to farm vehicle automatic guidance for agricultural tasks, *Autonomous Robots*, Vol. 21, No. 1, pp 79–97, 2006.

20. R. Lenain, B. Thuilot, C. Cariou, and P. Martinet, Mixed kinematic and dynamic sideslip angle observer for accurate control of fast off-road mobile robots, *Journal of Field Robotics*, Vol. 27, No. 2, pp. 181–196, 2010.
21. S. M. LaValle, Planning Algorithms, *Cambridge University Press*, pp. 722–726, 2006.
22. J. M. Snider, Automatic steering methods for autonomous automobile path tracking, *Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08*, 2009.
23. J. Kong, M. Pfeiffer, G. Schildbach and F. Borrelli, Kinematic and dynamic vehicle models for autonomous driving control design, *2015 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1094–1099, Seoul, South Korea, 2015.
24. H. Ferreau, T. Kraus, M. Vukov, W. Saeys and M. Diehl, High-speed moving horizon estimation based on automatic code generation, *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, pp. 687–692, Maui, HI, 2012.
25. C. V. Rao, J. B. Rawlings, and J. H. Lee, Constrained linear state estimation–a moving horizon approach, *Automatica*, Vol. 37, No. 10, pp. 1619–1628, 2001.
26. A. Voelker, K. Kouramas, and E. Pistikopoulos, Moving horizon estimation: error dynamics and bounding error sets for robust control, *Automatica*, Vol. 49, No. 4, pp. 943–948, 2013.
27. B. Morabito, M. Kogel, E. Bullinger, G. Pannocchia, and R. Findeisen, Simple and efficient moving horizon estimation based on the fast gradient method, *Proc. of 5th IFAC Conference on NMPC*, pp. 428–433, Seville, Spain, 2015.
28. M. Diehl, H. G. Bock, J. P. Schlöder, R. Findeisen, Z. Nagy and F. Allgöwer, Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations, *Journal of Process Control*, Vol. 12, No. 4, pp. 577–585, 2002.
29. B. Houska, H. J. Ferreau and M. Diehl, ACADO Toolkit – An open source framework for automatic control and dynamic optimization, *Optimal Control Applications and Methods*, Vol. 32, No. 3, pp. 298–312, 2011.
30. H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock and M. Diehl, qpOASES: A parametric active-set algorithm for quadratic programming, *Mathematical Programming Computation*, Vol. 6, No. 4, pp. 327–363, 2014.