Nonlinear model predictive control for improving range-based relative localization by maximizing observability

Shushuai Li*, Christophe De Wagter and Guido C. H. E. de Croon
Delft University of Technology, Kluyverweg 1, Delft

ABSTRACT

Wireless ranging measurements have been proposed for enabling multiple Micro Air Vehicles (MAVs) to localize with respect to each other. However, the high-dimensional relative states are weakly observable due to the scalar distance measurement. Hence, the MAVs have degraded relative localization and control performance under unobservable conditions as can be deduced by the Lie derivatives. This paper presents a nonlinear model predictive control (NMPC) by maximizing the determinant of the observability matrix in order to generate optimal control inputs, which also satisfy constraints including multi-robot tasks, input limitation, and state bounds. Simulation results validate the localization and control efficacy of the proposed MPC method for range-based multi-MAV systems with weak observability, which has faster convergence time and more accurate localization compared to previously proposed random motions.

1 INTRODUCTION

The use of multiple aerial robots has been studied deeply in recent years for more complicated tasks and challenging environments [1]. For example, a predictive control is proposed for flights of a swarm of five quadrotors despite cluttered obstacles [2]. In outdoor confined spaces, multiple drones are controlled with the evolutionary optimization method for flocking flights [3]. Multiple flying robots coordinate with simultaneous localization based on ranging measurements with beacons [4]. These recent studies show the state-of-art aerial swarm methods. However, most of them rely on extra positioning systems such as indoor optiTrack [2], outdoor GPS [3] or beacons [4].

To remove the dependence of the external infrastructure such as positioning systems, onboard sensors are deployed for developing an autonomous swarm of drones. For example, 3D relative direction can be estimated by sound-based microphone arrays and allows for leader-follower flights of micro aerial vehicles [5]. An array of infrared sensors can also enable relative positioning and inter-robot spatial-coordination [6]. However, these sensor arrays are too heavy and power-consuming for tiny flying robots. In [7], fully distributed and autonomous multiple tiny flying robots explore unknown environments with finite state machine. However, the relative localization is not very accurate due to the direct usage of signal strength, which may not fulfil the precise cooperative tasks.

Vision is the most widely used solution for multi-robot relative localization. Outdoor flocking of multiple drones localize each other with deep neural network and cameras for a safe navigation [8], which requires heavy AI hardware to run the deep network, also for [9] and [10]. Marker-based localization requires simple computation such as recognizing black circles [11] or April tags [12]. But these visual methods are easily influenced by the field of view or lighting conditions that lead to detection failure and localization disaster.

Wireless ranging sensors provide omnidirectional and low-cost ranging measurements, and recently have been used frequently for relative localization. It was initially proposed in [13], where use was still made of Bluetooth in order to fit on tiny MAVs. In [14], an ultra-wide band (UWB) based cooperative relative localization was proposed to estimate the neighbor drones' position based on the distance and self-displacement measurements under common orientation. Furthermore, [15] removes the orientation assumption and achieves the relative localization purely using the distance measurement and acceleration model. However, these experiments assumed high-order dynamic model and has low ranging frequency, which is not efficient for a large number of tiny robots.

In [16], a simplified velocity model and robust ranging protocol are designed for multiple tiny flying robots with self-regulated localization convergence. However, the initialization procedure with random velocity inputs is not efficient. Thus, this paper considers using nonlinear MPC to design the multi-robot controller by maximizing the task performance and degree of observability, while satisfying the constraints such as input velocity bounds and state bounds.

There are some related papers discussing the control of bearing-based or rang-based multi-robot systems [17]. Most papers use persistent excitation methods by setting specific active control patterns to maintain observability, which is not flexible nor optimal for other tasks or constraints.

The main contribution of this paper is leveraging weak

*Email address: s.li-6@tudelft.nl

observability theory to optimize the multi-robot control inputs, which has not yet been presented, to the best knowledge of authors. Specifically, the proposed NMPC framework maximizes the nonlinear observability condition derived by Lie derivatives, which is coupled with the velocity inputs and relative states. This leads to faster localization convergence and higher estimation accuracy even after convergence, compared to the random control inputs.

The rest of the paper is organized as follows. Section 2 states the problem including the range-based multi-MAV model, weak observability condition and the problem definition. Section 3 proposes the nonlinear MPC method with the cost function and corresponding constraints. Section 4 gives the simulation results of the proposed control with Acados, an integrated nonlinear MPC tool. The conclusion is discussed in Section 5.

2 PRELIMINARIES

This section briefly introduces the multi-MAV kinematic model and relative Kalman filter. Based on the relative model and distance observations, the observability matrix is determined with Lie derivatives. Finally, the control problem is defined by considering both the model and observability.

2.1 Relative multi-MAV model

The model of twin MAVs is described in this subsection, as the relative localization is distributed and triggered by the ranging event among arbitrary two MAVs. The simulated relative model has been tested in real experiments in our previous work, thus it has a small gap compared to the real-world multi-robot system. For details, consult in [16].

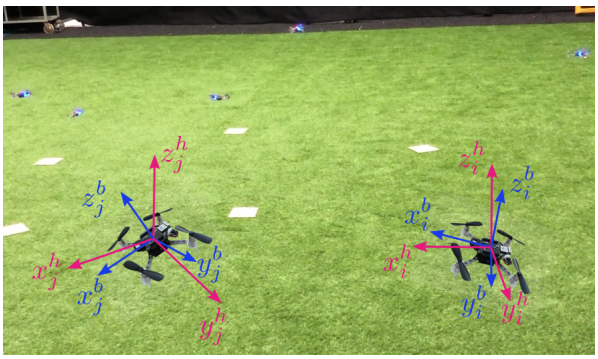

Figure 1: The diagram of a twin-MAV kinematic model, and two coordinated frames. Body frames and horizontal frames are shown with blue axes and red axes, respectively. Both frames are fixed to the robot, while the horizontal frames always have a vertical Z axis. The background images shows previous experiments of multi-MAV relative localization but without optimal control.

For simplicity, we assume the yaw rate of both robots to be zero. This assumption has no influence on the 3D movements of each robot. The control input vector $\boldsymbol{u} = [v_i^x, v_i^y, v_j^x, v_j^y]^T$ represents the XY-axis velocities of the i^{th} and j^{th} robots in their horizontal frames as shown in Fig. 1. The velocities in horizontal frame can be obtained by rotating the measured velocities in body frame, so that the Z axis in the horizontal frame aligns with gravity. The relative state is denoted by $\boldsymbol{x} = [x_{ij}, y_{ij}, \psi_{ij}]^T$, which represents the j^{th} robot's position and relative yaw in the horizontal frame of the i^{th} robot.

The nonlinear relative kinematic model can be derived from Newton formulas by considering the states $\boldsymbol{x}$ and velocity inputs $\boldsymbol{u}$, which can be written as follows [16]

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} cos(\psi_{ij})v_j^x - sin(\psi_{ij})v_j^y - v_i^x \\ sin(\psi_{ij})v_j^x + cos(\psi_{ij})v_j^y - v_i^y \\ 0 \end{bmatrix}. \quad (1)$$

A distance measurement d comes from the DWM1000 ranging sensors, and has the following relation to model states:

$$d = h(\boldsymbol{x}) = \sqrt{x_{ij}^2 + y_{ij}^2 + (h_j - h_i)^2}, \quad (2)$$

where h_i and h_j are the altitudes measured directly from the height sensors. The function $h(\cdot)$ represents the scalar nonlinear observation.

2.2 Relative estimation

This subsection briefly reviews the Extended Kalman filter (EKF) for the relative localization. The discrete prediction is formulated as:

$$\begin{aligned} \hat{\boldsymbol{x}}_{k+1|k} &= F(\hat{\boldsymbol{x}}_k, \boldsymbol{u}_k) = \hat{\boldsymbol{x}}_k + \dot{\boldsymbol{x}}_k \Delta t, \\ \boldsymbol{P}_{k+1|k} &= \boldsymbol{A}_k \boldsymbol{P}_{k|k} \boldsymbol{A}_k^T + \boldsymbol{B}_k \boldsymbol{Q}_k \boldsymbol{B}_k^T \end{aligned} \quad (3)$$

where Δt is the update interval, $\boldsymbol{P}$ is the error covariance, $\boldsymbol{A} = \partial F / \partial \boldsymbol{x}$ and $\boldsymbol{B} = \partial F / \partial \boldsymbol{u}$ are the Jacobians of states and inputs, and $\boldsymbol{Q}$ is the process noise covariance.

The final state estimation is estimated by using the distance measurement as shown below:

$$\begin{aligned} \boldsymbol{K}_k &= \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T (\boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T + \boldsymbol{R}_k)^{-1}, \\ \hat{\boldsymbol{x}}_k &= \hat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k (d_k - \boldsymbol{H}_k \hat{\boldsymbol{x}}_{k|k-1}), \\ \boldsymbol{P}_k &= (\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k) \boldsymbol{P}_{k|k-1} \end{aligned} \quad (4)$$

where $\boldsymbol{K}$ is the Kalman gain, $\boldsymbol{H} = \partial h(\boldsymbol{x}) / \partial \boldsymbol{x}$ is the observation Jacobian, $\boldsymbol{R}$ is the observation noise covariance, and $\boldsymbol{I}$ is the identity matrix.

Remark 1 *The kinematic model and EKF-based relative localization have been validated in real-world experiments [16].*

2.3 Observability constraint

Observability of nonlinear systems can be analyzed by Lie derivatives [18]. The corresponding observability matrix is defined as

$$\boldsymbol{O} = [\nabla \mathcal{L}_f^0 h, \nabla \mathcal{L}_f^1 h, \nabla \mathcal{L}_f^2 h]^T \quad (5)$$

where $\mathcal{L}_f h$ means the Lie derivative of function f. The iterations satisfy three conditions: 1) $\mathcal{L}_f^0 h = h(\boldsymbol{x})$; 2) $\mathcal{L}_f^{i+1} h = \nabla \mathcal{L}_f^i h \cdot f$; 3) $\nabla \mathcal{L}_f^i h = \partial \mathcal{L}_f^i h / \partial \boldsymbol{x}$.

Therefore, the relative states are observable only when the observability matrix $\boldsymbol{O}$ is full rank. That means that the determinant should be non-zero, which is expressed as:

$$\begin{aligned}|\boldsymbol{O}| = f_O(\boldsymbol{x}, \boldsymbol{u}) = -2[&-v_i^x v_j^x s(\psi) + v_i^y v_j^x c(\psi) \\ &- v_i^x v_j^y c(\psi) - v_i^y v_j^y s(\psi)] * [-v_j^x y_{ij} c(\psi) + v_j^y y_{ij} s(\psi) \\ &+ v_i^x y_{ij} + v_j^x x_{ij} s(\psi) + v_j^y x_{ij} c(\psi) - v_i^y x_{ij}]\end{aligned} \tag{6}$$

where $s(\cdot)$, $c(\cdot)$, and ψ are simplifications of $sin(\cdot)$, $cos(\cdot)$, and ψ_{ij}, respectively.

2.4 Problem statement

The optimal control problem P_O with respect to observability $|O|$ for this multi-MAV system is defined as:

$$\max_{u^*, k \in \{1,2 \ldots \mathrm{N}\}} P_O(|O_k|) = \sum_{k=1}^{\mathrm{N}} |f_O(\boldsymbol{x}_k, \boldsymbol{u}_k)| \tag{7}$$

where $\boldsymbol{u}^*$ is the optimal control vector at current time, which is normally taken from an control sequence. The appropriate control input sequence guarantees the strong observability of the multi-robot system in the future, which can improve the relative localization in both convergence speed and estimation accuracy. The problem is how to calculate the optimal control input sequence.

3 METHODOLOGY

This section proposes a nonlinear model predictive control for solving the optimal problem as described in (7). Then the cost function is further extended for multi-robot tasks such as formation control and motion tracking. In the end, the solver settings for the nonlinear problem (NLP) are presented.

3.1 Nonlinear MPC

The intuitive solution for NLP is MPC, which can achieve the target by minimizing the cost function. Hence, the nonlinear MPC for the proposed problem is designed as follows

$$\boldsymbol{u}_{0|t} := \min_{\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t}} J(\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t}) \tag{8a}$$

$$\text{s.t. } \boldsymbol{x}_{k+1|t} = f(\boldsymbol{x}_{k|t}, \boldsymbol{u}_{k|t})\delta_t + \boldsymbol{x}_{k|t}, \tag{8b}$$

$$\boldsymbol{x}_{0|t} = \mathrm{sat}_{\boldsymbol{x}_1}^{\boldsymbol{x}_\mathrm{u}}(\hat{\boldsymbol{x}}_\mathrm{t}), \tag{8c}$$

$$\|\boldsymbol{p}_{\cdot|t}\|_2 - d_{\mathrm{safe}} \geq 0, \tag{8d}$$

$$v_l \leq v_{i,\cdot|t}^x, v_{i,\cdot|t}^y, v_{j,\cdot|t}^x, v_{j,\cdot|t}^y \leq v_u, \tag{8e}$$

$$p_l \leq x_{ij,\cdot|t}, y_{ij,\cdot|t} \leq p_u \tag{8f}$$

where $\boldsymbol{x}_{\cdot|t}$ and $\boldsymbol{u}_{\cdot|t}$ stands for the sequence of states and control inputs in the prediction horizon. The first control value $\boldsymbol{u}_{0|t}$ is taken as the input for the robots. The relative position is denoted by $\boldsymbol{p} = [x_{ij}, y_{ij}]^T$. Saturation function $\mathrm{sat}()$ clips data with lower bound $\boldsymbol{x}_l$ and upper bound $\boldsymbol{x}_u$.

The overall objective function $J(\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t})$ is composed by several cost functions, which will be designed later. The remaining equations represent the constraints, which guarantee that the controller satisfies the system dynamic as (8b), the initial state condition related to the current estimated state as (8c), the safe distance for collision avoidance as (8d), the upper and lower bounds of input velocities as (8e), and the relative position bounds as (8f).

Remark 2 *The constraint of initial state is related to the estimated state which is not correct before localization convergence. Hence, the limitation of initial value is necessary to avoid singularity when solving the NLP. A saturation function is employed to limit the the initial value as shown in* (8c)*. This is reasonable as many nonlinear robust MPC methods for systems with uncertain states have their stability proof by assuming bounds on the uncertain state.*

3.2 Cost functions

A nonlinear least square (NLS) method is deployed for minimizing the objective function of (8), which is written as:

$$J(\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t}) = J_O(\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t}) + J_C(\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t}) \tag{9}$$

where $J_O(\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t})$ and $J_C(\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t})$ represent the reformulated observability cost and multi-robot formation coordination cost, respectively.

To maximize the observability with the NLS method, the observability objective (7) is reformulated as the following cost function.

$$J_O(\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t}) = \sum_{k=0}^{\mathrm{N}-1} \omega_O \| \frac{a_O}{f_O(\boldsymbol{x}_k, \boldsymbol{u}_k) + \epsilon_O} \| \tag{10}$$

where ω_O, a_O and ϵ_O denote the constant weight, amplitude of cost value, and a small value preventing the singularity.

The coordination cost of $J_C(\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t})$ is designed for multi-robot tasks such as motion tracking. Given the reference position sequence $\widetilde{\boldsymbol{p}}_{\cdot|t} = [\widetilde{x}_{ij,\cdot|t}, \widetilde{y}_{ij,\cdot|t}]^T$, the motion tracking cost function is designed as:

$$J_C(\boldsymbol{x}_{\cdot|t}) = \sum_{k=0}^{\mathrm{N}-1} \omega_C \| \boldsymbol{p}_{k|t} - \widetilde{\boldsymbol{p}}_{k|t} \| \tag{11}$$

where $\boldsymbol{p}_{\cdot|t}$ is the predicted relative position sequence in the proposed MPC, calculated by (8b). Specially, if the reference sequence is constant such that $\widetilde{\boldsymbol{p}}_{\cdot|t} \equiv [a_x, a_y]$, the multi-robot motion tracking reduces to formation control.

Remark 3 *Since the coordination task is inaccurate before the localization convergence, the weight ω_C can be set dynamically for the control stability according to the localization accuracy, e.g., the trace of the estimation error covariance $\mathrm{tr}(\boldsymbol{P})$. However, a constant $\omega_C = 2$ is enough for the following formation task.*

Sometimes, a penalty cost can be introduced to smooth the control inputs as follows:

$$J_U(\boldsymbol{u}_{\cdot|t}) = \sum_{k=1}^{\mathrm{N}-1} \omega_U \|\boldsymbol{u}_{k|t} - \boldsymbol{u}_{k-1|t}\| \tag{12}$$

Other multi-robot motion control can also be incorporated into the cost function of $J(\boldsymbol{x}_{\cdot|t}, \boldsymbol{u}_{\cdot|t})$. This paper does not discuss the details of those cost functions such as flocking, swarming, and cooperative coordination.

3.3 Acados solver

The nonlinear MPC solver we use in this paper is Acados, which is an open-source and high-performance library for fast optimal control [19]. This software supports Python and is finely tuned for multiple CPU. As for the model definition and differentiation, CasADi is employed to deal with the constraints and model calculations [20].

The brief process of the solver setting is summarized as below. First, the continuous optimal problem is discretized by the multiple shooting method. Furthermore, real-time iteration (RTI) is selected to solve the sequential quadratic programming (SQP). The corresponding Hessian approximation is based on Gauss-Newton. The quadratic problems (QP) in SQP are solved with the partial condensing HPIPM, which is based on linear algebra library BLASFEO. Overall, this solver has a competitive computation speed compared to other stat-of-the-art NMPC solvers.

4 SIMULATION RESULTS

This section shows the improvement of the proposed nonliear MPC on the relative localization performance compared to the stochastic initialization procedure studied in [16]. The statistics of the localization errors and convergence speed are analyzed to validate the efficiency of the proposed controller. In addition, adaptive formation flight of multiple MAVs is studied as example application.

4.1 Simulation set-up

The following simulation experiments are conducted on a Dell Latitude 7480 laptop with a i7-6600U CPU with 4 cores at 2.60GHz and 8GB of RAM. For the simulation experiments, the corresponding EKF parameters are chosen as $\Delta t = 0.01s$, $t_{\mathrm{sim}} = 40s$, $\boldsymbol{Q} = \mathrm{diag}([0.25, 0.25, 0.01])$, and $\boldsymbol{R} = \mathrm{diag}([0.1])$. The initial estimated relative states are set to zero. In contrast, the initial ground-truth positions of each robot are randomly generated, such that the EKF estimation has no prior knowledge of the initial state information. The error covariance is initialized as $\boldsymbol{P} = \mathrm{diag}([10, 10, 0.1])$.

As for the parameters of the proposed nonlinear MPC, the horizon is set to $\mathrm{N} = 50$ and prediction time to $T_f = 1s$, which means each control prediction takes $\delta_t = 0.02s$. Larger prediction horizon has long-term constraint guarantees but with more computation burden. In the observability

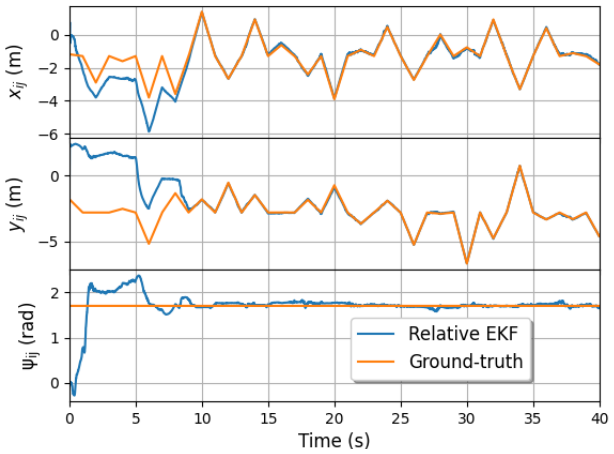

Figure 2: Relative state from EKF estimation and ground-truth between two MAVs under the random velocity inputs. The data consists of 2-axis relative positions and 1-axis relative orientation.

cost function, the parameters are $a_O = 0.021$, $\epsilon_O = 0.001$, and $\omega_O = 1$.

For the constraint settings, the saturation parameters for the initial state vector are chosen as $\boldsymbol{x}_l = [-4, -4, -15]$ and $\boldsymbol{x}_u = [4, 4, 15]$. The safe distance is set to $d_{\mathrm{safe}} = 0.1m$. The velocity input is bounded between $v_l = -2m/s$ and $v_u = 2m/s$. The minimum and maximum relative positions are set to $p_l = -4m$ and $p_u = 4m$, which prevents them flying far from each other.

4.2 Improvement on relative localization

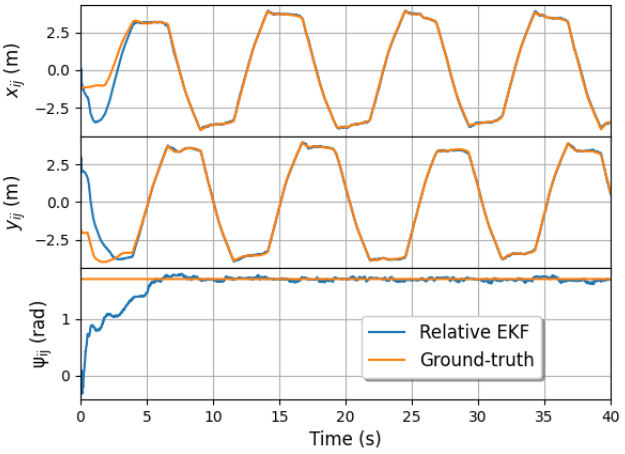

Figure 3: Relative state from EKF estimation and ground-truth between two MAVs under the nonlinear MPC controller. The data consists of 2-axis relative positions and 1-axis relative orientation.

This subsection compares stochastic initialization with nonlinear MPC, in order to verify that the proposed controller with consideration of pure observability cost has better localization performance than the former one. In this subsection,

the multi-robot task cost J_C is set to zero.

Fig. 2 and Fig. 3 shows the relative localization performance with the same initial relative states and same parameters for the EKF. Be notified that the three initial states are completely unknown for both the EKF and the controllers. Additionally, the maximum velocities for both controllers are set to be 2 m/s. From these two figures, we can see that the relative positioning with optimal controller has a faster convergence time (about 5s) compared to that of the random controller (about 9s). Especially, observability optimized NMPC has finite-time convergence in the axis of relative yaw, while the random control leads to overshooting as shown in the third subplot of Fig. 2. Therefore, the proposed controller with observability consideration excites all relative states which become more observable even with the unknown initial state errors.

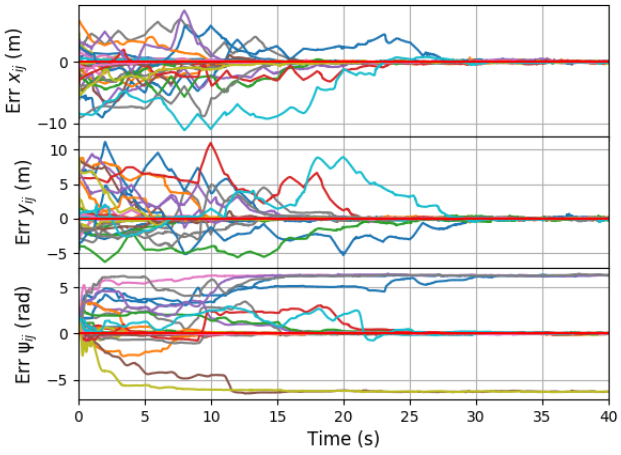

Figure 4: 30 simulation experiments of the stochastic controller from [16] with random initial MAV positions. This figure shows the estimation errors of 2-axis relative positions and 1-axis relative orientation. Note that yaw error with -2π or 2π offset has no influence on the relative localization due to the cos and sin operation.

In addition, after the localization is converged in Fig. 3, the optimal controller automatically generates a periodic motion pattern which is similar to the manual-designed persistent excitation motions. In addition, even with incorrect relative states, they still can avoid each other as shown in Fig. 3, because the observability cost penalizes the collision situation during which the observability determinant approximates zero.

To validate the general efficacy of the proposed NMPC, we gather more statistics on the performance. As shown in Fig. 4 and Fig. 5, 30 random simulation experiments are conducted for each controller. During each simulation epoch, the initial positions for both robots are generated randomly. Moreover, the velocity and distance measurement noise are also created randomly. Both figures imply that the proposed NMPC controller has in general a faster localization convergence speed.

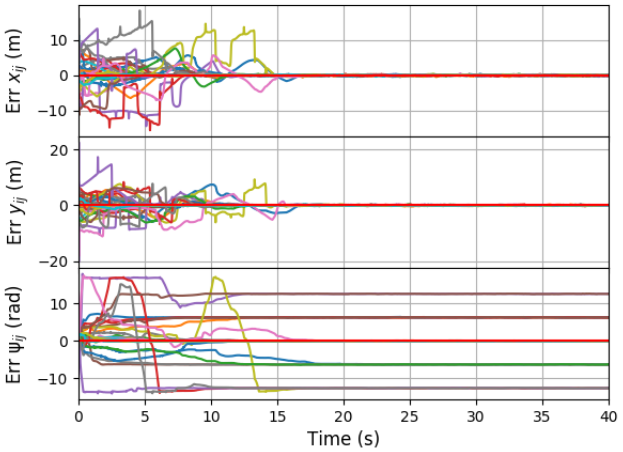

Figure 5: 30 simulation experiments of the optimal controller with random initial MAV positions. This figure shows the estimation errors of 2-axis relative positions and 1-axis relative orientation.

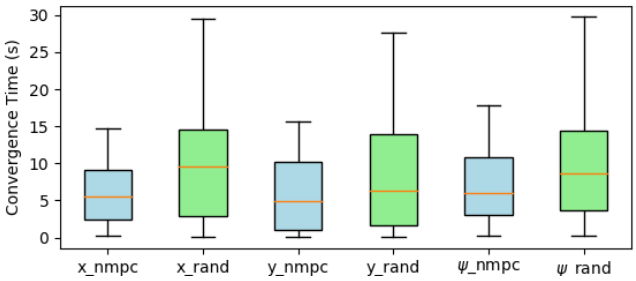

Figure 6: The statistics of convergence time of three-dimensional relative localization under 30 random tests. Blue: the proposed nonlinear MPC; Green: the stochastic control.

Fig. 6 shows the comparison of the detailed convergence time of two controllers with 30 random tests. From it we can see that the average convergence time of the NMPC on all axes is smaller than that of the random controller. Besides, NMPC with observability constraint has a lower maximum convergence time compared to random control inputs.

Another interesting result is the localization accuracy after estimation convergence. Fig. 7 shows the distributions of position estimation errors in the last 5 seconds of two controllers in the 30 random tests. Obviously, the proposed NMPC has lower averaged position estimation errors compared to the stochastic controller. Therefore, the behaviours after convergence are still meaningful to the localization performance. To study it, the control input $\boldsymbol{u}$ for two MAVs is shown in Fig. 8. From which we can see that all 4-channel velocities are approximating the maximum value of 2m/s. The

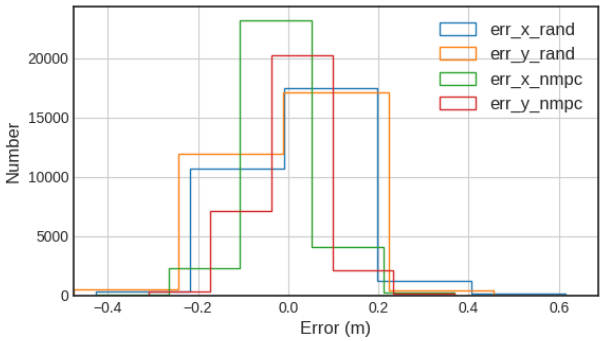

Figure 7: Localization error of two controllers after estimation convergence. Each distribution has total 15000 data on these 30 random tests, which is taken from the last 5 seconds when all estimators have converged.

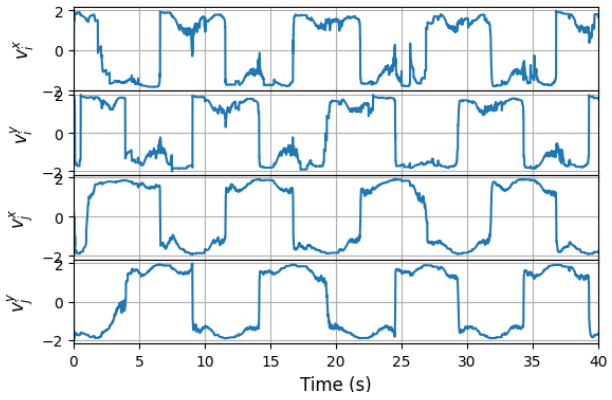

Figure 8: The control inputs generated by the proposed NMPC with observability optimization. These sequences show the velocity input values corresponding to the simulation in Fig. 3.

oscillations and changes of velocity direction occur due to the state bounds and velocity limitation. The four velocities are assigned different phases equally of the periodic motion pattern. This asynchronous behaviour has not been considered before, but NMPC can generate it automatically.

4.3 Formation control with NMPC

This subsection uses the NMPC controller for multi-robot tasks. Examples of formation flight and dynamic motion tracking are given below. At the beginning, a constant relative position is set in the task cost J_C, where $\widetilde{\boldsymbol{p}}_{\cdot|t} = [1, 1]m$. The other settings of the solver remain unchanged. After 15s, a variant relative motion reference is introduced, which is defined as $\widetilde{\boldsymbol{p}}_{\cdot|t} = [2cos(t), 2sin(t)]m$. This leads to a circle motion of the second MAV around the first MAV.

The corresponding control results are shown in the following figures. In Fig. 9 we can see that the proposed NMPC has fast and stable tracking performance given the formation and dynamic tracking tasks at $t = 5$ and $t = 15s$ respectively. In addition, the observability cost keeps being optimized simultaneously by the NMPC.

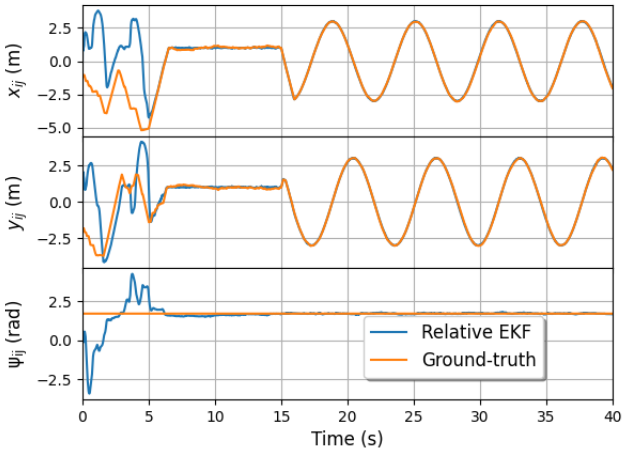

Figure 9: Relative localization and ground-truth between two MAVs under the proposed optimal control method and formation tracking multi-robot tasks. The target relative position is constant before $t = 15s$, and variant after $t = 15s$.

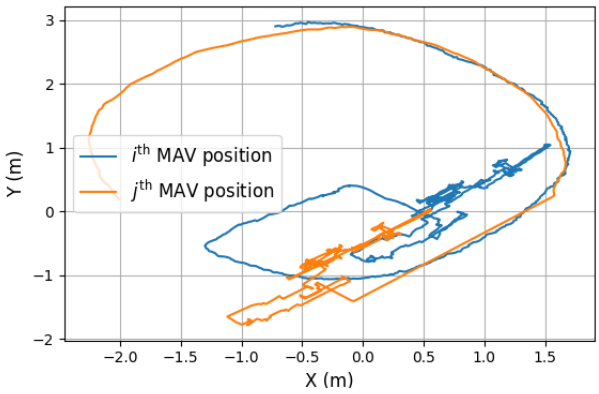

Figure 10: The world-frame trajectories of both MAVs under the proposed optimal control method and formation tracking multi-robot tasks. The time range of the data is between 10s and 20s.

To view the motion of each MAV in world-frame, the trajectories of both MAVs are plotted in Fig. 10. For the formation flight during 10-15s, both MAVs move slowly with constant relative positions. During 15-20s, both MAVs move to achieve the circle tracking and keep optimizing the observability according to the asynchronous behaviours. In addition, from trajetories after 15s in Fig. 9 we can see that introducing the multi-robot task cost eliminates the transients as shown in Fig. 8.

5 CONCLUSIONS

This paper proposes a novel nonlinear MPC controller with an observability cost to improve range-based multi-MAV relative localization. Simulation results demonstrate its faster localization convergence and lower estimation errors with respect to previously studied stochastic motion. Future work involves the implementation of this controller in real-world micro air vehicles for better localization and control.

REFERENCES

[1] M. Coppola, K. N. McGuire, C. De Wagter, and G. C. de Croon, "A survey on swarming with micro air vehicles: Fundamental challenges and constraints," *Frontiers in Robotics and AI*, vol. 7, p. 18, 2020.

[2] E. Soria, F. Schiano, and D. Floreano, "Predictive control of aerial swarms in cluttered environments," *Nature Machine Intelligence*, vol. 3, no. 6, pp. 545–554, 2021.

[3] G. Vásárhelyi, C. Virágh, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, 2018.

[4] M. Hamer and R. D'Andrea, "Self-calibrating ultra-wideband network supporting multi-robot localization," *IEEE Access*, vol. 6, pp. 22 292–22 304, 2018.

[5] M. Basiri, F. Schill, P. Lima, and D. Floreano, "On-board relative bearing estimation for teams of drones using sound," *IEEE Robotics and Automation letters*, vol. 1, no. 2, pp. 820–827, 2016.

[6] J. F. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano, "3-d relative positioning sensor for indoor flying robots," *Autonomous Robots*, vol. 33, no. 1, pp. 5–20, 2012.

[7] K. McGuire, C. De Wagter, K. Tuyls, H. Kappen, and G. C. de Croon, "Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment," *Science Robotics*, vol. 4, no. 35, 2019.

[8] F. Schilling, F. Schiano, and D. Floreano, "Vision-based drone flocking in outdoor environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2954–2961, 2021.

[9] H. Xu, L. Wang, Y. Zhang, K. Qiu, and S. Shen, "Decentralized visual-inertial-uwb fusion for relative state estimation of aerial swarm," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 8776–8782.

[10] A. Carrio, S. Vemprala, A. Ripoll, S. Saripalli, and P. Campoy, "Drone detection using depth maps," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1034–1037.

[11] J. Faigl, T. Krajník, J. Chudoba, L. Přeučil, and M. Saska, "Low-cost embedded system for relative localization in robotic swarms," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 993–998.

[12] M. Gassner, T. Cieslewski, and D. Scaramuzza, "Dynamic collaboration without communication: Vision-based cable-suspended load transport with two quadrotors," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 5196–5202.

[13] M. Coppola, K. N. McGuire, K. Y. Scheper, and G. C. de Croon, "On-board communication-based relative localization for collision avoidance in micro air vehicle teams," *Autonomous robots*, vol. 42, no. 8, pp. 1787–1805, 2018.

[14] K. Guo, Z. Qiu, W. Meng, L. Xie, and R. Teo, "Ultra-wideband based cooperative relative localization algorithm and experiments for multiple unmanned aerial vehicles in gps denied environments," *International Journal of Micro Air Vehicles*, vol. 9, no. 3, pp. 169–186, 2017.

[15] S. van der Helm, M. Coppola, K. N. McGuire, and G. C. de Croon, "On-board range-based relative localization for micro air vehicles in indoor leader–follower flight," *Autonomous Robots*, vol. 44, no. 3, pp. 415–441, 2020.

[16] S. Li, M. Coppola, C. De Wagter, and G. C. de Croon, "An autonomous swarm of micro flying robots with range-based relative localization," *arXiv preprint arXiv:2003.05853*, 2020.

[17] Z. Han, K. Guo, L. Xie, and Z. Lin, "Integrated relative localization and leader–follower formation control," *IEEE Transactions on Automatic Control*, vol. 64, no. 1, pp. 20–34, 2018.

[18] R. Hermann and A. Krener, "Nonlinear controllability and observability," *IEEE Transactions on automatic control*, vol. 22, no. 5, pp. 728–740, 1977.

[19] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "Acados: A modular open-source framework for fast embedded optimal control," *arXiv preprint arXiv:1910.13753*, 2019.

[20] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "Casadi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.