RCopterX - Experimental Validation of a Distributed Leader-Follower MPC Approach on a Miniature Helicopter Test Bed

Stephan M. Huck, Marvin Rueppel, Tyler H. Summers and John Lygeros

Abstract— **RCopterX is an indoor testbed for miniature remote controlled helicopters build at the Automatic Control Laboratory at ETH Zurich. The experimental setup includes modular custom control software, a hardware interface between a PC and radio-control signal transmitter, off-the-shelf miniature helicopters and an infrared vision system for global positioning indoors. Its purpose is to experimentally validate problems of UAVs performing multi-level controls, including stabilizing low level controls as well as testing of high level algorithms. The main features are the modular architecture allowing for multi-agent support and interchangeability of the vehicle hardware and control algorithms. For illustration, a reliable fast model predictive control (MPC) scheme for coaxial helicopters as well as the implementation of a formation control algorithm is described. The presented distributed leader follower MPC approach achieves reference tracking of the whole formation along with maintenance of the formation shape, by having the individual agents solve their own optimization problem based on information from their leader. Results of the experimental validation demonstrate the successful implementation and the multi-agent capabilities of the setup.**

I. INTRODUCTION

In recent years, autonomous vehicles have undergone a miniaturization, enabled by remarkable advances in sensing and computation technologies. This has led researchers to envision and study how individual vehicles can be pushed to their physical and dynamical limits and how cooperative teams of vehicles could be used to accomplish tasks such as aerial imaging or underwater exploration. There is now a large and increasing literature on various aspects of autonomous vehicle control and cooperative control of vehicle formations.

Much of the research on cooperative control of vehicle formations, however, is theoretical or methodological, and it is important to develop experimental testbeds that can validate control schemes and algorithms not just in simulation, but also on real hardware. Examples of such testbeds are described in [8], [15]. This paper describes RCopterX, an indoor testbed built for research on both, low-level vehicle control and multi-vehicle coordination algorithms using miniature helicopters. The experimental setup is small-scale, thus suitable for limited laboratory space, and still providing capabilities for multi-agent scenarios at the same time. We believe that it guarantees a flexible and efficient structure of low complexity while preserving many of the practical challenges of nonlinear and distributed control.

A fundamental problem in cooperative control of vehicle formations is formation shape control. The goal is for the individual vehicles to adjust their motion using limited relative state information about neighboring vehicles so that the formation achieves and maintains a specified geometric shape. Among possible control strategies, the model predictive control methodology (see e.g., [16]) is suitable to tackle possibly arising state constraints in such problems. Distributed model predictive control has been attracted research in the last decades [2], [3], [18] and specifically several algorithms for multi-vehicle formation control have been proposed in the literature [6], [10], [12], [13], [19]. To the best of our knowledge, only two papers [12], [13] describe experimental implementation of a distributed model predicted control algorithm on a multi-vehicle testbed. However, these algorithms are used for more sophisticated trajectory optimization with non-convex constraints on relatively slow time scales.

We experimentally validate a fast leader-follower model predictive control scheme, in which the solution to a global finite-horizon optimization problem is computed by distributing the computation over a set of communicating processors associated with each vehicle. The leader-follower structure we consider allows terms in the objective function to be split in a straightforward way so that each follower can independently optimize its own decision variables based on information from vehicles it follows. No iterative negotiation mechanism is required, and the formation control task can be executed at the same rate as for a single vehicle, with plans being updated every few milliseconds using custom solvers. All vehicles end up using reference tracking controllers with the same structure but transformed reference trajectories, which facilitates online changes to the formation topology, including leader changes. The leader can be given external commands such as set point changes and target trajectories, and the predictive mechanism allows followers to anticipate movements of the leader. The results of our experimental studies show that the proposed distributed algorithm tracks set point changes of the global position of the formation as well as changes of its geometrical shape. The approach is validated to work for sampling times in the range of milli-seconds. Furthermore, the achieved solver times are fast enough to trade off tracking errors with computation times.

The rest of the paper is structured as follows. Section II describes the testbed hardware and software setup. Section III describes the helicopter model. Section IV reviews a model predictive controller for a single vehicle using an automatically generated custom code, and Section V describes an experimental implementation and validation of the leader-follower model predictive control scheme. Concluding remarks are given in Section VI.

The authors are with the Automatic Control Laboratory, Department of Information Technology and Electrical Engineering, ETH Zurich, Zurich 8092, Switzerland. Email: {huck,summers,jlygeros}@control.ee.ethz.ch rueppelm@student.ethz.ch. This research was partially supported by the HYCON2 Network of Excellence

II. Testbed Description

Physical Setting

The RCopterX project [9] requires only a space of about $3\,\mathrm{m}\times3\,\mathrm{m}\times2.5\,\mathrm{m}$, where the 'flyable' volume, i.e., the space reliably covered by the vision system, is about $2\,\mathrm{m}\times2\,\mathrm{m}\times2.5\,\mathrm{m}$. The setup incorporates an infrared vision system for pose detection of the vehicles, the two PCs, running the vision and the control software separately and the helicopters with transmitters. For a more reliable detection of the objects the flyable space is shielded from sunlight and other infrared sources by black curtains. The cameras are mounted on height-adjustable rails on the walls to minimize disruption of the calibration.

Vision System

The 3D-position and orientation are recorded by a Vicon motion capture systemwhich consists of 4 Bonita B3 cameras with 0.3 megapixels and maximal field of view of $82°$ horizontal and $66°$ vertical. The collected data is processed by the Vicon Tracker v1.2 software, running on a normal Windows 7 desktop PC with an Intel Core i5-661, a dual core processor with hyper threading at $3.33\,\mathrm{GHz}$ and $16\,\mathrm{GB}$ RAM. The measurements are updated with 50Hz on a data stream, which is accessible through a TCP/IP port, to match the chosen sampling time of the entire system. In this setup reflective markers with a diameter of $9.5\,\mathrm{mm}$ are attached to the objects, which pushes the detection to its limit given the camera type and the covered volume. In the spirit of an easy to set up and affordable testbed, the authors would like to note, that the professional equipment could be replaced by inexpensive sensing systems relying on pose estimation and filtering. A good example for such a solution can be found, e.g., in [7].

Software Architecture

The core part of the system is the control software *Coaga* [1] written in C++, running on a second PC with an Intel Quad-Core i5-760 at $2.8\,\mathrm{GHz}$ and $8\,\mathrm{GB}$ RAM. A robust sampling time of $20\,\mathrm{ms}$ is chosen to acquire measurements via local area network, invoke the control computations and to interact with the transmission unit. The main feature is the modularity of the classes, e.g., supporting the switching between controllers during flight, online tuning and addition or removal of controlled agents during operation. Besides the control algorithms, Coaga provides a filtering module to implement different estimators, e.g., a Kalman Filter for velocities and accelerations. Furthermore, it incorporates an interface to external code written in other languages like Python or Matlab, thus supporting a rapid prototyping environment for low level controllers as well as high level-algorithms. A schematic of the software structure is given in Figure **??**, illustrating the interactions between the involved main components, classes and external communications.

Transmitter and Helicopter

The control signal is sent to the vehicles via transmitters, connected to the USB ports of the PC running the Coaga software. The system relies on the Spektrum DSM2 protocol, a $2.4\,\mathrm{GHz}$ direct sequence spread spectrum (DSSS) system, which is also compatible with DSMx receivers and therefore widespread in commercially available remote controlled devices. Ranging from different types of helicopters to quadrocopters, this infrastructure allows for diverse systems to be modeled and controlled. In this work, the Blade mCX2 coaxial helicopter is used, which is $20\,\mathrm{cm}$ long, has a rotor diameter of $19\,\mathrm{cm}$ and a weight of $28\,\mathrm{g}$. It has two linear servos to control the cyclic swash-plate movement, thus allowing for roll, pitch, yaw and thrust control actions. In addition, it comes with a Bell-type stabilizer bar and an on board heading hold gyroscope for yaw angle stabilization. Since the gyro can not be turned off in this model the yaw control will not be in the focus of this work. No additional sensors were added, except for a retrofitted bar carrying infrared markers.

III. Helicopter Model

Our MPC scheme uses a discrete time linear model for the helicopters dynamics. In this section the derivation from [11] is quickly recalled and extended.

a) Nonlinear model: The dynamical model is based on Newton-Euler mechanical laws for rigid bodies, which are fully described in, e.g., [17]. We obtain a reduced-order model by neglecting the pitch and roll angular states, which represents the observed behavior of the coaxial helicopter with flybar quite well.

Consider the nonlinear continuous-time system $\dot{\mathbf{x}} = f(\mathbf{x},\mathbf{u})$ with input vector $\mathbf{u} = \left[u_x,\, u_y,\, u_z,\, u_\Psi\right]$, the state vector $\mathbf{x} = \left[x_I,\, y_I,\, z_I,\, \psi,\, \dot{x}_B,\, \dot{y}_B,\, \dot{z}_B,\, \dot{\psi},\, x_{int},\, y_{int},\, z_{int}\right]$ and the function $f(\cdot,\cdot) : \mathbb{R}^{11}\times\mathbb{R}^4 \rightarrow \mathbb{R}^{11}$, which is given by the state dynamics (1)

$$\begin{aligned}
\dot{x}_I &= \cos(\Psi)\dot{x}_B - \sin(\Psi)\dot{y}_B, & \dot{z}_I &= \dot{z}_B,\\
\dot{y}_I &= \sin(\Psi)\dot{x}_B + \cos(\Psi)\dot{y}_B, & \dot{\Psi} &= \dot{\Psi}\\
\ddot{x}_B &= b_x u_x + k_x\dot{x}_B + \dot{\Psi}\dot{y}_B, & \ddot{z}_B &= b_z u_z - g\\
\ddot{y}_B &= b_y u_y + k_y\dot{y}_B - \dot{\Psi}\dot{x}_B, & \ddot{\Psi} &= b_\Psi u_\Psi + k_\Psi\dot{\Psi}\\
\dot{x}_{int} &= k_i(x_I - x_{\mathrm{ref}}), & \dot{z}_{int} &= k_i(z_I - z_{\mathrm{ref}})\\
\dot{y}_{int} &= k_i(y_I - y_{\mathrm{ref}})
\end{aligned} \tag{1}$$

Here, x_I, y_I, z_I denote the position of the helicopter in the inertial frame, $\dot{x}_B$, $\dot{y}_B$, $\dot{z}_B$ the velocities of the helicopter in the body frame and Ψ the yaw heading angle with its rotational velocity $\dot{\Psi}$. The body frame coordinate system of the helicopter is fixed to the helicopter center of gravity and x_B-axis pointing towards the nose and z_B-axis along the rotor axis. The model parameters are k_x, k_y, k_Ψ, k_i and the control inputs for pitch, roll, thrust and yaw are denoted by u_x, u_y, u_z, u_Ψ. In extension of [11] a third integral state z_{int} is augmented to the reduce steady-state error now in all three spatial dimension.

b) Discrete-Linear Model: A linear approximation of (1) is derived by the first-order terms of the Taylor series expansion around the set point $(\hat{\mathbf{x}}(t), \hat{\mathbf{u}}(t))$. The system is zero-order hold sampled with time $\mathrm{T_s}$ auch that the discrete representation is given by

$$\mathbf{x}_{k+1} = \mathbf{A}_{d,k}\mathbf{x}_k + \mathbf{B}_d\mathbf{u}_k - \mathbf{c}_{d,k}, \tag{2}$$

where the subindex k denotes states or matrices evaluated at $t = kT_s$. $\mathbf{A}_{d,k}$ is the first-order approximation of the system matrix, $\mathbf{B}_d$ the Euler-Integrations of the input matrix and

$\mathbf{c}_{d,k}$ the Euler-Integrations of $\mathbf{c}_k = \mathbf{A}_k\hat{\mathbf{x}}(k\mathrm{T_s})+\mathbf{B}\hat{\mathbf{u}}(k\mathrm{T_s})-\dot{\hat{\mathbf{x}}}(k\mathrm{T_s})$. Previous experiments [11] showed no significant benefit, when using more precise integration methods.

c) Hovering Simplification: In view of the formation control task presented in Section VI-C, the helicopter is assumed to be always in hovering mode, i.e., the model is linearized around set points with zero velocities. Experiments showed that this assumption is reasonable for set point changes that are not too large. The final model used for control is given as

$$\mathbf{A}_{d,k}(\hat{\Psi}_k) = \mathbf{I}^{11\times 11} +$$
$$\begin{bmatrix} 0&0&0&0&c_{\hat{\Psi}_k}&-s_{\hat{\Psi}_k}&0&0&0&0&0\\ 0&0&0&0&s_{\hat{\Psi}_k}&c_{\hat{\Psi}_k}&0&0&0&0&0\\ 0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&k_x&0&0&0&0&0&0\\ 0&0&0&0&0&k_y&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&k_\Psi&0&0&0\\ k_i&0&0&0&0&0&0&0&0&0&0\\ 0&k_i&0&0&0&0&0&0&0&0&0\\ 0&0&k_i&0&0&0&0&0&0&0&0 \end{bmatrix} \cdot \mathrm{T_s},$$

$$\mathbf{B}_d = \begin{bmatrix} & \mathbf{0}^{4\times 4} & & \\ b_x\mathrm{T_s}&0&0&0\\ 0&b_y\mathrm{T_s}&0&0\\ 0&0&b_z\mathrm{T_s}&0\\ 0&0&0&b_\Psi\mathrm{T_s}\\ & \mathbf{0}^{3\times 4} & & \end{bmatrix}, \mathbf{c}_{d,k} = \begin{bmatrix} \mathbf{0}^{6\times 1}\\ g\mathrm{T_s}\\ 0\\ k_i x_{\mathrm{ref,k}}\mathrm{T_s}\\ k_i y_{\mathrm{ref,k}}\mathrm{T_s}\\ k_i z_{\mathrm{ref,k}}\mathrm{T_s} \end{bmatrix}$$

where c_Ψ and s_Ψ denote the cosine and sine of Ψ. $\mathbf{I}^{n\times m}$ and $\mathbf{0}^{n\times m}$ are the identity matrix and zero matrix respectively in $\mathbb{R}^{n\times m}$. The derived model is time dependent through the yaw-angle set point $\hat{\Psi}_k$ and the position reference $\begin{bmatrix} x_{\mathrm{ref},k} & y_{\mathrm{ref},k} & z_{\mathrm{ref},k}\end{bmatrix}$. The treatment of these dependencies will be discussed later. Also, the identified model parameters are provided in Section VI-A.

IV. Single Helicopter MPC

A standard MPC scheme [16] is applied, where the controller solves at each time instance k a constrained multi-stage optimization problem over the horizon $i = [k, \cdots, k+N-1]$. From the obtained optimal input sequence, only the first input is applied to the system $u_k = u_1^*$. At the next step the horizon is shifted and the problem is solved again with an updated measurement.

Consider a stacked notation of the decision variables $\mathbf{z}_i = \begin{bmatrix}\mathbf{x}_i, \mathbf{u}_i\end{bmatrix}^\top$, $\mathbf{z}_N = \begin{bmatrix}\mathbf{x}_N, \mathbf{0}^{4\times 1}\end{bmatrix}$ and the symmetric cost matrices $\mathbf{Q} \succ 0$ and $\mathbf{R} \succ 0$ to penalize the deviations from the reference states and inputs $\mathbf{x}_{\mathrm{ref}}$ and $\mathbf{u}_{\mathrm{ref}}$ respectively. In addition, the solution $\mathbf{P}$ to the discrete algebraic Ricatti equation (DARE) is used as terminal cost matrix. The optimization problem is then given by

$$\min_{\mathbf{z}_i} \quad \frac{1}{2}\mathbf{z}_N^\top \mathbf{P}_z \mathbf{z}_N - \mathbf{z}_{\mathrm{ref},i}^\top \mathbf{P}_z \mathbf{z}_N + \sum_{i=1}^{N-1} \frac{1}{2}\mathbf{z}_i^\top \mathbf{Q}_{z,i}\mathbf{z}_i - \mathbf{z}_{\mathrm{ref},i}^\top \mathbf{Q}_{z,i}\mathbf{z}_i \qquad (3)$$

$$\text{subject to:} \quad \mathbf{x}_1 = \mathbf{x}$$
$$\mathbf{x}_{i+1} = \mathbf{A}_{d,i}\mathbf{x}_i + \mathbf{B}_d\mathbf{u}_i + \mathbf{c}_{d,i} \quad i = 1,...,N-1$$
$$\mathbf{z}_{\min} \leq \mathbf{z}_i \leq \mathbf{z}_{\max} \quad i = 1,...,N, \qquad (4)$$

where $\mathbf{Q}_z = \mathrm{diag}(\mathbf{Q}, \mathbf{R})$ and $\mathbf{P}_z = \mathrm{diag}(\mathbf{P}, \mathbf{0}^{4\times 4})$ are a block-diagonal matrix of the weight matrices. Note, that for the compact notation, the cost function was multiplied by the factor $\frac{1}{2}$ and constant term were dropped, which does not alter the optimization results. Assuming box constraints $\mathbf{z} \in [\mathbf{z}_{\min}, \mathbf{z}_{\max}]$ on the decision variables and positive definite matrices $\mathbf{Q}$ and $\mathbf{R}$, problem (3) is a convex quadratic and therefore applicable to fast convex solvers. It is well known [16], that under an appropriate choice of the terminal set, stability of the linearized system can be guaranteed. However, for the practical implementation, the derivation of the terminal set is omitted and the constraints are fixed over the whole horizon.

In the previous work [11] for the optimization problem was generated with CVXGEN [14]. Though this implementation was delivering good performance for the fast MPC problem, two main aspects had to be considered in view of a multi-agent formation control task. First, the modular and flexible structure of the RCopterX testbed, e.g., allowing insertion of vehicles at will, is not maintainable with the CVXGEN code, since it relies on global variables in the controller, making it necessary to hard-code the controllers for each helicopter. Second, the problem size is limited by the server time required to generate the problem. We used instead the FORCES code generator [5]. A complete comparison between the solvers is beyond the scope of this paper; however, the achieved solving times for the single helicopter problem with various horizon length are discussed in Section VI-C.

An important observation from simulations with FORCES was, that the number of iteration needed and therefore the solving times could be reduced significantly when the data was scaled such that all inputs and states are within the range of -1 to 1. It is straightforward to verify that the problem structure does not change, when applying the transformation

$$\tilde{\mathbf{z}} = \begin{bmatrix}\tilde{\mathbf{x}}\\ \tilde{\mathbf{u}}\end{bmatrix} = \mathbf{D}^{-1}(\mathbf{z}-\mathbf{s}) = \begin{bmatrix}\mathbf{D}_x^{-1}(\mathbf{x}-\mathbf{s}_x)\\ \mathbf{D}_u^{-1}(\mathbf{u}-\mathbf{s}_u)\end{bmatrix}, \qquad (5)$$

and dropping all constant terms in the cost function. The involved matrices become

$$\tilde{\mathbf{Q}}_{z,i} = \mathbf{D}^\top\mathbf{Q}_{z,i}\mathbf{D},\ \ \tilde{\mathbf{P}}_z = \mathbf{D}^\top\mathbf{P}_z\mathbf{D},\ \ \tilde{\mathbf{A}}_{d,i} = \mathbf{D}_x^{-1}\mathbf{A}_{d,i}\mathbf{D}_x,$$
$$\tilde{\mathbf{c}}_{d,i} = \mathbf{D}_x^{-1}(\mathbf{A}_{d,i}\mathbf{s}_x + \mathbf{B}_d\mathbf{s}_u + \mathbf{c}_{d,i} - \mathbf{s}_x),\ \tilde{\mathbf{B}} = \mathbf{D}_x^{-1}\mathbf{B}\mathbf{D}_u,$$

and the inequality (4) becomes $-1 \leq \tilde{\mathbf{z}}_i \leq 1$. The implemented values for the diagonal scaling matrix $\mathbf{D} \in \mathbb{R}^{15\times 15}$ and for the shift-vectors $\mathbf{s} \in \mathbb{R}^{15}$ ($\mathbf{s}_x \in \mathbb{R}^{11}$, $\mathbf{s}_u \in \mathbb{R}^4$) are provided in Section VI-B.

V. A Leader-Follower MPC Scheme for Formation Control

We implemented a leader-follower model predictive formation control scheme, in which global reference tracking is delegated to a single leader vehicle and the relative shape is maintained by the other vehicles by tracking a modified reference based on the predicted trajectory of the leader. The main objective of the algorithm is to maintain the relative positions of follower helicopters with respect to their designated leader, irrespective of planned or disturbed movements of the leader. In this way, a global optimization problem with terms in the objective coupling neighboring vehicles is solved by distributing the coupling terms amongst the followers. The desired relative state between two generic helicopters H_L and H_F is expressed by an offset vector $\mathbf{d}_{(L,F)} \in \mathbb{R}^{11}$. Every helicopter is assumed to measure its own global position and to receive information about the relative state and predicted trajectory of the vehicles it is following via a communication link. To simplify the exposition and implementation, every follower is assumed to follow only a single vehicle, though the scheme described here can in principle be modified to handle any directed acyclic communication graph.

The leader tracks a global reference using the MPC scheme described in Section IV and obtains at each time step k the optimal solution $\left[x_i^*,\, u_i^*\right]$ for $i = k, ..., k+N-1$ to the problem (3). This optimized predicted trajectory x_i^*, referred to as the *plan*, is communicated to the connected followers, which track a transformed version of this plan independently of the global reference or the other followers. The nominal reference at horizon step i for follower H_F of leader H_L is then given by

$$\mathbf{z}_{\mathrm{ref},(F),i} = \begin{bmatrix} \mathbf{\Omega}(\phi_{(L,F),i})\mathbf{x}^*_{(L),i} + \mathbf{d}_{(L,F)} \\ \mathbf{u}_{\mathrm{ref},(F),i} \end{bmatrix}. \tag{6}$$

The input reference is not altered, but for the state reference, the distance $\mathbf{d}_{(L,F)}$ is added to the rotated plan $\mathbf{x}^*_{(L),i}$ of the leader to introduce the desired offset. The rotation matrix $\mathbf{\Omega}$ maps the variables given in the body frame coordinate system of the leader into the followers frame. For the model derived in Section III, where pitch and roll angles are neglected, only the two translational velocities $\dot{x}_B$ and $\dot{y}_B$ need to be rotated. Thus, the rotation is

$$\mathbf{\Omega}(\phi_{(L,F),i}) = \begin{bmatrix} \mathbf{I}^{4\times 4} & \mathbf{0}^{4\times 2} & \mathbf{0}^{4\times 4} \\ \mathbf{0}^{2\times 4} & \begin{matrix} c_{\phi_{(L,F),i}} & s_{\phi_{(L,F),i}} \\ -s_{\phi_{(L,F),i}} & c_{\phi_{(L,F),i}} \end{matrix} & \mathbf{0}^{5\times 2} \\ \mathbf{0}^{5\times 4} & \mathbf{0}^{5\times 2} & \mathbf{I}^{5\times 4} \end{bmatrix},$$

with the relative orientation $\phi_{(L,F),i} = \Psi_{(L),i} - \Psi_{(F),i}$. Note that we assume the angles $\Psi_{(L),i}$ are communicated and therefore known to the follower. Practical choices for the required follower orientations will be discussed in Section VI-C.

Scaling: Since the optimization problems are scaled, as described in the previous section, and since in general two helicopters could have different scaling ranges and shift-vectors $(\mathbf{D}_L, s_L)$ and $(\mathbf{D}_F, s_F)$, the leader plan may require descaling, rotation, and rescaling. The scaled reference for the follower is then given by

$$\begin{aligned} \tilde{\mathbf{x}}_{\mathrm{ref},i} &= \underbrace{\mathbf{D}_{x,F}^{-1}\mathbf{\Omega}\mathbf{D}_{x,L}}_{\tilde{\mathbf{\Omega}}}\tilde{\mathbf{x}}^*_{(L),i} \\ &\quad + \underbrace{\mathbf{D}_{x,F}^{-1}\mathbf{d}_{(L,F)} + \mathbf{D}_{x,F}^{-1}(\mathbf{\Omega} s_{x,L} - \mathbf{s}_{x,F})}_{\tilde{\mathbf{d}}_{(L,F)}} \end{aligned} \tag{7}$$

There are three main advantages of this approach. First, the followers use both the current relative state and plan of the leader, allowing them to anticipate leader motion to provide better stabilization. Second, the leader-follower structure allows coupling terms in the objective to be straightforwardly distributed amongst the followers so that each follower can independently and in parallel optimize its own decision variables. No iterative negotiation mechanism is required, and the formation control task can be executed at the same rate as for a single vehicle, with all plans being updated every few milliseconds. Third, since the optimization problem structure is the same for follower and leader, the same generated code can be used. This facilitates changes to the formation structure, e.g., re-assignment of followers to other leaders or changing the leader of the formation.

VI. Experimental Results

A. Basic Parameter Settings

For the results presented in this paper, the following model parameters are taken from [11] and used for all helicopters. To match the update rate of the transmitter, the sampling time for the discrete system was set to $\mathrm{T_s} = 20\,\mathrm{ms}$.
The time-varying system matrix $\mathbf{A}_{d,k}(\hat{\Psi}_k)$ was set constant over the whole horizon, by linearizing around the desired operational point, i.e., the reference orientation $\hat{\Psi}_i = \Psi_{\mathrm{ref},k}$. This approach showed better performance than a linearization around the actual state, although the model approximation at the beginning of the horizon might be worse for large deviations from the reference orientation. This is likely to be attributed to the use of the terminal cost, which acts on the final state, where the approximation is good for sufficiently long horizons. Note that this is done to simplify the implementation, but in principle one can use time-varying trajectory tracking controllers as in [11]; we leave this for future work. Furthermore, the vector $\mathbf{c}_{d,k}$ in (2) depends on the reference as well, such that the dynamics are only re-linearized if the reference changes.

The controller for the MPC problem (3) was set up with the cost functions $\mathbf{R} = \mathrm{diag}\left(\left[5,\, 5,\, 5,\, 1\right]\right)$ and $\mathbf{Q} = \mathrm{diag}\left(\left[8,\, 8,\, 8,\, 4,\, 1,\, 1,\, 1,\, 1,\, 1,\, 1,\, 1\right]\right)$, where $\mathrm{diag}(\cdot)$ denotes the entries on the diagonal of a matrix with appropriate dimension. The constraints were chosen based on the physical bounds of the real helicopter and actuator limits, determined in [4]. Further, the spatial dimensions were limited as well, resulting in constraints for $\mathbf{x}_{\min} = \left[-1.2,\, -1.2,\, 0,\, -1,\, -1.3,\, -1,\, -25\right]$, $\mathbf{x}_{\max} = \left[1.2,\, 1.2,\, 2,\, 1,\, 1.3,\, 1,\, 25\right]$, $\mathbf{u}_{\min} = \left[-1,\, -1,\, 0,\, -0.4\right]$, $\mathbf{u}_{\max} = \left[1,\, 1,\, 1,\, 0.4\right]$.

In general the scaling of the variables is based on the ranges of the state and input constraints. However, not all states necessarily are constrained, hence the scales for these unconstrained variables were chosen such that they have approximately the same magnitude, based on empirical observations. The shift-vectors are selected as $\mathbf{s_x} = \left[0,\, 0,\, 1\,\mathrm{m},\, 0,\, 0,\, 0,\, 0,\, 0,\, 0,\, 0,\, 0\right]^\top$, $\mathbf{s_u} = \left[0,\, 0,\, 0.5,\, 0\right]^\top$, and the scaling matrix $\mathbf{D} = \mathrm{diag}(\left[1.2\,\mathrm{m},\, 1.2\,\mathrm{m},\, 1\,\mathrm{m},\, 10\,\mathrm{rad},\, 1.1\frac{\mathrm{m}}{\mathrm{s}},\, 1.3\frac{\mathrm{m}}{\mathrm{s}},\, \cdots\right.$ $\left.1\frac{\mathrm{m}}{\mathrm{s}},\, 25\frac{\mathrm{rad}}{\mathrm{s}},\, 1\,\mathrm{ms},\, 1\,\mathrm{ms},\, 1\,\mathrm{ms},\, 1,\, 1,\, 0.5,\, 0.4\right])$. The set point $\hat{u}_z = 0.55$ was chosen from experiments to keep the helicopters hovering at the same height, which is the reason for the entry in the shift vector and the limited input range. Furthermore, the nominal operation height z of the helicopters was shifted to $1\,\mathrm{m}$.

B. Single Helicopter

A comparison to the controller performance of [11], using the settings from Section VI-A, is not directly possible due to the additional use of a terminal cost in this work. However, using the previously reported weights, we can see in Figure 1, that the hovering performance for 18-step horizon problems is comparable and that the additional z-integrator has the expected impact.

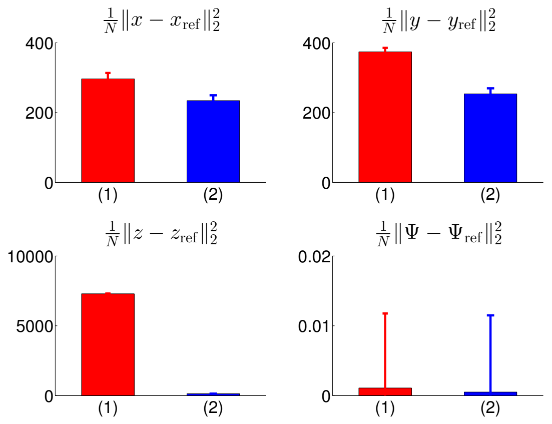

Fig. 1: Mean square error [in mm^2 and rad^2] of 3 independent hovering flights of $2\,\mathrm{min}$. *Red bars (1):* Implementation of [11]. *Blue bars (2):* Presented implementation with terminal cost and z-integrator.

The code implementation was optimized in a way such that the computation times of the control inputs are dominated by the solver time. Figure 2 shows the controller times for different horizons, where the maximal length considered was 110 steps. Though the average time for this case is $18\,\mathrm{ms}$ and still below the sampling time, the solver occasionally exceeded the $20\,\mathrm{ms}$, when tested with bad initial conditions or heavy disturbances. Measuring the error for a sequence of reference changes over the different horizons reveals the limits of the linearized and simplified model, as illustrated in Figure 2.

C. Formation Control

For the validation of the formation control approach presented in Section V, the following three implementation aspects were addressed. Note that the first two of these are implementation issues arising from the use of set point controller instead of trajectory tracking controllers, done to simplify and achieve fast computation times on the available resources.

Linearization: In the single MPC scheme the required linearization and recalculation of (2) was done only if changes in the reference occurred. From the follower perspective this happens every steps, since the plan of the leader is in general not constant. Again, for computational reasons the dynamics were linearized at each step and then

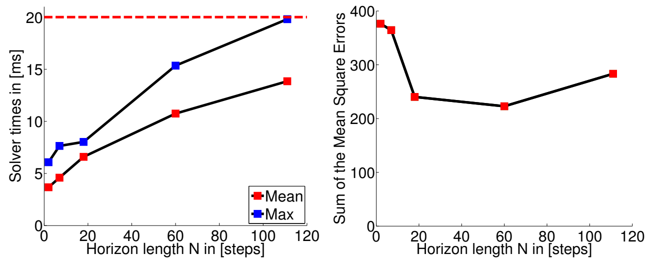

Fig. 2: *Left:* The curve with red squares indicate the mean solver times while the blue squares show the maximal encountered times. The results are obtained from 3 hovering flights of $45\,\mathrm{s}$ for each horizon length. The red dashes lines depicts the sampling time of the system. *Right:* Weighted sum of the mean square errors without integral states. The weights have been chosen such that a deviation of approximately $0.3\,\mathrm{rad}$ has the same weight as a deviation of $10\,\mathrm{mm}$. Here, every $4\,\mathrm{s}$ the position reference $(x_{\mathrm{ref}}, y_{\mathrm{ref}}, z_{\mathrm{ref}}, \Psi_{\mathrm{ref}})$ was changed along the points $(300, 0, 1400, 0)$, $(300, 300, 1400, 0)$, $(-500, 300, 1400, 0)$, $(-500, 300, 1000, 0)$, $(-500, 300, 1000, 3\pi)$. The sequence was flown three times for each horizon length.

kept constant over the horizon. Another change to the single MPC is, that the angular set point for the linearization has to be chosen differently. Not knowing the global reference, the follower should linearize around the best estimate of a future operation point it knows. Under the assumption of a sufficiently long optimization horizon, the last state in the leaders optimized plan will almost coincide with the global reference point. Therefore the shifted terminal state vector $\mathbf{x}_F^p = \mathbf{d}_{(L,F)} + \mathbf{x}^*_{(L),N}$ is used as the linearization point for the followers, i.e., $\hat{\Psi}_{(F),i} = \Delta\Psi_{(L,F)} + \Psi^*_{(L),N}$, where Δ denotes the components of $\mathbf{d}_{(L,F)}$. Similarly $c_{d,i}$ is calculated based on the first three components of $\mathbf{x}_F^p$.

Fixed Rotation: As mentioned earlier in Section V, the computation of the rotation matrix $\mathbf{\Omega}(\phi_{(L,F),i})$ requires knowledge of the follower orientations $\Psi_{(F),i}$ over the optimization horizon. One approach would be to start with an initial guess for the rotation matrices and then iteratively solve the optimization problem, to obtain a nominal orientation trajectory, and refine the rotation matrices. We chose a computational less demanding implementation by keeping $\mathbf{\Omega}$ constant over the whole horizon and calculate it with the desired orientation $\phi_{(L,F)} = \Delta\Psi_{(L,F)}$.

Homogeneous group: The group of agents is considered to be homogeneous with the same parameters and scaling. Furthermore, the shift-vector in (VI-A) does not affect the velocities, hence, since $\mathbf{\Omega}\mathbf{s}_x - \mathbf{s}_x = 0$, the last term of $\tilde{\mathbf{d}}_{(L,F)}$ in (7) becomes zero.

The approach was successfully implemented and tested with a formation of three helicopters, with the formation leader H_1 and the two direct followers H_2 and H_3. In this experiment, we only assign values to the first 4 possible formation distances $[\Delta x, \Delta y, \Delta z, \Delta\Psi]$ and set all other to zero. Figure 3 shows the results of a $5\,\mathrm{min}$ experiment, where several set-point changes were applied to the global reference as well as to the controlled distances $\mathbf{d}_{(1,2)}$ and $\mathbf{d}_{(1,3)}$. It can be seen, that not only the controlled distances are tracked very well, but also the distance $\mathbf{d}_{(2,3)}$ between the followers, even though it is not explicitly present in the problem objective. In general the deviations from the desired distances are not more than $20\,\mathrm{cm}$, which is about the length of the helicopters, since the follower anticipate the movements of the leader through the communicated plan. It is interesting to observe, that the largest tracking errors occur when reference changes of orientation are applied. Though the follower track changes in their nominal yaw distance well, they visibly deviate from their nominal distances for changes in the global yaw reference. However, this is not surprising because of the implementation choices described above. The linearization of the followers model depend on the projected orientations of the leader and the rotation matrix on their perfect relative orientation. While these simplifications reach their limit in case of large step changes, experiments showed a good performance for continuously varying changes. These results suggest the use of time varying matrices, as already proposed in [11], to further improve the movements of the whole formation.

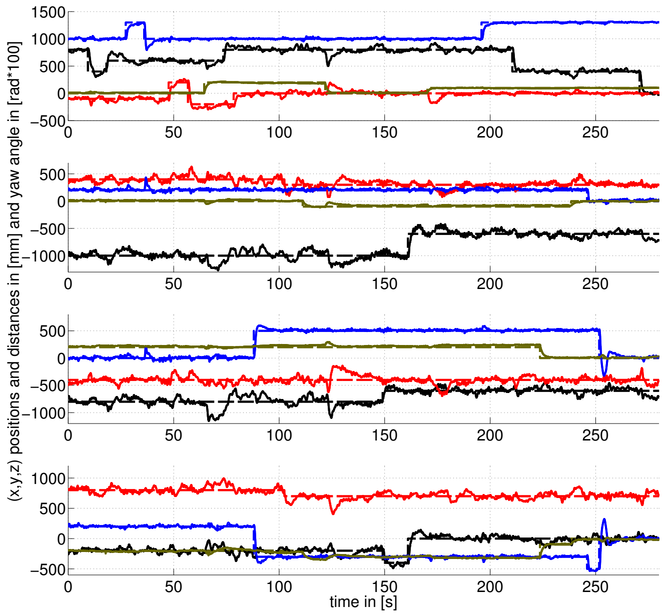

Fig. 3: The plots show the tracking performance of the formation for set point changes in the reference and nominal distances. *Top:* State trajectory of the leader and global formation reference. *Middle plots:* Distances from the two followers to the leader and their controlled distances. *Bottom:* Distance between the followers and the implicitly defined distance $\mathbf{d}_{(2,3)}$. The dashed lines represent the reference and nominal distances whereas the solid the leaders trajectory and the achieved distances respectively. The x-component of position and distances are shown in black, the y-component in red and z-components in blue. Note that the Ψ-components (green) are in radians but have been magnified by the factor 100 for visualization purpose.

VII. Conclusion

The setup of the RCopterX miniature helicopter testbed was presented, which allows for a wide range of experiments. The main advantages are its modular structure, the communication which makes it possible to incorporate different kinds of vehicles and the interchangeability of controllers during experiments. The successful implementation of a fast MPC scheme as low level control was described, which enables multi-agent experiments. Finally the implementation and experimental validation of a distributed formation control algorithm was shown. The relative simple structure of the algorithm keeps the computational demands low, while providing a flexible and robust leader-follower scheme.

References

[1] J. Amstuz, R. Bernhard, T. Wellerdieck, and F. Wermelinger. *Realization of Multi-Agent Support*. Group project, Automatic Control Laboratory, ETH Zurich, Zurich, Switzerland, 2012.

[2] E. Camponogara, D. Jia, B.H. Krogh, and S. Talukdar. Distributed model predictive control. *IEEE Control Systems*, 22(1):44–52, 2002.

[3] C. Conte, N.R. Voellmy, M.N. Zeilinger, M. Morari, and C.N. Jones. Distributed synthesis and control of constrained linear systems. In *American Control Conference (ACC), 2012*, pages 6017–6022, 2012.

[4] D. Daellenbach and V. Semeraro. *System Identification Routine for Miniature Helicopters*. Semester thesis, Automatic Control Laboratory, ETH Zurich, Zurich, Switzerland, 2012.

[5] A. Domahidi, A.U. Zgraggen, M.N. Zeilinger, M. Morari, and C.N. Jones. Efficient interior point methods for multistage problems arising in receding horizon control. In *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 668–674, 2012.

[6] William B. Dunbar and Richard M. Murray. Distributed receding horizon control for multi-vehicle formation stabilization. *Automatica*, 42(4):549–558, April 2006.

[7] S. Han, A.D. Straw, M.H. Dickinson, and R.M. Murray. A real-time helicopter testbed for insect-inspired visual flight control. In *IEEE International Conference on Robotics and Automation, 2009. ICRA '09*, pages 3055–3060, 2009.

[8] J.P. How, B. Bethke, A. Frank, D. Dale, and J. Vian. Real-time indoor autonomous vehicle test environment. *IEEE Control Systems*, 28(2):51–64, 2008.

[9] Huck et al. (2013, sept. 24.). Project RCopterX. [Online]. Available: http://www.rcopterx.ethz.ch/.

[10] T. Keviczky, F. Borrelli, K. Fregene, D. Godbole, and G.J. Balas. Decentralized receding horizon control and coordination of autonomous vehicle formations. *IEEE Transactions on Control Systems Technology*, 16(1):19–33, 2008.

[11] K. Kunz, S.M. Huck, and T.H. Summers. Fast model predictive control of miniature helicopters. In *European Control Conference*, Zurich, Switzerland, July 2013.

[12] Y. Kuwata and J.P. How. Cooperative distributed robust trajectory optimization using receding horizon MILP. *IEEE Transactions on Control Systems Technology*, 19(2):423–431, 2011.

[13] Y. Kuwata, A. Richards, T. Schouwenaars, and J.P. How. Distributed robust receding horizon control for multivehicle guidance. *IEEE Transactions on Control Systems Technology*, 15(4):627–641, 2007.

[14] J. Mattingley and S. Boyd. CVXGEN: a code generator for embedded convex optimization. *Optimization and Engineering*, 13(1):1–27, March 2012.

[15] Nathan Michael, D. Mellinger, Q. Lindsey, and V. Kumar. The GRASP multiple micro-UAV testbed. *IEEE Robotics Automation Magazine*, 17(3):56–65, 2010.

[16] J.B. Rawlings and D.Q. Mayne. *Model predictive control: theory and design*. Nob Hill Pub., Madison, Wis., 2009.

[17] D. Schafroth, C. Bermes, S. Bouabdallah, and R. Siegwart. Modeling, system identification and robust control of a coaxial micro helicopter. *Control Engineering Practice*, 18(7):700–711, 2010.

[18] B.T. Stewart, A.N. Venkat, J.B. Rawlings, S.J. Wright, and Gabriele Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8):460–469, August 2010.

[19] T.H. Summers and J. Lygeros. Distributed model predictive consensus via the alternating direction method of multipliers. In *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 79–84, 2012.