# UAV Motion Planning and Control for Multi-Coverage of 3D Environments

Matthias Weyrer*
Institute for Robotics and Mechatronics
Joanneum Research, Klagenfurt, Austria
matthias.weyrer@joanneum.at

Bernhard Rinner
Institute of Networked and Embedded Systems
Alpen-Adria-Universität Klagenfurt, Austria
bernhard.rinner@aau.at

***Abstract*—Multi-coverage (MC) represents an important problem for various tasks of unmanned aerial vehicles (UAVs). For the multi-coverage of a given environment, all visible surfaces must be covered by simultaneously captured images from at least $k$ different viewpoints where additional constraints on resolution, geometry and error may be imposed. In this paper we formulate the MC problem for robot missions and propose a model-predictive control structure to navigate a stereo-pair of UAVs in a leader-follower formation throughout the mission. Our simulation study demonstrates that the UAVs reach and maintain the formation with high spatial and temporal accuracy.**

***Keywords*—Multi-camera coverage; unmanned aerial vehicles (UAV); motion planning; model predictive control; leader-follower formation**

## I. Introduction

Unmanned aerial vehicles (UAVs) with onboard cameras are increasingly deployed in various applications including surveying, search and rescue, inspection as well as video production. In all these applications, the UAVs move (in many cases autonomously) to capture imagery of sufficient quality of the environment or objects of interest (e.g. [1]–[3]). In this paper, we focus on UAV mission planning and execution for *multi-coverage (MC)* which represents an important problem for surveying, surveillance and reconstruction tasks. For multi-coverage, all visible surfaces of a given environment must be covered by simultaneously captured images from at least $k$ different viewpoints. In order to solve the MC problem (cp. Figure 1), we first need to compute all camera viewpoints (or constellations), i.e., the position and orientation of $k$ cameras satisfying various constraints, and then to plan and execute the paths for the UAVs such that all constellations are visited. A prominent instance of the MC problem is stereo coverage with flexible baselines where two UAVs simultaneously cover the environment.

Our contribution is threefold. First, we introduce and formulate the MC problem for robot missions, which is distinct to the well-known problem of multi-robot coverage (e.g., [4]) due to its requirement on simultaneous image capturing. Second, we propose a method for computing the constellations and planning the UAV mission for the stereo case. Third, we realize a distributed, model-predictive controller for UAV motion execution and evaluate its performance for stereo coverage in a 3D simulation environment. This paper complements our previous work on multi-UAV systems[1], networks [5], [6] and coordination [7], [8] with a clear focus on path planning and motion control.

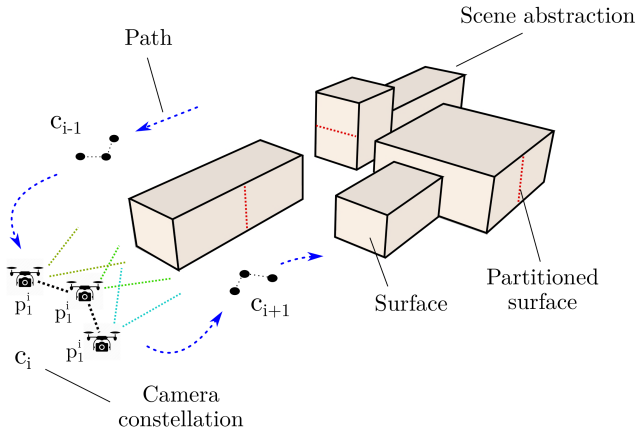


Fig. 1. Sketch of the MC problem. A team of UAVs autonomously flies through the environment such that each constellation $c_i$ is visited and the corresponding surface is covered by simultaneously captured images from $k$ UAVs.

The remainder of the paper is organized as follows. Section II discusses related work. Section III formulates the MC problem and Section IV describes our methods for constellation planning, path planing and control strategy for the stereo coverage case. Section V summarizes our simulation results and Section VI concludes this paper with a discussion about future work.

## II. Related Work

There are a couple of publications available that suggest to use multiple cameras to create stereo-pairs with different baselines generated by lateral displacements. Gallup et al. [9] present a multi-baseline, multi-resolution system that selects the displacement parameters to maintain a constant depth accuracy within the 3D reconstruction process. They investigate the depth error caused by pixel quantization and show, that selecting a proper baseline can significantly improve the accuracy of depth estimates while keeping the

[1] http://uav.aau.at

computational effort low by reducing the resolution. Matching performance is evaluated empirically and the algorithm is only tested within triangulation angles where the matching rate is sufficiently high. In 2005 Nakabo et al. [10] created a variable baseline stereo vision system that operates in real-time and uses two pan-tilt-cameras on linear sliders. They investigate the accuracy of 3D position estimates by considering the angular error caused by pixel quantization.

A set of stereo-camera parameters can also be interpreted as a formation of two cameras. A key challenge for a practical implementation is the determination of the relative pose of the cameras. This can be achieved with external positioning systems which imposes limitations on the applicability. An alternative approach is based on vision with two main techniques: In the first technique, one camera has a direct line of sight to the other camera and uses a known calibration pattern, markers or geometric properties for relative pose estimation [11], [12]. The second technique exploits the shared FoV of both cameras and uses matched keypoints of the two images to determine the relative orientation and direction of translation. This technique is also known from uncalibrated stereo algorithms.

Additional sensors, however, are needed to recover the scale factor. Achtelik et al. [13] fuse the vision results with readings from the inertial measurement unit (IMU) to compute a metric distance information. They include the scale factor in the state vector and show that their filter converges if there is sufficient relative acceleration. However, such relative acceleration requirement poses a limitation when aiming for a stable and synchronized formation flight. Piasco et al. [14] follow a similar approach but they recover the scale factor with the absolute altitude information measured by sonar sensors. Montijano et al. [15] use the IMU-readings to estimate roll and pitch, remove their effects from the image and determine the third rotation angle using a structure from motion approach. They show that their distributed controller converges to the desired formation up to scale.

To the best of our knowledge, this paper is the first work focusing on motion planning and control of optimal multi-coverage constellations for quad-copter UAVs.

## III. Problem Description

Figure 1 depicts the MC problem where the environment is abstracted by simple cuboids. Thus, all visible surfaces of the cuboids represent the overall coverage area. For each surface, a constellation of $k$ UAVs is computed satisfying various constraints, e.g., on the pixel resolution, depth error and UAV distances. In order to solve the MC problem, a team of $d \geq k$ UAVs must visit all constellations where $k$ UAVs capture simultaneously images of the corresponding surface.

We assume that each UAV has an onboard processing and communication unit with sufficient performance and is equipped with two cameras: a downward-facing camera that is used for relative localization and a camera for capturing the scene. Additionally, the UAVs are synchronized and are able to measure the distances between them in metric units (e.g., with onboard Ultra Wideband (UWB) modules). At least one UAV is able to navigate within the environment.

Solving the MC problem can be partitioned into three subproblems: First, all constellations must be determined to satisfy the constraints. A constellation is defined as the 3D positions and orientations of $k$ cameras to capture the synchronous multi-view shot of a particular surface. Second, routes for all UAVs must be planned such that all constellations are visited and synchronized images are captured. Third, a real-time control strategy must be executed to move the UAVs from one constellation to the next.

### A. Constellation Planning

The environment to be covered is represented by the set of $n \in \mathbb{N}$ visible surfaces of abstracted cuboids $S = \{s_1, ..., s_n\}$. The size of the surface which can be covered by a single constellation is limited by the camera's resolution and aperture angle. In a preprocessing step, surfaces exceeding this limit are partitioned into smaller parts resulting in a new set $S'$ with $m \geq n$ surfaces $S' = \{s'_1, ..., s'_m\}$.

For every surface $s'_i$, a camera constellation $c_i$ must be determined. Thus, we need to find $m$ constellations $C = \{c_1, ..., c_m\}$ to cover the whole scene. Each constellation contains $k$ camera-poses, consequently $c_i = \{p^i_1, ..., p^i_k\}$. with $p^i_j$ representing the pose (i.e., position and orientation) of camera $j$ in constellation $i$.

### B. Mission Planning

Mission planning is concerned with establishing the optimal capturing sequence among all constellations. In general, we must plan the routes of $d \geq k$ UAVs from their initial positions such that each constellation is simultaneously visited by $k$ UAVs (and synchronized multi-view images can be captured). The mission planning objective is to minimize the UAV routes wrt. some criterion (e.g., the overall flight time). The result of mission planning is thus a sequence of poses $p^i_j$ for each UAV. In the general case $d > k$, not every UAVs must visit every constellation.

### C. Mission Execution

Mission execution is concerned about sequentially navigating the UAVs to their desired constellation poses starting from their initial positions. When a UAV reaches pose $p^i_j$, it hovers there waiting for the remaining $k - 1$ UAVs of constellation $c_i$, performs the synchronized image capturing and moves then to the next pose in its corresponding mission plan. This is repeated until all camera constellations have been visited. Thus, a real-time control strategy for navigating the UAVs from one pose to the next pose location is required. Low-level control for stabilization and tracking a velocity or attitude signal is already well investigated [16], [17] as well as the input-output decoupling for quad-copters [18] and therefore not covered in this paper. Instead, we focus on the design of a high-level controller that determines the desired velocities and controls the orientation of heading of the UAVs. Since external positioning systems such as GPS are not always available or rather inaccurate, we explore relative localization of the UAVs resulting in precise relative poses of the constellation but less precise absolute poses due to the uncertainty of the external positioning system.

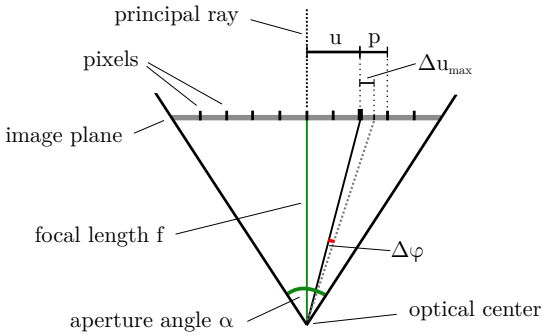


Fig. 2. 1D camera model and maximum angular error $\Delta\varphi(u)$ caused by quantization, with $u$ representing the position on the image plane (in length units) and $p$ the physical distance between two pixels. Focal length $f$, aperture angle $\alpha$ and the resolution $r$ are the camera parameters.

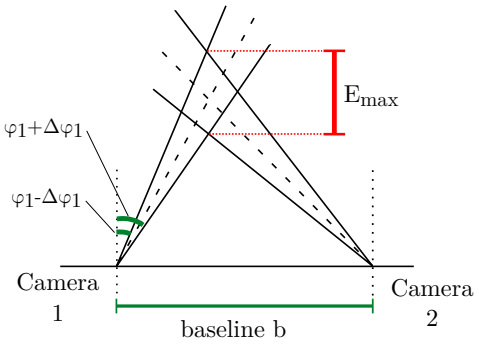


Fig. 3. Worst case depth error $E_{\text{max}}$ defined as the difference between longest and shortest distance estimates caused by the angular errors $\Delta\varphi_i$.

## IV. Methods

In the following we present our solution for the special case where two UAVs are capturing stereo images of the environment, i.e., $d = k = 2$. For the sake of simplicity, we assume only vertical surfaces with constant height and a common ground plane. We can therefore transform the MC problem to 2D environments where each surface $s_i$ can be represented by a line segment.

### A. Stereo Constellation Planning

We restrict the constellation planning to perpendicular oriented cameras and parallel baselines wrt. the considered surface. With the camera model depicted in Figure 2, we compute the maximal length $d_{\text{max}}$ of a captured surface still satisfying the minimum target pixel resolution $\delta$ as

$$d_{\text{max}} = \frac{r}{2\delta \tan\left(\frac{\alpha}{2}\right)} \tag{1}$$

with $\alpha$ as the aperture angle and $r$ the resolution of the camera. Since we are only interested in the overlap of both cameras' FoV for stereo coverage, we can determine the maximum length $l_{\text{max}}$ of any line segment $s_i$ and the maximum length of the baseline $b_{\text{max}}$. With the assumption of a 50 percent overlap, these limits are given as

$$l_{\text{max}} = b_{\text{max}} = d_{\text{max}} \cdot \tan\left(\frac{\alpha}{2}\right) \tag{2}$$

and serve as simple condition for partitioning line segments and transferring $S$ to $S'$ in the preprocessing step.

For each $s'_i \in S'$, a camera constellation $c_i$ is planned by incorporating the camera model (Figure 2), the depth-error model (Figure 3) and a cost function that reflects the matching performance of the downward facing cameras that are used for relative localization. A camera constellation contains $k = 2$ poses, and each pose is specified by the $x$ and $y$ coordinates as well as the orientation, which is the rotation around the $z$-axis.

The simple 1D camera model has three parameters: the focal length $f$, the aperture angle $\alpha$ and the resolution $r$. Since the image sensor has a finite number of pixels, quantization effects may lead to incorrect depth estimates. The worst case angular error $\Delta\varphi(u)$ depends on the position $u$ and the maximum lateral error $\Delta u_{\text{max}}$ on the image plane. As depicted in Figure 2, $\Delta u_{\text{max}}$ is defined as half of the distance $p$ between two pixels and can be calculated as

$$\Delta u_{\text{max}} = \frac{p}{2} = \frac{f \cdot \tan(\frac{\alpha}{2})}{r}. \tag{3}$$

This lateral error is constant over the whole image. The angular error $\Delta\varphi(u)$ is then given by

$$\Delta\varphi(u) = \arctan\left(\frac{u + \Delta u_{\text{max}}}{f}\right) - \arctan\left(\frac{u}{f}\right). \tag{4}$$

We compute the two camera poses, such that the cost function shown in Equation (5) is minimized. Two poses in 2D expose 6 degrees of freedom (two for the positions and one for the orientation of each camera). With the perpendicular orientation and the parallel baseline constraint we can reduce the constellation specification to three parameters and thus reduce the complexity of the optimization task, which is important for fast online planning. Each surface is represented by a line that is defined by its start and end point. We first transform the coordinate system such that the start point lies in the origin and the line is parallel to the positive $x$-axis. Thus, the constellation is given by $x_1, y_1$ and $x_2$.

The next step is to compute the camera poses in the transformed coordinates by performing the optimization that minimizes the cost function

$$J(x_1, y_1, x_2) = w_1 \cdot J_{\text{depth}} + w_2 \cdot J_{\text{match}}(\gamma) \tag{5}$$

where $w_1$ and $w_2$ are weight factors and $J_{\text{depth}}$ represents the aggregated depth error. We implemented $J_{\text{depth}}$ as the mean of the depth error calculated for a number of evenly spread samples across the line. The cost term $J_{\text{match}}$ depends on the relative view angle $\gamma$ of the downward facing cameras. This angle is derived using the height and the distance between the cameras. There are various publications that evaluate the matching performance of different feature types at various view-point conditions. Based on the results of [19] (Table 6.3) we fit a function

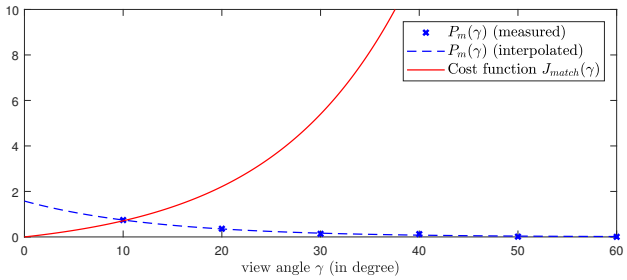


Fig. 4. Graph depicting the measured matching performance (blue crosses), the identified function $P_m(\gamma)$ as given in Equation (6) (blue dashed line) and the cost function $J_{match}(\gamma)$ as given in Equation (7) (red line) at different view angles. The parameters are chosen as $a = 1.576$ and $b = -0.07507$.

$$P_m(\gamma) = a \cdot \mathrm{e}^{(b|\gamma|)} \tag{6}$$

by identifying the parameters $a$ and $b$ that describe matching performance $P_m$ in terms of the view-angle $\gamma$ (see Figure 4). Since the sign of the angle is not relevant, the absolute value is taken. The cost-term is then defined as the inverse of this function shifted down, so that $J_{\text{match}}(0) = 0$ yields. This means that a relative view angle of 0°, consequently two identical images, does not result in any costs:

$$J_{\text{match}}(\gamma) = \frac{1}{a} \cdot \left(e^{-b|\gamma|} - 1\right) \tag{7}$$

Additionally, the solution of the optimization process is constrained since we have bounds on the baseline and the capturing distance and the whole line must be visible in both camera images. The final camera poses are determined by applying the inverse coordinate transformation to the camera positions determined by the optimization and the rotations are computed by adding the fixed 90° to the angle of this transformation (see Figure 5).

### B. Path Planning

Once the camera planning is completed for all $s'_i$, a set $C$ of all constellations is generated. In our special case with $d = k$ all UAVs have to participate to capture a multi-view shot, thus the planning process simplifies from looking for individual routes to finding a common capturing sequence defined as

$$C' = (c'_1, ..., c'_m) \tag{8}$$

which represents a sequence of all $c_i \in C$ satisfying the overall flight time objective.

With the known initial positions of the UAVs, a modified version of the classic Travelling Salesman Problem (TSP) can be used to solve this task. The starting node is fixed as the initial positions of the UAVs and the mission is completed after visiting all desired constellations. Additionally, the pairwise distance is specified as the longer route to get both UAVs from one constellation to the next.

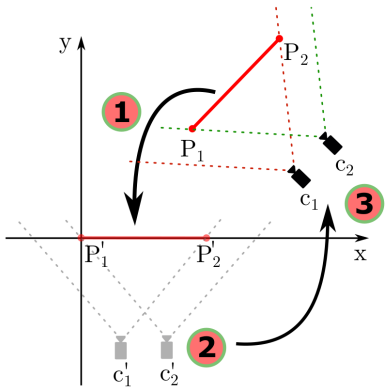


Fig. 5. Camera constellation computation. (1) a coordinate transformation is applied, such that the line starts in the origin and lies on the positive $x$-axis. (2) an optimization problem is solved to determine the camera poses in transformed coordinates. (3) the inverse transformation is applied.

### C. Control Strategy

The set of reference poses can also be seen as a set of formations, with a fixed position and orientation for one camera, and corresponding relative poses for the others. So, instead of treating each team member as an individual UAV that tracks a given trajectory, the problem can be reformulated as a formation flight problem: The team holds a given formation specified by the capturing constellation. Once the capturing process is finished for the actual surface, the UAVs move together to the next constellation and adjust their formation to the new one.

We apply a distributed control structure based on the leader-follower concept that uses *model predictive control (MPC)* techniques. We assume that the leader knows its absolute pose in the scene and the follower can determine its relative pose using the overlapping FoV of the downward-facing cameras and the measured distance (see Figure 6). The absolute pose of the follower is then given as the sum of these two poses. It is important to mention, that a certain amount of overlapping FoV is necessary for the visual relative localization. Inherent to its system design, a quad-rotor has to tilt or pan in order to generate forces for lateral acceleration or deceleration. Consequently, the poses of the mounted cameras vary which could result in loosing the overlap. It is therefore very important, that all formation members move synchronously. In our approach, the follower uses the leader's predicted trajectory to pro-actively set the control commands in order to achieve a synchronous motion. The MPC controller determines the control input based on optimization, therefore a stable formation is possible. In contrast, the output of a conventional controller, such as a PID-controller, is a function of the control error which means, that an error in the formation must occur, before the follower is able to react.

*1) Model:* Our MPC controller acts as a high-level controller, that steers the UAVs by determining the desired velocities. For the modelling we consider the whole quad-copter system as a block that tracks a given velocity signal, consequently it has a reference velocity as input. We assume that the UAV can localize itself, thus it has a measured position as output. We incorporate the dynamic response

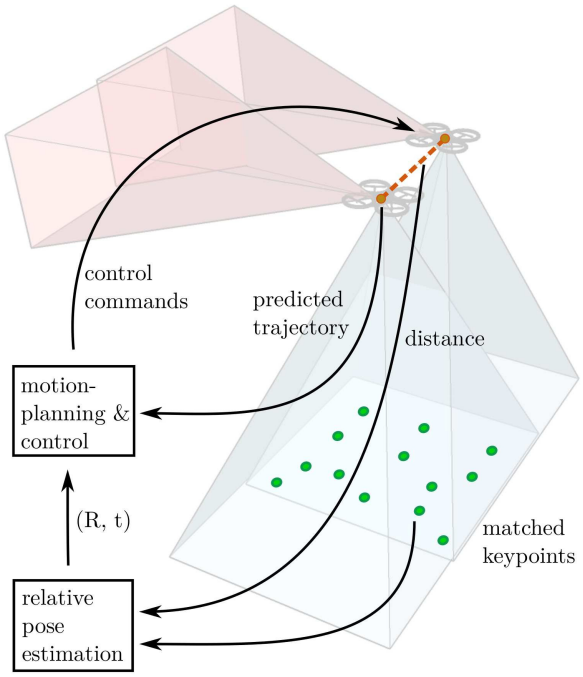


Fig. 6. Basic idea for relative localization: The overlapping FoV of the downward-looking cameras is used to determine the relative pose of the follower with respect to the leader up to a scale factor (e.g. with the 5-point algorithm). The absolute distance information is then used to recover the scale. We assume that this distance can be measure with ultra wideband technology (UWB). This information combined with the predicted trajectory of the leader is used to determine the follower's control commands. The forward-facing (red) cameras are used to capture the multi-view shots of the scene.

of the UAV to the reference velocities by modelling it as a decoupled linear time invariant (LTI) system. A comparison with the detailed non-linear model, that includes the non-linear state equations and motor-, attitude- and velocity-control, shows that a LTI-model of order three with constraints on the input signal is sufficient to approximate the system behavior in one coordinate direction. A model of the form

$$\begin{aligned} \dot{\mathbf{x}}_x &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ a_1 & a_2 & a_3 \end{bmatrix} \mathbf{x}_x + \begin{bmatrix} 0 \\ 0 \\ b_3 \end{bmatrix} u_x \\ y_x &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \mathbf{x}_x \end{aligned} \tag{9}$$

is identified with $u_x$ representing the reference velocity in $x$-direction as input, the position $x$, velocity $\dot{x}$ and acceleration $\ddot{x}$ as state variables, consequently $\dot{\mathbf{x}}_x = [x \quad \dot{x} \quad \ddot{x}]^T$, and the measured position $x$ as output. While many approaches restrict their state vector to position and velocity, our investigations showed, that this is not sufficient to cover the systems dynamics with sufficient precision. Therefore, the acceleration is added as additional state, which results in also modelling the jerk of the system. The height is assumed to be constant, thus the $z$-coordinate is not considered in the model. Further, to keep the MPC-problem linear we do not consider rotations at this point. The overall model is composed of the two sub-models of the $x$ and $y$ coordinates

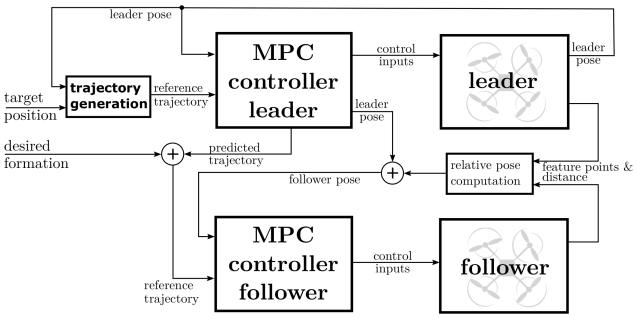


Fig. 7. Overview of our control structure with the target position for the leader and the relative position of the follower, specified desired formation, as inputs. The reference input for the leader MPC controller is generated by the "trajectory generation" block, whereas the reference trajectory of the follower is generated by the shifted predicted trajectory of the leader. The relative pose (computed by the 5-point algorithm [20] and the measured distance) is combined with the absolute pose of the leader to determine the follower pose. The control inputs determined by the MPC controller are the desired velocities in $x$ and $y$ direction of the UAV. To control the height and orientation of heading the provided controllers of the non-linear UAV model from [21] are used.

and can be given as

$$\begin{aligned} \dot{\mathbf{x}} &= \begin{bmatrix} \mathbf{A}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_y \end{bmatrix} \mathbf{x} + \begin{bmatrix} \mathbf{b}_x & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_y \end{bmatrix} \mathbf{u} \\ \mathbf{y} &= \begin{bmatrix} \mathbf{c}_x^T & \mathbf{0} \\ \mathbf{0} & \mathbf{c}_y^T \end{bmatrix} \mathbf{x} \end{aligned} \tag{10}$$

with $\mathbf{x} = [\mathbf{x}_x \quad \mathbf{x}_y]^T$, $\mathbf{u} = [\mathbf{u}_x \quad \mathbf{u}_y]^T$ and $\mathbf{y} = [\mathbf{y}_x \quad \mathbf{y}_y]^T$.

*2) Control Structure:* As shown in Figure 7, the target position for the leader UAV and the relative pose for the follower are the inputs to the control structure. Based on the leader's current position a discrete reference trajectory is computed specifying the next $h$ desired positions of the leader, where $h$ is the prediction horizon of the MPC controller. Within this trajectory generation we follow a very simple approach by defining a straight line between start and destination as the ideal path. Note that more complex trajectory generation concepts may be applied to generate the reference trajectory. The desired velocity of the leader defines the distance between the points of the reference trajectory.

As already mentioned, we aim for a stable and synchronized formation flight. Therefore, the reference trajectory of the follower is determined by shifting the predicted trajectory of the leader by the desired relative position. Our approach uses the downward facing cameras to compute the relative pose up to scale with feature based methods (e.g. with the 5-point algorithm [20]). The scale factor is recovered with the distance information provided by the UWB modules. In Figure 7, this is represented with the "relative pose computation" block. The absolute pose of the follower is then determined as the sum of the leader's absolute and the follower's relative pose.

Each controller uses its measured 2D-position and its reference trajectory that includes the next $h$ desired positions to determine the control inputs of the UAV, which are

the reference velocities in $x$ and $y$-direction. These control inputs are determined in global coordinates, thus a rotation must be applied to translate them to the UAV's body frame. The angle of rotation is given by the actual facing direction of the UAV. This rotation is not included in the prediction model, but as our results show, errors are only introduced when the UAVs are rotating. Constant angles of rotation do not cause any problems.

## V. Results

We evaluate our proposed solution for the MC problem with a simulation study. We have therefore implemented the three parts of our approach in Matlab/Simulink. Constellation planning is implemented as a constrained non-linear optimization task. Mission planning is based on an online available Matlab function, called "Fixed Start Open Traveling Salesman Problem - Genetic Algorithm" [22]. For the model identification and flight simulation we use the open source Simulink model of a quad-copter [21] which includes the non-linear state equations, motor dynamics and control as well as attitude and velocity controllers. It also contains inputs for external disturbances that can be used to evaluate the robustness of the controllers. The Matlab toolbox for predictive control is exploited to implement our MPC controllers. Within the MPC controllers, the identified linear model is used and the constraints for the control inputs are set. Based on a simple demonstrative example we perform the constellation and mission planning and simulate the flights of both UAVs.

### A. Constellation and Mission Planning

Figure 8 depicts the results of constellation and mission planning on a small scenario composed by two rectangles with a total of 8 line segments. We specified the camera parameters as a focal length of $f = 4.7$ mm, an aperture angle of $\alpha = 60°$ and a resolution of $r = 1024$ pixels. With the choice of a minimum resolution of $\delta = 25$ pixels per meter on the target we get an upper bound on the capturing distance with $d_{\max} = 35.4$ m and the limit of $l_{\max} = b_{\max} = 20.5$ m. As no line of the model exceeds this limit, no further partitioning was required. Planning was conducted for $m = 8$ constellations and achieved a path for the leader with an overall length of $464$ m.

### B. Modelling and Flight Simulation

Figure 9 shows a comparison of the linear prediction model with the detailed non-linear model for one axis. Further simulations show, that the accuracy of this model is sufficient, since both UAVs come up with a very stable formation flight, once they took the desired formation. In advance, using a linear model allows a higher frame-rate of the MPC because of lower computational complexity compared to non-linear MPCs. This is important for implementation on real UAVs with limited resources.

The results of the simulated mission execution with a total flight time of 115 seconds are shown in Figure 10. The UAVs start at their initial positions. The inputs, given by the target position of the leader and the desired relative position of the follower, as well as the rotations of both UAVs, are defined by the capturing constellations. They start

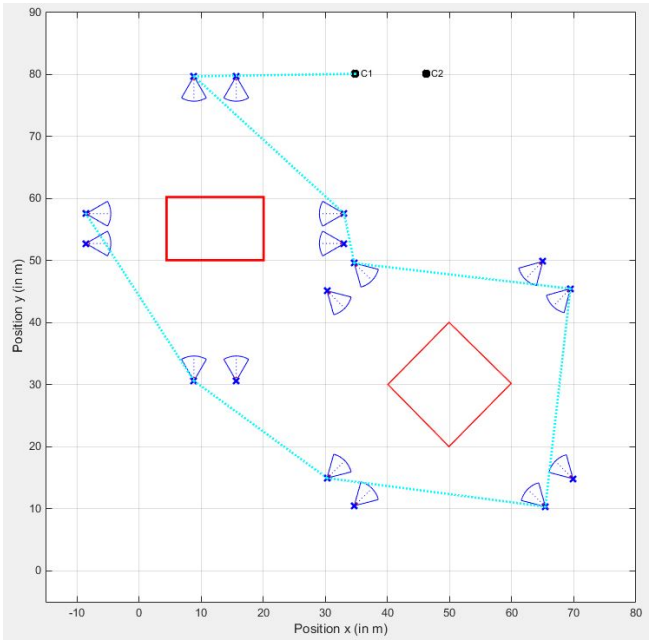


Fig. 8. Results of the planning process. Input: The given scene abstraction (red) and the initial positions of the UAVs (black crosses). Output: The constellations (blue) and the planned path of the leader (cyan line)

with the values of the first capturing constellation $c'_1$ and are kept constant until the multi-view image is is captured. We assume that the capturing is complete when both, the positional error of both UAVs are below $T_{\text{pos}} = 0.2$ m and their velocities are less than $T_{\text{vel}} = 0.1$ m/s for at least 2 seconds. After these conditions have been fulfilled, the reference signals jump to the values of the next constellation. This process is repeated for all $c'_i \in C'$ with $i = 1, \ldots, m$. The reference signal for the height is kept at a constant value during the whole simulation. Figure 10 and 11 show that all control errors vanish when the UAVs approach the constellation. This means that the leader UAV accurately reaches the intended poses and the desired formation is taken by the follower. In Figure 12 the temporal evolution of the 2D positions is shown. It can be seen, that the desired formation is taken while moving to the next target. However, significant errors are imposed immediately after the reference values are set to the next constellation. The UAVs start with a rotation resulting in poor tracking performance of the leader until the

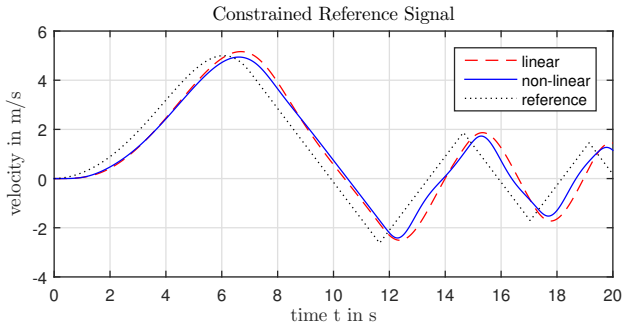


Fig. 9. Response of the linear (dashed red line) and the nonlinear (blue line) prediction model to a varying velocity reference input. Velocity and acceleration are limited by 5 m/s and 1.5 m/s$^2$, respectively.

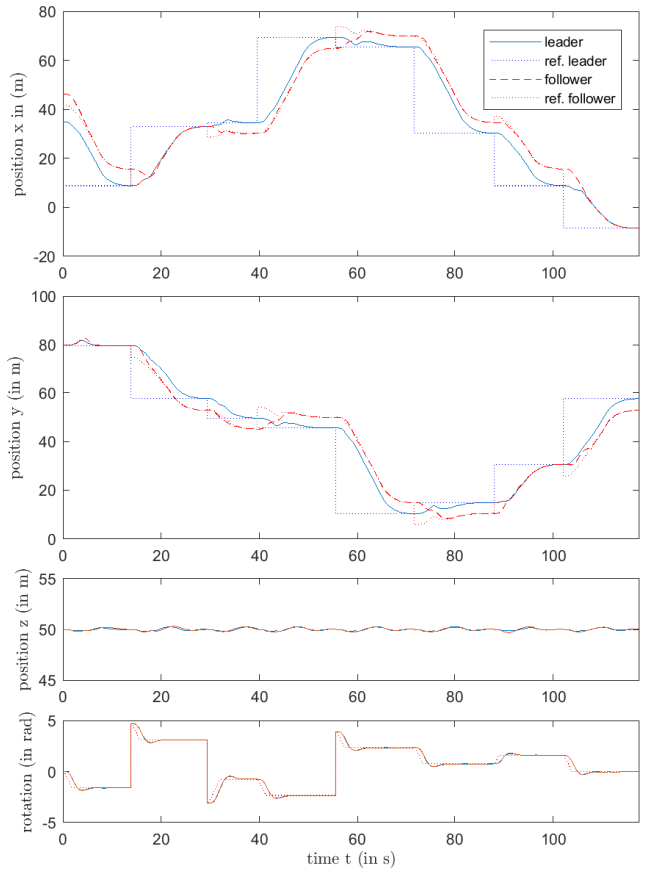


Fig. 10. Performance of the MPC controller during the overall mission. The upper two plots show the $x$ and $y$ positions of the leader (blue solid) and the follower (dashed red) UAVs whereas the dotted lines represent the reference values. The third plot depicts the heights of the UAVs which have been set to 50 m. Deviations in the height tracking are caused by uncompensated coupling effects whenever a UAV changes its pan or tilt angle. The fourth plot shows the orientations of leader and follower.

reference orientation of heading is reached. This behavior is a consequence of our linear prediction model which does not cover UAV rotations. A further consequence of the rotation maneuver is that the movement of the follower looses synchronicity. However, once the formation error has vanished, synchronicity is maintained again.

## VI. Conclusion

In this paper, we introduced the multi-coverage problem and presented a distributed model-predictive controller for the special case of flexible stereo coverage. Our approach computes the required camera poses wrt. a given depth error model and a cost-function that takes the relative view angle of the downward facing cameras into account. Our MPC control structure achieves accurate relative positioning and a stable formation flight when the UAVs are approaching their constellation poses which represent important phases of the mission. However, synchronicity and formation deteriorate immediately after the next constellation poses are set.

Smooth formation transitions are one strategy to compensate this behaviour and would result in a sufficient FOV overlap of the downward facing cameras at *any* time throughout the mission. In our future work we extend the planning into 3D and implement the system on real UAVs to conduct real-world experiments.

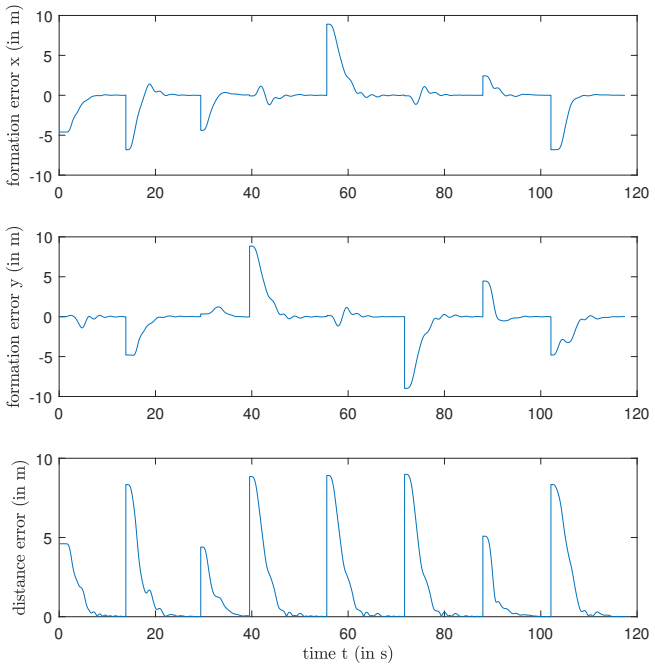


Fig. 11. Formation error: The first and second plots show the formation error in x and y direction, respectively. The third plot shows the absolute formation error, which is the Euclidean distance between the actual and reference position of the follower in the $xy$-plane.

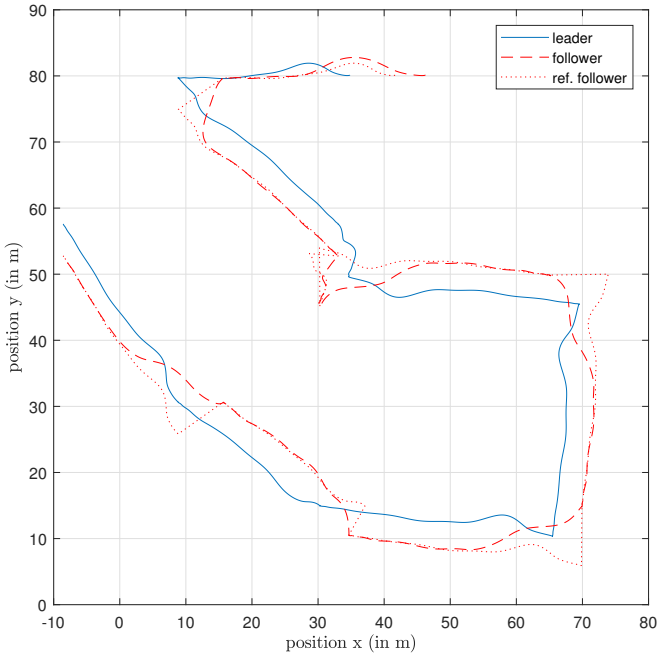


Fig. 12. Plot of the positions on the $xy$-plane: The plot shows the 2D position of the leader (blue), the follower (dashed red) and the reference input of the followers MPC controller (dotted red). The results show that whenever the reference values jump to those of the next camera constellation, there is an error in the follower's position which converges and the UAVs take the desired formation while moving towards their target.

## ACKNOWLEDGMENT

This work was conducted while Matthias Weyrer was with the Institute of Networked and Embedded Systems, Alpen-Adria-Universität Klagenfurt, Austria and was supported by the research initiative 'Intelligent Vision Austria' with funding from the Austrian Federal Ministry of Science, Research and Economy and the Austrian Institute of Technology.

## REFERENCES

[1] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative Robots to Observe Moving Targets: A Review," *IEEE Transactions on Cybernetics*, vol. 48, no. 1, pp. 187–198, 2018.

[2] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.

[3] D. Wischounig-Strucl and B. Rinner, "Resource Aware and Incremental Mosaics of Wide Areas from Small-Scale UAVs," *Machine Vision and Applications*, vol. 26, no. 7-8, pp. 885–904, 2015.

[4] S. Bhattacharya, R. Ghrist, and V. Kumar, "Multi-robot coverage and exploration on Riemannian manifolds with boundaries," *International Journal of Robotics Research*, vol. 33, no. 1, pp. 113–137, 2014.

[5] E. Yanmaz, S. Yahyanejad, B. Rinner, H. Hellwagner, and C. Bettstetter, "Drone Networks: Communications, Coordination, and Sensing," *Ad Hoc Networks*, vol. 68, pp. 1–15, 2018.

[6] A. Khan, B. Rinner, and A. Cavallaro, "Multiscale observation of multiple moving targets using Micro Aerial Vehicles," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, 2015, pp. 4642–4649.

[7] A. Khan, E. Yanmaz, and B. Rinner, "Information Exchange and Decision Making in Micro Aerial Vehicle Networks for Cooperative Search," *IEEE Transactions on Control of Network Systems*, vol. 2, no. 4, pp. 335–347, 2015.

[8] J. Scherer and B. Rinner, "Short and Full Horizon Motion Planning for Persistent multi-UAV Surveillance with Energy and Communication Constraints," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and System*, 2017, pp. 230–235.

[9] D. Gallup, J. M. Frahm, P. Mordohai, and M. Pollefeys, "Variable baseline/resolution stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[10] Y. Nakabo, T. Mukai, Y. Hattori, Y. Takeuchi, and N. Ohnishi, "Variable baseline stereo tracking vision system using high-speed linear slider," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2005, pp. 1567–1572.

[11] K. Hausman, J. Müller, A. Hariharan, N. Ayanian, and G. S. Sukhatme, "Cooperative multi-robot control for target tracking with onboard sensing," *The International Journal of Robotics Research*, vol. 34, no. 13, pp. 1660–1677, 2015. [Online]. Available: http://dx.doi.org/10.1177/0278364915602321

[12] F. Lin, K. Peng, X. Dong, S. Zhao, and B. M. Chen, "Vision-based formation for UAVs," in *Proceedings of the IEEE International Conference on Control Automation*, 2014, pp. 1375–1380.

[13] M. W. Achtelik, S. Weiss, M. Chli, F. Dellaerty, and R. Siegwart, "Collaborative stereo," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, 2011, pp. 2242–2248.

[14] N. Piasco, J. Marzat, and M. Sanfourche, "Collaborative localization and formation flying using distributed stereo-vision," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1202–1207.

[15] E. Montijano, E. Cristofalo, D. Zhou, M. Schwager, and C. Sagüés, "Vision-Based Distributed Formation Control Without an External Positioning System," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 339–351, 2016.

[16] X. Zhang, X. Li, K. Wang, and Y. Lu, "A survey of modelling and identification of quadrotor robot," in *Abstract and Applied Analysis*, vol. 2014. Hindawi Publishing Corporation, 2014.

[17] R. Mahony, V. Kumar, and P. Corke, "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," *IEEE Robotics Automation Magazine*, vol. 19, no. 3, pp. 20–32, Sept 2012.

[18] H. Voos, "Nonlinear control of a quadrotor micro-uav using feedback-linearization," in *Proceedings of the IEEE International Conference on Mechatronics*, April 2009, pp. 1–6.

[19] J. Morel and G. Yu, "ASIFT: A New Framework for Fully Affine Invariant Image Comparison," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 438–469, 2009. [Online]. Available: https://doi.org/10.1137/080732730

[20] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 756–770, June 2004.

[21] D. Hartman, K. Landis, M. Mehrer, S. Moreno, and J. Kim. (2014) Quadcopter Dynamic Modeling and Simulation (Quad-Sim) v1.00. [Online]. Available: http://github.com/dch33/Quad-Sim

[22] J. Kirk. (2014) Fixed Start Open Traveling Salesman Problem - Genetic Algorithm. [Online]. Available: https://de.mathworks.com/matlabcentral/fileexchange/21198-fixed-start-open-traveling-salesman-problem-genetic-algorithm