MPC controlled Multirotor with suspended slung Load: System Architecture and Visual Load Detection

Markus Zürn
Australian Research Centre for Aerospace Automation
Queensland University of Technology
University of Stuttgart
22 Boronia Rd, Brisbane Airport
QLD 4009, Australia
+61 7 3138 1772
zuern.markus@gmail.com

Aaron McFadyen
Australian Research Centre for Aerospace Automation
Queensland University of Technology
22 Boronia Rd, Brisbane Airport
QLD 4009, Australia
+61 7 3138 1772
aaron.mcfadyen@qut.edu.au

Kye Morton
Australian Research Centre for Aerospace Automation
Queensland University of Technology
22 Boronia Rd, Brisbane Airport
QLD 4009, Australia
+61 7 3138 1772
kye.morton@qut.edu.au

Stefan Notter
Institute for Flight Mechanics and Flight Controlling (iFR)
Pfaffenwaldring 27
70569 Stuttgart, Germany
+49 711 685 66618
stefan.notter@ifr.uni-stuttgart.de

Alexander Heckmann
University of Stuttgart
Stuttgart, Germany
alexander.heckmann@outlook.com

Felipe Gonzalez
Australian Research Centre for Aerospace Automation
Queensland University of Technology
22 Boronia Rd, Brisbane Airport
QLD 4009, Australia
+61 7 3138 1772
felipe.gonzalez@qut.edu.au

Abstract—**There is an increased interest in the use of Unmanned Aerial Vehicles for load transportation from environmental remote sensing to construction and parcel delivery. One of the main challenges is accurate control of the load position and trajectory. This paper presents an assessment of real flight trials for the control of an autonomous multi-rotor with a suspended slung load using only visual feedback to determine the load position. This method uses an onboard camera to take advantage of a common visual marker detection algorithm to robustly detect the load location. The load position is calculated using an onboard processor, and transmitted over a wireless network to a ground station integrating MATLAB/SIMULINK and Robotic Operating System (ROS) and a Model Predictive Controller (MPC) to control both the load and the UAV. To evaluate the system performance, the position of the load determined by the visual detection system in real flight is compared with data received by a motion tracking system. The multi-rotor position tracking performance is also analyzed by conducting flight trials using perfect load position data and data obtained only from the visual system. Results show very accurate estimation of the load position (~5% Offset) using only the visual system and demonstrate that the need for an external motion tracking system is not needed for this task.**

978-1-4673-7676-1/16/$31.00 ©2016 IEEE

TABLE OF CONTENTS

1. INTRODUCTION

Civilian Unmanned Aerial Vehicle Systems (UAS), especially small multirotors, offer a multitude of possible applications from automation in farming to load transportation and many more [1] [2] [3] [4]. For this reason there is an increased interest in industry and research. In particular, the combination of a multirotor UAV with a suspended slung load [5] represents a system of great benefit. Due to its simplicity such a system can be assembled quickly and offers the possibility to pick up and deliver very fast and access areas where it would be an immense effort with conventional methods. Furthermore a suspended load can protect the load from the wash of the propellers due to its distance to them. This is an advantage in both outdoor use, e.g. precise pest spraying in agriculture, moving material in construction or parcel and book delivery, and indoor use.

A common problem in controlling such a system is the detection of the loads or rather the objects position. Many approaches work with a motion tracking system [6] [7]. This might be beneficial in research for control algorithms, however, this cannot be applied in real world applications. Since the system, especially the load, is meant to be simple, neither extra sensors nor other electronics should be attached to the load. Hence, the use of a camera in combination with an object or marker detection algorithm seems likely. Thus almost any kind of load can be used and the load is detached of a motion tracking system and ready for real world applications.

For this purpose, the contributions of this paper are:

1. Design of an operational and fast system architecture with the necessary network between different models.
2. Implementation of an onboard marker detection algorithm.
3. Flight test and evaluation of the effect of the visual feedback on the system and evaluation of the precision of the visual detection.

This paper is structured as follows: Section two provides an overview over the system dynamics and the Model Predictive Control in general. The system architecture and the visual load detection is presented in section 3 and 4 respectively. The flight test and results are outlined in section 5 followed by a discussion and conclusion in section 6.

2. System Dynamics and Model Predictive Control

2.1 Coordinate Systems

In order to model the system dynamics the coupled system is presented in two different coordinate systems. The inertial frame of reference is the geographical coordinate system and the body fixed frame of reference is the coordinate system of the UAV with the *z*-axis pointing downwards. Both earth rotation and curvature are neglected.

Figure 1 illustrates the coordinate systems used. The inertial frame of reference $I = \{\mathbf{x}_I, \mathbf{y}_I, \mathbf{z}_I\}$ is related to the body fixed frame of reference $F = \{\mathbf{x}_F, \mathbf{y}_F, \mathbf{z}_F\}$, by the translation matrix $\mathbf{T}(r)$ and the transformation matrix $\mathbf{R}$:

$$\mathbf{R} = \left[\mathbf{R}_x(\phi)\mathbf{R}_y(\theta)\mathbf{R}_z(\psi)\right]^{\boldsymbol{T}}, \qquad (2.1)$$

where $\mathbf{R}_i$ are the sub sequential rotations by the roll angle ϕ, pitch angle θ and yaw angle ψ, which are represented in the attitude vector of the UAV $\boldsymbol{\phi} = (\phi, \theta, \psi)$.

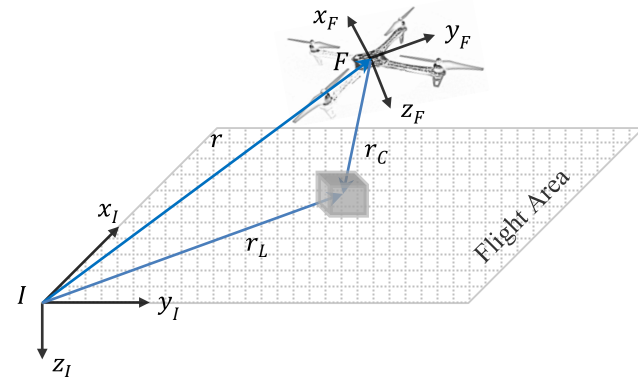


Figure 1. Coordinate Systems and Position Vectors

2.2 Multirotor and Load dynamics control

Neglecting the low-level controller dynamics due to the fact that they are reacting quasi-instantaneously [7], the equations of motion for the high-level control of the multirotor UAV and the equations of motion of the load [8] result in the state vector of the coupled system

$$\mathbf{x} = (\mathbf{r}, \mathbf{v}, \boldsymbol{\phi}, x_L, y_L, u_L, v_L)\,, \qquad (2.20)$$

where $\mathbf{r}$ is the position vector of the UAV, $\mathbf{v}$ is the velocity of the UAV, x_L and y_L are the *x* and *y* position of the load in the inertial frame of reference, u_L and v_L are the velocity of the load in *x* and *y* direction in the inertial frame of reference, taking into account that the *z* component of the load position can be derived from the kinematic constraint

$$||\mathbf{r} - \mathbf{r}_L||_2 = l\,, \qquad (2.21)$$

where l is the length of the cord with which the load is attached to the UAV plus the distance from the top of the load to the center of the load.

The control vector is given by

$$\mathbf{u} = (\omega, f_T), \qquad (2.22)$$

where ω are the attitude rates of the UAV and f_T is the collective thrust of the multirotor motors.

2.3 Model Predictive Control (MPC)

MPC is based on solving an open-loop control problem online. With the implemented model of the system dynamics and the actual measurement, the behavior of the system can be predicted over a certain time-horizon T_P. An input sequence $\bar{\mathbf{u}}$ over a control horizon T_C is obtained by minimizing a quadratic cost function $J(\cdot)$. The bar symbol denotes a controller internal variable. It is assumed that at least one new sensor update is available within the prediction step size δ. Therefore only the first element of the input sequence is implemented.

The basic mathematical formulation presented in this subsection is based on [9] and the principle of MPC, which is

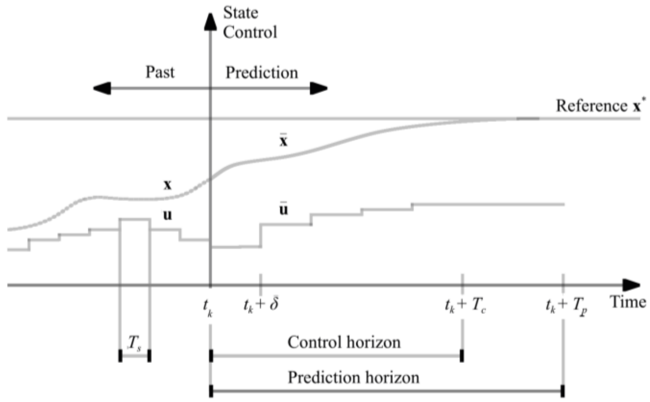


Figure 2. Principle of MPC [10]

illustrated in Figure 2, and the translation to discrete state and input vectors is from [10], [5].

The time-discrete system to be controlled is described by a nonlinear set of difference equations

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k), \tag{2.30}$$

where the index $k \in \mathbb{N}_0$ denotes the state control input of the system at the kth sampling instant. The nonlinear system is linearized in the context of this paper. Both the control vector $\mathbf{u}$ and the state vector $\mathbf{x}$ are constrained by predefined constant vectors.

$$\begin{aligned} \mathbf{u} &\in \mathcal{U}, \forall k, \\ \mathbf{x} &\in \mathcal{X}, \forall k, \end{aligned} \tag{2.31}$$

with

$$\begin{aligned} \mathcal{U} &:= \{\mathbf{u} \in \mathbb{R}^m \,|\mathbf{u}_{min} \le \mathbf{u} \le \mathbf{u}_{max}\}, \\ \mathcal{X} &:= \{\mathbf{x} \in \mathbb{R}^m \,|\mathbf{x}_{min} \le \mathbf{x} \le \mathbf{x}_{max}\}, \end{aligned} \tag{2.32}$$

where m denotes the dimension of the input vector $\mathbf{u}$ and n denotes the dimension of the state vector **x.** In the context of this paper the constraints for **x** and **u** are given by the UAV and the flight area. The control law is represented by the discrete finite horizon open-loop optimal control problem of finding an input sequence $\bar{\mathbf{u}}$ that minimizes a quadratic cost function

$$\min_{\bar{\mathbf{u}}} J\,(\mathbf{x}_k, \bar{\mathbf{x}}, \bar{\mathbf{u}}; T_C, T_P), \tag{2.33}$$

This leads to

$$\bar{\mathbf{x}}_{j+1} = \mathbf{f}(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_j), \bar{\mathbf{x}}_0 = \bar{\mathbf{x}}_k, \tag{2.34}$$

with,

$$\begin{gathered} \bar{\mathbf{u}}_j = \begin{cases} \in \mathcal{U}, & N \le M \\ \bar{\mathbf{u}}_M, & N > M \end{cases}, \\ \bar{\mathbf{x}}_j \in \mathcal{X}, \end{gathered} \tag{2.35}$$

where the index $j \in \mathbb{N}_0$ denotes the jth prediction instant for the kth sampling instant of the internal state vector $\bar{\mathbf{x}}$ or input vector $\bar{\mathbf{u}}$. The number of steps of the prediction horizon

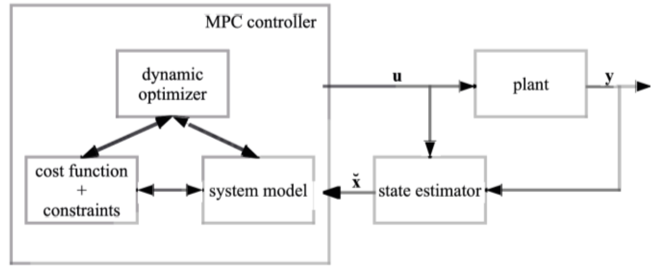


Figure 3. Basic MPC control loop [9]

is $N = T_P / \delta$, and $M = T_C / \delta$ is the number of steps in the control horizon.

The basic sequence of an MPC instance works as follows:

- Read the current state vector of the system by either measuring or estimating
- Solve the optimal control problem (2.31) to obtain an optimal input sequence
- Implement the first element of the computed sequence
- Continue by repeating

The basic MPC control loop is illustrated in Figure 3.

3. System Architecture

The complete system exhibits two major parts. One of them is the groundstation and the other is the UAV and its onboard equipment. Figure 4 illustrates the general layout of the system.

3.1 Groundstation

The groundstation is an off board computer which runs three separate models in parallel computing in MATLAB/SIMULINK. These models are the sensor, the controller (e.g. MPC and state estimator) and the radio.

The sensor's function is to receive the attitude and position of the UAV. It is delivered by the motion tracking system "VICON" [11]. The VICON system also determines the attitude and position of the load to provide a comparison with the camera load detection. The camera load detection data is transmitted via WIFI to the Sensor. Both, the state of the UAV as well as the state of the load is forwarded to the controller model.

The controller model incorporates the MPC algorithm (Section 2) and receives the current states $(\mathbf{r}, \mathbf{v}, x_L, y_L, u_L, v_L)$ of both UAV and load. It produces the command $\mathbf{u} = (\omega, f_T)$, (roll, pitch, yaw and collective thrust) to control the UAV and load states with respect to a given reference trajectory. These commands are forwarded to the radio model.

The Radio model maps the given commands to PWM values and transmits them to the platform via a serial-wireless

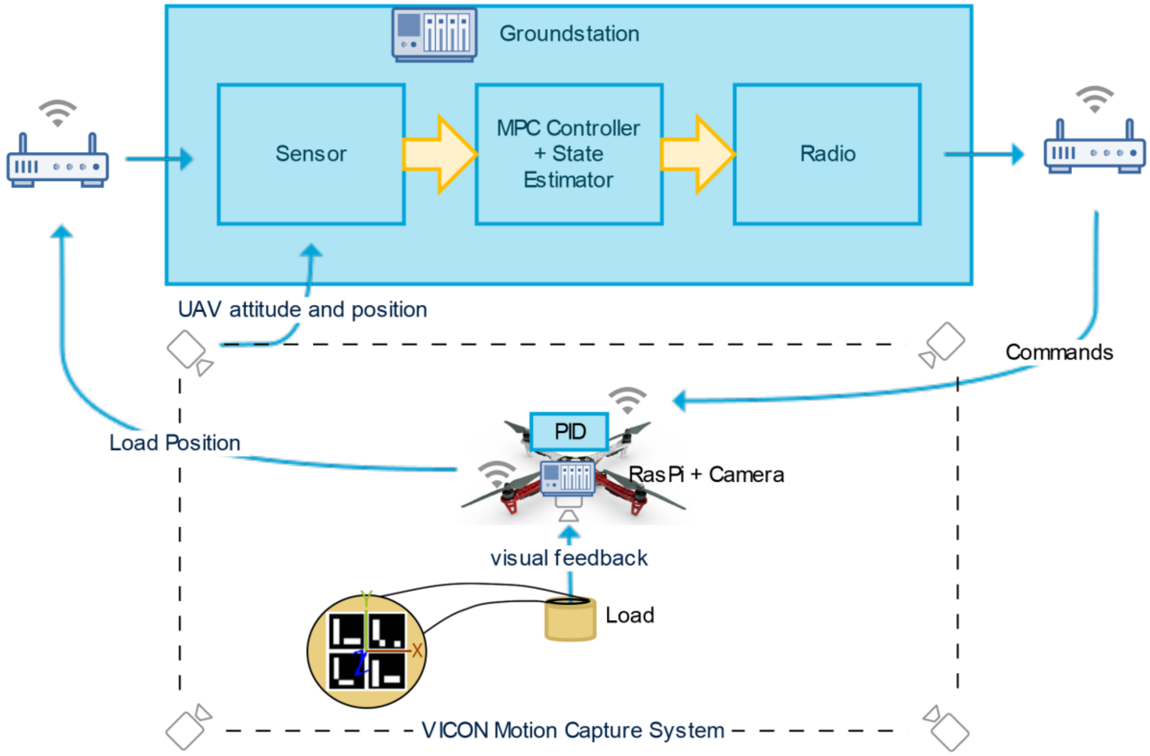


Figure 4. System Architecture

interface. All models are running in parallel at an average frequency of 175 Hz. However, the slowest part of the control loop, which is the onboard receiver, is running at 25 Hz. For this reason, the Controller model is also coupled with a state estimator.

The state estimator and latency compensator, as can be seen in Figure 3, receives the measured data of the current state of UAV and load and the control vector **u.** It also knows the average system latency with N sampling steps. The estimator works as introduced in [7], [12]. At time step t_k the measured state is $\mathbf{x}_k$ and the command is $\mathbf{u}_k$. The estimate of the state t_k taking measurements up to the time step t_j into account is represented by $\hat{\mathbf{x}}_j^k$. With a current time step t_k the estimated state is $\hat{\mathbf{x}}_k^{k+N}$. In order to predict the state, the current measurement and the previous estimate are weighted to obtain a new estimate

$$\hat{\mathbf{x}}_k^k = \mathbf{C}\hat{\mathbf{x}}_{k-1}^k + (\mathbf{I} - \mathbf{C})\mathbf{x}_k, \tag{3.10}$$

where **I** is the identity matrix and **C** is a diagonal matrix, where its elements $\mathbf{C}_{ii} \in [0,1]$ are the relative weights of the predicted state. Afterwards, the N most recent commands are recursively applied to the nonlinear system dynamics

$$\hat{\mathbf{x}}_k^{j+1} = \mathbf{f}(\hat{\mathbf{x}}_k^j, \hat{\mathbf{u}}_{j-N}), \tag{3.11}$$

From $j=k$ to $j=k+N-1$. Based on the predicted state $\hat{\mathbf{x}}_k^{k+N}$, the controller computes the next command $\mathbf{u}_k$ which arrives at the UAV when its true state is $\mathbf{x}_{k+N}$.

2.2 Unmanned Aerial Vehicle (UAV) platform

There are four main components onboard the unmanned aerial vehicle. The quadrotor frame (with battery and rotors), the load, a RaspberryPi with downward facing camera module, and a low-level attitude controller (set of PID controllers).

The attitude controller receives commands from the radio model and calculates the required change in rotor speed to map the commands into actions. The load is attached to the UAV frame with a thin cord of a given length. It has two degrees of freedom. A marker is attached on top of the load, such that the marker can be seen by the downward facing camera (Figure 4). This camera is connected to the RaspberryPi where the marker tracking algorithm is running and calculating the position of the load. This position is transmitted to the groundstation via a wireless network and received by the sensor model.

4. Load Detection

The ROS node "ar_sys" [13] is running on the RaspberryPi and is used to keep track of the marker on the load, and calculate its position. "ar_sys" uses the "openCV" library [14] in combination with the "ArUco" [15] library to provide an algorithm for robust marker detection and tracking. Every "ArUco" marker has its own ID. It can be chosen from 1024 different IDs. Since it could be difficult to detect a single marker, because of poor light conditions and other issues, "ArUco" allows to use marker boards. A board of four

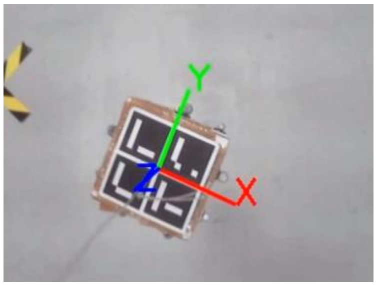


Figure 5. In flight load detection. The image shows the detected load from the point of view of the UAV.

"ArUco" markers arranged in a square (as can be seen in Figure 5) combined with a low camera resolution (320x240), ensures fast image processing and enables frequent load position updates. The side length of the marker used in the context of this paper is 11cm.

The position of the center of the marker board, calculated by using standard photo-geometric procedures, is published in the ROS network at 25fps in body fixed frame coordinates ($\mathbf{r}_{L,F} = (x_{L,F}, y_{L,F}, z_{L,F})^T$). The field of view of the camera is ~54° horizontal and ~41° vertical [16]. It is assumed that the load is always in the field of view of the camera. This is a reasonable assumption considering the length of the slung load and the speed of the UAV.

The groundstation software which includes the sensor model and the RaspberryPi are connected to the ROS network. With the new "Robotics System Toolbox" [17] of MATLAB, ROS topics such as the published load position can be subscribed directly within SIMULINK. This makes the connection between the detected load position and the controller straightforward. Now, the sensor model has the position $\mathbf{r}$ in the inertial frame and attitude $\boldsymbol{\phi}$ of the UAV as well as the position of the load in the body fixed frame $\mathbf{r}_{L,F}$. With the attitude vector of the UAV $\boldsymbol{\phi}$ the load position can be transferred to the controller model in inertial frame coordinates by

$$\mathbf{r}_L = \mathbf{r} + \mathbf{R}\mathbf{r}_{L,F}, \tag{4.1}$$

where $\mathbf{R}$ is the transformation matrix given in (2.1) depending on $\boldsymbol{\phi}$. As described in (2.20) only x_L and y_L of $\mathbf{r}_L$ are necessary for the controller and must be returned. Furthermore the velocity of the load in u_L and v_L can be calculated with the derivative $\dot{\mathbf{r}}_L$, whereat the minimal time step is $\Delta t = 25\ Hz$. Hence all relevant data of the load part of the state vector is available to process the control algorithm.

5. Flight Tests

Three flight tests were conducted at the flight area of the Australian Research Centre for Aerospace Automation (ARCAA) (Figure 6). The purpose of the flight tests is to demonstrate the accuracy and robustness of the visual object detection compared to load tracking using the VICON motion

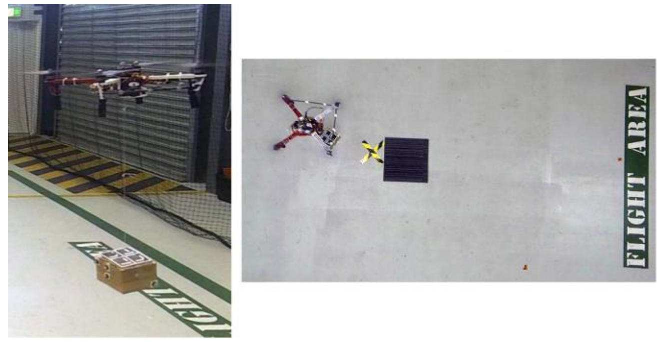


Figure 6. Still images of a flight test. Side view on the left hand side, bird's eye view on the right hand side.

tracking system. The test area is provided with daylight as well as artificial light. In all three flight tests the reference trajectory is a planar figure eight.

Table 1. Flight Time

Test #	Flight time [s]
1	22.5
2	20
3	20

For comparison purposes the load position in the inertial frame is detected by the VICON system as well. In test one and two, the load position detected by the camera is used for the control algorithm, whereas in test three the load position detected by the VICON system is used. In Table 1 the flight time of each flight test is displayed. The *z* Position over time is calculated using (2.21) and a cord length *l* of 60 cm. The load is a box with 14 cm x 14 cm surface and a depth of 10 cm. The quadrotor has a diagonal wheelbase of 450mm.

Figure 7 to Figure 14 shows the result for the system using the visual feedback of the camera to control the load (Test #1 and #2), Figure 15 to Figure 18 (Test #3) show the result when the system uses the VICON data for the location of the load. The solid blue line in each of these figures illustrates the load position detected by the camera, whilst the dashed red line illustrates the load position detected by the VICON system. Abrupt changes in the VICON load position are from a short failure in the object orientation in the VICON's coordinate frame. The dashed-dotted black line illustrates the reference trajectory of the load.

All figures (Figure 8 till Figure 18) show a good accuracy between the camera and the VICON detection of the load. It is also visible that the trajectory of the load in the *x*-plane (Figure 8, Figure 12 and Figure 16) is closer to the reference than in the *y*-plane (Figure 9, Figure 13 and Figure 17). This is due to the fact that for a planar figure 8 flight, there has to be one period in one plane and two periods in the second plane but both in the same time. The *z*-plane is the most accurate (Figure 10, Figure 14 and Figure 18) concerning the distance between the load position detected by the VICON system and the load position detected by the camera.

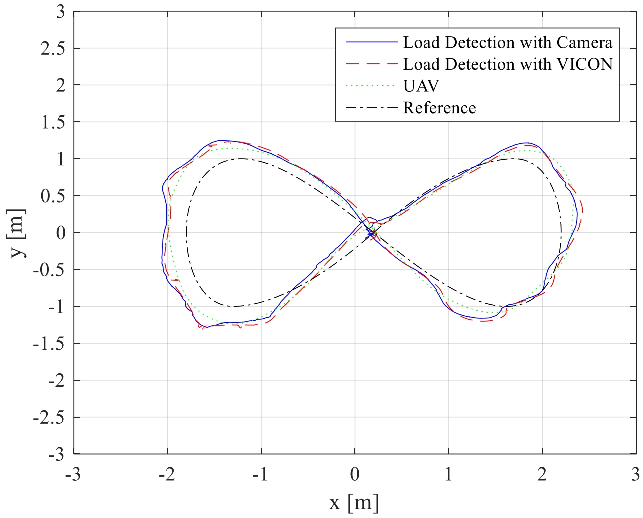


Figure 7. Flight test with load detection by camera in 2D. (Test #1)

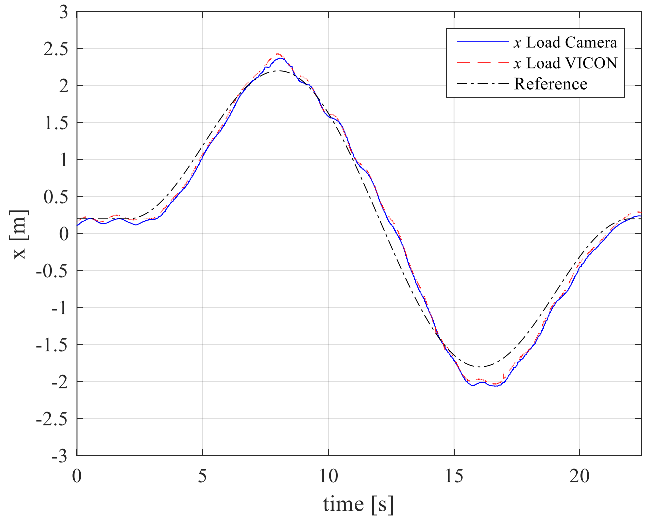


Figure 8. Flight test with load detection by camera. *X* position over time. (Test #1)

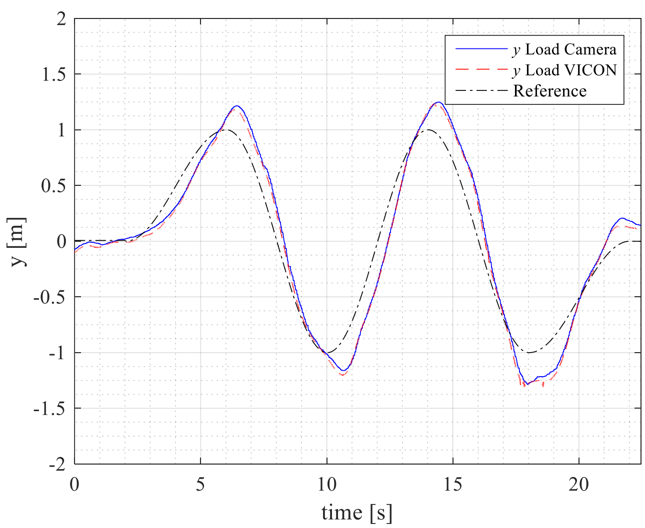


Figure 9. Flight test with load detection by camera. *Y* position over time. (Test #1)

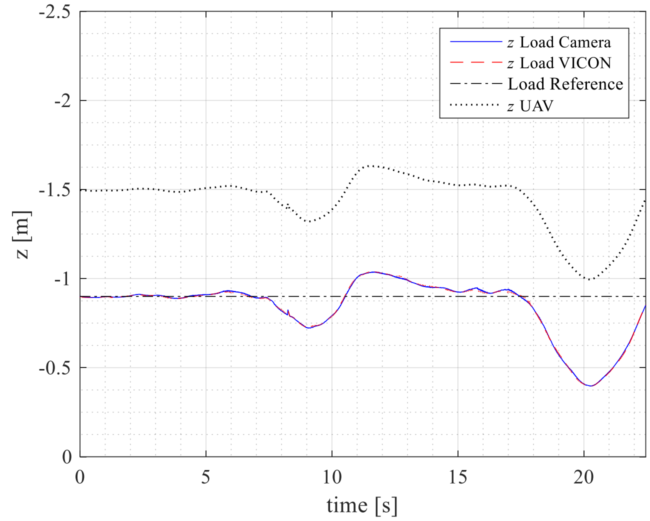


Figure 10. Flight test with load detection by camera. *Z* position over time. (Test #1)

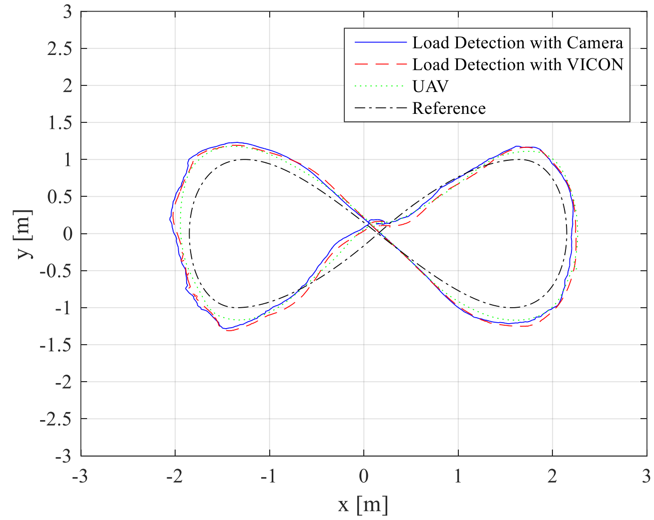


Figure 11. Flight test with load detection by camera in 2D. (Test #2)

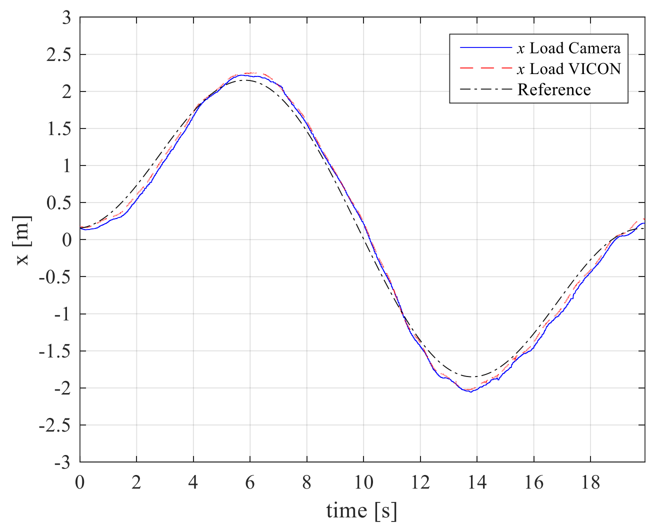


Figure 12. Flight test with load detection by camera. *X* position over time. (Test #2)

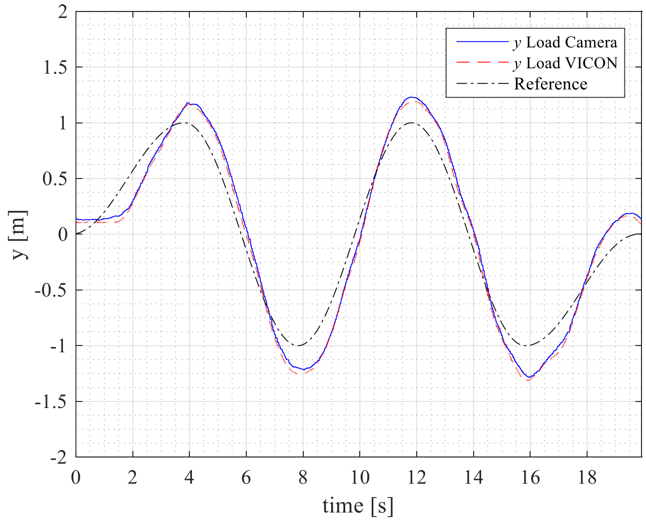


Figure 13. Flight test with load detection by camera. *Y* position over time. (Test #2)

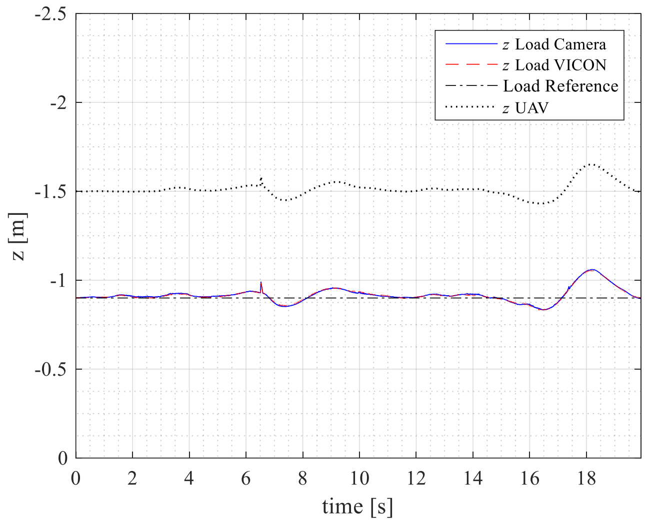


Figure 14. Flight test with load detection by camera. *Z* position over time. (Test #2)

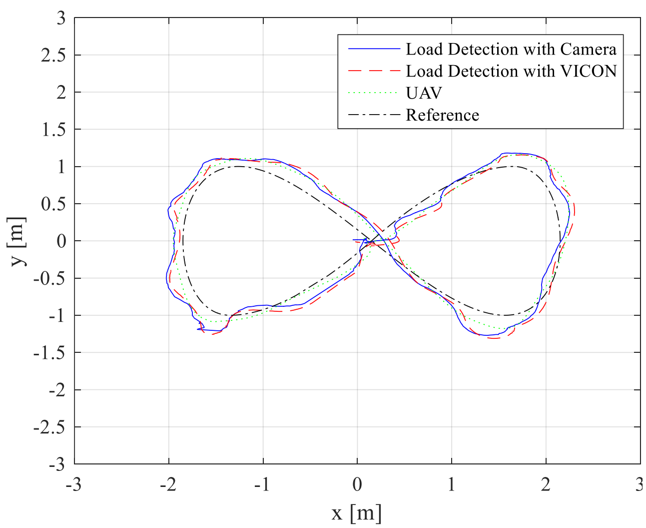


Figure 15. Flight test with load detection by VICON in 2D. (Test #3)

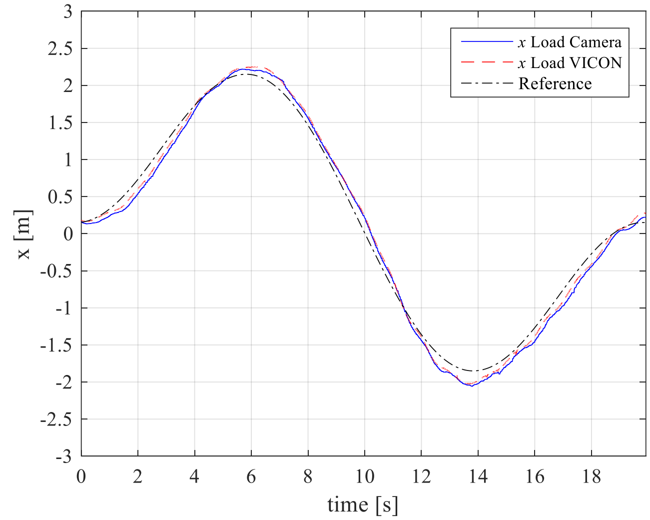


Figure 16. Flight test with load detection by VICON. *X* position over time. (Test #3)

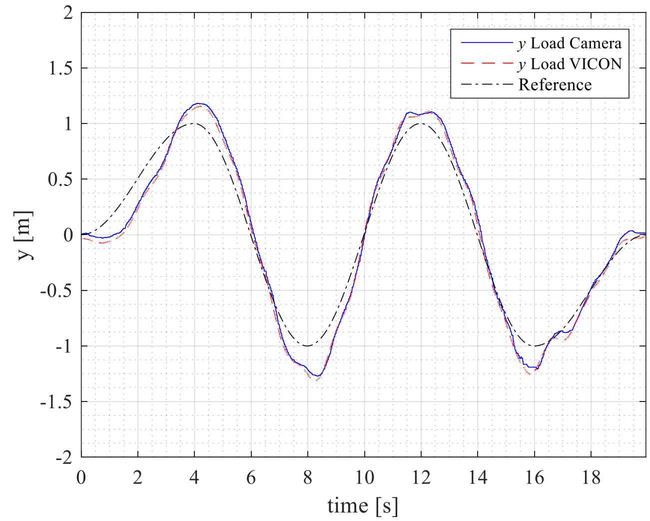


Figure 17. Flight test with load detection by VICON. *Y* position over time. (Test #3)

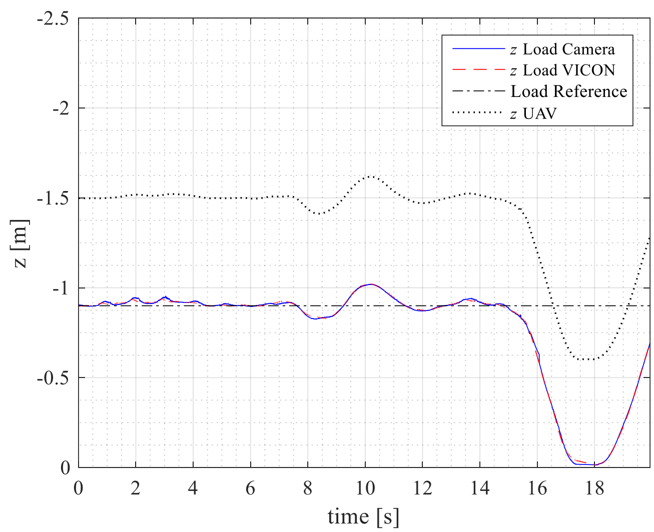


Figure 18. Flight test with load detection by VICON. *Z* position over time. (Test #3)

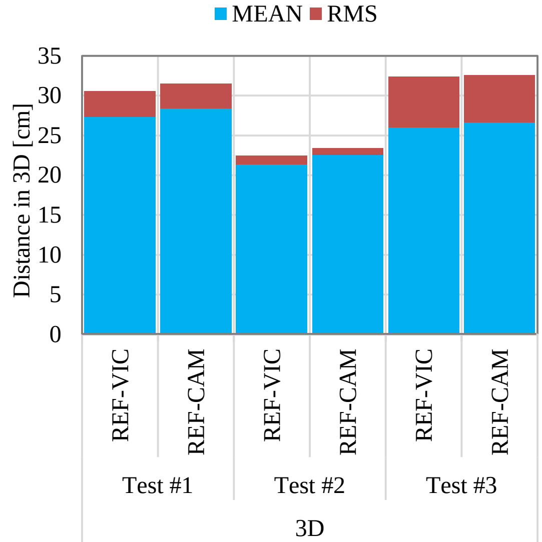


Figure 19. Mean and RMS values of the distance between the load position detected by the VICON and the load position detected by the camera in the inertial frame of reference

Figure 19 illustrates the distance between the load position detected by the VICON system and the reference (REF-VIC) and the load position detected by the camera and the reference (REF-CAM) in the inertial frame of reference in mean (blue) and root mean squared (RMS) (red) values. It can be seen that the controlling with visual feedback has no noticeable influence compared to the controlling with the VICON data for the load position (comparison of Test #1 and #2 with Test #3)[1].

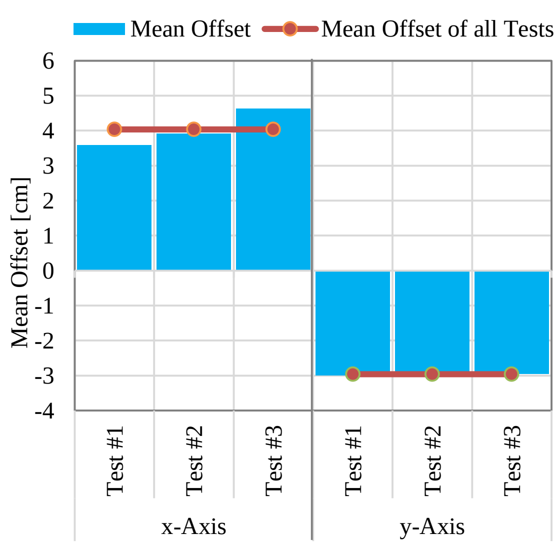


Figure 20. Mean offset between the load position detected by the VICON and the load position detected by the camera in x and y

[1] The table in the Appendix displays the separated data for *x*-, *y*-, and *z*-plane.

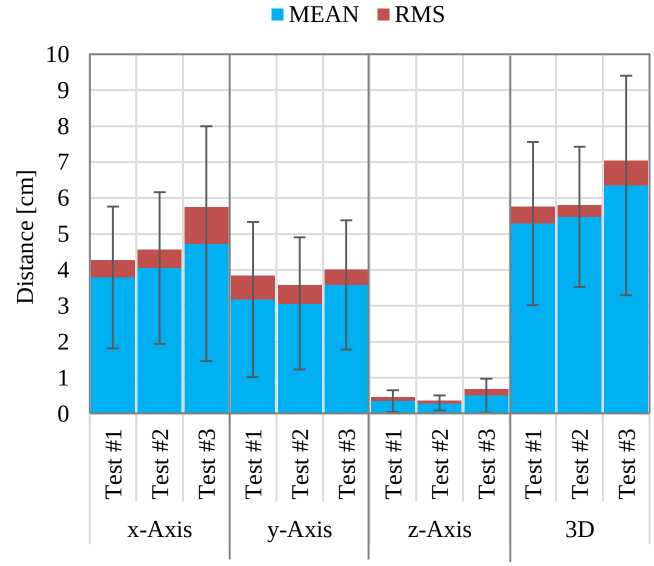


Figure 21. Mean distance between the load position detected by the VICON and the load position detected by the camera in x, y, z and the inertial frame of reference (3D)

However the distance between the load position detected by the VICON and the reference is a little bit smaller. In order to investigate this, the mean and RMS values between the load position detected by the VICON and the load position detected by the camera is considered.

Figure 20 shows the mean offset values of all tests respectively (blue) and the mean offset of all test together

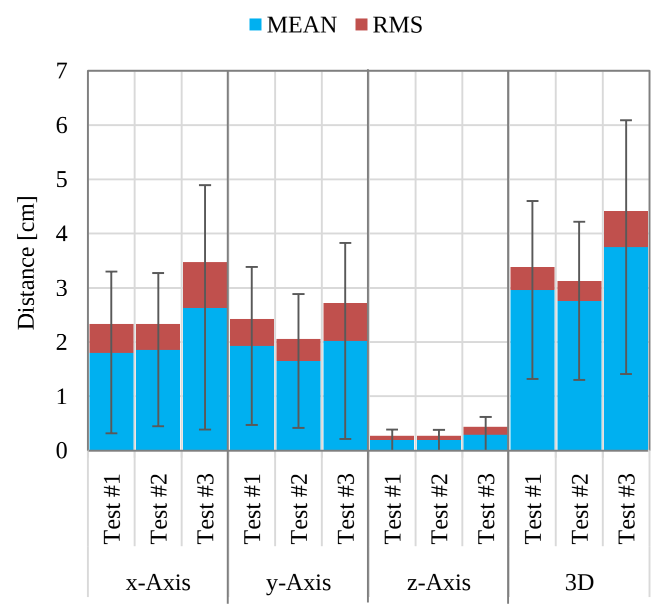


Figure 22. Simulated Mean distance between the load position detected by the VICON and the load position detected by the camera in x, y, z and the inertial frame of reference (3D) with mean offset in x and y considered.

(red) in *x* and *y*. Figure 21 illustrates the mean (blue) and RMS (red) distance in *x*, *y*, *z*, in the inertial frame of reference (3D) as well as the standard deviation (StD).

In Figure 20 it can be seen that the mean offset of all tests is almost equal to the mean offset of each test respectively. Therefore, it can be concluded that the mean offset of all tests is an offset in the hardware itself which means an offset between the center of the load in the VICON frame and center of the load in the Camera frame. This leads to the conclusion that it is an issue of calibration and can be implemented in the sensor model. Knowing that, the distance values of Figure 21 can be re-calculated considering the overall mean offset in *x* and *y* (Figure 20) between the load position detected by the VICON and the load position detected by the Camera to simulate a precise calibration. Adding these offset values to the load position detected by the camera, as it can be done in the sensor model, leads to the results shown in Figure 22. These results show that a precise calibration can improve the performance.

The overall values of the precision of the camera load detection are displayed in Table 2.

Table 2. Average Mean and RMS values of Test #1, #2 and #3 between the load position detected by the VICON and the load position detected by the Camera

		x [cm]	y [cm]	z [cm]	3D [cm]
Results	MEAN	4.19	3.27	0.38	5.7
	RMS	4.68	3.81	0.5	6.2
Results with simulated Offset	MEAN	2.1	1.86	0.22	3.15
	RMS	2.71	2.39	0.32	3.64

It can be seen that with a precise calibration a precision of up to 3.15 cm can be reached with the visual detection system. With a cord length *l* of 60cm this means a deviation of the center of the suspended load of ~3° degrees based on the suspension point or ~5% in relation to the cord length.
This result shows that the visual tracking method is very accurate.

6. Conclusion

This paper presented a new method to obtain the position of the load in a coupled system of UAV with suspended slung load.

The comparison between the load detection with the camera and the detection with the VICON system shows, that a simple camera system could be used as an alternative for the motion tracking system. The tests revealed that the difference between the tracking methods lies in a reasonable area with insignificant effects on the controller and system behavior. These initial results suggest that the visual object detection system would be capable for both, indoor and outdoor use. In order to get fully independent from any motion tracking system data different approaches are possible. On the one hand, a fully visual predictive control approach can be implemented for indoor use [18], on the other hand a combination of the presented visual tracking method and the use of GPS/IMU to obtain the UAVs state can be developed for outdoor use. The latter is a subject of ongoing research.

Acknowledgements

The authors would like to acknowledge ARCAA (Australian Research Centre for Aerospace Automation) for the use of the indoor flight test area and equipment. They also want to thank the iFR (Institute of Flight Mechanics and Control of the University of Stuttgart) for the exchange and cooperation.

References

[1] F. Gonzalez, MPG Castro, P. Narayan, R. Walker and L. Zeller, Development of an autonomous unmanned aerial system to collect time-stamped samples from the atmosphere and localize potential pathogen sources, in *Journal of Field Robotics*, pp. 961--976, 2011.

[2] D. S. Lee, L. F. Gonzalez, J. Périaux and S. K., Evolutionary Optimisation Methods with Uncertainty for Modern Multidisciplinary Design in Aeronautical Engineering, in *100 Volumes of 'Notes on Numerical Fluid Mechanics', Springer Berlin Heidelberg*, pp. 271-284, 2009.

[3] A. Malaver, F. Gonzalez, A. Depari, P. Corke and N. Motta, Towards the development of a gas sensor system for monitoring pollutant gases in the low troposphere using small unmanned aerial vehicles, in *Proceedings of Workshop on Robotics for Environmental Monitoring*, Sydney, Australia (Vol. 19), 2012.

[4] L. F. Gonzalez, Robust evolutionary methods for multi-objective and multdisciplinary design optimisation in aeronautics, University of Sydney, 2005.

[5] J. E. Trachte, L. F. Gonzalez and A. McFadyen, Multi-rotor with suspended load: System dynamics and control toolbox., in *Proceedings of 2015 IEEE Aerospace Conference*, Yellowstone Conference Center, Big Sky, Montana, 2015.

[6] J. How P, B. Bethke, A. Frank, D. Dale, J. Vian, Real-time indoor autonomous vehicle test environment, in *Control Systems, IEEE,* Nr. 28, pp. 51--64, 2008.

[7] S. Lupashin, M. Hehn, M. W. Mueller, A. P. Schoellig, M. Sherback and R. D'Andrea, A platform for aerial robotics research and demonstration: The Flying Machine Arena, in *Mechatronics*, no. 24, pp. 41--54, 2014.

[8] S. Notter, A. Heckmann, A. McFayden and F. Gonzalez, Modelling, Simulation and Flight Test of a Model Predictive Controlled Multirotor with Heavy

Slung Load, in *Submitted to IFAC Symposium on Automatic Control in Aerospace, Sherbrooke, Quebec, Canada 2016.*, 2016.

[9] R. Findeisen, F. Allgöwer, An introduction to nonlinear model predictive control, in *21st Benelux Meeting on Systems and Control*, Veldhofen, 2002.

[10] S. Notter, Modelling, Simulation and Flight Test of a model predictive controlled Multirotor with heavy Slung Load, Master Thesis, 2015.

[11] VICON Motion Capture System, URL: http://www.vicon.com.

[12] A. Heckmann, FLICK: A Flexible Framework for Multirotor, Master Thesis, Master Thesis, 2015.

[13] S. Hamdi, S. Rafael Muñoz und M. Bence, "ar_sys", in Robotic Operating System (ROS), URL: http://wiki.ros.org/ar_sys, 2015.

[14] OpenCV, URL: http://opencv.org, 2015

[15] S. Garrido-Jurado, R. Munoz-Salinas, F.J Madrid-Cuevas, M.J. Marin-Jimenez, Automatic generation and detection of highly reliable fiducial markers under occlusion, 2015

[16] Raspberry Pi, www.raspberrypi.org, URL: https://www.raspberrypi.org/documentation/hardware/camera.md.

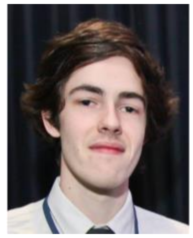
[17] MATLAB, Robotics System Toolbox, URL: http://de.mathworks.com/products/robotics/?refresh=true.

[18] A. McFadyen and e. al., Aircraft collision avoidance using spherical visual predictive control and single point features, in *Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, 2013.

BIOGRAPHY

Markus Zürn *received his B.Sc. degree in Aerospace Engineering from the University of Stuttgart in 2014. He was a research exchange student at ARCAA and QUT in 2015 and is currently studying a master's degree at the University of Stuttgart.*

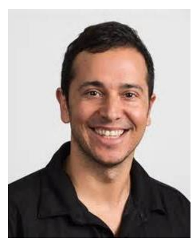
Aaron McFadyen *received the B.Eng. and Ph.D. degrees in robotics and aerospace from Queensland University of Technology (QUT). He holds a private pilot's license, has professional flight simulation engineering experience (CAE), and is currently a Research Fellow at the Australian Research Centre for Aerospace Automation (ARCAA) at QUT. His research interests include the application of machine learning, control and optimization theories to airspace integration problems related to unmanned aircraft systems.*



Kye Morton *is currently studying a doctorate of philosophy at the Queensland University of Technology (QUT). He holds a Bachelor of Engineering in Aerospace and Avionics from QUT in 2014. His current developments are made through ARCAA and QUT in UAV control methods.*

Stefan Notter *received his M.Sc. in Aerospace Engineering from University of Stuttgart. He wrote his master's thesis on model predictive control of quadrotor UAVs at ARCAA in 2014. Currently, he is working as a Research Assistant at the Institute of Flight Mechanics and Control (iFR) at University of Stuttgart. His research is focused on developing efficient control and path planning algorithms for unmanned aircraft systems.*

Alexander Heckmann *received his M.Sc. degree in Aerospace Engineering from University of Stuttgart. He wrote his master's thesis on model predictive control of quadrotor UAVs at ARCAA in 2014 and is currently working for Daimler AG.*



Felipe Gonzalez *is a Senior Lecturer (UAV) in the Science and Engineering Faculty, QUT and Team leader for UAV Remote Sensing at ARCAA. He holds a BEng (Mech) and a PhD from the University of Sydney. His research explores bioinspired optimization, uncertainty based UAV path planning and UAVs for environmental monitoring. He leads the CRC Plant Biosecurity project Evaluating Unmanned Aerial Systems for Deployment in Plant Biosecurity and the Green Falcon Solar Powered UAV. Felipe is a Chartered Professional Engineer and member of professional organizations including the RAeS, IEEE and AIAA.*

APPENDIX

Distance between the load position detected by the VICON or the load position detected by the camera and the reference in 3D. MEAN, RMS and StD. (All values are in cm).

			MEAN	**RMS**	**StD**
x-Axis	Test #1	REF-VIC	13.16	16.3	9.62
		REF-CAM	14.14	17.14	9.69
	Test #2	REF-VIC	9.95	11.25	5.26
		REF-CAM	11.91	13.23	5.75
	Test #3	REF-VIC	9.29	11.08	6.03
		REF-CAM	10.16	12.08	6.53
y-Axis	Test #1	REF-VIC	16.54	20.03	11.31
		REF-CAM	17	20.79	11.96
	Test #2	REF-VIC	17.27	18.94	7.79
		REF-CAM	17.25	18.83	7.54
	Test #3	REF-VIC	13.4	15.87	8.51
		REF-CAM	13.38	15.56	7.95
z-Axis	Test #1	REF-VIC	9.82	16.37	13.11
		REF-CAM	9.86	16.3	12.98
	Test #2	REF-VIC	2.76	4.25	3.22
		REF-CAM	2.9	4.35	3.24
	Test #3	REF-VIC	11.92	25.95	23.05
		REF-CAM	11.98	25.95	23.1
3D	Test #1	REF-VIC	27.32	30.58	13.74
		REF-CAM	28.38	31.49	13.66
	Test #2	REF-VIC	21.37	22.44	6.84
		REF-CAM	22.58	23.42	6.21
	Test #3	REF-VIC	26	32.37	19.29
		REF-CAM	26.63	32.58	18.77