



Minimum Swing Control of a UAV with a Cable Suspended Load

M. (Martijn) Weijers

MSc Report

Committee:

Prof.dr.ir. S. Stramigioli

Dr. R. Carloni

Dr.ir. R.G.K.M. Aarts

Dr.ir. J.F. Broenink

November 2015

035RAM2015

Robotics and Mechatronics

EE-Math-CS

University of Twente

P.O. Box 217

7500 AE Enschede

The Netherlands

CONTENTS

Introduction to the project	5
-A Report Outline	5
I Introduction	6
II Dynamics	6
II-A A quadrotor UAV with a cable suspended Load	6
III Control Design	7
III-A Passivity Based Control	7
III-B Control of a quadrotor UAV with a cable suspended load	7
III-C Lyapunov Stability	7
IV Simulations	8
V Experiments	8
V-A Discussion on experiments	9
VI Conclusions & Recommendations	11
VI-A Conclusion	11
VI-B Recommendations for future work	11
References	11
Appendix A: Control Design with System Decoupling	12
A-A Lyapunov Analysis	13
A-B Simulations	13
A-C Experiments	13
A-D Conclusion	13
Appendix B: Manual	15
B-A Prerequisites	15
B-B Installation	15
B-C Usage of package	15
B-D How to fly with the load	16

INTRODUCTION TO THE PROJECT

Transportation by unmanned aerial vehicles (UAVs) is of growing interest in the last couple of years, Especially in the field of search and rescue. The SHERPA project is an example this. The aim of this project is to develop a robotic platform with several ground robots and UAVs to help a team of people to find for example a person who is buried under the snow due to an avalanche. The UAVs will help here by doing specific tasks. For example is designed to make a map of the surrounding or scan the environment to find the person to be rescued. Also they can help transporting certain payloads to dangerous sites which are not reachable by the ground vehicles. During transportation by a UAV, swings can happen in the payload which can cause damage to both the UAV and the payload.

The aim of this project is the design of a passivity based controller such that the swing of the cable suspended payload is not destabilizing the UAV during transportation. Also the controller should be able to meanwhile control the UAV to a desired position. A Lyapunov stability analysis is done to show stability in the controller. The designed controller is verified by simulation and in experiment by using the Crazyflie 2.0.

A. Report Outline

In Section I to Section VI a paper is written where the passivity based controller is derived and validated by simulations and experiments. In Appendix A an other controller for stabilizing the load swing is proposed which uses decoupled dynamics of the UAV and the load dynamics. A manual of how to use this controller and how to install the necessary software can be found in Appendix B.

Minimum swing control of a UAV with a cable suspended load

Martijn Weijers and Raffaella Carloni

Abstract—This paper focuses on the modelling and control of an unmanned aerial vehicle (UAV), which is transporting a cable suspended load. A cable suspended load when not controlled drastically changes the dynamics of the flying UAV which can result in an unstable system. A passivity based control technique is used to control the UAV such that minimum swing is achieved. To design the controller a 2D model is considered which is implemented and tested in a 3D simulation environment. The control architecture is finally validated in simulation and during experimental tests.

I. INTRODUCTION

The research community is devoting a growing interest towards aerial service robots, a new generation of unmanned aerial vehicles (UAVs) that can support human beings in activities such as inspection [1] and search and rescue [2]. In such complex activities, the UAVs are often required to transport certain payloads such as equipment to dangerous sites which are inaccessible by ground vehicles. The transportation by UAVs can be done in different ways: by cooperative aerial manipulation [3] on the load or by using cables attached to the UAVs and the payload [4], [5].

This paper focuses on the transportation of a cable suspended load and on reducing the swing when going to a desired location. In [4], [6] a non-linear controller is designed based differentially flat system properties. These properties are used to find executable trajectories for the load. In [7], the swing is minimized by considering the cable as multiple rigid links and they achieve an asymptotically stable position via non-linear control. In [5], the control is performed by using a saturation controller.

This paper contributes by designing an energy-based passive controller that is able to asymptotically stabilize the load swing during transportation. With a Lyapunov-based energy function the control parameters can be obtained such that a stable equilibrium is guaranteed. The controller is designed in a way that moving from point to point is smooth and without any aggressive movement of the UAV.

For experiments and validation a small agile UAV is used. The advantage for using small UAVs for load transportation is that multiple UAVs can cooperate with each other to transport payloads that can not be carried by one single UAV. From these simulations and experiments can be seen that a single UAV is

able to transport a small payload with ensuring a minimal swing in the load.

The paper is organized as follows. In Section II the dynamics of one UAV with load is derived and discussed. In Section III the passivity based control design is described and Lyapunov stability is shown. In Section IV the results obtained by simulation are presented. The results obtained by experiment are shown in V. Conclusions and recommendations for future work are drawn in Section VI.

II. DYNAMICS

In this section the dynamics of a UAV carrying a load is derived. The is constructed in a 2D environment to reduce the mathematical complexity but still able to secure the main features of the 3D case.

A. A quadrotor UAV with a cable suspended Load

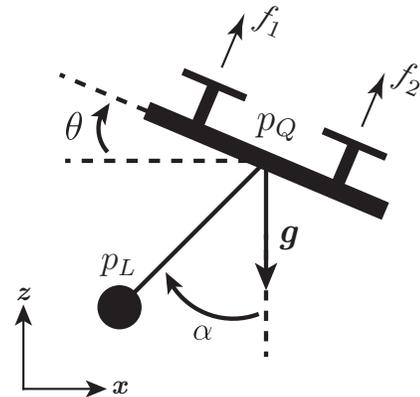


Fig. 1. 2D UAV model with a cable suspended load. f_1 and f_2 are the forces generated by the two propellers. p_Q is a vector representing the position of the UAV and p_L the position of the load. θ is the pitch angle with respect to the horizontal position. α is the angle of the load with respect to the z -axis and g is the gravity constant

In Fig. 1 a sketch is shown of the UAV with cable suspended load that need to be modelled and controlled. The center of mass position of the quadrotor and load can be described in inertial frame with $p_Q = [x, z]^T \in \mathbb{R}^2$ the quadrotor position and $p_L = [x_L, z_L]^T \in \mathbb{R}^2$ the position of the load. The attitude is described by $\theta \in \mathbb{R}$ with respect to the horizontal position. The rotation of the load with the z -axis is described by $\alpha \in \mathbb{R}$. The UAV is actuated by two propellers generating the forces f_1 and $f_2 \in \mathbb{R} \geq 0$. The assumption is that the cable is rigid, inextensible and massless. Using the Euler-Lagrange

This work has been funded by the European Commission's Seventh Framework Programme as part of the project SHERPA under grant no. 600958.

The authors are with the Faculty of Electrical Engineering, Mathematics and Computer Science, CTIT Institute, University of Twente, The Netherlands. Emails: m.weijers@alumnus.utwente.nl, r.carloni@utwente.nl

formalism the dynamics of the UAV and load are obtained [5], i.e.,

$$(M + m)(\ddot{z} + g) + mL(\cos \alpha \dot{\alpha}^2 + \sin \alpha \ddot{\alpha}) = u_1 \cos \theta \quad (1a)$$

$$(M + m)\ddot{x} + mL(\sin \alpha \dot{\alpha}^2 - \cos \alpha \ddot{\alpha}) = u_1 \sin \theta \quad (1b)$$

$$mL^2\ddot{\alpha} + mL \sin \alpha \dot{\alpha}^2 - mL \cos \alpha \ddot{\alpha} + mgL \sin \alpha = 0 \quad (1c)$$

$$J\ddot{\theta} = u_2 \quad (1d)$$

Where:

- M, m : are the mass of quadrotor and load, respectively
- $u_1 = f_1 + f_2$: is the total thrust generated by the propellers
- L : is the length of the cable
- J : is the inertia of the quadrotor
- $u_2 = (f_1 - f_2)b$: is the moment due to the propellers where b is the distance between the center of mass and the motors.

III. CONTROL DESIGN

In this section the passivity based control design of the system is presented and the stability is analysed by using Lyapunov theory. The goal of the controller is to move from point to point while the swing in the load stay bounded near the equilibrium.

A. Passivity Based Control

The idea behind passivity based control is that each subsystem consist of an certain energy. By adding these energies together the energy of the overall system is obtained. By controlling this energy such that the used energy is minimized a stabilized system can be obtained [8]. Due to the passivity property there will be no more energy stored then that is supplied to the system.

B. Control of a quadrotor UAV with a cable suspended load

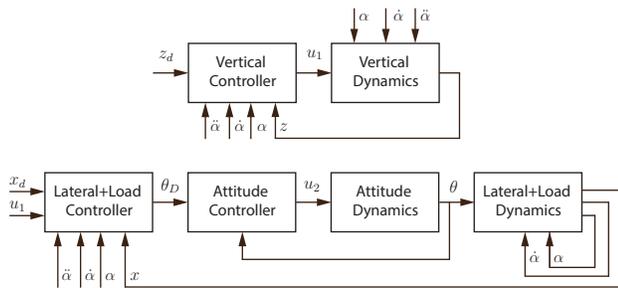


Fig. 2. Control architecture of the UAV with the cable suspended load

To control the UAV with the cable suspended load the control architecture as described in Fig. 2 is used. It can be seen that a fast high gain attitude controller is used such that $\theta \approx \theta_D$ where θ_D is the desired pitch angle. Furthermore is assumed that during movement the UAV stays near the hovering position: This means that $\theta_D \approx 0$, and therefore that $\cos \theta_D \approx 1$ and $\sin \theta_D \approx \theta_D$. From Fig. 2 it can also be seen that the lateral and load dynamics are taken together. This is because both dynamics are coupled with each other (1b),(1c). Within this controller can be assumed that $\ddot{z} \rightarrow 0$ because the

vertical position is assumed to be at a constant altitude during movement to a desired location. This controller is controlling α to zero and x to x_d . With this the reduced load dynamics is obtained:

$$\lim_{\alpha \rightarrow 0} \ddot{\alpha} = \frac{1}{L} \ddot{x} \quad (2)$$

The vertical dynamics are affected by the load dynamics and the controller is controlling z to z_d with the input u_1 . The control inputs u_1 and θ_D should be designed such that:

$$\ddot{z} = -k_p^z(z - z_d) - k_d^z \dot{z} \quad (3)$$

$$\ddot{x} + \ddot{\alpha} = -k_p^x(x - x_d + \alpha) - k_d^x(\dot{x} + \dot{\alpha}) \quad (4)$$

This is obtained by choosing the inputs as:

$$u_1 = (M + m)(u_z + g) \quad (5)$$

$$\theta_D = \frac{ML + mL \sin^2 \alpha}{u_1(L + \cos \alpha)} (-k_p^x(x - x_d + \alpha) - k_d^x(\dot{x} + \dot{\alpha}) + \frac{mL \sin \alpha(L + \cos \alpha)}{ML + mL \sin^2 \alpha} \dot{\alpha}^2 - \frac{(M + m)(M + m + mL \cos \alpha)}{ML + mL \sin^2 \alpha}) \quad (6)$$

This can be proofed by the following. When considering the hovering position the vertical dynamics can be approximated by [9]:

$$\ddot{z} = u_z + d_z \quad (7)$$

with u_z the input to the system and d_z an exogenous disturbance. From this, u_z can be described by the following to obtain PD dynamics in the vertical dynamics.

$$u_z = -k_p^z(z - z_d) - k_d^z \dot{z} + \frac{mL}{M + m} (\cos \alpha \dot{\alpha}^2 + \sin \alpha \ddot{\alpha}) \quad (8)$$

The combined dynamics of x and α need to be described with the following equation to also obtain PD dynamics.

$$\ddot{x} + \ddot{\alpha} = \frac{u_1(L + \cos \alpha)}{ML + mL \sin^2 \alpha} \theta_D - \frac{mL \sin \alpha(L + \cos \alpha)}{ML + mL \sin^2 \alpha} \dot{\alpha}^2 - \frac{(M + m)(M + m + mL \cos \alpha)}{ML + mL \sin^2 \alpha} \quad (9)$$

C. Lyapunov Stability

To see if the designed controller will result in a stable system a proof of stability is performed by using a Lyapunov function [10]. For this system, the following candidate Lyapunov function is chosen:

$$V = \frac{1}{2}(\dot{x} + \dot{\alpha})^2 + \frac{1}{2}\dot{z}^2 + \frac{1}{2}k_p^x(x - x_d + \alpha)^2 + \frac{1}{2}k_p^z(z - z_d)^2 \quad (10)$$

The following is obtained when this function is differentiated.

$$\dot{V} = (\dot{x} + \dot{\alpha})(\ddot{x} + \ddot{\alpha}) + \dot{z}\ddot{z} + k_p^x(x - x_d + \alpha)(\dot{x} + \dot{\alpha}) + k_p^z(z - z_d)\dot{z} \quad (11)$$

$\frac{1}{2}(\dot{\cdot})^2$ denotes the kinetic energy and $\frac{1}{2}k_p^\beta(\dots)^2$ the potential energy of the system [9]. In (12) the result is shown when the

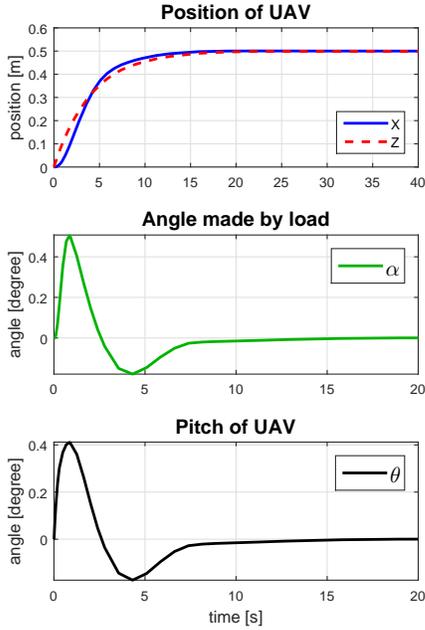


Fig. 3. Position response of the system if a setpoint of 0.5 is given in x and z. The second plot shows that the load stabilises after this position change. The pitch angle of the UAV stays near zero.

lateral and vertical dynamics are included, which results in cancelling out some terms.

$$\dot{V} = -k_d^x(\dot{x} + \dot{\alpha})^2 - k_d^z\dot{z}^2 \leq 0 \quad (12)$$

This result shows that the Lyapunov function is negative semi-definite which means that x_d and z_d are stable setpoints. With this derivation, stability is guaranteed for the lateral, vertical and load dynamics. Using LaSalle's Theorem asymptotically stability is shown.

IV. SIMULATIONS

This section discusses the results obtained via simulations. To validate the designed controller two simulations in 20sim [11], a modelling and simulation program, are performed. In the first experiment a setpoint in x and z of 0.5m is given to validate the setpoint response of the proposed controller. The gains are chosen such that the resulting swing is bounded and that the setpoint response is smooth without an overshoot. The result of this simulation is shown in Fig. 3. It can be noticed that the setpoint is reached smoothly without an overshoot. In the load angle part can be seen that the angle is kept bounded during movement. The same holds for the pitch angle θ .

In the second test a hovering position is kept and a disturbance is given to load. In this way the behaviour can be seen of how the system responds to a large swing in the load. The result of this simulation is shown in Fig. 4. To compensate for the swing in the load a small disturbance can be observed in the x position of the UAV. The plot shows also that the angle in the load is reduced quickly and that the pitch angle θ stays near zero. It also show that the UAV is counteracting the swing by moving in the other directions.

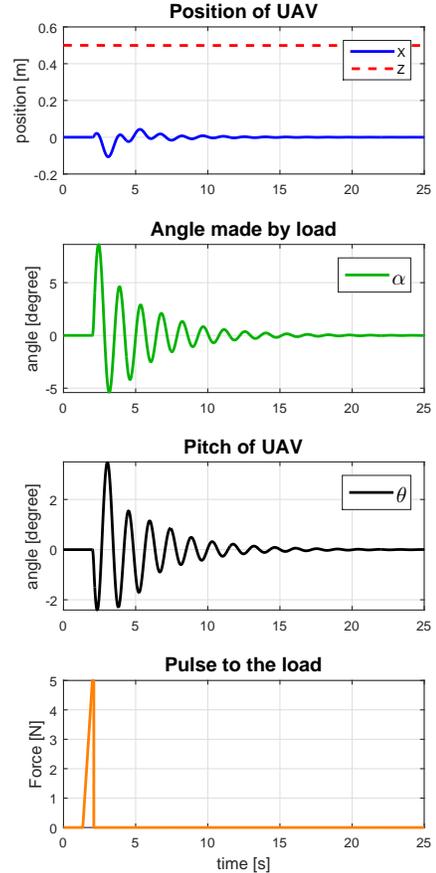


Fig. 4. Result when a disturbance after 2 seconds is given to the load. To compensate for the swing the UAV need to change his x-position. The angle with the load is asymptotically going to his equilibrium due to the counteracting of the UAV.



Fig. 6. The CrazyFlie 2.0 [12]

V. EXPERIMENTS

In this section the results obtained by experiment will be discussed. In Fig. 5 an overview of the overall setup is shown. The Crazyflie 2.0 [12] as shown in Fig. 6 will be used to test the designed controllers in experiment. The communication of the controller will go via ROS [13] which is a flexible framework for writing robot software and makes communication possible between different platforms. The optitrack system is used to track the Crazyflie and the load [14]. With this system the position and attitude of the Crazyflie can accurately be

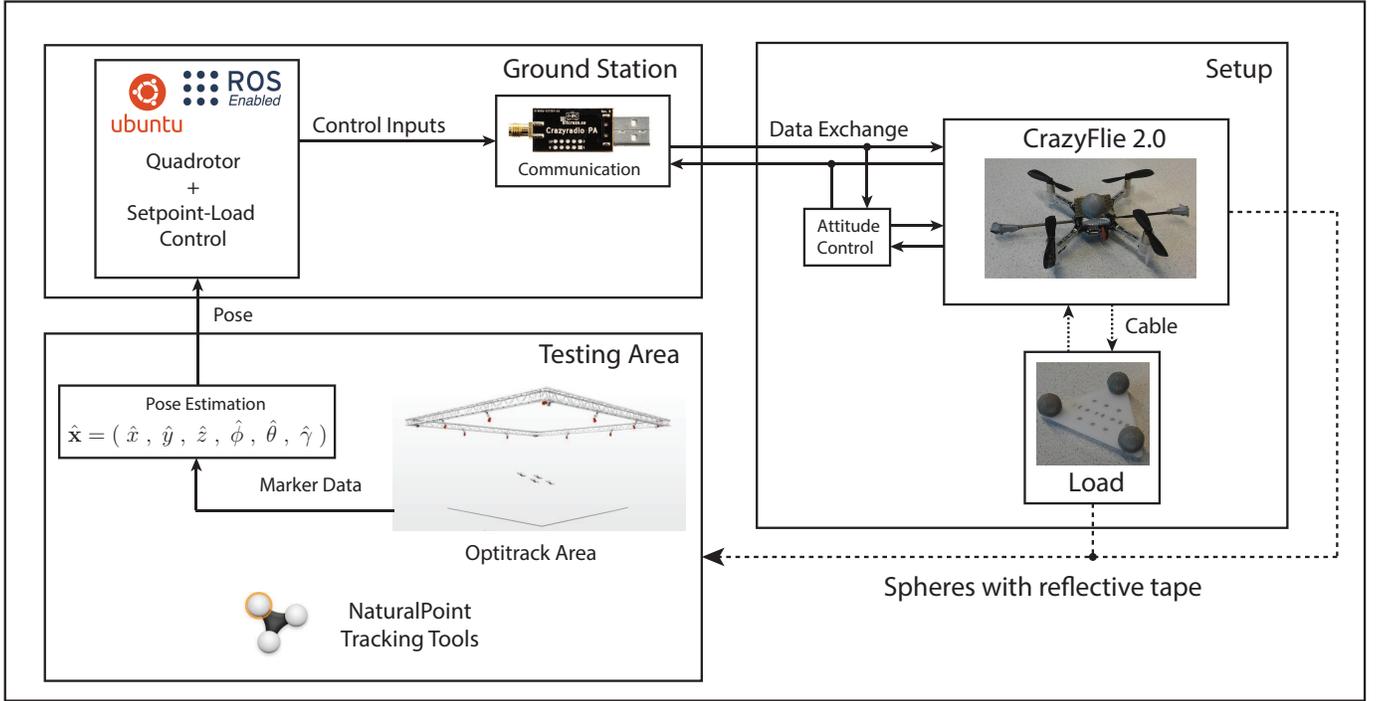


Fig. 5. System Overview: the overall system consisting of one CrazyFlye 2.0 to which a load is connected. The setup to which reflective markers are attached is placed in an optitrack area. A motion tracking system is used to detect the trackable position from which the absolute pose is estimated. The ground station is used to communicate with the UAV via a radio link and to run the setpoint and load controller. Over this radio link the control inputs can be sent. The robotic operating system (ROS) is used on ubuntu to communicate between the different software modules and the Crazyflye.

measured. The maximum recommended amount of payload to be carried is limited to 15g [15]. In this experiment a payload of 12.4g is used. This is the combination of 8.6 g load and 3.8 g for the optitrack markers and the carbon tube to attach the reflective markers on. For the cable a length of 1m is taken which is 95cm after attaching to the Crazyflye.

The Crazyflye itself is controlled via an internal attitude controller and a thrust for the vertical direction. Via ROS only a velocity can be sent as an input for the lateral direction to the UAV. For this it is assumed that for small angle changes it can be considered as setting a velocity. For the error the difference is taken between the current position and desired position. To obtain the velocity and acceleration of the angle a state variable filter is used based on a butterworth low pass filter [16]. To obtain the acceleration a third order filter is needed. The transfer function of this filter is as follows:

$$H(s) = \frac{\omega_c^3}{s^3 + 2\omega_c s^2 + 2\omega_c^2 s + \omega_c^3} \quad (13)$$

Where ω_c is the desired cut-off frequency. By using this filter a phase shift is introduced in the outputs. By increasing the cut-off frequency this shift is lowered but more noise is allowed on the outputs. For this a consideration need to be made on how much noise or delay is allowed. For this experiment a cut-off frequency of 1.5Hz is taken. With this a phase delay of 0.3 seconds is introduced.

In Fig. 8 the result of the experiment with the UAV and load can be seen. It can be noticed that controller is able to make a stable flight but that the position of the UAV does

not converge to the desired setpoint. The swing of the load is kept to a minimum of 28° in the x-direction and 27° in the y-direction. This can be seen in Fig. 7. In the x-direction it has a setpoint error of 72cm and in the y-direction of 98cm. From Fig. 9 can be concluded that the assumption that the pitch and roll are small during control is achieved since they stay within 10° for the biggest part of the experiment. To see if the controller it self is stable an experiment is performed where no load is attached and the angle α is set to zero. In Fig. 10 the result of this experiment is shown. This plot shows that the UAV itself is able to stay in his desired position but that the load adds to much disturbance to achieve a stable system. A big offset can be noticed in the vertical direction due to the fact that the controller still compensates for the mass of the load with his initial thrust offset.

A. Discussion on experiments

In the result of the experiment while flying with the load can be seen that the UAV is unable to reach his desired setpoint. A reason for this can be that the controller is designed in 2D space and that experiment is performed in 3D. Due to this the setpoint is only fully controlled in the x-direction. To control in the y-direction, the x-direction is translated to the y-direction and by this not optimal. Also using the same controller gains for the pitch and roll direction is probably a problem since the moment in the roll direction is higher due to the added optitrack markers. Another reason for this problem is that the Crazyflye 2.0 can not generate enough thrust to compensate for

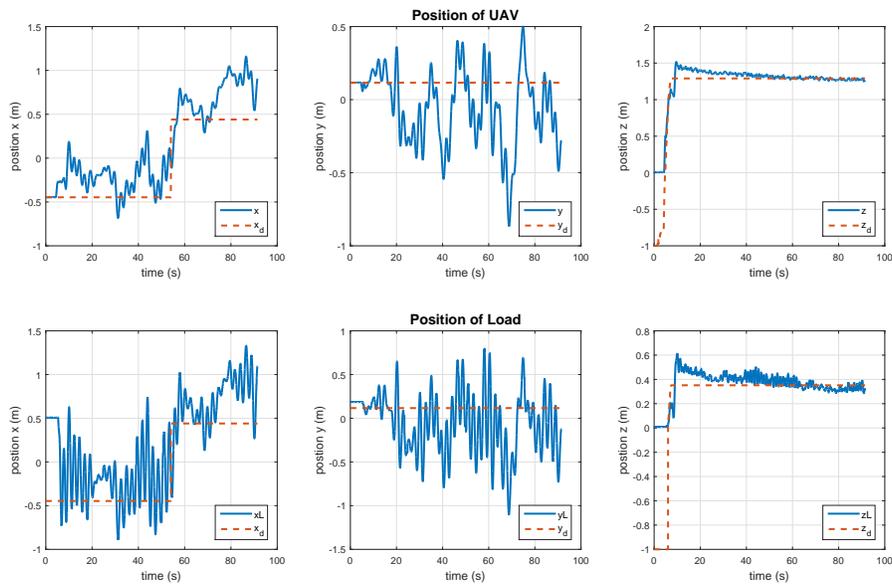


Fig. 8. Recorded Position of UAV and load during experiment. Setpoint z is -1 at the start to guarantee that the UAV stays one the ground when the communication between the UAV and PC is started.

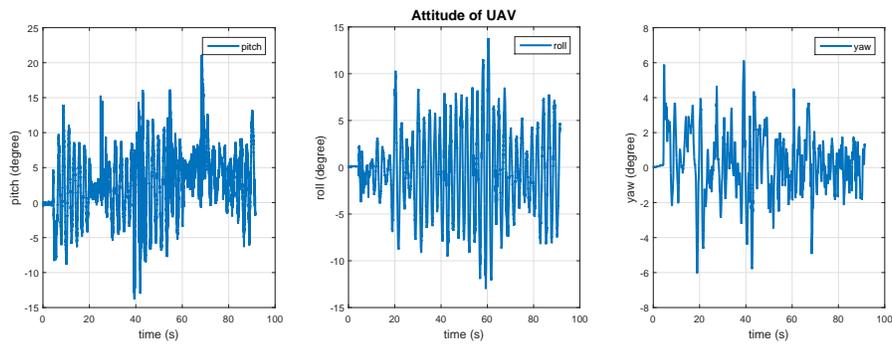


Fig. 9. Measured attitude of UAV during experiment.

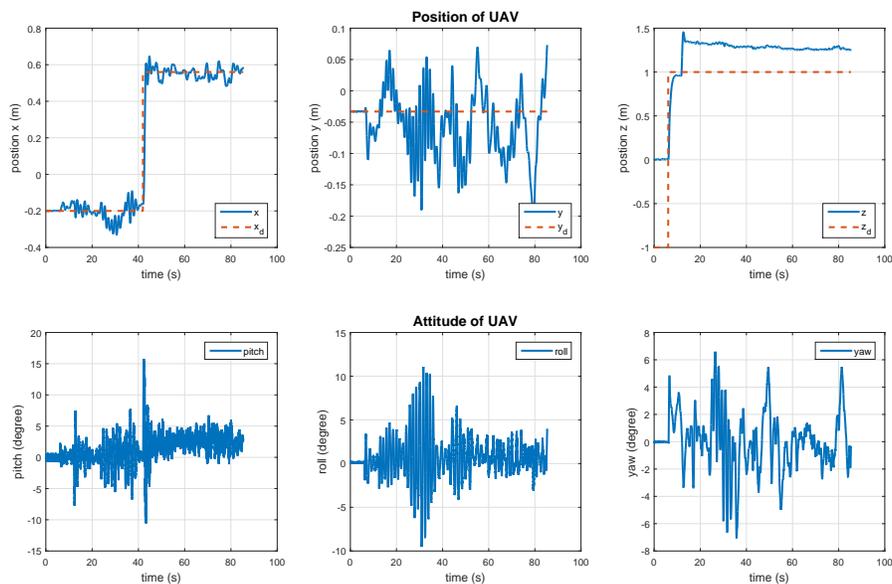


Fig. 10. Result experiment when no load is attached. With this can be seen that the controller is able to stay in a desired position and reach stably another position. A big offset in z -position is due to the fact that the controller still compensates for the load mass.

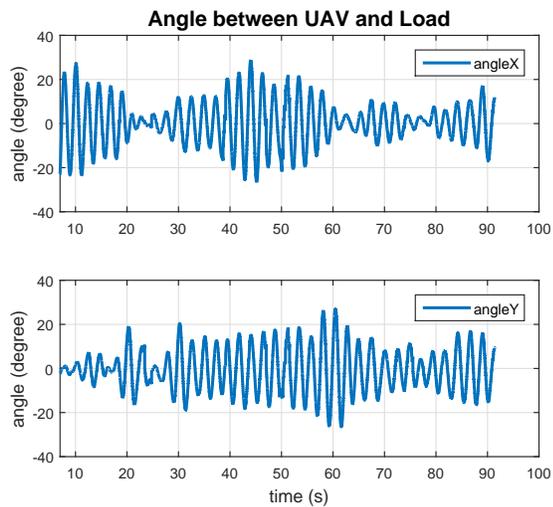


Fig. 7. Angle between the load and the UAV during the experiment. The plot starts at 7 seconds since on that moment the load controller is enabled. Further it shows that the angles stay bounded overtime

errors in the lateral directions when carrying a load. Since most of this thrust is already needed to stay in his vertical position. Also due to the small size of the Crazyflie the life time of the battery is not long and by this a recharge is needed after a couple of minutes flight time. This decay goes more rapidly when a load is attached. Also the controller performance starts to decrease when the battery is discharging. Taking the rope length to small will cause that the load is pushed away to much by the thrust of the UAV which is exerting a moment on the UAV which is also influencing the position of the UAV.

VI. CONCLUSIONS & RECOMMENDATIONS

A. Conclusion

In this work, a passivity based controller has been designed such that minimum swing is achieved in the load. Both controllers in simulation are performing as expected and can control the system to a desired setpoint with ensuring minimal swing. During experiment the UAV is able to damp out most of the swing after lift off. The system is also holding this bounded swing overtime but the system is not able to stabilize his position around the desired position. By not using the load and setting α to zero the load controller is able to stabilize around the desired position. Extensive controller tuning is performed to improve the setpoint response but without better results.

B. Recommendations for future work

In this project the controller is designed in a 2D space but tested in 3D space. Therefore to improve the controller it is recommended to model and design the controller in 3D space. To achieve better simulation Since the Crazyflie 2.0 are not performing well while carrying a load due to there limit payload and low weight it will be a good experiment to see how this kind of controller performs on a slightly bigger quadrotor. In this way the load will probably not be affected by the thrust to much. Also designing a separate controller for the pitch and roll direction can improve the performance.

Extending the Gazebo model such that a load can be added to the UAV is also recommended since then more accurate simulation can be performed. Extend work by using multiple drones is also recommended since in this way the advantages of small drones is utilized.

REFERENCES

- [1] L. Marconi, R. Naldi, A. Torre, J. Nikolic, C. Huerzeler, G. Caprari, E. Zwicker, B. Siciliano, V. Lippiello, R. Carloni, and S. Stramigioli, "Aerial service robots: An overview of the AIRobots activity," in *Proceedings of IEEE International Conference on Applied Robotics for the Power Industry*, 2012, pp. 76–77.
- [2] L. Marconi, C. Melchiorri, M. Beetz, D. Pangercic, R. Siegwart, S. Leutenegger, R. Carloni, S. Stramigioli, H. Bruyninckx, P. Doherty, A. Kleiner, V. Lippiello, A. Finzi, B. Siciliano, A. Sala, and N. Tomatis, "The SHERPA project: Smart collaboration between humans and ground-aerial robots for improving rescuing activities in alpine environments," in *Proceedings of IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2012, pp. 1–4.
- [3] G. Gioioso, A. Franchi, G. Salvietti, S. Scheggi, and D. Prattichizzo, "The flying hand: A formation of uavs for cooperative aerial tele-manipulation," in *Proceedings of IEEE International Conference on Robotics and Automation*, 2014, pp. 4335–4341.
- [4] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Robotics: Science and Systems (RSS)*, 2013.
- [5] M. Nicotra, E. Garone, R. Naldi, and L. Marconi, "Nested saturation control of an uav carrying a suspended load," in *Proceedings of American Control Conference*, June 2014, pp. 3585–3590.
- [6] K. Sreenath, T. Lee, and V. Kumar, "Geometric control and differential flatness of a quadrotor uav with a cable-suspended load," in *Proceeding of IEEE 52nd Annual Conference on Decision and Control*, 2013, pp. 2269–2274.
- [7] T. Lee, "Geometric control of multiple quadrotor uavs transporting a cable-suspended rigid body," in *Proceedings of IEEE 53rd Annual Conference on Decision and Control*, 2014.
- [8] R. Ortega, A. Van der Schaft, I. Mareels, and B. Maschke, "Putting energy back in control," *IEEE Control Systems*, vol. 21, no. 2, pp. 18–33, 2001.
- [9] M. Fumagalli, R. Naldi, A. Macchelli, F. Forte, A. Keemink, S. Stramigioli, R. Carloni, and L. Marconi, "Developing an aerial manipulator prototype: Physical interaction with the environment," *IEEE Robotics Automation Magazine*, vol. 21, no. 3, pp. 41–50, 2014.
- [10] H. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002. [Online]. Available: https://books.google.nl/books?id=t_d1QgAACAAJ
- [11] Controllab Products, "20-sim: The power in modeling." [Online]. Available: <http://www.20-sim.com/>
- [12] Bitcraze. (2015) The Crazyflie 2.0. [Online]. Available: <https://www.bitcraze.io/crazyflie-2/>
- [13] ROS, "ROS: An open-source Robot Operating System." [Online]. Available: <http://www.ros.org/about-ros/>
- [14] Naturalpoint, Inc, "Optical Tracking Solutions, optitrack: motion capture systems. Version 2.3.3." [Online]. Available: <http://www.optitrack.com/>
- [15] Bitcraze Wiki. (2015) Crazyflie 2.0 hardware specification. [Online]. Available: <https://wiki.bitcraze.io/projects:crazyflie2:hardware:specification>
- [16] Butterworth, S, "On the Theory of Filter Amplifiers," October 1930. [Online]. Available: <http://www.gonascient.com/papers/butter.pdf>
- [17] M. de Roo, "Optimal Event Handling By a UAV Network," Master's thesis, University of Twente., 2015.

APPENDIX A
CONTROL DESIGN WITH SYSTEM DECOUPLING

To obtain a passivity based controller with system decoupling the system should first be decoupled [5]. Due to this the system can be divided in two loops: The inner loop and outer loop. The inner loop will control the attitude of UAV by using the input u_2 . With the inputs u_1 and θ the outer loop will be controlled. With this is assumed that the vertical dynamics are not influenced by the lateral and load dynamics and controller by u_1 . These will be controlled by the desired pitch angle θ_D . For this the attitude controller will be designed fast such that $\theta \approx \theta_D$. In Fig. 11 the control architecture can be seen to achieve this goal [9]. The symbols x_d and z_d are the desired positions of the UAV. α is the angle between the UAV and the load.

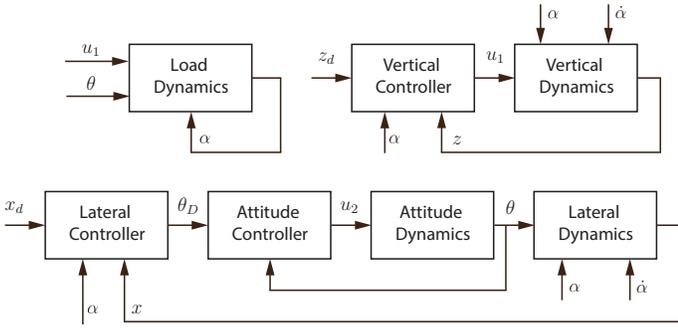


Fig. 11. Control architecture of the 1 UAV with the cable suspended load

The goal of this controller is to choose the controlling input u_1 and θ_D such that the resulting system is able to move stably from point to point and meanwhile bound the swing in the load near his equilibrium.

During the derivation of the controller the trigonometric identities in 14 and 15 will be used. For this is assumed that the UAV stays near his hovering position during his flight. With small θ_D it can be approximated that $\cos \theta_D \approx 1$ and $\sin \theta_D \approx \theta_D$. with this:

$$\begin{aligned} \cos(\theta - 2\alpha) &= \cos \theta \cos 2\alpha + \sin \theta \sin 2\alpha \\ &= \cos 2\alpha + \theta_D \sin 2\alpha \end{aligned} \quad (14)$$

$$\begin{aligned} \sin(\theta - 2\alpha) &= \sin \theta \cos 2\alpha - \cos \theta \sin 2\alpha \\ &= \theta_D \cos 2\alpha - \sin 2\alpha \end{aligned} \quad (15)$$

In (16) the derivation is shown of the dynamics in the z -direction. For simplification (14) is used.

$$\begin{aligned} \ddot{z} &= -g - \frac{m}{M+m}L \cos \alpha \dot{\alpha}^2 + \frac{1}{M+m}(\cos \theta \\ &+ \frac{m}{2M}(\cos \theta - \cos(\theta - 2\alpha)))u_1 \end{aligned} \quad (16a)$$

$$\begin{aligned} &= -g - \frac{m}{M+m}L \cos \alpha \dot{\alpha}^2 \\ &+ \frac{1}{M+m}(1 + \frac{m}{2M}(1 - \cos 2\alpha - \theta_D \sin 2\alpha))u_1 \end{aligned} \quad (16b)$$

$$= -g + \frac{u_1}{M+m} + \frac{u_1 m}{2M(M+m)}(1 - \cos 2\alpha - \theta_D \sin 2\alpha)$$

$$- \frac{m}{M+m}L \cos \alpha \dot{\alpha}^2 \quad (16c)$$

With (15) the lateral dynamics in (17) are simplified.

$$\begin{aligned} \ddot{x} &= -\frac{m}{M+m}L \sin \alpha \dot{\alpha}^2 + \frac{1}{M+m}(\sin \theta \\ &+ \frac{m}{2M}(\sin \theta - \sin(\theta - 2\alpha)))u_1 \end{aligned} \quad (17a)$$

$$\begin{aligned} &= -\frac{m}{M+m}L \sin \alpha \dot{\alpha}^2 + \frac{1}{M+m}(\theta_D \\ &+ \frac{m}{2M}(\theta_D - \theta_D \cos 2\alpha - \sin 2\alpha))u_1 \end{aligned} \quad (17b)$$

$$\begin{aligned} &= g\theta_D + \frac{gm}{2M}(\theta_D - \theta_D \cos 2\alpha) \\ &- \frac{m}{M+m}L \sin \alpha \dot{\alpha}^2 - \frac{gm}{2M} \sin 2\alpha \end{aligned} \quad (17c)$$

By defining A and B as follows the z and x dynamics can be simplified:

$$A = (1 + \frac{m}{2M}(1 - \cos \alpha - \theta_D \sin 2\alpha)) \quad (18)$$

$$B = (g + \frac{gm}{2M}(1 - \cos 2\alpha)) \quad (19)$$

This results in the following dynamics for \ddot{z} and \ddot{x} :

$$\ddot{z} = -g - \frac{m}{M+m}L \cos \alpha \dot{\alpha}^2 + \frac{1}{M+m}Au_1 \quad (20a)$$

$$\ddot{x} = B\theta_D - \frac{m}{M+m}L \sin \alpha \dot{\alpha}^2 - \frac{gm}{2M} \sin 2\alpha \quad (20b)$$

The resulting dynamics can be approximated by (21) when assuming the hovering position of the UAV [9].

$$\ddot{\gamma} = u_\gamma + d_\gamma \quad \text{with } \gamma \in [z, x] \quad (21)$$

Here u_γ is the control input to the system and d_γ exogenous disturbances. Leaving out the disturbances, the inputs u_1 and θ_D are chosen as:

$$u_1 = \frac{(u_z + g)(M+m)}{1 + \frac{m}{2M}(1 - \cos 2\alpha - \theta_D \sin 2\alpha)} \quad (22)$$

$$\theta_D = \frac{u_x}{g + \frac{gm}{2M}(1 - \cos 2\alpha)} \quad (23)$$

To obtain a PD-controller the control inputs u_z and u_x need to be chosen as in (24a).

$$u_z = -k_p^z(z - z_d) - k_d^z \dot{z} + \frac{m}{M+m}L \cos \alpha \dot{\alpha}^2 \quad (24a)$$

$$u_x = -k_p^x(x - x_d) - k_d^x \dot{x} + \frac{m}{M+m}L \sin \alpha \dot{\alpha}^2 - \frac{gm}{2M} \sin 2\alpha \quad (24b)$$

When the inputs u_1 and θ_D are included in (20) they result in the following PD dynamics:

$$\ddot{z} = -k_p^z(z - z_d) - k_d^z \dot{z} \quad (25a)$$

$$\ddot{x} = -k_p^x(x - x_d) - k_d^x \dot{x} \quad (25b)$$

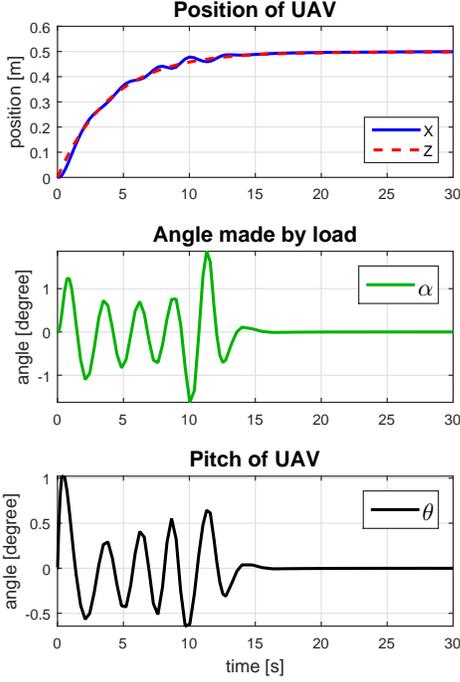


Fig. 12. Simulation result of a UAV with a cable suspended load

A. Lyapunov Analysis

A Lyapunov function will be used to check stability of the position. The Lyapunov function in (26a) is used for this case. $\frac{1}{2}\dot{\gamma}^2$ denotes the kinetic energy and $\frac{1}{2}k_p^\gamma(\gamma - \gamma_d)^2$ the potential energy of the system. Taking the derivative, results in (26d).

$$V = \frac{1}{2}\dot{x}^2 + \frac{1}{2}\dot{z}^2 + \frac{1}{2}k_p^x(x - x_d)^2 + \frac{1}{2}k_p^z(z - z_d)^2 \quad (26a)$$

$$\dot{V} = \dot{x}\ddot{x} + \dot{z}\ddot{z} + k_p^x(x - x_d)\dot{x} + k_p^z(z - z_d)\dot{z} \quad (26b)$$

$$= -k_p^x(x - x_d)\dot{x} - k_d^x\dot{x}^2 + k_p^x(x - x_d)\dot{x} - k_p^z(z - z_d)\dot{z} - k_d^z\dot{z}^2 + k_p^z(z - z_d)\dot{z} \quad (26c)$$

$$\dot{V} = -k_d^x\dot{x}^2 - k_d^z\dot{z}^2 \leq 0 \quad (26d)$$

This result shows that the derivative is negative semi-definite and by this the system is stable for his position. Since stability for the load dynamics can not be guaranteed with a Lyapunov function it can not be said that the whole system is stable as well. With this derivation, stability is guaranteed for the lateral and vertical dynamics. Now only the load angle dynamic need to be checked for stability but this is the most difficult part to do since there is no direct control on these dynamics.

B. Simulations

In Fig. 12 and Fig. 13 the simulation results can be observed of the designed controller. These simulations are performed in 20sim. One simulation is done when a setpoint of 0.5 is given in the x- and z-direction. In the other simulation a pulse is given to the load to see if the UAV is still able to damp out the caused swing. In both cases a small disturbance can be noticed in the x-position to counteract the swing in the load. Furthermore can be seen that the angle α and θ stay bounded overtime.

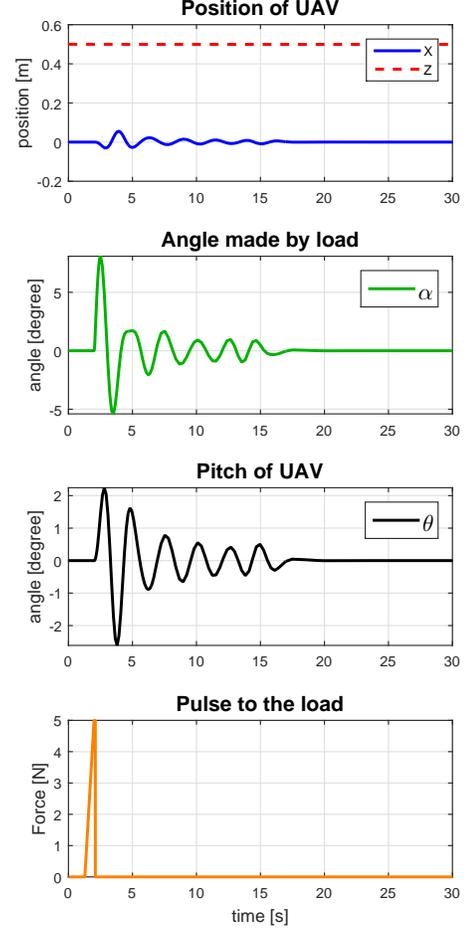


Fig. 13. Simulation result of a UAV with a cable suspended load when a pulse is added to the load

C. Experiments

The controller is also tested in an optitrack environment with a real setup. The same setup as in the main paper is used to perform this experiment. In Fig. 8 the position results can be seen. It can be noticed that the system is not able to go to the desired setpoint. Especially in the y-direction. Here it is drifting from the desired position. The vertical position is not reached due to probably the attached load and that the minimal offset thrust to compensate for gravity is set to low. The resulting load angles are shown in Fig. 14. It shows that the controller is able to bound the load angle within 10° . Fig. 16 shows that the attitude stays small during movement of the UAV.

D. Conclusion

The conclusion that be drawn from this appendix is that this controller is able to damp out the swing more then the controller of the paper. But the controller of the paper is more reliable in following the desired setpoint. This is probably due to the fact that in the paper the lateral and load dynamics are taken together for designing the controller.

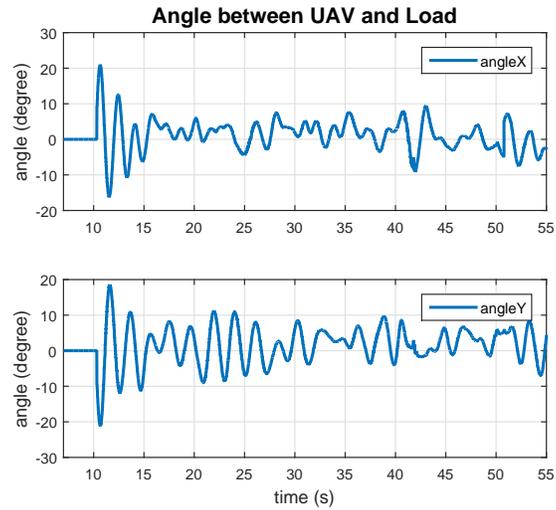


Fig. 14. Resulting angle between UAV and load during the experiment. Plot stays zero until 10 seconds since on that moment the load controller is enabled.

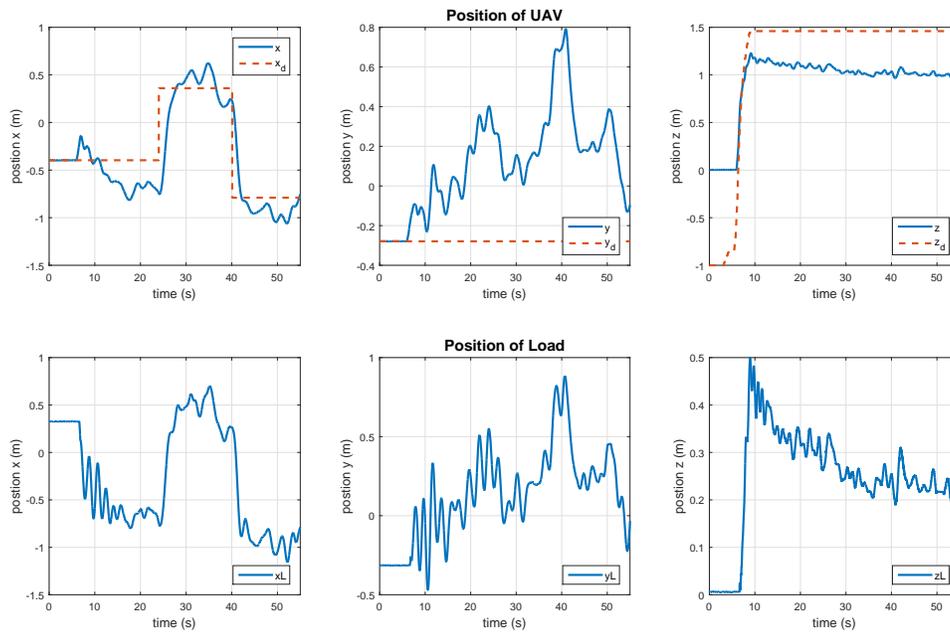


Fig. 15. Recorded Position of UAV and load during experiment. Setpoint z is -1 at the start to guarantee that the UAV stays on the ground when the communication between the UAV and PC is started. The minimal thrust offset to compensate for gravity was set to low to reach the desired altitude.

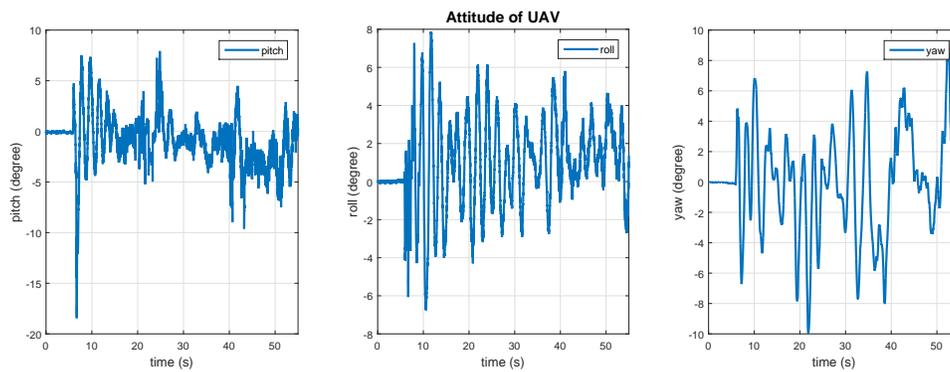


Fig. 16. Measured attitude of UAV during experiment

APPENDIX B MANUAL

In this appendix the use of the software package will be explained in chronological order. The package is available through the Robotics and Mechatronics group of the University of Twente. The manual is based on the manual written in [17] with some additions and clarifications.

A. Prerequisites

The software package is written for ROS. At the date that this package is developed the current version was ROS indigo. Ubuntu 14.04 is used as operating system to run ROS. The following prerequisites are assumed for this manual:

- 1) Ubuntu 14.04 or any derivative based on this (Xubuntu 14.04, Lubuntu 14.04, etc).
- 2) ROS Indigo installation according to <http://wiki.ros.org/indigo/Installation/Ubuntu> with a Desktop-Full Install.
- 3) Crazyflie client (<https://github.com/bitcraze/crazyflieclients-python>)
- 4) clean catkin workspace
- 5) Knowledge on how to use Optitrack

B. Installation

The `ram_crazy` package requires three external packages:

- `hector_quadrotor`: <http://wiki.ros.org/hectorquadrotor>
- `crazyflie_ros`: <http://wiki.ros.org/crazyflie>
- `mocap_optitrack`: <http://wiki.ros.org/mocapoptitrack>

The `hector_quadrotor` package is used to simulate a quadrotor in gazebo. To connect to the crazyflie the `crazyflie_ros` package is used. The data obtained from the optitrack system need to be translated to a ROS message. This is handled in the `mocap_optitrack` package. The current versions of these packages are included by the `ram_crazy` such that compatible issues are prevented. But using the current version might be interesting since more and updated options might be available. After unpacking the packages in the catkin workspace, execute the following commands in a terminal to solve the dependencies.

```
$ cd ~/path_to_catkin_workspace
$ rosdep install --from-path
  src --ignore-src
```

This will install all the required packages, the installation will request confirmation of installing the packages several times. Run the following command inside your workspace through a terminal to compile the catkin workspace.

```
$ catkin_make
```

To compile the package completely it might be necessary to run this command several times.

C. Usage of package

Before the package can be used make sure that the following is done to make everything work properly. In Ubuntu all the python `.py` files need to be executable before they work. To do this run the following commands for all the python files:

```
$ cd ~/catkin_workspace/src/ram_crazy/py
$ sudo chmod +x filename.py
```

Also make sure that the following lines are included in bottom of the `./bashrc` file. Since otherwise this command need always be executed after opening a new terminal. To access this file execute the following command and paste the two lines.

```
$ gedit ~/.bashrc
source /opt/ros/indigo/setup.bash
source ~/catkin_ws/devel/setup.bash
```

By placing these lines in the `./bashrc` they will be executed automatically after opening a new terminal.

Now everything should be set to start using this package. There are two launch files available. One which starts only the `interface.py` script and one which starts this script and the gazebo program Gazebo. For only launching the `interface.py` script execute the following (tab can be used to auto complete):

```
$ roslaunch ram_crazy
  interface_launch.launch
```

After this gazebo can be started separately with the following command:

```
$ roslaunch ram_crazy
  quadrotor_empty_world.launch
```

Executing the following command will do both at once:

```
$ roslaunch ram_crazy
  ram_crazy_empty_world.launch
```

In Fig. 17 the interface is shown after start-up. With the *Simulation* toggle button can be specified if the test is performed in simulation or in experiment. Depending on the selection, ROS will start-up different publishers and subscribers. Four columns can also be noticed. The *Address* column displays the radio address that is used to connect to that drone. The first value shows the radio ID, the second the drone ID and the last one shows the used bandwidth for communication. The second column displays the prefix of the drone used by ROS to distinguish each drone during control. In the optitrack environment each drone needs to be given an ID. This one is specified by the *Trackable ID* column. The value present here needs to be given to the trackable in optitrack. The last column shows if the controller for that drone is active. This allows to send the pose from optitrack to ROS. With the *Add Drone* button an arbitrary number of drones can be added. The increment is set to 60 but can be changed. The Load has been given the trackable ID of 12. After adding one drone the interface will look like as in Fig. 18. To make the controller active double click that row. This will start-up the controller shown in Fig. 19. This interface has different buttons and these are explained below. Moving the sliders will change the values. Changing the setpoint sliders and offset thrust slider will directly change the controller. After changing the sliders in the gain section the *Set Gains* button needs first to be clicked before these values will be used.

Take-off:

gets the current position and sets the z-setpoint to one.

Land:

Lands the UAV.

Get Setpoint:

Changes the setpoints of the interface to the current position measured by optitrack.

Go to setpoint:

adds a value to the current position and makes this a new setpoint.

Toggle Radio:

Activates the communication with the drone

Spawn Simulation

Spawns the drone in Gazebo

Save Data:

Saves the data to a specified location. default location is `./ros/` folder. In the `.cpp` file of the controller the location and what need to be saved can be set.

Publish Setpoint:

Enables the publishing of the setpoint.

Lissajous test:

Starts the path following of a Lissajous figure.

Setpoint drone0:

Uses the setpoint of drone0 with an distance in the x-direction specified by the slider next to it. Used for load transport with 2 UAVs.

1/2 Drone(s):

Specifies with how many drones the load transportation is performed.

Enable Load Control:

Enables the load control part and also sets the gains to be used.

Get Current:

Allows the current I-action to be read and sent to the interface, in this way the offset can be determined using the I-action of the PID controller.

Set Gains Control Without Load:

Sets the default gains for controlling without load.

Set Gains Control With Load:

Sets the default gains for controlling with load. Gains are restored depending on which toggle is active of the one or two drones.

Set Gains:

Sets the gains depending on how the sliders are set.

In Fig. 20 an example of gazebo is shown when two drones are used. RViz can be used to visualize the current setpoint and pose of the UAV. It can be launched by executing the following command:

```
$ roslaunch ram_crazy
rviz_ram_launch.launch number:=TrackableID
```

In Fig. 21 the result of this command is shown. In this case also the Lissajous path is enabled and by this also shown in RViz. The two arrows display the orientation of the UAV and the velocity vector.

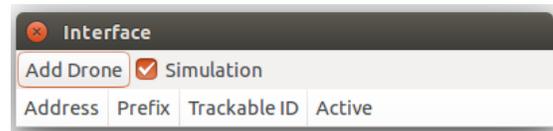


Fig. 17. Interface after starting up the interface.py script

Address	Prefix	Trackable ID	Active
radio://0/0/2M	drone0	0	No
radio://1/60/2M	drone60	60	No

Fig. 18. Example of the interface with two drones added

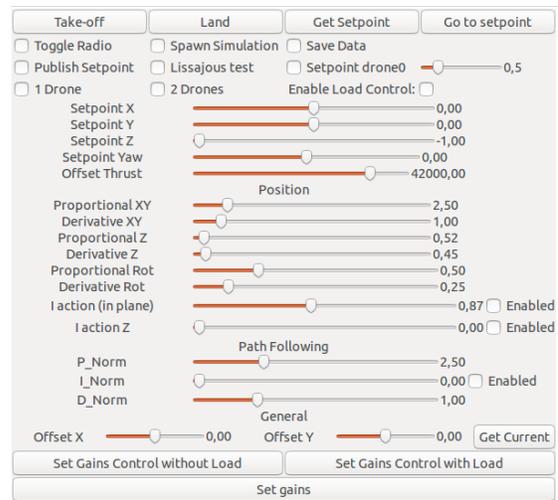


Fig. 19. Controller interface to fly with the crazyflie. In this interface the control parameters can be changed and the desired setpoint can be given. It is also possible to enable the control to fly with a load.

D. How to fly with the load

The following can be done to be able to fly with a cable suspended load.

- place the drone and load in the optitrack area such that the load is along the x- or y-axis.
- start-up the interface and the controller for drone0
- Press *Get Setpoint*
- Enable *1 Drone*
- Press *Set Gains Control without Load*
- Enable *Toggle Radio*, *Publish Setpoint* will enable automatically
- Slide the slider of *Setpoint Z* carefully up until the drone and load are of the ground
- Enable *Enable Load Control*

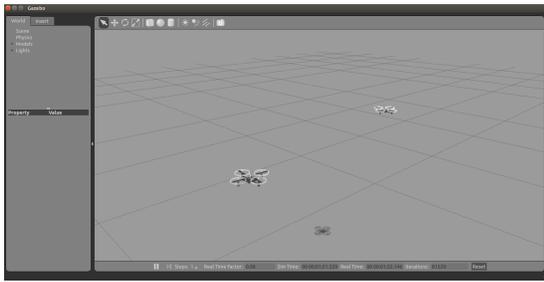


Fig. 20. Gazebo after two drones are added

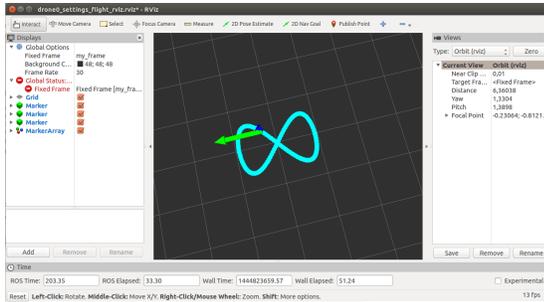


Fig. 21. RVIZ when path following is enabled. In this window the position of the setpoint and the UAV can be seen. The arrows show the orientation vector of the UAV and the calculated command velocity vector.