Non-Linear Model Predictive Control for UAVs with Slung/Swung Load

Felipe Gonzalez [1], Alexander Heckmann[2], Stefan Notter [2], Markus Zürn[2], Jan Trachte[2], Aaron McFadyen[1]

[1]Queensland University of Technology
[2]Institute of Flight Mechanics, University of Stuttgart

Abstract—Suspended loads on UAVs can provide significant benefits to several applications in agriculture, law enforcement and construction. The load impact on the underlying system dynamics should not be neglected as significant feedback forces may be induced on the vehicle during certain flight manoeuvres. Much research has focused on standard multi-rotor position and attitude control with and without a slung load. However, predictive control schemes, such as Nonlinear Model Predictive Control (NMPC), have not yet been fully explored. To this end, we present software and flight system architecture to test controller for safe and precise operation of multi-rotors with heavy slung load in three dimensions.

1. INTRODUCTION

Heavy suspended loads, no matter if containing a sensor, water for firefighting or a pesticide fluid [1,2], may significantly influence flight dynamics due to a high mass-ratio of the load to the vehicle. The nonlinear dynamics of an aerial suspended load are well studied by a vast amount of publications, such as [3–6]. In most cases, the controller is derived by a linearisation of the dynamical model. An overview of the generic 3-dimensional pendulum control problem is mentioned in [3]. A fuzzy controller for load swing compensation while the vehicle is tracking a position trajectory at low attitude angles near hover is introduced in [4]. Detailed studies on the slung load model and a trajectory tracking slung load controller using linear quadratic regulation are presented in [5] and [6]. A dynamic programming approach is developed, capable to generate swing free trajectories for agile manoeuvres. Nevertheless, there hasn't been research on including the slung load dynamics to the model of an online Nonlinear Model Predictive Control (NMPC). NMPC provides an optimal controller for highly nonlinear dynamics whilst accounting for constraints and enabling a state feedback loop that involves disturbance and model uncertainties. NMPC is an established tool for relatively slow elapsing processes since the Nineties and has mostly been applied in the field of industrial process engineering. However, due to rising computational power on micro controllers, NMPC becomes applicable for agile lightweight aerial systems, too. To this end, the contributions of this paper are:

1. Design of a NMPC for a quadrotor with suspended slung load, capable of stabilising or tracking the load and quad movement over a large flight envelope
2. Performance comparison of the NMPC to a Linear Quadratic Regulator (LQR) with respect to robustness, time varying reference and aggressive control
3. Preparation of a NMPC algorithm in C++ code for field tests including assessment of computational demand

2. SYSTEM DYNAMICS

A precise derivation of the highly non-linear system dynamics of the quad-load-combination is essential for an efficient control over the entire flight envelope.

The kinematics of the load depend on the moving pivot and herewith the quad acceleration. The acceleration resulting from the cable force impact on the pivot is now added to the quad acceleration. With Newton's second law of motion, this is ac = fM, where f is the cable force vector and M is the quad mass. The complete system of equations for the combined dynamics for model Fig.1 is shown in (1).

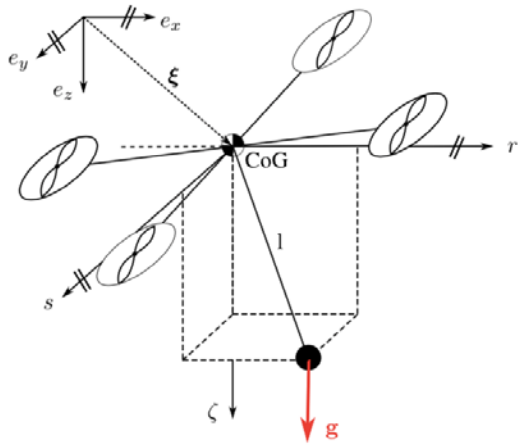

Figure 1 Load coordinates in frame O related to inertial frame I by the quad position ξ_

It remains to mention that, as a result, the quad dynamics implicitly depend on the load's motion and vice versa. The yaw angle is considered to be constant zero such that a feed forward term can be added to control the quadrotor orientation

Details on the system dynamics can be found in literature [3-8]

$$
\begin{aligned}
\ddot{x} &= \left(T\cos\phi\sin\theta - \mathrm{m}(\ddot{x}+\ddot{r})\right)/\mathrm{M} \\
\ddot{y} &= -\left(T\sin\phi - \mathrm{m}(\ddot{u}+\ddot{s})\right)/\mathrm{M} \\
\ddot{z} &= \left(T\cos\phi\cos\theta + \mathrm{m}((\ddot{r}r+\dot{r}^2+\ddot{s}s+\dot{s}^2)/\zeta - \ddot{z} + (\dot{r}r+\dot{s}s)^2/\zeta^3 + \mathrm{g}(\zeta/\mathrm{l}))\right)/\mathrm{M} + \mathrm{g} \\
\ddot{r} &= (\zeta^4\ddot{x} - r\zeta^3\ddot{z} + rs\zeta^2\ddot{s} + (r\mathrm{l}^2 - rs^2)\dot{r}^2 + (r\mathrm{l}^2 - r^3)\dot{s}^2 + 2\dot{r}\dot{s}r^2s + rg\zeta^3)/((s^2-\mathrm{l}^2)\zeta^2) \\
\ddot{s} &= (\zeta^4\ddot{y} - s\zeta^3\ddot{z} + rs\zeta^2\ddot{r} + (s\mathrm{l}^2 - sr^2)\dot{s}^2 + (s\mathrm{l}^2 - s^3)\dot{r}^2 + 2\dot{r}\dot{s}s^2r + sg\zeta^3)/((r^2-\mathrm{l}^2)\zeta^2)
\end{aligned} \tag{1}
$$

We assume

- No aerodynamic effects
- No cable mass
- No cable strain
- No free fall of the load (cable force ≥ 0)
- Pivot with no offset to multi-rotor CoG

3. CONTROL DESIGN

The control design is separated into the description of the NMPC and the LQR, both of which are used and compared in this work. The two controls consider the slung load dynamics to derive the corresponding controller.

3.1 Nonlinear Model Predictive Control (NMPC)

The mathematical model defined by (1) represents the system dynamics and is used as an internal model f for the predictive controller. An optimal control problem is repeatedly solved at each control step over a finite time horizon [9]. The state vector contains the three Cartesian positions and velocity of the vehicle, the pitch and roll angle as well as the two slung load Cartesian coordinates and their derivatives. The third coordinate is omitted due to the spherical surface constraint of (2). The control vector contains the vehicle's roll rate, pitch rate and collective thrust. A cost function J is minimized with respect to u. The cost function is used to penalise deviation from the reference flight and slung load condition. The optimal control problem can be defined as

$$
J^{\star}(\mathbf{x},\mathbf{u}) = \underset{\mathbb{U}}{\operatorname{argmin}}\, J(\mathbf{x},\mathbf{u})
$$

$$
\begin{aligned}
J(\mathbf{x},\mathbf{u}) = \Delta\mathbf{x}_N^T P \Delta\mathbf{x}_N + \sum_{k=1}^{N-1} \Delta\mathbf{x}_k^{\mathrm{T}} Q \Delta\mathbf{x}_k \\
+ \Delta\mathbf{u}_k^{\mathrm{T}} R \Delta\mathbf{u}_k \,,
\end{aligned}
$$

$$
\begin{aligned}
&\text{where} && \Delta\mathbf{x}_k = \mathbf{x}_k - \mathbf{x}_k^* \\
& && \Delta\mathbf{u}_k = \mathbf{u}_k - \mathbf{u}_k^* \\
&\text{subject to} && \mathbf{x}_k \in \mathbb{X} \quad \mathbb{X} \in \mathbb{R}^{12} \\
& && \mathbf{u}_k \in \mathbb{U} \quad \mathbb{U} \in \mathbb{R}^{3} \\
& && \mathbf{x}_{k+1} = f\left(\mathbf{x}_k, \mathbf{u}_k\right)
\end{aligned} \tag{2}
$$

with the errors Δx_k and Δu_k of the current state and control to the reference trajectory, denoted by the asterisk. The matrices Q and R are positive semi-definite weighting matrices on the quadratic state error and control error respectively. P defines the terminal cost, i.e. the positive cost of the error at the last step of the prediction horizon. The stability of the open loop can be significantly influenced by this parameter.

The ACADO toolkit [10] provides a comprehensive C++ code library suitable for creation of an algorithm to solve the optimal control problems arising from the NMPC formulation.

The optimal control problem is subject to the model's differential system of equations and a set of inequality constraints. These constraints capture the bounds on the control inputs and platform limitations, or state constraints. For example, they can be used to prevent the platform from flying upside down (through roll and pitch angle bounds) or limit the roll rate based on the maximum torque that the rotors can generate.

The collective thrust is limited to the maximum available power and to avoid an unrecoverable drop in altitude. The constraint domain is derived from experience or specification of the X-4 Flyer Mark II in [11], with numerical values given in Table 1.

Table 1. Inequality constraints for NMPC

Description	Constraint
Roll angle limit	$\pi/2 \leq \phi \leq \pi/2$
Pitch angle limit	$\pi/2 \leq \theta \leq \pi/2$
Sph. surface constr. (r)	$-1 \leq r/\sqrt{\mathrm{l}^2 - s^2} \leq 1$
Sph. surface constr. (s)	$-1 \leq s/\sqrt{\mathrm{l}^2 - r^2} \leq 1$
Roll rate limit	$-\pi \leq \dot{\phi} \leq \pi$
Pitch rate limit	$-\pi \leq \dot{\theta} \leq \pi$
Collective thrust range	$-1.6\mathrm{Mg} \leq T \leq -0.4\mathrm{Mg}$

The reference trajectories and weighting matrices are set dynamically An overview of the prediction parameters and their numerical values are given in Table 2. The values were chosen empirically and based on literature, such as [12,13] or examples from the ACADO toolkit.

Table 2. Overview of NMPC settings

Parameter	Value
Number of time-steps N	30
Time step size δ	0.4 s
Discretisation type	Multiple Shooting
Integrator type	Gauss-Legendre 3rd Order
Integrations per step N_i	4

4. SIMULATION FRAMEWORK

The simulation framework can be structured into six major components (Fig 2.) . These are:

- Top-level controller (i.e. NMPC or LQR)
- Low-level proportional attitude control
- Control mixer to provide motor-speeds
- Quadrotor system dynamics
- Slung load system dynamics
- Visualisation tool

The top-level control is embedded to the framework by an outer feedback loop, where the states are directly forwarded to the control block, neglecting measurement deviations, hence $y = x$. Whilst the LQR is included by putting the gain matrix K on the state error, the NMPC uses a MATLAB S-function [14] and C++ compiler (mex) to define the controller. Reference trajectories and weighting matrices are dynamically allocated in the SIMULINK environment.

5. FLIGHT SYSTEM ARCHITECTURE

An overview of the system architecture is shown in Fig 3. The motion capture system provides position and attitude of the Vehicle. We use a cascaded flight control and a consumer PC runs the computationally intensive off-board Controller. The on-board controller tracks the commands based on its on-board IMU. In the system the user can change the flight mode and nominal trajectory

The motion Capture System uses five infrared cameras track the markers attached to the vehicle visually. The rigid geometry of the vehicle is introduced to the motion capture system software and the position of the COG and the attitude of the vehicle can be determined by the system

Figure 4 shows the off board architecture where the motion captures system provides position and attitude of the vehicle. The software module is implemented in MATLAB/Simulink and includes the off-board controller and also provides the user interface. The commands are transmitted to the vehicle wirelessly via XBee-Pro radio modules

The on-board system is also shown in Fig 4. In the on-board system the Arduino decodes the radio messages and converts the commands to PWM signals which are fed into the APM. The APM (ATmega2560, IMU) runs the ArduCopter firmware. The APM has two main flight modes: a) Stabilize: control of roll, pitch, yaw rate and thrust and b) Acro: control of angular rates and thrust

The software consists of three Simulink models running three instances of MATLAB with a) Sensor Model Outputs the state vector b) Controller Model Off-board controller, user interface Radio Model Communication protocol, redundancy management and c) a shared memory is used for inter-process communication

The core modules of the software system are detailed in figure 5

a) Calibration: Gives the user the capability to move the origin of the inertial frame to the current position of the vehicle and to align the inertial frame with the body axis
b) Latency Compensation: In order to compensate for the system latency, a state estimator is used to predict the future state of the vehicle
c) Ground Control: Provides the user interface and monitors and logs the flight data\
d) Reference Management Provides the nominal trajectory
e) Controller: Off-board controller
f) Flight Mode Management: Overwrites the output of the off-board controller depending on the flight mode

The system Latency is shown in Table 3

Table 3 System Latency

Component	Average latency
Motion capture system	10 ms
Software module	Controller dependent (MPC: 20-40 ms)
Radio link	8 ms
Vehicle dynamics	Vehicle dependent (TBD)

The flight failure modes included

a) The on-board radio receives no radio messages : After the timeout, the on-board Arduino initiates the emergency landing
b) The Radio model receives no command updates : The Radio model transmits an emergency landing command
c) The motion capture system fails to track the vehicle: The Radio model transmits an emergency landing command
d) The motion capture system provides misleading position or attitude information: This failure mode cannot be detected by the current system. An on-board estimator needs to be implemented to detect unsafe manoeuvres

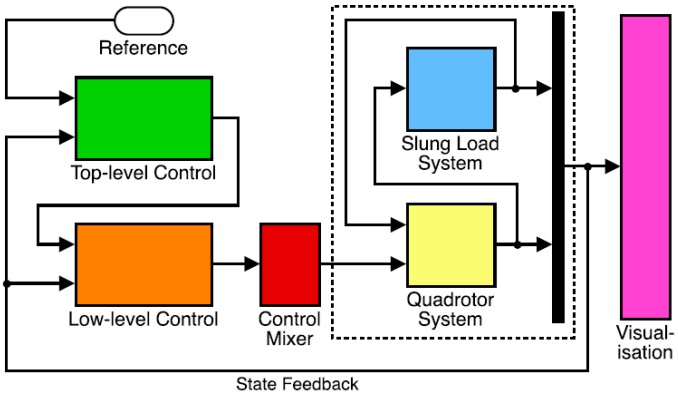

Figure 2. Simulation framework components

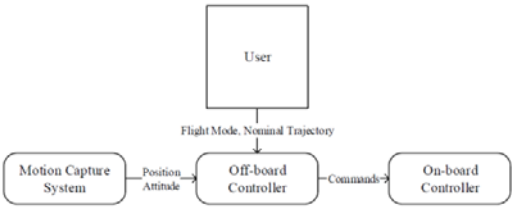

Figure 3. Overview of the Flight System Architecture

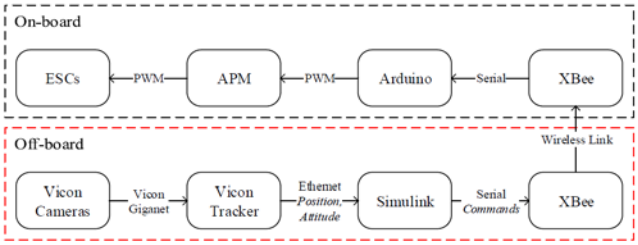

Figure 4. Overview of the on board and off-board system Architecture

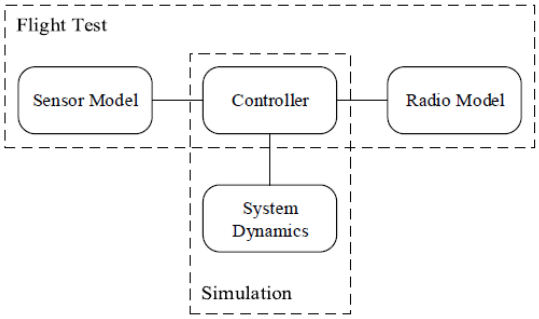

Figure 5. Overview of Software Module

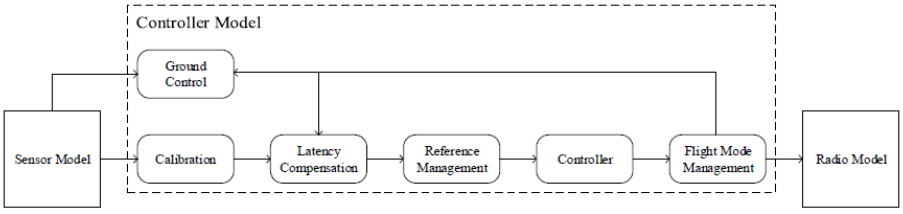

Figure 6. Core modules

6. RESULTS

Simulation

The NMPC and the LQR are evaluated for their capability of managing stabilisation problems and trajectory tracking. Three different scenarios are analysed. The first scenarios involve stabilisation problems with a deflected initial condition. The second and third scenario involve reference trajectory tracking for the quad rotor and the load position.

The weighting matrices of both control algorithms are the same in each simulation. Table 4 shows the entries of the matrices that show variations to the identity matrix. The weighting matrix P for the NMPC terminal state error is set equal to the corresponding values of Q for simplification.

Table 4. Weighting matrix variations

Matrix Cell	Value	Corresp. state/control
$Q(3,3)$	1×10^3	quad position z
$Q(7,7)$	1×10^2	load position r
$Q(8,8)$	1×10^2	load position s

Stabilisation of Load Displacement

In this scenario the goal is to actively damp an oscillation of the load. The quad is at stable hover and the load is initially displaced from its equilibrium position by 1m to the positive r-direction. As the control reference is set to maintain a stable hover with no load movement, active damping of the swinging load is required. Fig. 7 compares the vehicle's x-position, the load's position r and the corresponding control input i.e. pitch-rate, for the NMPC and the LQR control designs. The LQR forces the control limits to be exceeded and a motor-speed beyond the possible limit is demanded on some rotors. As a result, the load contacts the ground and the yaw angle deviates from the reference value. Both can be compensated after a few seconds, when the pitch angle and corresponding thrust demand start decreasing. The NMPC ensures the control constraints are respected, such that the load no longer impacts the ground. Of note, both controllers are able to damp the load within 4 s, whilst returning the platform to the reference flight configuration.

Waypoint Tracking

In this scenario, the goal is to track a specific reference trajectory for the platform whilst avoiding a swinging of the load. An inclined square circuit is used as the reference trajectory for the vehicle, where each corner forms a waypoint at a different reference altitude. The square's sides are 1m in length (only lateral direction measured) with an altitude variation of _2 m. The load reference is zero deflection to the equilibrium state, i.e. no swinging. The initial position is at stable hover with non-deflected slung load and an altitude of 2 m. Results for an example simulation are depicted in Fig. 9, where quad and load position are shown.

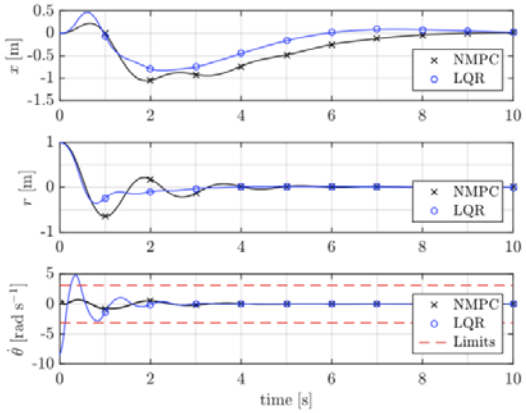

Figure 7. Quad position, load position and commanded pitch rate over time with an initial load deflection.

The results (Fig. 8) suggest that the NMPC is able to effectively track the reference trajectory of both the quad and the load, with a Root Mean Squared Displacement (RMSD) of 0:064m and 0:091m respectively. The LQR control design shows poorer tracking performance, approximating a more circular trajectory with position and load RMSD of 0:569m (governed by the altitude deviation) and 0:021m respectively. This can be attributed, in part, to the fact a single reference value is required at each time step for the LQR, compared to the full time varying reference used in the NMPC. The result implies that the NMPC approach can better manage more complex reference trajectories.

Load Position Tracking

In this scenario, the goal is to track a specific load reference trajectory, whilst the platform maintains hover. The load reference describes a circular pattern of radius 1m and a period of 3 s. The initial position of the platform and load are (0,0,-2) and (0,0,2+l) respectively. Results for an example simulation are given in Fig. 9, where load and platform position are depicted. To track the reference load position, the quad must first leave the stable hover in order to move or upswing the load. The flight path for both control algorithms describes loops, where a larger loop radius is initially required to force the load to adopt the reference trajectory. Subsequent motion is a relatively constant radius circular trajectory to maintain the load's reference circular path. The NMPC's performs with an overall RMSD of quad and load position of 0:779m and 0:915m respectively. The LQR comes to 0:781m and 0:924 m.

The load trajectory generated by the LQR touches the ground at 0:8 s simulation time, shown by the dashed line. The actual ground contact is not part of the simulation and the depicted flight neglects the impact to continue the flight and recover from the altitude drop at 2:2 s simulation time. The reason for this impact is again the disregarding of control input constraints,

putting the vehicle to an attitude where total thrust becomes insufficient for maintaining a level flight.

The average computational cost of the NMPC is 5:0 ms per real time iteration which corresponds to 200 Hz control sample time. This was verified using 1000+ iterations on the first stabilisation scenario run on a 1.7 GHz Intel Core i7-4650U dual-core processor on an Apple OS X 10.9.5 operating system and a test-simulation in C++ code. The SIMULINK environment was removed to ensure efficiency of the control algorithm. Similar computation times were observed for other scenarios which suggest the controller is suitable for real applications, and could be implemented onboard real hardware configurations.

Flight Test

Figure 10 shows the x-y position and altitude of the reference load trajectory and the vehicle. We can observe good match between the command and the flight logs. Figure 11 shows the ratio of MPC weighting for vehicle and load position deviation from the reference

7. CONCLUSION

A NMPC algorithm for slung load quadrotor control including comparative performance assessment under a range of operating conditions was presented. The results show the importance of explicit consideration of the platform constraints and nonlinear dynamics of slung load systems in control design. Especially when heavy slung loads generate significant cable forces, wise trajectory planning is required. This work shows that the LQR may violate these constraints, e.g. leading to ground contact. The NMPC strictly avoids such events and significantly decreases the overall control effort through predictive management of the actuating control elements. This is a great advantage when power is limited, e.g. carrying loads close to the specification limit, resulting in short control margins. A state observer for the slung load must be derived, which could be accomplished using visual sensor to then realise an integrated visual predictive control solution [15]. Investigation on the NMPC robustness, e.g. Lyapunov stability using the end term penalty, is advisable to assure robustness for all operating conditions before preparing a field deployable solution. Furthere work is also focused on the use of robust methods for multi-objective and multdisciplinary design of the slung load, UAV system [16]

ACKNOWLEDGMENTS

This work was kindly supported by Queensland University of Technology (QUT), Australian Research Centre for Aerospace Automation (ARCAA) . The authors would also like to thank Walter Fichter and Christoph Seiferth at the Institute of Flight Mechanics and Control (IFR), University of Stuttgart in Germany.

REFERENCES

[1] C. Hillnhutter et al., "Remote sensing for the detection of soil-borne plant parasitic nematodes and fungal pathogens," in Precision Crop Protection-the Challenge and Use of Heterogeneity, Springer, 2010, pp. 151–165.

[2] L.F. Gonzalez, DS Lee, R.A Walker, J. Periaux, "Optimal mission path planning (MPP) for an air sampling unmanned aerial system", Australasian Conference on Robotics and Automation (ACRA), December 2-4, 2009, Sydney, Australia

[3] J. Shen et al., "Dynamics and control of a 3d pendulum,"in Conference on Decision and Control, The Bahamas, 2004, pp. 323–328.

[4] H. M. Omar, "New fuzzy-based anti-swing controller for helicopter slung-load system near hover," in Int. Symp. Computational Intelligence in Robotics and Automation,Daejeon, Korea, 2009, pp. 474–479.

[5] M. Bisgaard, "Modeling, estimation, and control of helicopter slung load system," Ph.D. dissertation, Fac. Eng. and Science, Aalborg Univ. Space Center, 2007, pp. 204–221.

[6] I. Palunko et al., "Trajectory generation for swingfree maneuvers of a quadrotor with suspended payload: A dynamic programming approach," in Int. Conf. on Robotics and Automation, St. Paul, MN, 2012, pp.2691–2697.

[7] M. Burri et al., "Aerial service robots for visual inspection of thermal power plant boiler systems," in Int. Conf. on Applied Robotics for the Power Industry, Zurich,Switzerland, 2012, pp. 70–75.

[8] M. Hehn and R. D'Andrea, "A flying inverted pendulum," in Int. Conf. on Robotics and Automation, Shanghai, China, 2011, pp. 763–770.

[9] R. Findeisen, and F. Allg¨ower, "An introduction to nonlinear model predictive control," in Benelux Meeting on Systems and Control, Veldhoven, The Netherlands, 2002, pp. 119-141.

[10] B. Houska et al., "ACADO toolkit—An open-source framework for automatic control and dynamic optimization," Optimal Control Applications and Methods, vol. 32, no. 3, pp. 298–312, 2011.

[11] P. Pounds et al., "Modelling and control of a large quadrotor robot," IFAC Control Engineering Practice, vol. 18, no. 7, pp. 691–699, 2010.

[12] P. Bouffard, "On-board Model Predictive Control of a Quadrotor Helicopter: Design, Implementation, and Experiments," Univ. of California, Berkeley, CA, 2012.

[13] K. Alexis et al., "Model predictive quadrotor control: attitude, altitude and position experimental studies," IET Control Theory and Applications, vol. 6, no. 12, pp. 1812–1827, 2012.

[14] Matlab User's Guide, The Math Works Inc., 2013.
[15] A. Mcfadyen et al., "Aircraft collision avoidance using spherical visual predictive control and single point features,"in Int. Conf. on Intelligent Robots and Systems,Tokyo, Japan, 2013, pp. 50–56.
[16] L.F Gonzalez, Robust evolutionary methods for multi-objective and multdisciplinary design optimisation in aeronautics, The University of Sydney , 2006

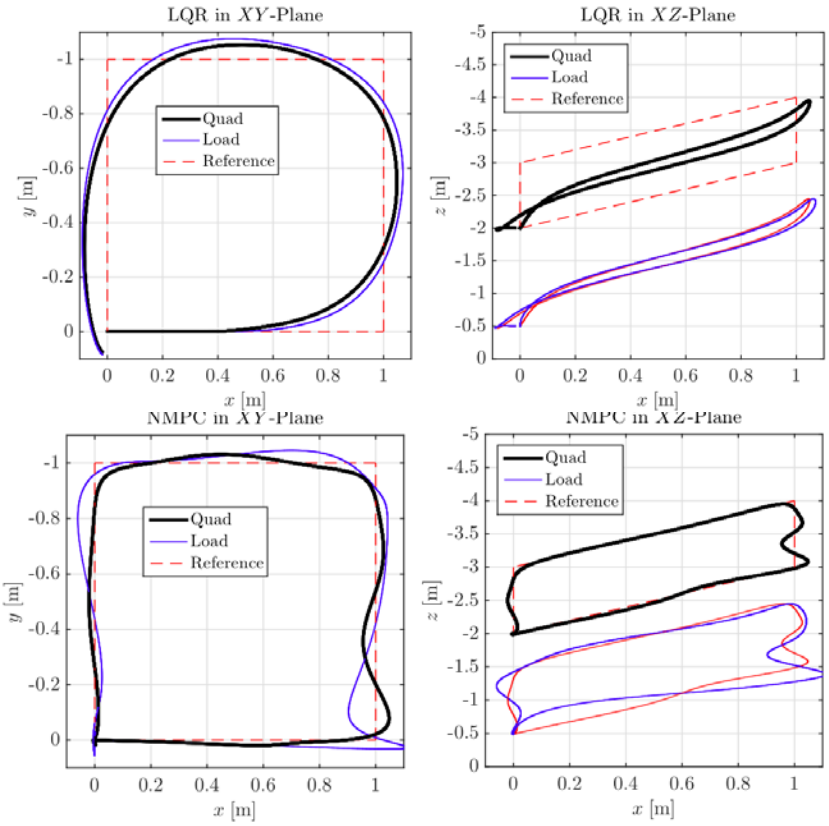

Figure 8. Reference trajectory tracking of LQR and NMPC in comparison.

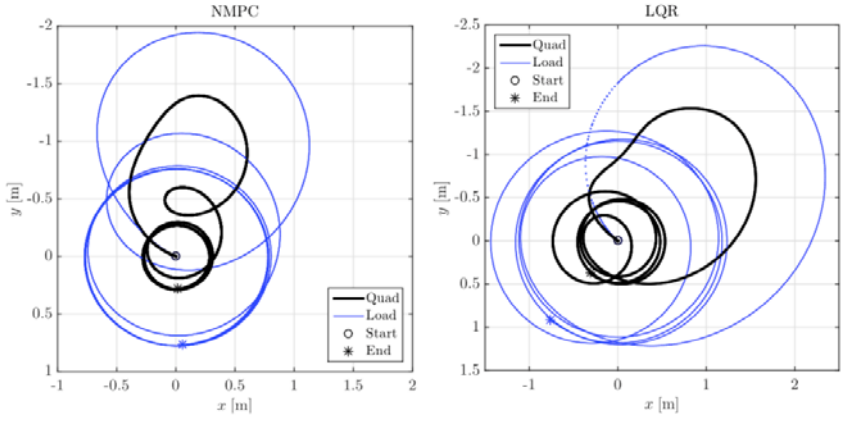

Figure 9. Tracking a circular load reference of radius 1m to the quad and period 3 s while quad reference is at origin.

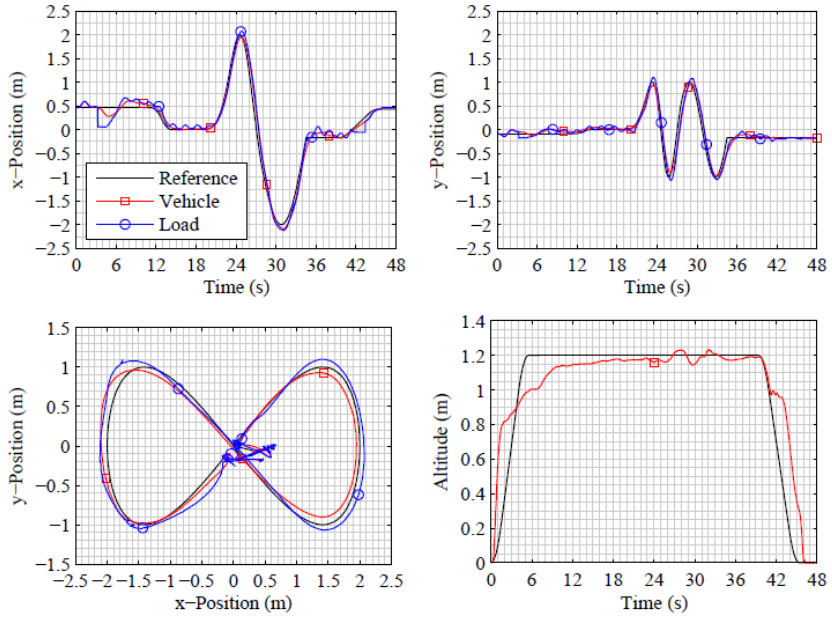

Figure 10. x-y position and altitude of the reference load trajectory and the vehicle

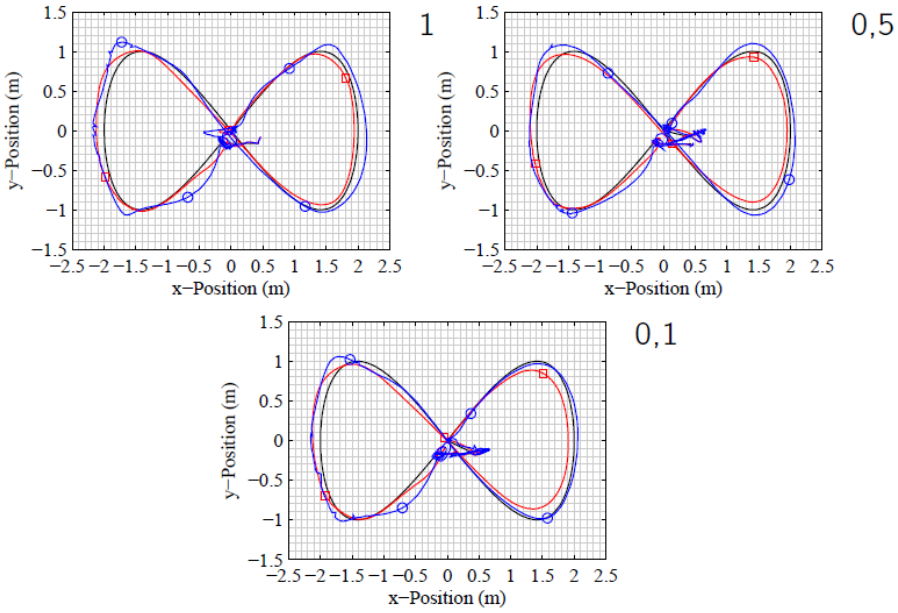

Figure 11. Ratio of MPC weighting for vehicle and load position deviation from the reference