Energy-Based Control of an Underactuated Quadrotor
using a Geometrical Spring-Damper approach

J.J. (Jelmer) Graat

MSc Report

Committee:
Prof.dr.ir. S. Stramigioli
Dr.ir. G.A. Folkertsma
R.A.M. Rashad, MSc
Dr.ir. A.Q.L. Keemink

February 2018

003RAM2018
Robotics and Mechatronics
EE-Math-CS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Contents

List of acronyms

PBC Passivity-Based Control

IPC Intrisically Passive Control

RCC Remote Center of Compliance

IDA-PBC Interconnection and Damping Assignment Passivity-Based Control

UAVs Unmanned Aerial Vehicles

PID Proportional-Integral-Derivative

ROS Robot Operating System

SITL Software-in-the-loop

IMU Inertial Measurement Unit

Abstract

Nowadays, quadrotors are getting more applications where it is necessary to interact with the environment which is in general unknown. For guaranteed safe interaction, the energy in the system should be considered explicitly. Using energy-based control, virtual dynamics are added to the plant such that the desired dynamics are achieved.

An energy-based controller using a virtual spring-damper has been developed for a underactuated quadrotor. A virtual spring with a Remote Center of Compliance (RCC) is presented to induce a coupling between the error in the non-actuated plane and the quadrotor tilting torque. To guarantee safe interaction, the controller should be passive. It is shown that passivity is destroyed by the projection of the control wrench on the subspace of implementable wrenches if done naively. An energy-tank is proposed to recover passivity.

Simulations and experiments have been performed to study the passivity and the behavior of the controller. It has been concluded that underactuation and the rotor dynamics are problematic for the passivity of the controller. Furthermore, with the currently available quadrotor hardware, energy-based control can only be implemented by approximation. However, RCC-based control is found to be promising, especially for interaction control.

1 Introduction

1.1 Context

Earlier robots were commonly made for specific tasks in a known environment. For example, assembly tasks where high-speed and accuracy are important. Nowadays, robots are getting more common in everyday life where it is necessary to interact with the environment which is in general unknown. For guaranteed safe interaction, the energy in the system should be considered explicitly. The classical control approach is to control either the position or force, neglecting the energy in the system. Interaction control explicitly takes energy into account by shaping the energy in the controlled system by using a combination of position and force control (impedance control, Hogan (1984)). The controller acts like a dynamical system that interacts with the robot to achieve the desired behavior.

A quickly growing field in robotics is in the use of autonomous Unmanned Aerial Vehicles (UAVs). Especially quadrotors are a popular research topic. Their mechanical simplicity and ability to hover make them useful for civilian applications like surveillance, inspections and maintenance. In many applications, the quadrotor will have to physically interact with the environment and in most cases, this environment is not fully known beforehand. The Proportional-Integral-Derivative (PID) controller scheme is widely used for quadrotor control. A PID controller is signal-based and calculates the control outputs as function of control errors. The signals cannot be directly related to physics and the energy in the system is not considered. As energy is not considered, stability when in contact with an unknown environment cannot be guaranteed (Folkertsma, 2017). Therefore, for safety concerns, an PID controller is not suitable for interaction control. An energy-based approach on quadrotor control is needed to guarantee safe interaction.

1.2 Related work

The laws of nature are governed by energy. Energy and energy flows in a system define it's behavior. With energy-based control, the behavior of the system is controlled by controlling the energy and energy flows in the system explicitly. Where the goal of a PID controller is to control a given error to zero, an energy-based controller shapes the dynamics of the robot to control its behavior. In other words, designing an energy-based controller can be seen as finding a dynamic system (controller) which shapes the total energy function of the controlled system as desired. This energy-shaping approach is discussed thoroughly in Ortega et al. (2001).

To guarantee safety of a system, the total energy in the system should be considered explicitly (Stramigioli, 2015), (Folkertsma, 2017). Limiting the total energy in the system will put an upper limit on the kinetic energy of the plant. Willems (1972) has shown that when two systems (e.g. controller and plant) which are both passive are connected, no energy can be generated inside the connected system. In this case, the system is passive and the energy in the system can only increase due to energy injection by the environment. Stability will always be guaranteed and does not depend on the environment, making passivity an important property for interaction control.

Different approaches can be taken to shape the dynamics of a system. Sieberling et al. (2010) use non-linear dynamic inversion to artificially kill the dynamics of the plant. This allows the dynamics to be chosen freely by adding a controller. Another approach is not to kill, but to reshape the dynamics of the plant by artificially change the interconnection and damping matrices of the plant (Interconnection and Damping Assignment Passivity-Based Control (IDA-PBC)). This approach has been successfully used by Yüksel et al. (2014) to reshape the physical properties of the quadrotor to obtain desired dynamics. However, an external control-

ler is still needed to be able to control the quadrotor. In Yüksel et al. (2014), this controller is not passive and therefore not suitable for interaction. Finally, the dynamics of the system can be reshaped by control by interconnection, where another dynamical system (the controller) is connected to the plant via power ports. Stramigioli (2001) has proposed the Intrisically Passive Control (IPC) scheme to implement control by interconnection. The IPC consists of a passive controller that is connected to the plant and a supervisor controller. The supervisor controller can inject energy in the passive controller to let the system perform work.

1.3 Project goal

The goal of this thesis is to design an Intrisically Passive Control (IPC) for an underactuated quadrotor suitable for interaction. During this thesis the supervisor controller will not be considered. The effects of underactuation and actuation limits on the passivity of the IPC will be considered explicitly. Furthermore, for the controller to be suitable for interaction, the influence of the parameters on the controller response should be intuitive. Finally, the controller should be implemented on a real quadrotor to perform experiments on the controller performance and stability.

1.4 Notations

The following notations will be used in the remainder of this thesis:

Ψ_a Right-handed reference frame with label 'a'.

p_a^b Position of origin of frame 'a' expressed in frame 'b'.

R_a^b Rotation matrix of frame 'a' to frame 'b'.

H_a^b Homogeneous transformation matrix from frame 'a' to frame 'b':

$$H_a^b = \begin{pmatrix} R_a^b & p_a^b \\ 0_3{}^T & 1 \end{pmatrix} \in SE(3)$$

$v_a^{c,b}$ Linear velocity vector of the body attached to frame 'a' with respect to the body attached to frame 'b' expressed in frame 'c'.

$\omega_a^{c,b}$ Angular velocity vector of the body attached to frame 'a' with respect to the body attached to frame 'b' expressed in frame 'c'.

$T_a^{c,b}$ Twist of the body attached to frame 'a' with respect to the body attached to frame 'b' expressed in frame 'c':

$$T_a^{c,b} = \begin{pmatrix} \omega_a^{c,b} \\ v_a^{c,b} \end{pmatrix} \in se(3)$$

F^a Force vector expressed in frame 'a'.

τ^a Torque vector expressed in frame 'a'.

W^a Wrench expressed in frame 'a':

$$W^a = \begin{pmatrix} \tau^a & F^a \end{pmatrix} \in se^*(3)$$

$\widetilde{\alpha}$ Skew-symmetric tilde form of vector α:

$$\widetilde{a} = \begin{pmatrix} 0 & -\alpha_3 & \alpha_2 \\ \alpha_3 & 0 & -\alpha_1 \\ -\alpha_2 & \alpha_1 & 0 \end{pmatrix}$$

$Ad_{H_a^b}$ Adjoint of the homogeneous matrix H_a^b:

$$Ad_{H_a^b} = \begin{pmatrix} R_a^b & 0 \\ \widetilde{p}_a^b R_a^b & R_a^b \end{pmatrix}$$

$ad_{T_a^{c,b}}$ Small adjoint of the Twist $T_a^{c,b}$:

$$ad_{T_a^{c,b}} = \begin{pmatrix} \widetilde{\omega}_a^{c,b} & 0 \\ \widetilde{v}_a^{c,b} & \widetilde{\omega}_a^{c,b} \end{pmatrix}$$

1.5 Thesis structure

The remainder of this thesis is structured as follows: In Chapter 2, the mathematical background of energy-based control will be treated. Next, in Chapter 3, the controller design is presented. In Chapter 4, simulations on the passivity and behavior of the controller performed in a 20-sim[1] simulation environment will be presented. In Chapter 5, the approach to implement wrench control on the given platform (ROS and Pixhawk flight controller) is explained. In Chapter 6, Software-in-the-loop (SITL) simulations and experiments are treated to evaluate the performance and stability of the implemented controller. Finally, in Chapter 7, conclusions and recommendations for future work will be given.

[1] http://www.20sim.com

2 Energy-based control

The laws of nature are governed by energy. The dynamics of a system can be completely described by the energy as function of the state, the Hamiltonian. The behavior of a system can be controlled by shaping its energy. In this energy-based approach, the controller can be described as a dynamical system itself whose behavior is intuitively known. This approach is more intuitive than the classical approach of treating controllers as signal processors, especially for interaction control problems. Furthermore, energy in the system is explicitly taken into account, which can help to prevent instabilities during interaction and ensure operation within safe limits.

2.1 Port-based modeling

The behavior of dynamical systems can be described using the port-Hamiltonian formalism. This formalism describes the dynamics in terms of energy functions and energy flows through power ports between elements in the system. A more extensive overview of this formalism can be found in van der Schaft et al. (2014). The formalism is suitable for the design of energy-based controllers as the energy and energy exchange is considered explicitly. The two power port variables of a mechanical system are the generalized force and velocity. Graphically, port-Hamiltonian systems expressed in fundamental elements can be visualized in bond graphs (Paynter, 1961), (Breedveld, 1984). A short introduction on bond graphs can be found in Appendix B. Bond graphs will be used in this thesis to model the quadrotor and the controller in a 20-sim[1] simulation environment.

2.2 Passivity

Consider a dynamical system Σ with state x, input u and output y and a real-valued function $w(u,y)$ called the supply rate. If a non-negative storage function $S(x)$ can be found such that (Willems, 1972):

$$S(x_0) + \int_{t_0}^{t_1} w\left((u(t), y(t)\right) \mathrm{d}t \geq S(x_1) \tag{2.1}$$

then the system is said to be dissipative. When describing a mechanical system using the port-Hamiltonian formalism, the storage function $S(x)$ can be chosen as the Hamiltonian $\mathscr{H}(x)$, and the supply rate $w(u,y)$ as $y^T u$ where u and y are the generalized velocity and force of the power port respectively. Substituting and taking the differential form of Equation 2.3 gives Equation 2.2 (Folkertsma, 2017):

$$\dot{\mathscr{H}}(x) = y^T u - \frac{\partial \mathscr{H}}{\partial x}^T (x) R(x) \frac{\partial \mathscr{H}}{\partial x} \tag{2.2}$$

In words, the system is dissipative if the total increase in stored energy in the system is equal to the energy injected in the system through its port minus the internal loss due to dissipation. Considering the dissipation $R \geq 0$, Equation 2.2 can be further simplified to:

$$\dot{\mathscr{H}}(x) \leq y^T u \tag{2.3}$$

If the equality doesn't hold, power is generated in the system and the system is said to be active. When $R(x)$ is equal to zero, the system is conservative and does not dissipate any power. When $R(x) > 0$ the system is dissipative.

[1] http://www.20sim.com

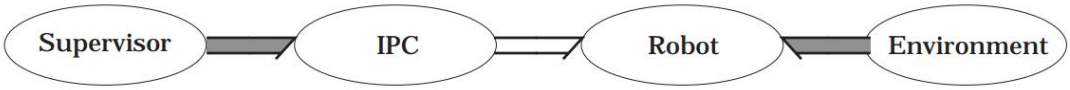


Figure 2.1: IPC controller scheme (Stramigioli and Bruyninckx, 2001). The Supervisor controller can inject energy in the (passive) IPC to let the robot perform work.

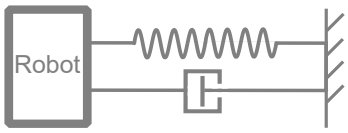


Figure 2.2: Example of a an IPC consisting of a virtual passive spring and damper. The Supervisor controller can inject energy by altering the rest state of the virtual spring.

2.2.1 Passivity of a quadrotor

In case of a quadrotor modeled as a rigid body, the Hamiltonian is a function of gravitational potential energy and kinetic energy:

$$\mathscr{H}(z,p) = \frac{1}{2} p^T M^{-1} p - mgz \tag{2.4}$$

The system is said to be passive if Equation 2.3 holds and $\mathscr{H}(x)$ is bounded from below (García-Canseco et al., 2010). The gravitational potential energy term in the Hamiltonian $\mathscr{H}(z,p)$ is unbounded, so passivity cannot be proven for a quadrotor. However, cyclo-passivity can be proved for the quadrotor as the system cannot generate energy over closed paths in the state space. Cyclo-passivity, in contrast to the stronger passivity property, cannot prove stability of the system as the Hamiltonian is unbounded (Intuitively, a free-falling quadrotor will never reach an equilibrium point, which is consistent with this conclusion).

Passivity guarantees that the total energy in the system will always be bounded by the total energy injected by the environment. Passivity is a strong property that is especially important for interacting systems as it ensures stable behavior during interaction with another, passive environment. To guarantee stability of the controller, the controller should be passive.

2.3 Intrinsically passive control

Intrisically Passive Control (IPC) has been proposed by Stramigioli (2001) and Stramigioli and Bruyninckx (2001). The IPC control scheme as proposed in Stramigioli and Bruyninckx (2001) is depicted in Figure 2.1. The robot is directly connected to an IPC. This is a low-level controller, which is passive by definition. This IPC can be described in bond graph terms as a network of passive elements, which can be translated to virtual components like a spring, mass and a damper. This visualization of the IPC makes the controller highly intuitive. An example of an IPC and its visualization is shown in Figure 2.2: the robot is connected via a virtual controller spring and damper to the setpoint. The Supervisor is a high-level controller that can inject energy in the IPC to enable the robot to perform a given task. For example, this supervisor could change the spring rest state the IPC shown in Figure 2.2 to let the robot reach a given setpoint. This high-level controller is non-passive such that it can inject energy in the system to let the robot perform work.

2.4 Projections

When implementing a control law, the wrench calculated by the controller has to be projected to the space of implementable wrenches spanned by the actuators. This space can be restricted by underactuation or actuator limits of the robot. Consider a controller wrench e_1 and actuator

wrench e_2. If $e_1 > e_{2\max}$, than the controller wrench is scaled (or projected) to $e_{2\max}$. Often, this control wrench projection is done implicitly and does only affect the wrench and not the dual twist which is non-passive. for the projection to be power conservative (passive), the input power (wrench times twist) should be equal to the output power as shown in Equation 2.5. The projection will scale the effort e and flow f according to ratio P_α and P_β respectively as shown in Equation 2.6. Rewriting and substituting Equation 2.6 into 2.5 gives Equation 2.7.

$$e_1{}^T f_1 = e_2{}^T f_2 \tag{2.5}$$

$$\begin{aligned} e_1 &= P_\alpha e_2 \\ f_2 &= P_\beta f_1 \end{aligned} \tag{2.6}$$

$$e_1{}^T f_1 = e_1{}^T P_\alpha^{-T} P_\beta f_1 \tag{2.7}$$

From this equation we can conclude that the projection is only power conservative if $P_\alpha = P_\beta{}^T$, where it behaves like a transformer. A power conservative projection affecting the wrench exerted on the body, will also affect the dual twist and vice versa.

2.5 Energy tank

Every control law can be made passive artificially by actively monitoring and controlling the energy injected by the controller. By allocating an energy budget beforehand, an upper limit on the total energy in the system is set. This concept can be implemented in the form of an energy tank (Stramigioli, 2015), (Dietrich et al., 2016).

By initially filling the energy-tank, the controller can use the energy from the tank to perform a given non-passive action. As the energy was already in the system in the form of energy tank content, no energy is generated within the system so the system will remain passive. During actuation, the energy tank will inject energy and deplete. When tank is empty, no energy can be injected in the system anymore such that passivity is preserved. The controller behavior will change, as only passive actions are allowed. The energy tank can be re-filled by a supervisor controller. By putting an upper limit on the energy tank content, an upper limit on the total energy in the system can be set, assuring safety and stability.

3 Controller design

In this chapter, an energy-based controller for an underactuated quadrotor will be presented. The controller and quadrotor model are described in terms of geometrical twists and wrenches. A short introduction of twists and wrenches can be found in Appendix A. A more comprehensive description is given by Stramigioli and Bruyninckx (2001). The controller and quadrotor model are implemented in bond graphs (Appendix B) using 20-sim simulation software. The flow and effort are respectively the twists and wrenches. The framework and notation used are depicted in Figure 3.2. Frame Ψ_b is the controller setpoint (desired position and orientation) and Ψ_a the quadrotor body-fixed frame. The body-fixed right-handed reference frame is chosen in the center of mass of the quadrotor, as shown in Figure 3.3.

An overview of the controller is depicted in Figure 3.1. An IPC controller scheme (as introduced in Section 2.3) has been used. Note that the high-level supervisor controller is not considered in this thesis, as it is out of the scope of this research (to test the controller, the initial conditions of the controller are set non-zero). The 'Projection' subsystem implements the interface between the IPC and the quadrotor. This projections maps the controller wrench to the subspace of implementable actuator wrenches. This projection can be implemented either passively or non-passively. In Section 3.1, a basic quadrotor model description will be treated. In Section 3.2, the design of the IPC will presented. In the last section of this chapter, Section 3.3, several approaches for implementing the projection subsystem will be treated.

3.1 Quadrotor model

A basic quadrotor model has been implemented in 20-sim to be able to simulate the controller performance and energetic behavior. The quadrotor has been simplified to a single rigid body. The rotor dynamics have been ignored such that the controller wrench can be exerted on the quadrotor body directly.

3.1.1 Body

The bond graph model of the quadrotor body is depicted in Figure 3.4. The 1-junction in the middle of the model has the body-fixed twist $T_a^{a,0}$ as common flow. The I_6 element is the inertia, MGY the fictitious wrench, Se the gravity and the lower MTF the transformation of the power port to the world frame. The controller is connected to the body via the power port W_a. The full mathematical derivation of the model can be found in Appendix C.

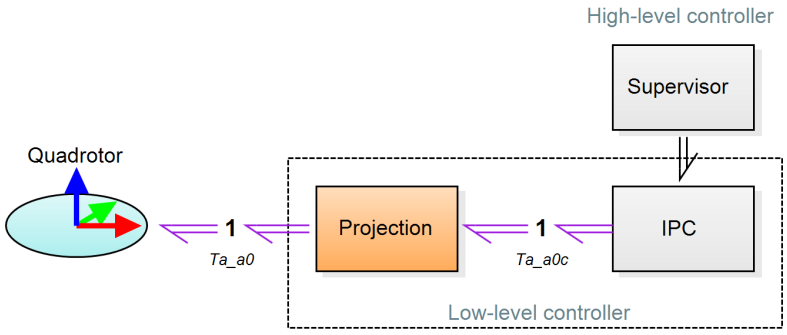


Figure 3.1: Overview of the controlled quadrotor. The Projection subsystem maps the controller wrench from the IPC to the subspace of implementable wrenches. The high-level Supervisor controller is not considered in this thesis.

Figure 3.2: The framework of quadrotor and setpoint. Frame Ψ_a is the body-fixed reference frame and Ψ_b is the setpoint.

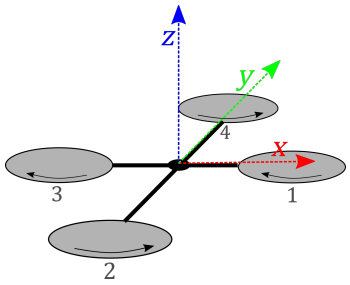


Figure 3.3: Body fixed reference frame Ψ_a of the quadrotor ('+' configuration).

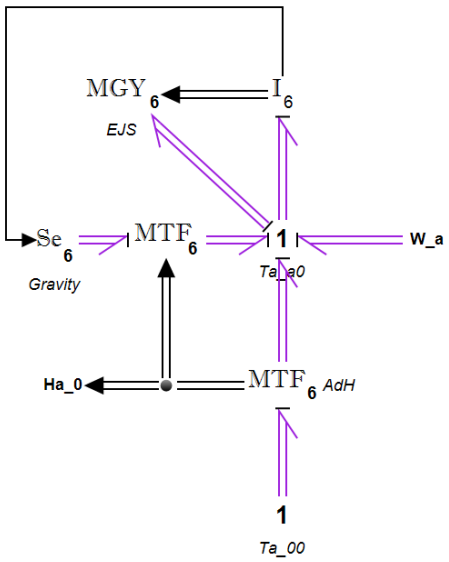


Figure 3.4: Bond graph model of a rigid body. The 1-junction in the middle of the model has the body-fixed twist $T_a^{a,0}$ as common flow. The I_6 element is the inertia, MGY the fictitious wrench, Se the gravity, W_a the power port from the controller and the lower MTF the transformation of the power port to the world frame.

3.1.2 Actuation

The controller outputs a control wrench which should be exerted on the quadrotor body by the actuators. The actuators consist of four rotors with configuration as shown in Figure 3.3. The arrows defines the positive rotation direction of each rotor. Each rotor produces thrust along and a torque around the z-axis. As the wrench exerted by each rotor cannot be controlled or measured directly, the wrench is approximated by controlling the rotor angular velocities. The thrust and torque produced by each rotor are dependent on the relative airspeed, the rotor angular velocity and the thrust and drag factor respectively. If the relative airspeed is assumed to be zero and the thrust and drag factor are assumed to be constant, the relation between the rotor angular velocities squared (Ω^2) and the body-fixed wrench is simplified to a linear map:

$$(W^a)^T = \begin{pmatrix} 0 & -bl & 0 & bl \\ -bl & 0 & bl & 0 \\ d & -d & d & -d \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ b & b & b & b \end{pmatrix} \begin{pmatrix} \Omega_1|\Omega_1| \\ \Omega_2|\Omega_2| \\ \Omega_3|\Omega_3| \\ \Omega_4|\Omega_4| \end{pmatrix} \tag{3.1}$$

where Ω_i is the rotor angular velocity of rotor i (expressed in $\mathrm{rad\,s^{-1}}$), b the thrust factor (expressed in $\mathrm{Nm/(rad/s)^2}$), d the drag factor (expressed in $\mathrm{Nm/(rad/s)^2}$) and l the arm length of the quadrotor in meters. As can be seen in Equation 3.1, the F_x and F_y component of the wrench are unaffected by the rotor angular velocities: the implementable wrench space spanned by the rotors is limited, as the quadrotor is underactuated. If the non-actauted F_x and F_y component of W^a are left out, then Equation 3.1 is invertible such that the rotor speeds can be calculated as function of the requested control wrench. Currently available quadrotor hardware does not measure the actual wrench exerted by the rotors, so the wrench can only be controlled by feed-forward. The error between the control wrench and the actual wrench depends on the accuracy of the map. The actual energy flow between the controller and the quadrotor body **cannot be guaranteed**, limiting the application of the energy-based control scheme.

For a realistic quadrotor model, the thrust limits of the rotors have to be incorporated in the model. As the maximum thrust and torques are all dependent on the rotor speeds, the absolute maximum thrust or torque is obtained when all others are set to zero. These absolute limits are dependent on the minimum and maximum rotor angular velocities of the four rotors and are show in Equation 3.2.

$$\tau_{x_{\max}} = -bl(\Omega_{\min}|\Omega_{\min}|) + bl(\Omega_{\max}|\Omega_{\max}|) \tag{3.2a}$$
$$\tau_{y_{\max}} = -bl(\Omega_{\min}|\Omega_{\min}|) + bl(\Omega_{\max}|\Omega_{\max}|) \tag{3.2b}$$
$$\tau_{z_{\max}} = -2d(\Omega_{\min}|\Omega_{\min}|) + 2d(\Omega_{\max}|\Omega_{\max}|) \tag{3.2c}$$
$$F_{z_{\max}} = 4b(\Omega_{\max}|\Omega_{\max}|) \tag{3.2d}$$

3.2 Controller

The controller consists of an IPC and gravity compensation. The IPC can be described as an impedance, visualized by a network of transformers, a spring, and a damper. The gravity compensation acts to counteract the gravity wrench. In the first subsection, a geometrical spring and damper are presented. In Subsection 3.2.2, a Remote Center of Compliance (RCC) will be introduced. This RCC provides a coupling between the translation and orientation, which provides controllability of the quadrotor in the horizontal world plane. In Subsection 3.2.3, gravity compensation is added to the controller. Finally, in Subsection 3.2.4, a bond-graph model of the complete controller is derived.

3.2.1 Spatial spring and damper

The IPC consists of a spring and a damper. The spatial spring acts to align the frame attached to the end of the spring to a virtual equilibrium frame. The spatial spring can be defined in 6 degrees of freedom. In Fasse and Broenink (1997), it is shown that the choice of compliance parameters is non-trivial, and a spatial compliance family is introduced. This family is fully determined by a virtual equilibrium frame ($SE(3)$), a center of compliance ($\mathbb{R}^3$) and three stiffness-matrices ($\mathbb{R}^{3x3}$). The center of compliance is the point where the rotational and translational elasticity is maximally decoupled. The stiffnesses are described by the translational (K_t), rotational (K_o) and coupling (K_c) stiffness-matrix. The translational stiffness acts to coincide the position of the end-effector to the position of the virtual equilibrium frame and the rotation stiffness the orientation. The coupling stiffness defines the coupling between position and torques, and orientation and forces.

The controller damper is a simple six dimensional damper :

$$W = -BT \tag{3.3}$$

Only the diagonal terms of the damping matrix B are used as they are the most intuitive. Coupling between the rotation and translation will be introduced by the RCC (Subsection 3.2.2).

3.2.2 Remote center of compliance

As a result of the underactuated design of a quadrotor, no force can be generated by the actuators in the body-fixed x/y-plane. To be able to control the quadrotor to a setpoint in the horizontal world plane, the quadrotor should be tilted. Once the quadrotor is tilted, a part of the spring force is decomposed to the actuated z-direction and the quadrotor will start moving towards the setpoint. To accomplish the tilting motion, a coupling between the translation and orientation must be present.

For the spring, this coupling can be achieved by defining a Remote Center of Compliance (RCC). By locating the center of compliance above the quadrotor body, the correct coupling is achieved. This RCC can be visualized with two rigid poles with length l (Figure 3.5). Frame Ψ_b is the frame which is attached to the setpoint and frame Ψ_a is the body-fixed quadrotor frame. Frame Ψ_j is rigidly attached to Ψ_b and Ψ_i to Ψ_a at a distance l in the z-direction. The spring is connected to Ψ_i and Ψ_j. A pure spring force in frame Ψ_i will transform to a force and torque in the body-fixed frame Ψ_a. For the damper, the same coupling between orientation and translation is desirable. A motion in the horizontal world plane should be damped by tilting the quad back from the direction of motion. This behavior is achieved by placing the damper on the same location as the spring, between frame Ψ_j and frame Ψ_i.

The controller as shown in Figure 3.5 can be parameterized in two different ways. The first approach is to connect the spring and damper between frame Ψ_a and Ψ_b and take spring and damper parameters with non-zero coupling terms. The second approach is to connect the

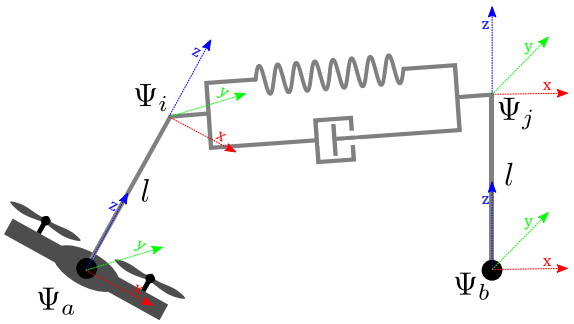


Figure 3.5: Visualization of the proposed controller. A virtual, massless arm with length l is rigidly connected to both the quadrotor (Ψ_a) and the setpoint frame (Ψ_b). The IPC (spring and damper in parallel) are connected to the frames Ψ_i and Ψ_j attached to the end of the arms such that a coupling between the translation and orientation is realized. Note that this is a simplified visualization, as the damper and spring are 6-dimensional in the real controller.

spring and damper to frame Ψ_i and Ψ_j. The second approach introduces the coupling with the arm l such that no coupling terms are needed for the spring and damper parameters. In the author's opinion, this way of visualizing and parameterizing the controller is the most intuitive. Furthermore, the coupling can be tuned with a single parameter, the arm length l. For this reason, this approach will be used in the remainder of this thesis.

Extension to variable RCC

The controller behavior is strongly dependent on the arm length l. The controller could be extended by varying l on-the-fly to improve performance. If the quadrotor is located far away from the setpoint, a large value for l will limit the tilt angle of the quadrotor. When the quadrotor is close to the setpoint, l can be shortened such that the controller dynamics become be faster, which can help to prevent overshoot and oscillations. Stramigioli and Duindam (2001) have shown that changing the center of compliance is passive. However, in general, changing distance l will not only change the center of compliance but will also affect the potential energy stored in the spring. Therefore, changing l is in general not a passive operation (Figure 3.6a).

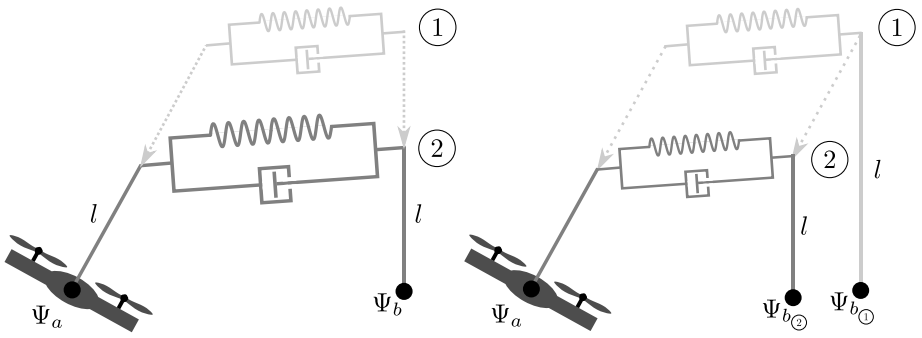
One option to guarantee passivity would be to add a power port to the controller to change the arm length l. A supervisor controller would then be able to inject or extract the energy necessary for the displacement of frame Ψ_i and Ψ_j.

The other option is to change the operation of varying l to a passive operation. The operation is only passive if it does not affect the potential energy in the spring. Mathematically, the twist $T_j^{i,i}$ extending the spring should not be affected by the change of l. By combining the equations of Equation 3.4, Equation 3.5 is obtained. When only the RCC of the spring is allowed to be modified, $T_b^{i,j}$ and $T_b^{i,a}$ should be equal. Only when this is the case, the spring elongation will not be affected and the spring modification is purely a change of the RCC.

$$\begin{aligned} T_j^{i,i} &= T_a^{i,i} - T_a^{i,j} \\ T_a^{i,j} &= T_b^{i,j} - T_b^{i,a} \end{aligned} \tag{3.4}$$

$$T_j^{i,i} = T_a^{i,i} - T_b^{i,j} + T_b^{i,a} \tag{3.5}$$

The passive spring behavior modification is shown graphically in Figure 3.6b. For $T_b^{i,j}$ and $T_b^{i,a}$ to be equal, Ψ_j and Ψ_i should be moved parallel over the same distance. In Figure 3.6a, it is clearly visible that the total spring elongation H_i^j changes. By defining the twist $T_a^{i,i}$ as being always in the z-direction, the passivity preserving twist $T_b^{j,j}$ can be calculated. In all



(a) Non-passively **(b)** Passively

Figure 3.6: Altering controller arm length l. Passivity is lost if the spring state is affected (a). Passivity can be recovered by moving the setpoint such that the spring state remains constant (b).

cases except when Ψ_j and Ψ_i are oriented with the z-axis parallel, a non-zero $T_a^{i,i}$ will need a $T_b^{j,j}$ with a component in the lateral direction w.r.t Ψ_b to preserve passivity. This twist will move frame Ψ_j with respect to the setpoint and will result in an steady-state error between the setpoint and quadrotor position in the end.

Varying the RCC by varying the arm length l is shown to be non-passive as it can affect the spring elongation. The controller could be extended with a power port, such that the supervisor controller could inject energy to change the arm length l passively. As the supervisor controller is out of the scope of this research, a fixed arm length wil be considered in the remainder of this thesis.

3.2.3 Gravity compensation

Gravity exerts a wrench on the quadrotor body. When gravity is not actively compensated, then the gravity wrench will extend the spring in the z-direction until equilibrium is reached. Therefore, the quadrotor will always have a steady-state position error in the z-direction, depending on the spring stiffness and quadrotor mass. Because the quadrotor is underactuated, not the full gravity wrench can be compensated for depending on the tilt angle. Three approaches for gravity compensation will be treated here.

Spring pre-tension

The first option is to place the controller setpoint above the real setpoint, such that the steady-state error compensated for. The spring extension in the z-direction in equilibrium can be calculated as follows:

$$\Delta z = \frac{mG_n}{K_z} \tag{3.6}$$

Where Δz is the spring extension in equilibrium in the z-direction, m the mass of the quadrotor, G_n the gravitational constant and K_z the spring constant in the z-direction. By moving frame Ψ_j for Δz in the z-direction, the steady-state error is compensated for. This method is illustrated in Figure 3.7.

However, this approach has a disadvantage. The pre-tension Δz influences the behavior of the controller by increasing the apparent arm length at the setpoint ($l + \Delta z$). The pre-tension Δz is dependent on K_z, so when one changes K_z, the apparent arm length is also affected. This dependency makes tuning of K_z less intuitive.

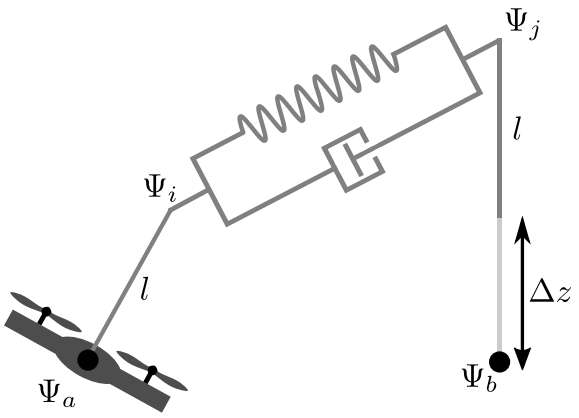


Figure 3.7: Gravity compensation by spring force. Frame Ψ_j is placed Δz higher to put pre-tension on the spring. This influences the behavior of the controller, as the position of frame Ψ_j is modified. Note that Δz is a function of the mass and the spring stiffness in the z-direction and can become large when a large mass or small stiffness is used.

Compensating force in frame Ψ_a

The second option is to compensate the gravity force by a compensating force in frame Ψ_a. Adding a effort source (force) is not clearly non-passive. However, the effort source could also be exchanged by a spring with passive projection and an energy tank for passive compensation. The quadrotor is underactuated, so depending on the orientation of the quadrotor, not the full gravity force can be compensated. Expressed in frame Ψ_a, the gravity force F_g can be decomposed in F_{g_x}, F_{g_y} and F_{g_z}, where only F_{g_z} can be compensated by a thrust F_{c_z}. This is shown in 2D in Figure 3.8a. F_{c_z} and F_{g_z} cancel, but F_{g_x} and F_{g_y} cannot be compensated and remain. Assuming that the controller will keep frame Ψ_i and Ψ_j roughly on the same position, the quadrotor will behave like a pendulum with Ψ_i as pivot point and start swinging. This oscillating behavior is undesired.

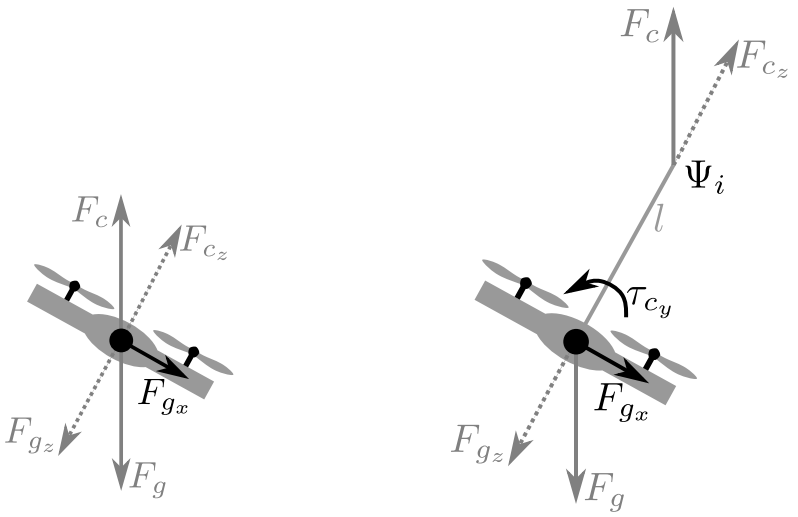
Compensating force in frame Ψ_i

The gravity compensating force F_c can also be defined at the end of the arm in frame Ψ_i as shown in 2D in Figure 3.8b. Adding a effort source (force) is not clearly non-passive. However, the effort source could also be exchanged for a spring with passive projection and an energy tank for passive compensation. Again F_{c_z} and F_{g_z} will cancel, but an additional torque τ_{c_y} will be present. This torque is non-zero if the quadrotor is tilted and will act to level the quadrotor. Together with the controller damping, this torque will improve the stability of the quadrotor and will counter-act the oscillatory behavior. As a result, this method of gravity compensation is chosen.

3.2.4 Controller model

The 20-sim model of the controller is shown in Figure 3.9. The model has the following ports:

- **Taa0c** - Power port carrying the projected $T_a^{a,0}$ entering the controller and the dual wrench. $T_a^{a,0}$ is the twist of quadrotor body fixed frame Ψ_a with respect to the the inertial frame Ψ_0, expressed in Ψ_a.
- **Ha_0** - Signal carrying H_a^0, homogeneous transformation matrix from Ψ_a to Ψ_0.
- **Tb00** - Power port carrying the twist $T_b^{0,0}$ of the setpoint frame Ψ_b with respect to the world frame Ψ_0 expressed in Ψ_0 and the dual wrench.



(a) Compensating in frame Ψ_a. **(b)** Compensating in frame Ψ_i.

Figure 3.8: Gravity compensation by force F_c. The quadrotor is underactuated, so the body-fixed x and y component of F_c cannot be realized. If the compensation is done in the center of mass (a), a netto force will be present. If the compensation is performed at the end of the arm, an additional stabilizing torque will be present.

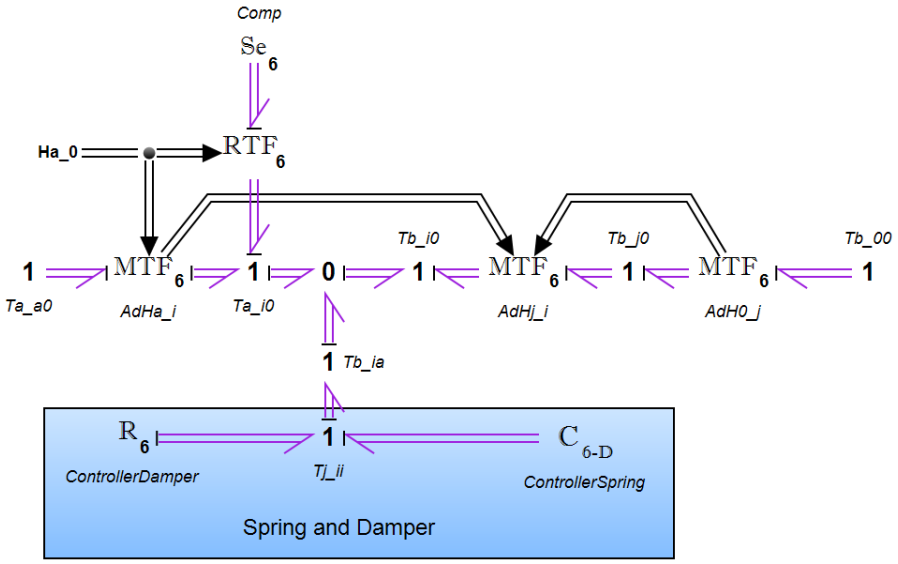


Figure 3.9: 20-sim model of the controller. The spring and damper are shown in the blue box. The gravity compensation is performed in frame Ψ_i by the effort source 'Comp' shown in the upper part of the figure. The MTF's handle the coordinate transformations between frames Ψ_a, Ψ_i, Ψ_b and Ψ_j.

The power port *Taa0c* connects to the quadrotor body via de 'Projection' submodel which will be treated in Section 3.3. This 'Projection' submodel may affect the twist $T_a^{a,0}$ so in general, $T_a^{a,0} \neq T_a^{a,0}{}_c$. As a result, H_a^0 is not integrated using $T_a^{a,0}{}_c$ but given as a signal. The power port *Tb00* can be used to control the setpoint passively by a supervisor controller.

The twists and wrenches carried by *Taa0c* and *Tb00* are first transformed to frame Ψ_i and Ψ_j respectively in the MTF submodels (using Adjoints, see Appendix A). Next, the setpoint twist and wrench is transformed from frame Ψ_j to frame Ψ_i. Now that the body and setpoint twist/wrench are both expressed in frame Ψ_i, the twist of the setpoint w.r.t. the body expressed in frame Ψ_i can be calculated:

$$T_b^{i,a} = T_a^{i,0} - T_b^{i,0} \tag{3.7}$$

which is represented in the 20-sim model as a 0-junction. Finally, the twist $T_j^{i,i}$ can be calculated:

$$T_j^{i,i} = T_b^{i,a} + T_a^{i,i} - T_b^{i,j} \tag{3.8}$$

As frame Ψ_i is rigidly connected to Ψ_a, and Ψ_j to Ψ_b (the arm length l is constant), $T_a^{i,i}$ and $T_b^{i,j}$ are both equal to zero. Equation 3.8 then simplifies to:

$$T_j^{i,i} = T_b^{i,a} \tag{3.9}$$

This twist is represented in the model by the 1-junction in the lower part of Figure 3.9. The controller spring and damper are connected to this 1-junction. The effort source 'Comp' on the top of the model is the gravity compensating effort source. This force is expressed in the gravity frame attached to Ψ_i. This frame has the same origin as Ψ_i but the orientation of the world frame Ψ_0. The RTF transforms the force to Ψ_i.

3.3 Projection

Because the implementable wrench space (spanned by the actuators) is smaller than the controller wrench space, a projection between the controller and quadrotor is needed. The quadrotor cannot exert a force in the x/y-plane and the all the thrust and torques are limited by the maximum motor speed (see Section 3.1.2). The projection limits the effort co-vector from the controller before entering the quadrotor model and is shown in Figure 3.1. This projection can be performed passively or non-passively (see Section 2.4).

3.3.1 Non-passive projection

An option would be to saturate the wrench from the controller, while the twist is left unaffected. In general this is not energy-conservative, as the projection is only performed on the effort and not on the dual flow. In other words, $P_\alpha \neq P_\beta^T$ (see Section 2.4). In an experiment, the loss of passivity can be shown. Think of the quadrotor located on a steady-state setpoint. Next, an environment could fix the orientation of the quadrotor, and then start moving the quadrotor in the non-actuated plane. The controller will not be able to exert a force in this plane, but the spring will extend as only the force is affected by the projection. Assuming no air drag, the environment will not need to consume any energy to move the quadrotor to another position in the non-actuated plane. The controller spring will extend, so the total energy in the system has increased, while no external energy has been injected.

3.3.2 Passive projection

The projection is only power conservative if $P_\alpha = {P_\beta}^T$, where it behaves like a transformer (Section 2.4). In words, a power conservative projection affecting the wrench (effort) exerted on the body, will also affect the dual twist (flow) and vice versa.

In the actuated directions, this means that when the wrench is limited, the twist will be limited too. If the wrench exceeds the maximum value and the wrench is limited, the twist entering the controller will be different than the actual body twist as the dual twist is limited too. In the end, this will result in a steady-state position error.

In the non-actuated directions, the limit is always zero. When passive projection is used, the twist in these directions will also be set to zero. Any motion in the non-actuated x/y-plane will therefore never be able to enter the controller, so will not be 'felt' by the controller spring or damper. Think of a quadrotor in steady-state and an external force is exerted in the non-actuated plane by the environment such that the quadrotor will start moving. This motion will not enter the controller, so the spring will not be extended and no damping force is generated. The quadrotor can move freely in the non-actuated plane. Furthermore, when the quadrotor tilts, the gravitational force will also have a component in the non-actuated plane (as shown in 2D in Figure 3.10). In this figure, F_z is in the actuated direction and will extend the spring until it reaches equilibrium, but F_x is in the non-actuated direction so will not result in controller damping/spring extension. The quadrotor will start 'free falling' in the body-fixed x-direction. The quadrotor will never be able to reach the setpoint.

3.3.3 Energy-tank passivized projection

It is possible to compensate for the part of the twist which is projected out by the energy-conservative projection. This compensating twist T_{balance} can be generated by an energy tank containing a limited amount of energy, without loss of passivity (Stramigioli, 2015). When the energy tank is filled, the energy tank will compensate the projection by injecting T_{balance}. When the energy tank is empty, T_{balance} is set to zero to retain passivity.

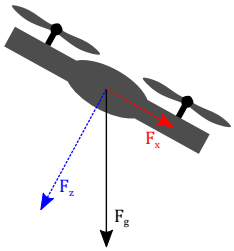


Figure 3.10: Gravity force acting on the quadrotor decomposed. If the quadrotor is tilted, the gravity force will have a component in the non-actuated plane (F_x). If passive projection is used, the quadrotor will start sliding in the non-actuated plane and will never stop, as the motion does not enter the controller.

The compensation should equalize the twist of the quadrotor body and the twist entering the controller (Equation 3.10). The twist entering the controller is equal to the projected twist with the compensating twist added (Equation 3.11 and 3.12), provided by the energy tank.

$$T_a^{a,0}{}_{\text{controller}} = T_a^{a,0} \tag{3.10}$$

$$T_a^{a,0}{}_{\text{controller}} + T_{\text{balance}} = T_a^{a,0}{}_{\text{proj}} \tag{3.11}$$

$$T_{\text{balance}} = T_a^{a,0}{}_{\text{proj}} - T_a^{a,0} \tag{3.12}$$

The projection with the energy tank added is shown in Figure 3.11. In bond graph terms, Equation 3.11 is translated in a 0-junction. The energy tank has a flow-out causality. The energy tank behavior is described in the Equations shown in 3.13. Here E is the stored energy, W the input wrench, T the output twist and S the state. I is chosen such that the stored energy does not depend on I, and is zero when the energy tank is empty and the power flow is not towards the energy tank.

$$E = \frac{1}{2}S^2 \tag{3.13a}$$

$$\frac{\partial S}{\partial t} = I^T W_{\text{balance}} \tag{3.13b}$$

$$T_{\text{balance}} = I\frac{\partial E}{\partial S} \tag{3.13c}$$

$$I = \begin{cases} (T_a^{a,0}{}_{\text{proj}} - T_a^{a,0})/\frac{\partial E}{\partial S} & S > 0 \text{ or } W^T T < 0 \\ 0 & S \leq 0 \text{ and } W^T T \geq 0 \end{cases} \tag{3.13d}$$

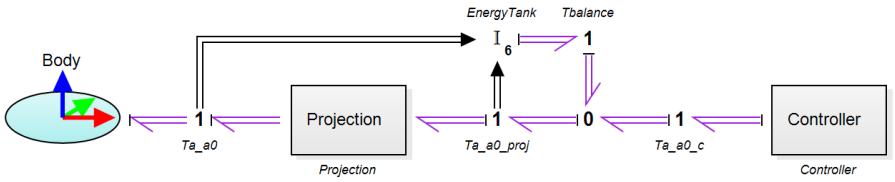


Figure 3.11: The controller with passive projection extended with an energy tank. The energy tank acts to equalize the projected twist entering the controller and the body twist by injecting a balancing twist while energy is available. If the energy tank is depleted, the balancing twist is set to zero to preserve passivity.

4 Simulations

Simulation have been performed on the controller in 20-sim. The body-fixed reference frame is chosen as in Figure 3.3. The quadrotor is modeled as a single rigid body (Section 3.1). No rotors have been simulated and zero air drag is assumed. The physical parameters of the quadrotor used in the simulations in this chapter are reported in Table 4.1. The effect of the passive or non-passive projection will be shown in Section 4.1. The controller behavior will be treated in Section 4.2.

4.1 Controller passivity

In this section, the effects of different projection approaches on the passivity of the system will be shown. The system consists of the quadrotor and controller combined. The total energy in the system E_{tot} is defined as energy stored in both controller and plant added. In the controller, potential energy is stored in the virtual spring. In the quadrotor body, energy is stored as kinetic energy of the rigid body and as potential energy. The system is said to be passive if the increase in E_{tot} is always equal or less than the power injected externally by the environment E_{ext}. In other words, the system is passive if and only if:

$$\frac{\mathrm{d}}{\mathrm{d}t}E_{\text{tot}} \leq \frac{\mathrm{d}}{\mathrm{d}t}E_{\text{ext}}, \qquad \forall\, t \tag{4.1}$$

Rewritten, $E_{\text{tot}} - E_{\text{ext}}$ should never increase:

$$\frac{\mathrm{d}}{\mathrm{d}t}(E_{\text{tot}} - E_{\text{ext}}) \leq 0, \qquad \forall\, t \tag{4.2}$$

For the passivity simulations, the rotational K_o and coupling spring stiffnesses K_c are set to zero. The translational spring stiffness K_t is a diagonal matrix with all elements equal to $5\,\mathrm{Nm}^{-1}$. The diagonal terms of the controller damping B are set to $0\,\mathrm{Nm/(rad/s)}$ for the rotational damping elements and $5\,\mathrm{N/(m/s)}$ for the translational damping elements. The arm length l is set to 1 m. The air drag has been assumed negligible and is set to zero.

4.1.1 Underactuation

In this simulation, the effect on the passivity of the system for both the the passive- and non-passive projection approaches will be shown. The quadrotor is non-actuated in the body-fixed x/y-plane, so the controller wrench terms in this plane are always projected to zero. When only the wrench is projected, passivity will be lost. When both the wrench and dual flow are projected, passivity is preserved. This loss or preservation of passivity will be shown by moving the quadrotor in the non-actuated x-direction.

The environment fixes the orientation of the quadrotor and then starts moving the quadrotor from the origin over the x-axis to $x = 1\,\mathrm{m}$. The controller will not be able to exert a force in this non-actuated direction, so no energy is injected by the environment. With non-passive

Name	Parameter	Value	Unit
Mass	m	1.15	kg
Moment of Inertia about x axis	I_x	0.026	kgm^2
Moment of Inertia about y axis	I_y	0.038	kgm^2
Moment of Inertia about z axis	I_z	0.052	kgm^2

Table 4.1: Quadrotor physical parameters used in simulation

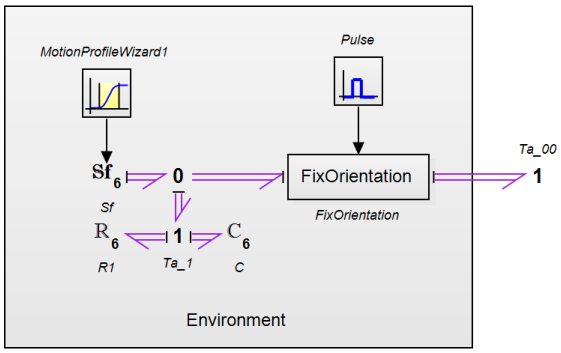


Figure 4.1: Environment for underactuation passivity simulation. The environment fixes the orientation of the quadrotor and moves the quadrotor slowly to $x = 1$ m. Next, the quadrotor orientation is detached from the environment and the environment moves the quadrotor slowly back to $x = 0$.

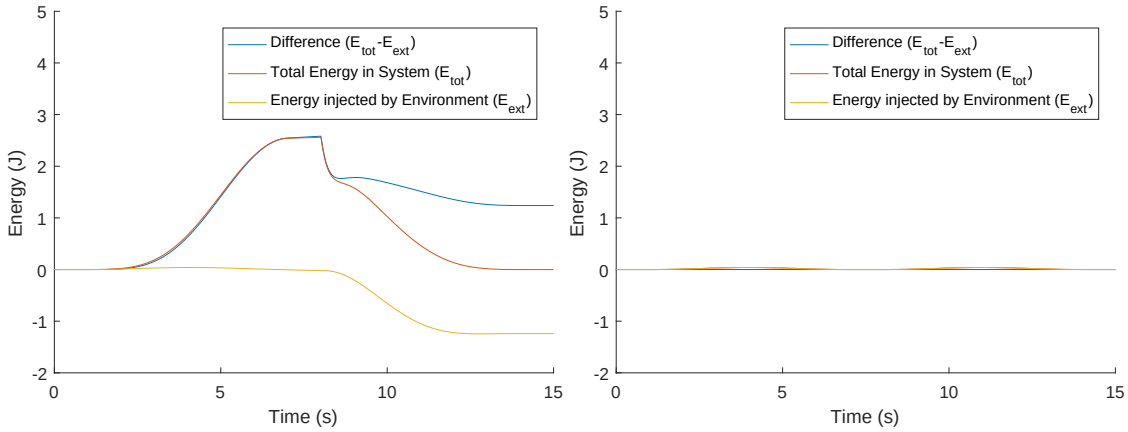
projection, the spring will extend as the flow is unaffected. Next, the environment stops fixing the orientation of the quad. As the spring is extended, the quadrotor will pitch towards the origin. Finally, the environment moves the quadrotor back to the origin. As the quadrotor is pitched, it will be able to exert a force and therefore will release energy to the environment while no energy has been injected by the environment.

Simulation

The environment fixes the orientation using a flow source connected to the quadrotor body via a stiff spring and damper as shown in Figure 4.1. The spring and damper in the external environment are added to prevent causality conflict, as both the flow source and inertia have flow-out causality. The flow source is set zero except for the velocity in the world frame x-direction, where the flow is positive during the period $1\,\text{s} < t < 7\,\text{s}$, moving the quadrotor to $x = 1$ m. From $t = 7$ s, the quadrotor orientation is disconnected from the environment, such that the orientation is free again. The flow source is negative during $8\,\text{s} < t < 14\,\text{s}$ moving the quadrotor slowly back to the origin. The simulation will be performed for both the passive- and non-passive projection.

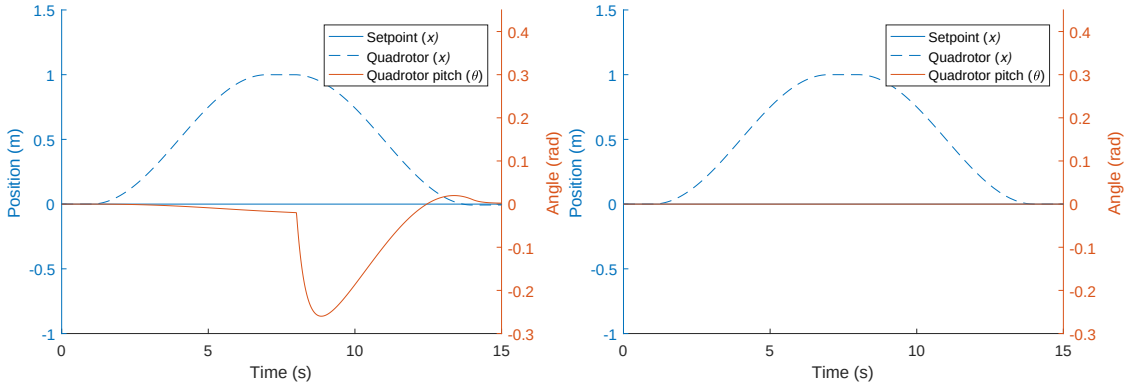
Results

The simulation results are presented in Figure 4.2, 4.3 and 4.4. The simulation performed with non-passive projection is shown on the left of the figures, and the passive projection on the right. Figure 4.2 shows the energy in the system. If non-passive projection is used (Figure 4.2a), the total energy in the system is able to increase without energy being injected by the environment. When the quadrotor has returned to the origin, energy has been extracted by the environment. This non-passive behavior is as expected, as the spring is able to extend while no external energy is injected. For the projection to be passive, if the wrench is projected, the dual twist should be projected too (Figure 4.4b). The velocity in the x-direction does not enter the controller, so the spring does not extend and the quadrotor will not move (Figure 4.3b). The difference between E_{tot} and E_{ext} (Figure 4.2b) is zero which is indeed passive.



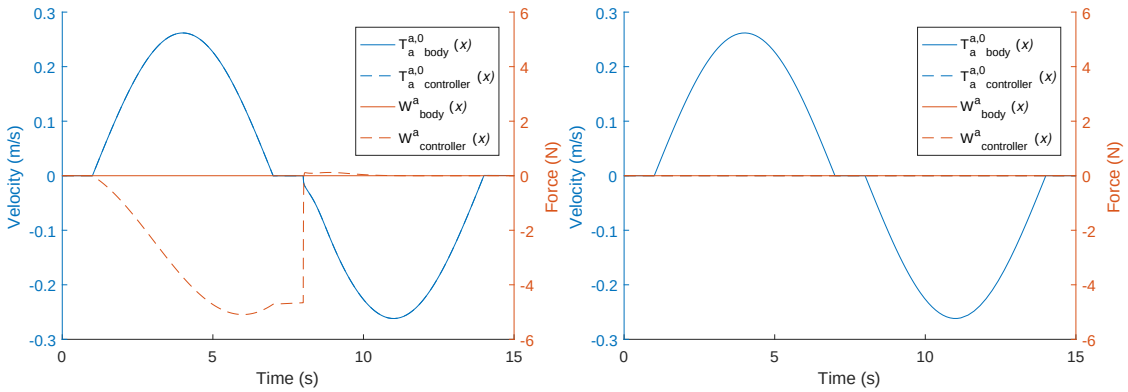
(a) Non-passive projection **(b)** Passive projection

Figure 4.2: Energy plot of underactuation simulation. When non-passive projection is used, energy is extracted from the system by the environment (a). Between 1 s $< t <$ 7 s, the spring extends without the environment injecting energy. The orientation of the quadrotor is released from $t = 8$ s. The quadrotor tilts to the origin and releases energy to the environment. With passive projection, the spring will not extend as both the effort and dual flow are affected (b).



(a) Non-passive projection **(b)** Passive projection

Figure 4.3: Position plot of underactuation simulation. From $t = 8$ s, the orientation of the quad is released. The spring tilts the quadrotor towards the origin if non-passive projection is used (a). With passive projection, the spring is not extended as both the effort and dual flow are affected (b).



(a) Non-passive projection **(b)** Passive projection

Figure 4.4: Velocity and force plot of underactuation simulation. When non-passive projection is used, only the wrench is affected (a). With passive projection, both the wrench and dual twist are affected (b).

4.1.2 Underactuation with energy tank

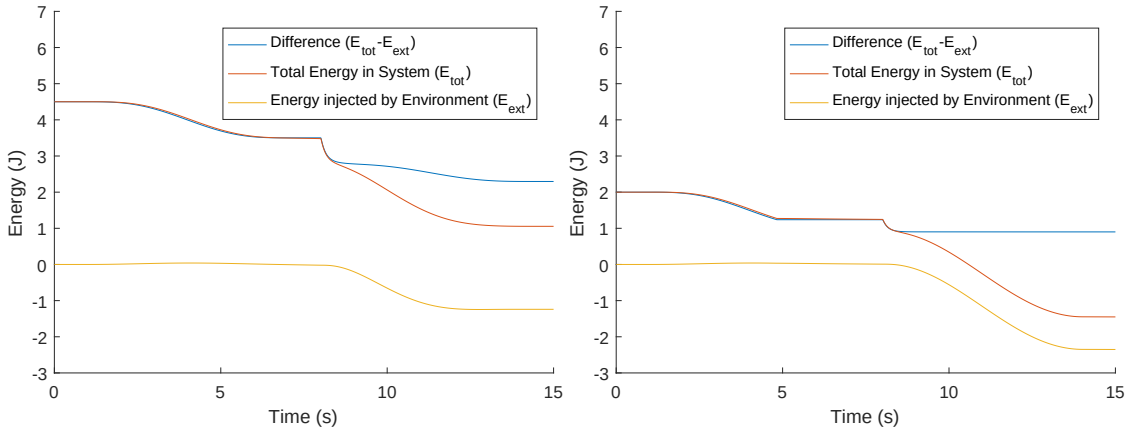
The underactuation simulation (Subsection 4.1.1) has shown that passive projection affects both the wrench and twist. Because the twist is also projected, the quadrotor is not controllable in the body-fixed x/y-plane as the motion will not enter the controller. In Subsection 3.3.3 an energy tank has been proposed to 'balance' the twist which is projected-out by the passive projection. In this simulation, the underactuation experiment of Subsection 4.1.1 will be repeated, but with only passive projection and with an energy tank added.

Simulation

The environment is identical to the environment of Subsection 4.1.1 and is depicted in Figure 4.1. Only passive projection is considered. An energy tank has been added to the controller to balance the twist (Figure 3.11). Two simulations have been performed for two different values for the initial state of the energy tank, such that the behavior of both a filled and depleted energy tank is visualized. In the first simulation, the initial state S_{init} is set to $3\,\text{kg}\,\text{m}\,\text{s}^{-1}$ which translates to 4.5 J of initial energy in the tank. In the second simulation, S_{init} is reduced to $2\,\text{kg}\,\text{m}\,\text{s}^{-1}$ which translates to 2 J.

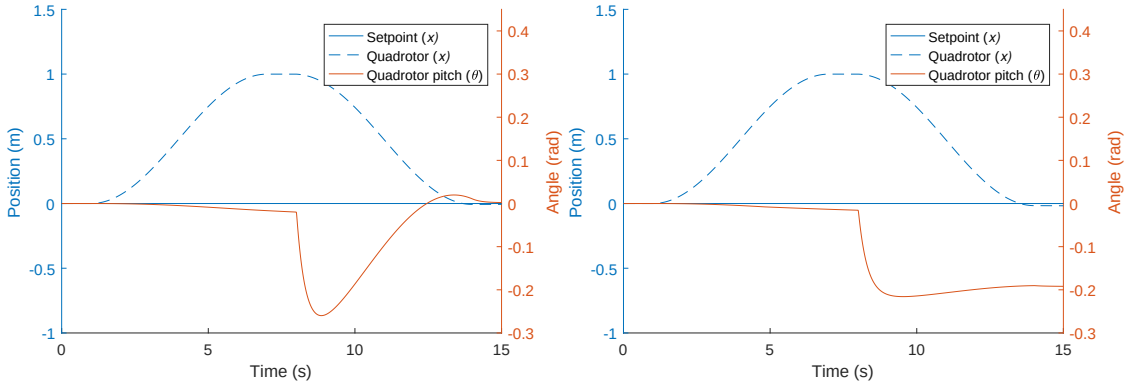
Results

The results of the simulation are presented in Figure 4.5, 4.6 and 4.7. The left subfigures have $S_{\text{init}} = 3\,\text{kg}\,\text{m}\,\text{s}^{-1}$, the right subfigures $S_{\text{init}} = 2\,\text{kg}\,\text{m}\,\text{s}^{-1}$. The difference in energy in both Figure 4.5a and 4.5b are always decreasing, so the system with the energy tank is passive. In Figure 4.7a, it can be seen that the energy tank equalizes ${T_a^a}^0{}_{\text{body}}$ and ${T_a^a}^0{}_{\text{controller}}$ by injecting energy. In Figure 4.7b, the initial state of the energy tank is not sufficient and the energy tank equalizes the twists until the tank is depleted at $t \approx 5\,\text{s}$.



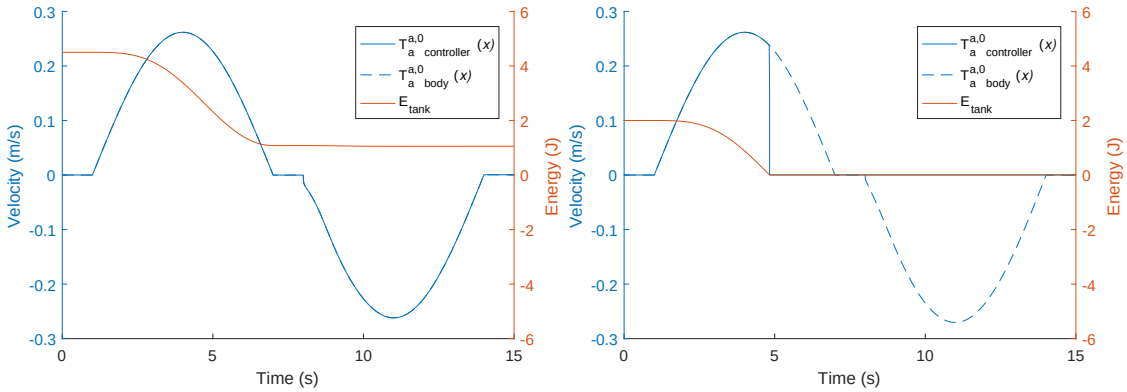
(a) $S_{\text{init}} = 3\,\text{kg}\,\text{m}\,\text{s}^{-1}$ **(b)** $S_{\text{init}} = 2\,\text{kg}\,\text{m}\,\text{s}^{-1}$

Figure 4.5: Energy plot of underactuation simulation with energy tank. The system is passive, as $E_{\text{tot}} - E_{\text{ext}}$ is always decreasing. The energy required for extending the spring is provided by the energy tank.



(a) $S_{\text{init}} = 3\,\text{kg}\,\text{m}\,\text{s}^{-1}$ **(b)** $S_{\text{init}} = 2\,\text{kg}\,\text{m}\,\text{s}^{-1}$

Figure 4.6: Position plot of underactuation simulation with energy tank. Subfigure a shows similar behavior to Figure 4.3a. In Subfigure b, the tank depletes at $t \approx 5\,\text{s}$. When the orientation is released, the quadrotor tilts towards the origin but the motion in the underactuated plane does not enter the controller as the energy tank is empty. Therefore, the pitch remains constant.



(a) $S_{\text{init}} = 3\,\text{kg}\,\text{m}\,\text{s}^{-1}$ **(b)** $S_{\text{init}} = 2\,\text{kg}\,\text{m}\,\text{s}^{-1}$

Figure 4.7: Velocity and energy tank plot of non-actuation simulation with energy tank. The energy tank balances the twist until the tank is depleted, which happens at $t \approx 5\,\text{s}$ in Subfigure b. When the tank is empty, the balancing twist is set to zero to preserve passivity.

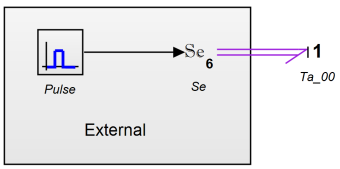


Figure 4.8: Environment for limiter passivity simulation. Between 1 s $\leq t \leq$ 5 s, the environment pushes the quadrotor down with a constant force in the world z-direction extending the controller spring such that the spring force exceeds the actuator limits ($F_z > F_{z_{\max}}$). From t = 5 s, the environment force is set to zero.

4.1.3 Wrench limiter

In this subsection, actuation limits in the actuated direction will be simulated. The wrench which can be exerted in the actuated direction is limited. When the controller wrench exceeds this limit, the wrench is effectively projected. When only the wrench is projected, passivity will be lost. When both the wrench and dual flow are projected, passivity is preserved. This loss or preservation of passivity will be shown by moving the quadrotor in the actuated z-direction in such a way that the controller thrust exceeds the maximum thrust $F_{z_{\max}}$.

The quadrotor will be pulled downwards in the z-direction such that the spring extends. At a certain point, the spring force will be larger than the maximum thrust $F_{z_{\max}}$ that can be produced by the actuators. At this point, the wrench will be limited (projected). If non-passive projection is used, only the wrench is affected. If passive projection is used, also the dual flow entering the controller will be limited.

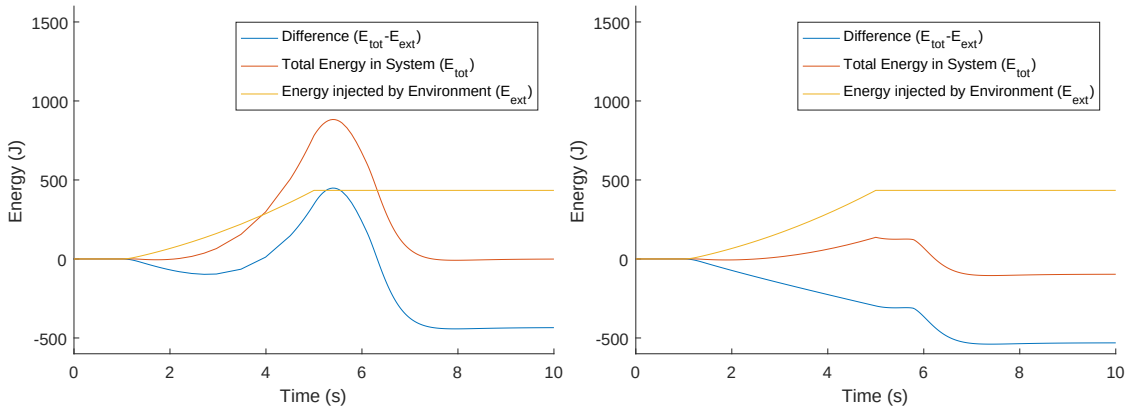
Simulation

The environment is shown in Figure 4.8. The environment exerts a constant force of −22 N in the z-direction (such that it is adds up with the gravity force) in the period 1 s $\leq t \leq$ 5 s and a force of 0 N from t = 5 s. The controller wrench in the z-direction is limited to $F_{z_{max}}$ = 32 N. The simulation will be performed for both the passive- and non-passive projection. With non-passive projection, the twist is unaffected. With passive projection, when the wrench is limited the twist is also affected (Equation 4.3) such that the projection is passive.

$$v_{z_{\text{controller}}} = \begin{cases} v_{z_{\text{body}}} & F_z \leq F_{z_{\max}} \\ \frac{F_{z_{\max}}}{F_z} v_{z_{\text{body}}} & F_z > F_{z_{\max}} \end{cases} \tag{4.3}$$

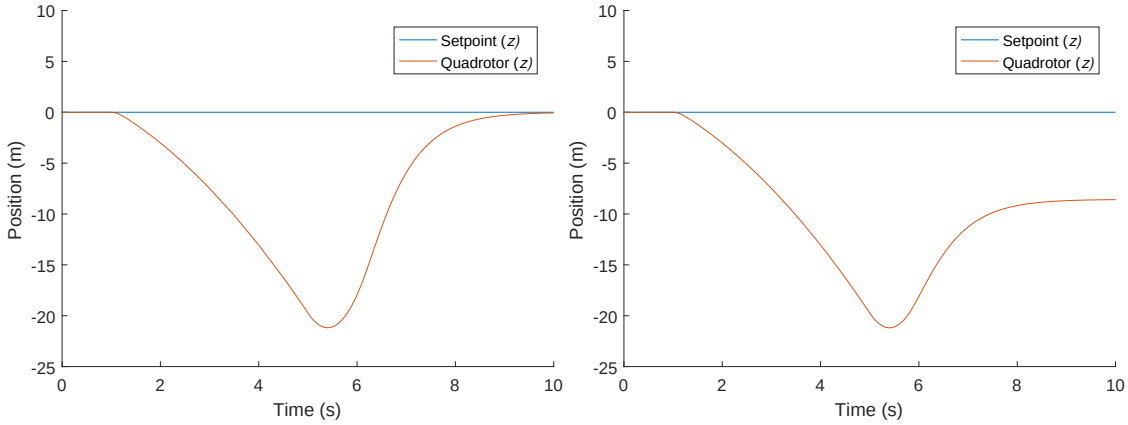
Results

The results of the simulation are presented in Figure 4.9, 4.10 and 4.11. If non-passive projection is used, than the difference between energy in the system E_{tot} and the energy injected by the environment E_{ext} is able to increase (Figure 4.9a) as only the force and not the velocity is affected by the projection (Figure 4.11a). When passive projection is used, the velocity is also affected (Figure 4.11b). Because the velocity is also affected, a steady-state error will be introduced (Figure 4.10b).



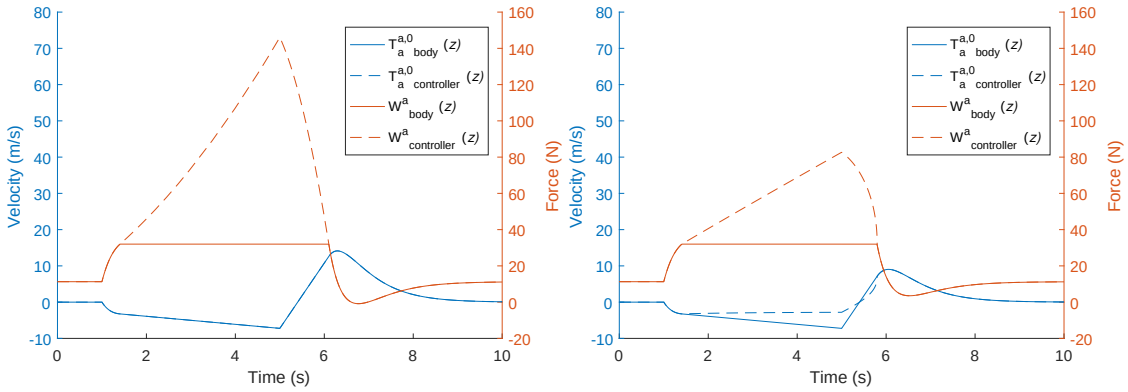
(a) Non-passive projection **(b)** Passive projection

Figure 4.9: Energy plot of limiter simulation. When non-passive projection is used, $E_{\text{tot}} - E_{\text{ext}}$ is momentary increasing when the controller force F_z exceeds the limit $F_{z_{\max}}$ which is not passive.



(a) Non-passive projection **(b)** Passive projection

Figure 4.10: Position plot of limiter simulation. With passive projection (Subfigure b), both the force and velocity in the z-direction are affected by the projection. Less energy enters the spring, which results in a steady-state error in the z-position when the quadrotor is released by the environment.



(a) Non-passive projection **(b)** Passive projection

Figure 4.11: Velocity and force plot of limiter simulation. When non-passive projection is used, only the wrench is affected (a). With passive projection, both the wrench and dual twist are affected (b).

4.1.4 Conclusion

In this section both passive and non-passive approaches have been simulated to map the controller wrench to the implementable wrench subspace. The implementable wrench subspace is zero in the non-actuated plane, and bounded in the actuated space. If non-passive projection is used, energy can be generated in the controller while no energy needs to be injected by the environment. For the system to be passive, the projection limiting the controller wrench should also affect the twist entering the controller. In the actuated space, this limiting of the twist will introduce a steady-state error. In the non-actuated plane, the limit is zero. A motion in this plane will not enter the controller making the quadrotor **uncontrollable** in this plane.

The real body twist can be recovered by adding an energy tank which injects a balancing twist after the passive projection. The energy tank is filled with a given amount of energy, and will balance the twist until the energy tank is empty. By limiting the amount of energy in the tank, an upper limit is set on the total energy in the system as the system is passive. Besides the spring initial conditions, the energy tank state could be set by a supervisor controller giving the IPC an energy budget to perform a given task.

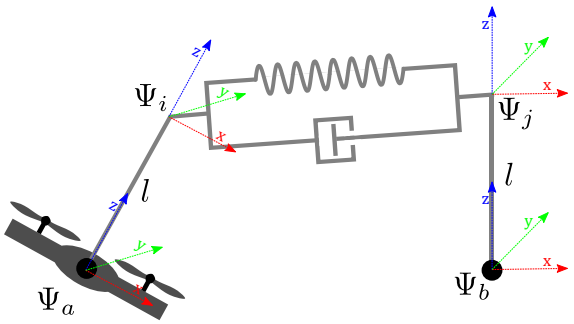


Figure 4.12: Simplified visualization of the proposed controller. Tuning parameters are the spring stiffnesses, the damping and the arm length l. By tuning l, the coupling between the translation and orientation is modified.

4.2 Controller behavior

Simulations have been performed in 20-sim to gain insight in the controller behavior. In this thesis, the supervisor controller that controls the energy budget is not considered. Therefore, in the simulations in this section, passive controller projection with an infinite energy tank budget is considered (note that this is equivalent to non-passive projection) and a non-passive effort source is used for gravity compensation (which could be passivized by using a spring, passive projection and an energy tank). In Subsection 4.2.1, the controller parameters are introduced. Next, in Subsection 4.2.2, simulations will be performed to evaluate the behavior of the controller.

4.2.1 Parameters

In this subsection the controller parameter will be introduced. The proposed controller is visualized in Figure 4.12. The tuning parameters are the spring stiffnesses, the damping and the arm length l. The spring stiffnesses and damping are matrices. Only the diagonal terms of the matrices will be considered non-zero as they are the most intuitive terms. As the quadrotor is underactuated, the reachable configuration space of the quadrotor in steady-state is limited to the position and the orientation around the body-fixed z-axis (yaw). The roll and pitch (orientation about the x- and y-axis) of the quadrotor are coupled to the change in position and are therefore not free.

Spring

In Subsection 3.2.1 it is shown that the controller spring is described by three stiffness matrices ($\mathbb{R}^{3x3}$) defined as the translational (K_t), rotational (K_o) and coupling matrix (K_c). Only the diagonal terms are considered as variables. The translational stiffness K_t acts to coincide the origins of frame Ψ_i and Ψ_j. Because these frames are defined above the body-fixed frame Ψ_a and setpoint frame Ψ_b respectively, k_{t_x} and k_{t_y} will act to tilt and pull the quadrotor towards the setpoint. If the translational stiffnesses k_t are chosen non-equal, the spring becomes an anisotropic spring. The spring is still passive, but an additional torque will be present depending on the quadrotor configuration. Therefore, an isotropic spring will behave more intuitive.

$$K_t = \begin{bmatrix} k_{t_x} & 0 & 0 \\ 0 & k_{t_y} & 0 \\ 0 & 0 & k_{t_z} \end{bmatrix} \tag{4.4}$$

The rotational stiffness K_o acts to orientate frame Ψ_i as frame Ψ_j. Non-zero terms for k_{o_x} and k_{o_y} would counteract the tilting action of the translational spring and damper. Only the term k_{o_z} is set non-zero to be able to control the yaw.

$$K_o = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & k_{o_z} \end{bmatrix} \tag{4.5}$$

The coupling stiffness matrix (K_c) describes the coupling between the orientation and translation. This coupling is already achieved by placing the spring at the end of the arm between frame Ψ_i and Ψ_j, so the coupling matrix is set to zero.

Damper

The damping is described by the damping matrix B ($\mathbb{R}^{6x6}$). Only the diagonal terms are considered as variables. The rotational terms b_{ω_x}, b_{ω_y} and b_{ω_z} act to dampen the angular velocity of the quadrotor. Due to the arm, the translational terms b_{v_x} and b_{v_y} act to dampen the translational velocity of the quadrotor by tilting the quadrotor back. The translational velocity in the z-direction is damped by b_{v_z}. Non-zero terms for b_{ω_x} and b_{ω_y} would counteract the tilting action of the translational spring and damper and are set to zero. The term b_{ω_z} is set non-zero to be able to control the yaw. The damping matrix with the tuning parameters is shown in Equation 4.6.

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & b_{\omega_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{v_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{v_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & b_{v_z} \end{bmatrix} \tag{4.6}$$

Arm length

With the arm length l (Figure 4.12), the coupling between the translation and the orientation of the quadrotor is tuned. Furthermore, a large arm length l limits the quadrotor tilt. If l is chosen small, the quadrotor will tilt more easily and have a faster and more aggressive behavior. With this single parameter, the behavior of the quadrotor can be tuned intuitively.

4.2.2 Simulation

In this subsection the behavior of the controller will be treated. The effect of the spring stiffness, the damping and the arm length on the behavior will be shown. The quadrotor initial position is set to $x = 5\,\text{m}$ and the setpoint is set to the origin. The quadrotor will move through the 2D x/z-plane, so only the parameters B_{v_x}, B_{v_z}, K_{t_x}, K_{t_z} and l will have an effect on the behavior. First, the spring and damper parameters are varied. Next, the arm length l is modified. The physical parameters of the quadrotor used in the simulation can be found in Table 4.1. The air drag has been assumed negligible and is set to zero.

Spring and damper

In Figure 4.13 the result of the simulation for the damping $B_{v_x}, B_{v_z} = 3.3\,\text{N/(m/s)}$ and stiffnesses $K_{t_x}, K_{t_z} = 2.9\,\text{N m}^{-1}$ are presented. The spring pitches the quadrotor towards the origin and the damper pitches the quadrotor back when the quadrotor is close to the origin to prevent overshoot. The parameters are well-tuned and show quick convergence. In Figure 4.14, only the spring stiffnesses K_{t_x} and K_{t_y} are both increased from $2.9\,\text{N m}^{-1}$ to $5\,\text{N m}^{-1}$. The quadrotor pitches more aggressively towards the origin resulting in an overshoot in the position. In Figure

	$l = 1$	$l = 10$
Settling time (s)	2.47	7.26

Table 4.2: Settling time for two arm lenghts. The longer the arm length, the longer the settling time. The settling time is defined as the time it takes for the x-position of the quadrotor to stay within 1% bounds of the setpoint.

4.15, only the damping B_{v_x} and B_{v_z} are increased from 3.3 N/(m/s) to 5 N/(m/s). The position slowly converges while the pitch oscillates. The quadrotor is susceptible to oscillations as it is not damped in the body-fixed x/y-plane.

Remote center of compliance

The arm length l defines the position of the Remote Center of Compliance (RCC). The same simulation is repeated while keeping the spring stiffness and damping constant, and varying the arm length. The parameters for the damping are B_{v_x}, B_{v_z} = 3.3 N/(m/s) and stiffnesses $K_{t_x}, K_{t_z} = 2.9\,\mathrm{N\,m^{-1}}$. The simulation is performed for l=1, 3 and 10 m. The results are presented in Figure 4.16, 4.17 and 4.18 and show respectively the x-position, z-position and pitch angle for the three arm lenghts. The results show that longer arm lengths (or higher placed RCC) have a direct impact on the behavior. For the given parameters can be concluded that the larger the arm length, the slower the response and the larger the overshoot.

The spring stiffness and damping should be re-tuned for each arm length to prevent overshoot. The simulation has been repeated for arm length $l = 1$ m and $l = 10$ m. For both arm length, the spring and damper parameters have been re-tuned to give critically damped convergence. For $l = 1$ m, the parameters are tuned to $B_{v_x}, B_{v_z} = 3.3$ N/(m/s) and $K_{t_x}, K_{t_z} = 2.9\,\mathrm{N\,m^{-1}}$. For $l = 10$ m, the parameters are reduced to $B_{v_x}, B_{v_z} = 1.05$ N/(m/s) and stiffnesses $K_{t_x}, K_{t_z} = 0.29\,\mathrm{N\,m^{-1}}$. The results of the simulation are presented in Figure 4.19, 4.20 and 4.21. Both arm lenghts show convergence without overshoot. The settling time has been calculated for both arm lengths and are presented in Table 6.2. For $l = 1$ m, the maximum pitch angle is 0.32 rad and the position converges quickly (≈ 2.5 s). For $l = 10$ m, the maximum pitch angle is reduced to 0.036 rad and converges much slower (≈ 7.26 s).

Conclusion

In this subsection the behavior of the controller for position control has been treated. The damping and stiffness should be re-tuned for each mass and arm length to avoid damped oscillations in the non-actuated plane. The air drag has been assumed negligible so there is no damping in the non-actuated plane. Therefore, the quadrotor behaves like a pendulum, with frame Ψ_i as moving pivoting point (Figure 4.22). When the quadrotor is pitched, the gravity component in the non-actuated plane will start the quadrotor to 'swing' around frame Ψ_i. This 'swing' motion will be counter-acted by the stabilizing torque provided by the gravity compensation in frame Ψ_i and the controller damping. For each mass and arm length (defining the eigen frequency of the pendulum motion), there is only one stiffness and damping which results in a perfect motion towards the setpoint such that no oscillation occurs.

The arm length is a key parameter for altering the behavior of the controller. The arm length defines the location of the RCC and controls the 'aggressiveness' of the controller. The shorter the arm length, the quicker the controller reacts and the larger the maximum pitching angle is. At longer arm lenghts, the controller reacts slower and the maximum pitching angle is reduced. Especially when the quadrotor is to interact with an environment, the ability to tune the 'aggressiveness' of the controller with this single parameter is powerful. However, as treated before, for each arm length the stiffness and damping should be re-tuned to get a perfect response without oscillation. This is a disadvantage when one wants to control the position of the quadrotor. However, this might be less important when the primary objective of the controller is

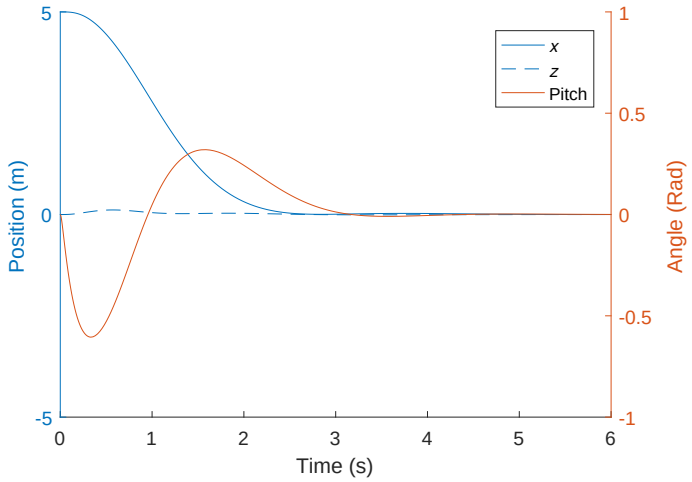


Figure 4.13: Behavior of the controller with well-tuned parameters showing quick convergence. The spring pitches the quadrotor towards the origin and the damper pitches the quadrotor back when the quadrotor is close to the origin to prevent overshoot.

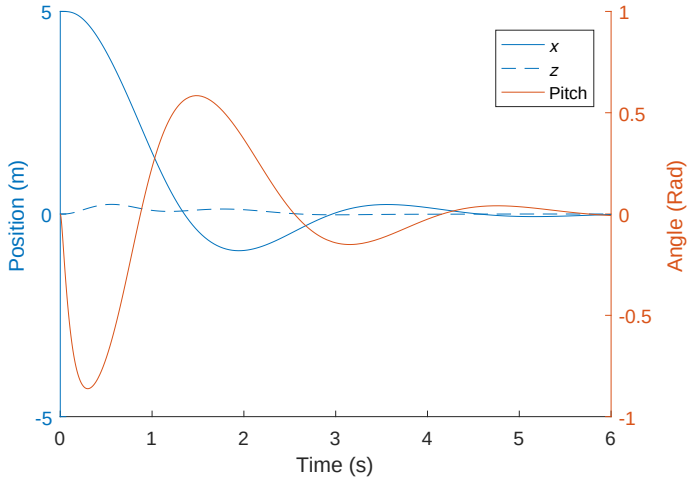


Figure 4.14: Behavior of the controller with a stiff spring. The quadrotor pitches torwards the origin aggressively resulting in an overshoot.

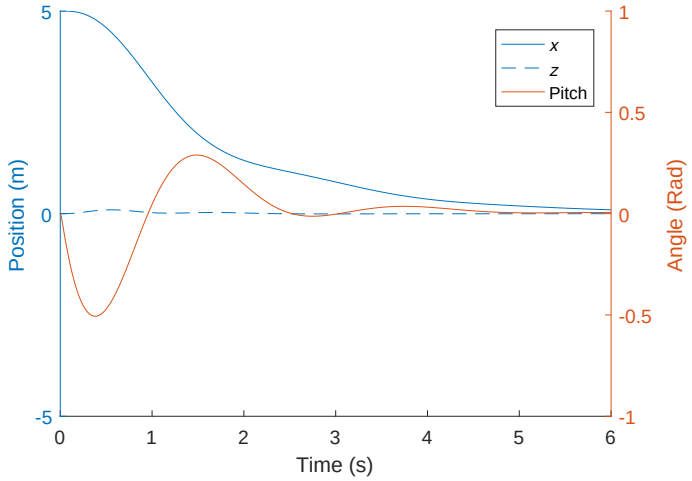


Figure 4.15: Behavior of the controller with to high damping. The quadrotor pitches towards the origin and is quickly damped by the damper, resulting in a damped oscillation of the pitch.

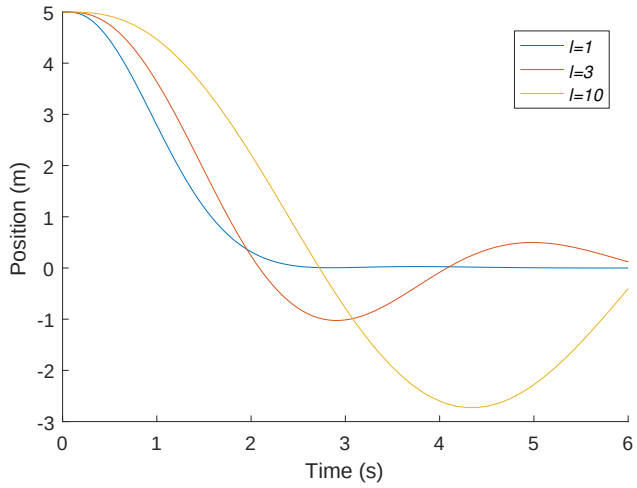


Figure 4.16: x-position of the quadrotor for three different arm lengths l. The initial position of the quadrotor is $x = 5\,\text{m}$. For $l = 1\,\text{m}$, the x-position converges to zero without overshoot. For larger arm lengths l, the converging gets slower and the overshoot increases showing a pendulum-like motion.

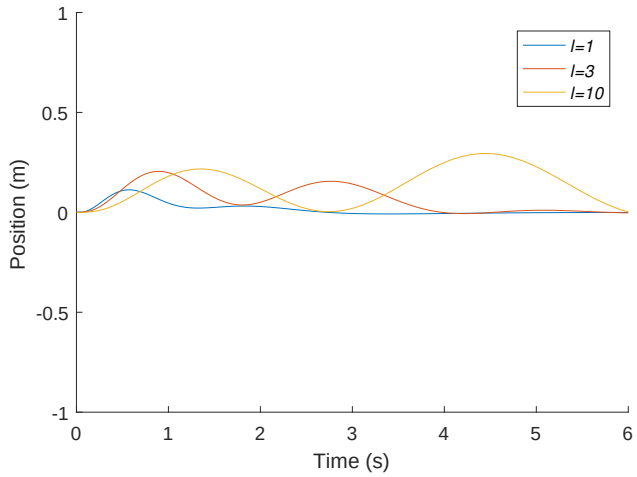


Figure 4.17: z-position of the quadrotor for three different arm lengths l.

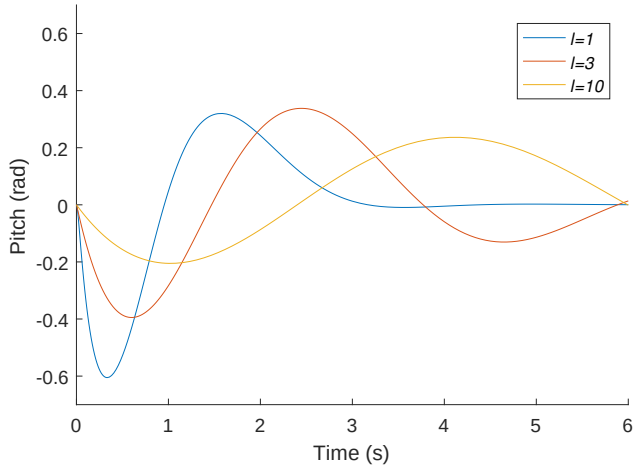


Figure 4.18: Pitching angle of the quadrotor for three different arm lengths l. For small arm lenghts, the controller pitches more aggressive. For longer arm lenghts, the maximum pitch angle gets smaller and oscillations in the pitch due to the pendulum-like motion is introduced.

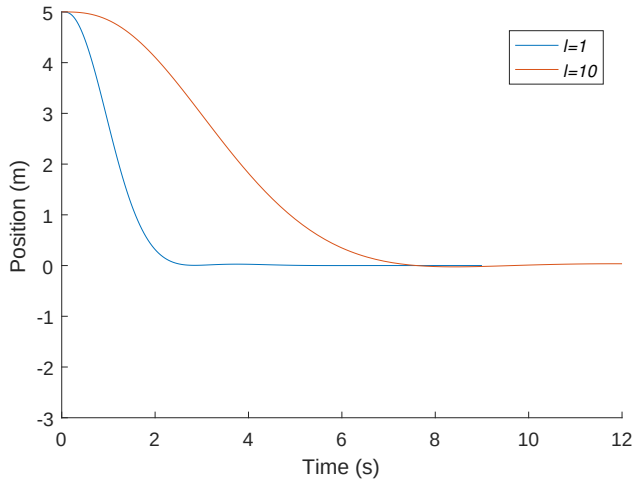


Figure 4.19: x-position of the quadrotor for $l = 1$ m and $l = 10$ m. The spring and damper parameters are re-tuned for each arm length. Both show convergence without overshoot. The controller with the shorter arm length $l = 1$ m converges more quickly.

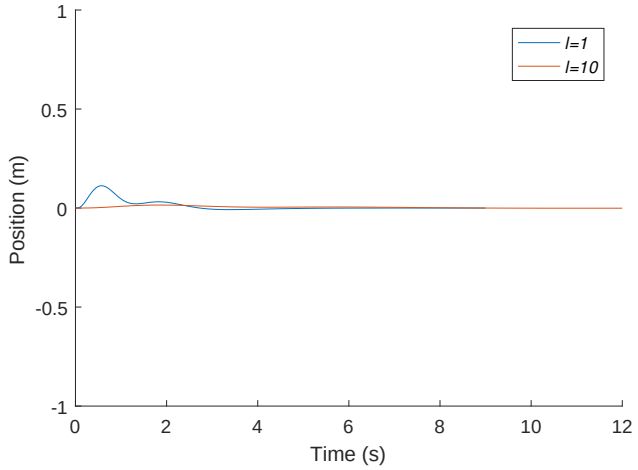


Figure 4.20: z-position of the quadrotor for two different arm lengths l.

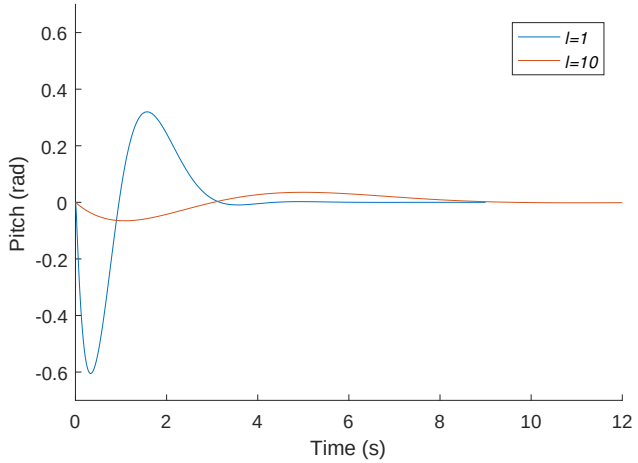


Figure 4.21: Pitching angle of the quadrotor for $l = 1$ m and $l = 10$ m. The spring and damper parameters are re-tuned for each arm length. The controller with the shorter arm length $l = 1$ m has a much larger maximum pitch angle, so acts quicker and more aggressive.

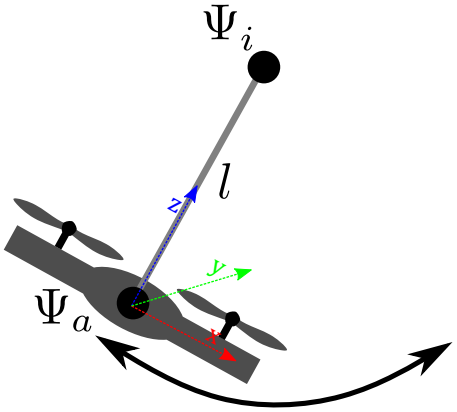


Figure 4.22: The quadrotor is prone to oscillations in the non-actuated plane. As there is no damping in this plane, the quadrotor behaves like a pendulum with frame Ψ_i as (moving) pivot point.

to interact. In this case, the ability to tune the 'aggressiveness' of the controller with a single parameter may outweigh the introduction of the oscillation.

5 Implementation

To be able to perform Software-in-the-loop (SITL) simulations and real-world experiments on the controller, the controller has been implemented on a quadrotor. In this thesis, the supervisor controller that controls the energy budget is not considered. Therefore, non-passive projection has been implemented, which is equivalent to passive projection with an infinite energy tank budget. Furthermore, a non-passive effort source is used for gravity compensation (which could be passivized by using a spring, passive projection and an energy tank). The controller has been implemented using a PC running Robot Operating System (ROS) and a Pixhawk flight controller running PX4 firmware. An overview is depicted in Figure 5.1. The implementation allows running the rotational damping onboard on the Pixhawk to increase the stability of the system.

5.1 Wrench control on a quadrotor

The output of the controller is a wrench that should be realized by the actuators. The actuators (rotors) are speed-controlled, so it is not possible to send 'wrench'-commands to the actuators directly. The controller wrench can be approximated by calculating the rotor speeds as a function of the wrench by inverting the fully-actuated part of Equation 3.1. There are no sensors on the rotors to measure the actual exerted wrench so the system is lacking feedback. The applied wrench is an approximation of the control wrench. This is a fundamental limitation on quadrotor wrench control with current hardware.

5.2 Wrench control by attitude commands

The ROS controller node calculates the control wrench and sends a control command to the Pixhawk accordingly. The Pixhawk is connected to the ROS node via Mavlink. To approximate a wrench, rotor speeds can be sent to the Pixhawk by sending an actuator control message directly. However, to be able to run a part of the control loop onboard on the Pixhawk, the onboard attitude controller has been used. Rotational damping can be injected by the onboard attitude controller, which runs on a much higher rate than the ROS controller node, improving the stability of the controller (see Subsection 5.2.1).

The control wrench is normalized and sent to the Pixhawk in an attitude message as body rates and thrust. By scaling the control wrench and by calculating the necessary gain of the onboard attitude controller, an approximated wrench is sent using the attitude controller. The signal path of the controller wrench is shown in Figure 5.2. The high-level controller running in ROS is shown on the left and the onboard attitude controller on the right. The onboard controller

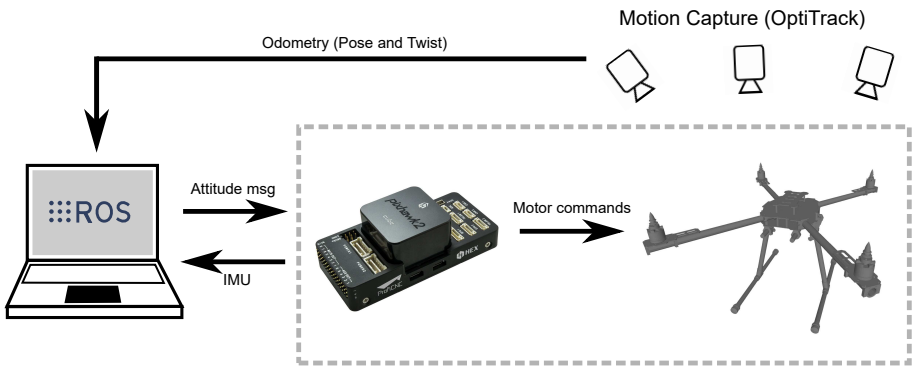


Figure 5.1: Overview of the implementation. The controller calculating the control wrench is implemented in ROS and communicates via Mavlink with the onboard Pixhawk flight controller.

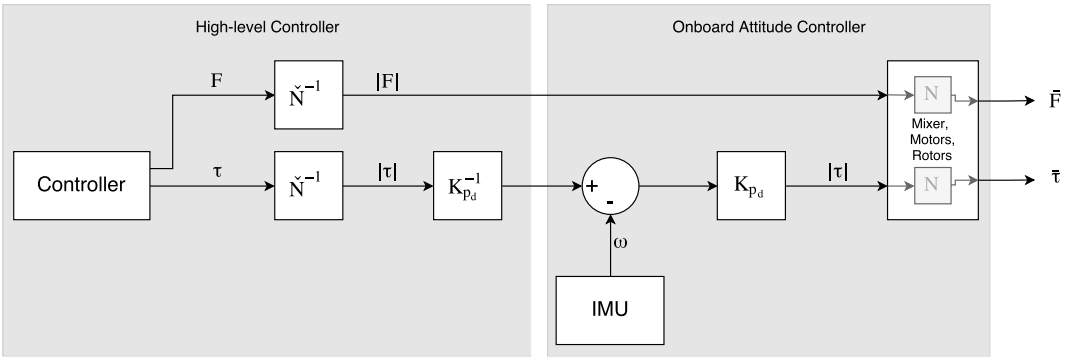


Figure 5.2: Signal path from the controller wrench through the onboard attitude controller. By using the onboard attitude controller, wrench control can be realized with additional angular damping running onboard improving stability.

consists of a P controller on the body rates ω and a direct feed-through on the thrust. The high-level controller outputs force F and a torque τ. Following the signal path, the actual exerted force $\bar{F}$ and torque $\bar{\tau}$ can be calculated:

$$\bar{F} = F\check{N}^{-1}N \tag{5.1a}$$

$$\bar{\tau} = (\tau\check{N}^{-1}K_{p_d}^{-1} - \omega)K_{p_d}N \tag{5.1b}$$

where K_{p_d} is the onboard attitude controller proportional gain, N the mixer de-normalization factor and $\check{N}$ the estimated value for N. The input of the mixer are the normalized body torques and thrust, and these are converted to rotor speeds. If a linear relation between the mixer input and squared of the rotor speeds is assumed[1], then N is linear and depends on the minimum and maximum torques and thrust. If all actuators are used below their limits, then the maximum torques and thrust are constant. If an actuator reaches a limit, the torques and thrust will be scaled by the Pixhawk mixer algorithm. In this case, the actuator wrench and controller wrench are not equal and passivity may be lost.

If all actuators are kept beyond the limits and a linear relation between the squared of the rotor speeds and the wrench is assumed, N can be estimated by $\check{N}$ where $\check{N}$ depends on the absolute maximum torques and thrust. Assuming that N is well-estimated by $\check{N}$ such that $\check{N}$=N holds, Equation 5.1 reduces to:

$$\bar{F} = F \tag{5.2a}$$

$$\bar{\tau} = \tau - (K_{p_d}\check{N})\omega \tag{5.2b}$$

The force produced by the propellers $\bar{F}$ is equal to the controller force F. The produced torque $\bar{\tau}$ however, depends on the controller torque τ and an additional term $-(K_{pd}\check{N})\omega$. This additional term can be seen as rotational damping injected by the onboard controller, with a damping coefficient equal to $K_{p_d}\check{N}$. The onboard rotational damping can be tuned by the parameter K_{p_d}.

[1] During experiments, this assumption has been found to be incorrect. The Pixhawk mixer input has a linear relation with the (non-squared) rotor speed output (see Subsection 6.1.4).

5.2.1 Onboard rotational damping

The onboard attitude control loop runs on a much higher rate than the control loop of the ROS node, so the stability is increased when the rotational damping runs on the onboard control loop. The body-fixed wrench exerted by the controller damper depends on the body twist $T_a^{a,0}$, the Adjoint of the transformation matrix from frame Ψ_a to Ψ_i $Ad_{H_a^i}{}^T$ and the damping matrix K_d:

$$W^a = -Ad_{H_a^i}^T K_d Ad_{H_a^i} T_a^{a,0} \tag{5.3}$$

Writing out he matrices gives:

$$W^a = -\begin{pmatrix} I_3 & -\tilde{p}_a^i \\ 0 & I_3 \end{pmatrix} \begin{pmatrix} K_{d_\omega} & 0 \\ 0 & K_{d_v} \end{pmatrix} \begin{pmatrix} I_3 & 0 \\ \tilde{p}_a^i & I_3 \end{pmatrix} \begin{pmatrix} \omega_a^{a,0} \\ v_a^{a,0} \end{pmatrix} \tag{5.4}$$

Performing the matrix multiplications gives the damping matrix:

$$\begin{pmatrix} \tau^a \\ F^a \end{pmatrix} = -\begin{pmatrix} K_{d_\omega} - \tilde{p}_a^i K_{d_v} \tilde{p}_a^i & -\tilde{p}_a^i K_{d_v} \\ K_{d_v} \tilde{p}_a^i & K_{d_v} \end{pmatrix} \begin{pmatrix} \omega_a^{a,0} \\ v_a^{a,0} \end{pmatrix} \tag{5.5}$$

The top-left part of equation 5.5 is the rotational damping:

$$\tau^a = -(K_{d_\omega} - \tilde{p}_a^i K_{d_v} \tilde{p}_a^i)\omega_a^{a,0} \tag{5.6}$$

Where K_{d_ω} and K_{d_v} are diagonal matrices and $p_a^i = [0\ 0\ -l]^T$. This is simplifies to the following set of equations describing the rotational damping:

$$\tau_x = -(K_{d_{w_x}} + K_{d_{v_y}} l^2)\omega_x \tag{5.7a}$$

$$\tau_y = -(K_{d_{w_y}} + K_{d_{v_x}} l^2)\omega_y \tag{5.7b}$$

$$\tau_z = -K_{d_{w_z}} \omega_z \tag{5.7c}$$

The rotational damping equations shown in Equation 5.7 are uncoupled and therefore can be implemented onboard by tuning K_{p_d} of the onboard attitude controller. By combining Equation 5.7 and Equation 5.2b, the onboard gain K_{p_d} can be calculated:

$$K_{p_{d_x}} = \frac{K_{d_{w_x}} + K_{d_{v_y}} l^2}{\check{N}_{\omega_x}} \tag{5.8a}$$

$$K_{p_{d_y}} = \frac{K_{d_{w_y}} + K_{d_{v_x}} l^2}{\check{N}_{\omega_y}} \tag{5.8b}$$

$$K_{p_{d_y}} = \frac{K_{d_{w_z}}}{\check{N}_{\omega_z}} \tag{5.8c}$$

Name	Parameter	Value	Unit
Rotor thrust factor	b	$1.49 \cdot 10^{-5}$	$\mathrm{N/(rad/s)^2}$
Rotor drag factor	d	$8.94 \cdot 10^{-7}$	$\mathrm{Nm/(rad/s)^2}$
Maximum rotor speed	$\Omega_{\max}$	800	$\mathrm{rad\,s^{-1}}$

Table 5.1: Quadrotor rotor parameters

Name	Parameter	Value	Unit
Thrust (z-direction)	$F_{z_{\max}}$	38	N
Roll-moment	$\tau_{x_{\max}}$	4.7	Nm
Pitch-moment	$\tau_{y_{\max}}$	4.7	Nm
Yaw-moment	$\tau_{z_{\max}}$	2.3	Nm

Table 5.2: Absolute maximum thrust and roll, pitch, yaw moments

5.2.2 Offboard rotational damping

Running the rotational damper onboard increases the stability because the onboard control loop runs on a much faster rate than the offboard ROS controller. However, the attitude proportional gain factor K_{p_d} depends on the controller parameters l and K_d. When one changes l or K_d, then K_{p_d} should be recalculated and updated in the Pixhawk firmware. For easy controller parameter reconfiguration, K_{p_d} should be independent of the controller parameters. This can be realized by running the complete control loop (including the rotational damping) on the offboard ROS controller. When K_{p_d} is set to a very small value, Equation 5.2b approaches $\bar{\tau} = \tau$, such that there is no additional damping injected by the attitude controller. The control loop can run fully offboard and the controller parameters can be reconfigured without the need to change K_{p_d} in the Pixhawk firmware. This enables dynamic reconfiguring of the parameters and decreases the time-to-test period. However, rotational stability of the controller will reduce because the full control wrench is calculated by the slower, offboard control loop. Therefore, offboard rotational damping has only be used for quick tests in SITL. During the real experiments, onboard rotational damping has been used.

5.2.3 Normalization

To calculate $\check{N}$ (the estimate of N), the absolute maximum torques and thrust should be calculated. Using Equation 3.1 and the maximum rotor speed of the quadrotor $\Omega_{\max}$ presented in Table 5.1, the theoretical maximum thrust and roll, pitch, yaw moments can be calculated. The values are calculated in Equation 3.2. It is assumed that the rotors cannot reverse the spinning direction, so $\Omega_{\min} = 0$. The results are presented in Table 5.2. The normalization factors $\check{N}$ are then given by $\check{N}_F = F_{z_{\max}}$, $\check{N}_{\omega_x} = \tau_{x_{\max}}$, $\check{N}_{\omega_y} = \tau_{y_{\max}}$, $\check{N}_{\omega_z} = \tau_{z_{\max}}$.

6 Experiments

Experiments have been conducted to study the controller performance on a real quadrotor. The controller has been implemented in ROS and a Pixhawk as shown in Chapter 5. First, the controller has been simulated using Software-in-the-loop (SITL) which will be treated in Section 6.1. Next, experiments have been performed on a real quadrotor (Section 6.2).

6.1 Software-in-the-loop validation

Before running the controller on a real quadrotor, simulations have been performed using Software-in-the-loop (SITL). The Pixhawk flight controller used on the quadrotor runs PX4 firmware. PX4 provides a SITL simulation framework in which the Pixhawk is simulated such that the full controller can be tested in a simulation environment before running it on a real quadrotor.

In Subsection 6.1.1, an overview of the simulation setup is given. Next in Subsection 6.1.2, the stability of the controller is examined for both full-offboard control mode and the onboard rotational damping mode. In Subsection 6.1.3, the behavior of the controller is studied and compared to the 20-sim simulation.

6.1.1 Setup

An overview of the used setup is depicted in Figure 6.1. The 'Controller node' is the central ROS node calculating the control wrench. The controller parameters can be set by using the 'Dynamic Reconfigure GUI' and the controller setpoint H_b^0 can be set by the 'Setpoint GUI'. As explained in Chapter 5, the Controller node converts the wrench to body rate and thrust setpoints that are sent to the Pixhawk in an attitude message. The 'MAVROS' node provides a ROS interface to communicate with the Pixhawk through MAVLINK. The SITL node simulating the Pixhawk flight controller is provided by PX4 and shown in Figure 6.1 as 'PX4 SITL'. The 'PX4 SITL' sends actuator commands to a quadrotor model running in the Gazebo simulator. Plugins have been added to the Gazebo model such that the quadrotor odometry (pose and twist) and Inertial Measurement Unit (IMU) data is published to a ROS topic. The Controller node is subscribed to these topics such that it can calculate the control wrench accordingly. During simulation, all topics are recorded in a rosbag such that the data can be analyzed.

6.1.2 Controller stability

Simulations have been performed to examine the stability of the controller. The same controller parameters have been chosen that gave good results for $l = 1\,\text{m}$ in the behavior sim-

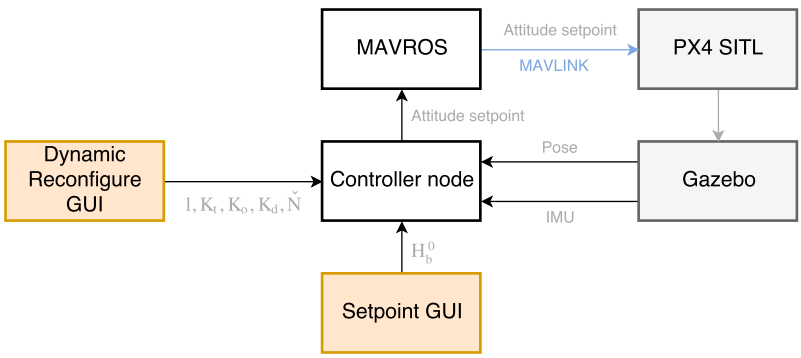


Figure 6.1: Overview of the Software-in-the-loop (SITL) implementation. The simulation is controlled via the 'Setpoint' and 'Dynamic Reconfigure' GUI.

ROS::Transporthints	Offboard	Onboard
Default (empty)	0.1305	0.0092
TCP_NODELAY	0.0227	0.0097

Table 6.1: RMS of the roll angle (in rad) calculated from the data presented in Figure 6.2 and 6.3. The TCP_NODELAY Transporthint affects the stability of the controller the most if the whole control loop runs offboard (no onboard rotational damping).

ulations (Figure 4.13) in Section 4.2. The damping is set to $B_{v_x}, B_{v_y}, B_{v_z} = 3.3\,\mathrm{N/(m/s)}$, $B_{\omega_z} = 1\,\mathrm{Nm/(rad/s)}$ and the spring constants to $K_{t_x}, K_{t_y}, K_{t_z} = 2.9\,\mathrm{N\,m^{-1}}$, $K_{o_z} = 1\,\mathrm{N\,m\,rad^{-1}}$. The normalization factors $\check{N}$ are shown in Table 5.2. The quadrotor setpoint and initial position are both set to $[0\ 0\ 0]^T$ such that the quadrotor hovers. The control rate is set to 200 Hz. The simulation is performed for both offboard control and onboard rotational damping and the roll angle is recorded. The simulation results are shown in Figure 6.2. When the whole controller runs offboard, the controller roll angle shows oscillatory behavior. When the rotational damping runs onboard on the Pixhawk, the controller is more stable as the oscillatory behavior is suppressed.

The strong oscillatory behavior of the offboard controller shown in Figure 6.2 was not expected at the control rate of 200 Hz. Further research has been performed, and it was found that the control wrench was indeed sent at 200 Hz but the wrench value stayed constant for ~ 10 messages. Therefore, the wrench value was updated at only ~ 20 Hz. This can be explained by an algorithm found in the transport layer of the ROS messages. ROS messages are sent over TCP, which uses an algorithm (Nagle's algorithm) to reduce the number of packets on the network. The ROS node publishing the quadrotor odometry was set to 200 Hz, so 200 TCP packets should be sent per second. Nagle's algorithm reduces the number of packets by buffering and aggregating multiple TCP packets in a single packet. After each TCP packet, the sender side waits untill it receives an acknowledgment before sending the next packet. If the receiver waits for a period before sending the acknowledgment (delayed ack), the sender starts buffering and aggregating TCP packets. During the simulation, about 10 odometry messages where aggregated in a single TCP message which was sent at a rate of ~ 20 Hz. As only the last odometry message is useful, the apparent rate was reduced to ~ 20 Hz. Nagle's algorithm can be disabled by forcing the receiver to send an acknowledgment immediately. In ROS, Nagle's algorithm is enabled by default but it can be disabled by setting the ros::TransportHints().tcpNoDelay() argument when the subscriber is initialized. More information on Nagle's algorithm can be found in Cheshire (2005).

Nagle's algorithm has been disabled for the odometry subscriber in the controller node and the Attitude subscriber in the MAVROS node to enable faster control rates. The stability simulation described at the begin of this subsection has been repeated. The results are presented in Figure 6.3. While the controller running the rotational damping onboard is still more stable, the stability of the controller with full offboard control has improved significantly. To compare the stability numerically, the RMS value of the roll has been calculated for both on-and offboard and Nagle's algorithm enabled and disabled (Table 6.1). From the RMS values, it can be concluded that the stability of the offboard controller is affected the most by Nagle's algorithm. When Nagle's algorithm is disabled, the RMS value reduces with a factor of ~ 6. When running the rotational damping onboard, the stability is is less affected by Nagle's algorithm as the rotational damping runs onboard on a high rate, independent of the offboard ROS control loop.

6.1.3 Controller behavior

Simulations have been performed to examine the behavior of the controller. Onboard rotational damping has been used and Nagle's algorithm has been disabled. The same control-

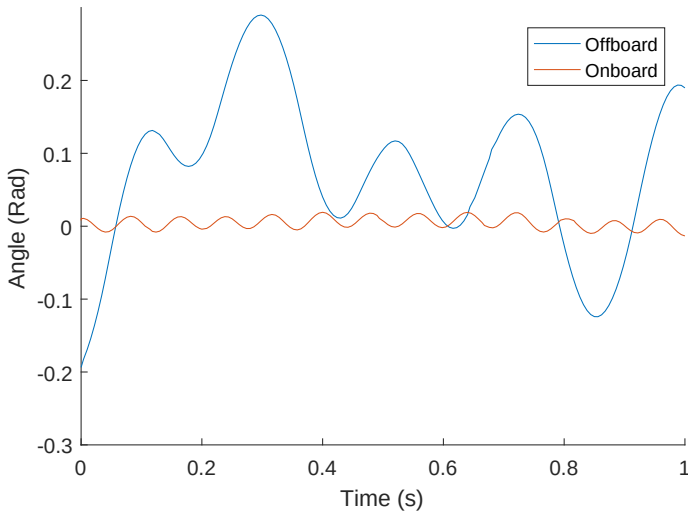


Figure 6.2: Roll angle for hovering quad running the rotational damping offboard and onboard. Default ROS subscriber initialization. The quadrotor roll is more stable if the rotational damping runs onboard on the Pixhawk.

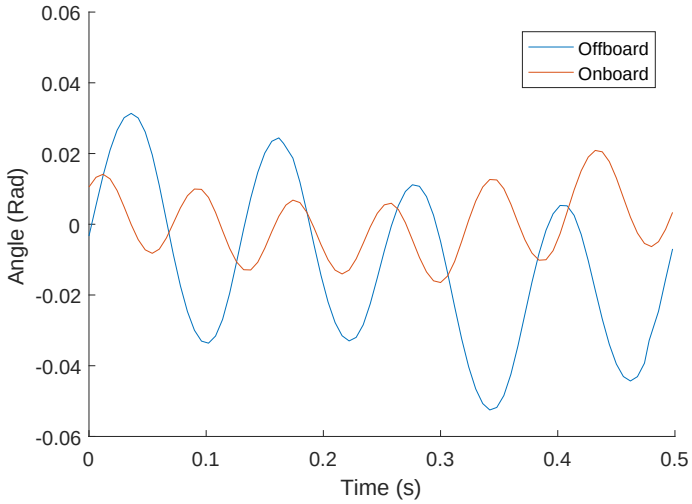


Figure 6.3: Roll angle for hovering quad running the rotational damping offboard and onboard. Subscribers initialized with ROS::Transporthints argument TCP_NODELAY. The offboard stability has improved significantly but the quadrotor roll is still more stable if the rotational damping runs onboard on the Pixhawk.

Simulation	$l = 1\,\text{m}$	$l = 10\,\text{m}$
20-Sim	2.47	7.26
SITL	3.86	did not settle
SITL (re-tuned $\check{N}_{F_z}$)	2.44	did not settle

Table 6.2: Settling time (in s) for both the 20-sim and SITL simulations. After re-tuning N_{F_z}, the settling time in SITL is very close the the settling time in 20-sim. The settling time is defined as the time it takes for the x-position of the quadrotor to stay within 1% bounds of the setpoint.

ler parameters have been chosen which gave good results in the behavior simulations in Section 4.2 (Figure 4.13). The parameters are an arm length of $l = 1\,\text{m}$, damping of $B_{v_x}, B_{v_y}, B_{v_z} = 3.3\,\text{N/(m/s)}$, $B_{\omega_z} = 1\,\text{Nm/(rad/s)}$ and the spring constants of $K_{t_x}, K_{t_y}, K_{t_z} = 2.9\,\text{N m}^{-1}$, $K_{o_z} = 1\,\text{N m rad}^{-1}$. The normalization factors $\check{N}$ are shown in Table 5.2. To be able to compare the behavior in SITL with the simulated behavior in 20-sim, the behavior simulation has been repeated (setpoint: $x, y, z = 0$, initial position: $x = 5\,\text{m}$, $y, z = 0$. The results of the SITL simulation are presented in Figure 6.4. The SITL simulation (Figure 6.4) shows slower convergence than the 20-sim simulation (Figure 4.13) and the z-position has a steady-state error. From the steady-state error it can be concluded that the control wrench is unequal to the actuator wrench ($\check{N} \neq N$) and this inequality has to be compensated for by the spring. $\check{N}_{F_z}$ is re-tuned such that the steady-state error reduces to zero and simulation is repeated. Note that $\check{N}_{\tau_{x,y,z}}$ are kept constant. The results are presented in Figure 6.5 and show smooth convergence to the setpoint. The behavior is compared to the 20-sim behavior in Figure 6.6. The 1% settling time has been calculated for Figure 6.5 and is equal to $\sim 2.4\,\text{s}$. For comparison, the settling times of both the 20-sim, SITL and SITL with re-tuned $\check{N}_{F_z}$ are presented in Table 6.2. The SITL with re-tuned $\check{N}_{F_z}$ and 20-sim simulations show similar behavior. Small differences in the pitch angle and z-position might be caused by an error in the estimation of N by $\check{N}$. Note that $\check{N}_{F_z}$ has been re-estimated by tuning the steady-state z-position error to zero, but the torque factors $\check{N}_\tau$ have not been re-tuned. As $\check{N}_{F_z}$ and $\check{N}_\tau$ are coupled, it is likely that $\check{N}_\tau$ should be re-tuned too. Furthermore, N might show non-linear behavior.

6.1.4 Evaluation

Disabling Nagle's algorithm shows more stable behavior for the offboard controller, but both the on- and offboard controller still show oscillatory behavior. This oscillatory behavior is found to be caused by the rotor inertial dynamics. The inertial dynamics behave like a low-pass filter on the rotor angular velocity. In the SITL, the rotor dynamics are also being simulated. The rotors are configured to have time constants of $\tau_{\text{up}} = 0.0125\,\text{s}$ and $\tau_{\text{down}} = 0.025\,\text{s}$. The rotor dynamics introduce low-pass behavior in the control loop which will result in oscillations or even instabilities when the control loop gains are to high. The oscillation can be mitigated by lowering the loop gains. In Figure 6.7, the control gains have been reduced to $l = 0.5\,\text{m}$, $B_{v_x}, B_{v_y}, B_{v_z} = 2.5\,\text{N/(m/s)}$, $B_{\omega_z} = 1\,\text{Nm/(rad/s)}$ and $K_{t_x}, K_{t_y}, K_{t_z} = 1.5\,\text{N m}^{-1}$. No oscillation is visible anymore. The RMS values reduce to close to zero: $1.43 \cdot 10^{-5}$ s for the offboard and $1.63 \cdot 10^{-4}$ s for the onboard controller.

In Chapter 5, it has been assumed that the relation between the mixer normalized input and mixer wrench output is linear (so there is a linear relation between the mixer input and the squared of the rotor speed). However, during experiments is has been found out that this not the case. In the PX4 implementation, the mixer relation between the input and non-squared rotor speed is linear. For the thrust, the relation is shown in Figure 6.8. The 'Pixhawk mapping' is the real mapping during SITL. The bias of $200\,\text{rad s}^{-1}$ was defined in the motor model of the SITL model. Thanks to this bias, the Pixhawk mapping and expected mapping do approximate each other pretty close. However, there is a non-linear relation between the mixer input and

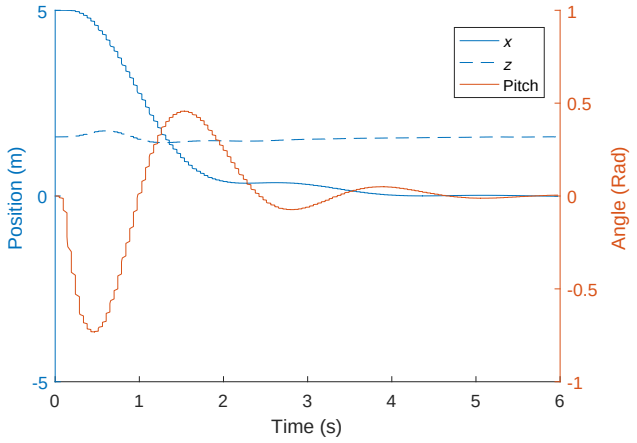


Figure 6.4: Behavior SITL simulation with arm length $l = 1$ m using the same parameters as in earlier simulation (Figure 4.13). In SITL, the controller converges slower to the setpoint. Settling time: 3.86 s.

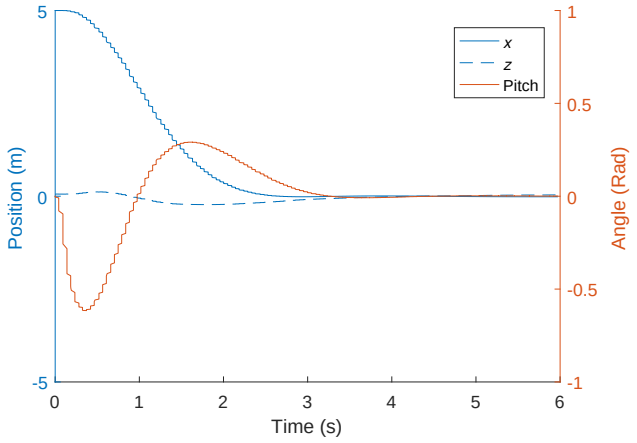


Figure 6.5: Behavior SITL simulation with arm length $l = 1$ m using the same parameters as in earlier simulation (Figure 4.13) but with $\check{N}_{v_z}$ re-tuned. The controller converges smoothly to the setpoint. Settling time: 2.44 s.

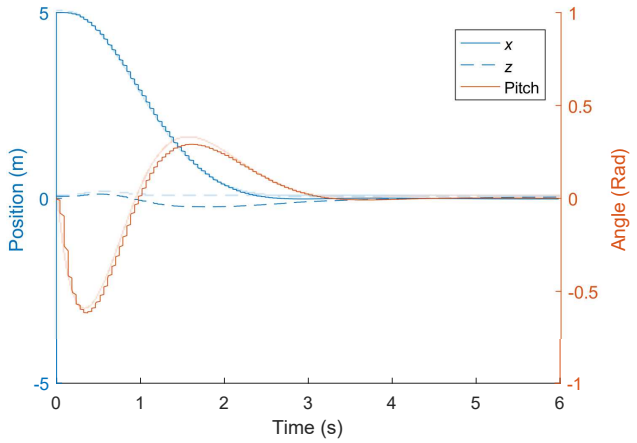


Figure 6.6: 20-sim and SITL behavior compared for $l = 1$ m (SITL Figure 6.5 on top and the 20-sim Figure 4.13 transparently overlayed). The SITL and 20-sim show similar behavior.

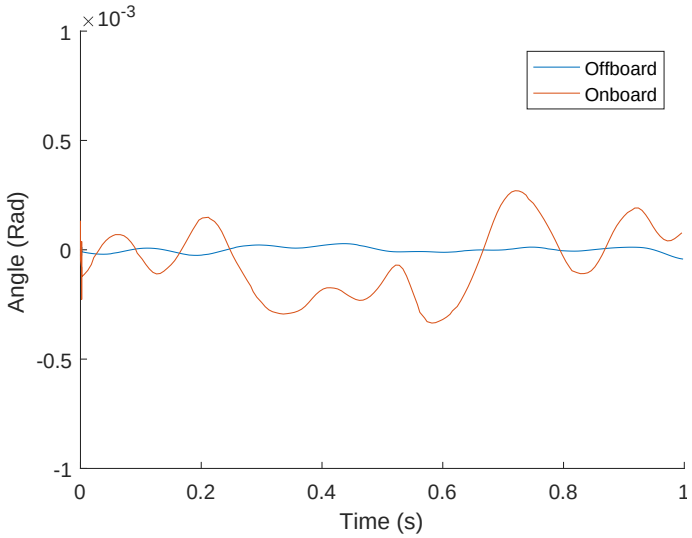


Figure 6.7: Roll angle for hovering quad running the rotational damping offboard and onboard. Subscribers initialized with ROS::Transporthints argument TCP_NODELAY. Controller gains have been lowered to remove oscillatory behavior as a result of the rotor dynamics.

the wrench output. As a linear relation is assumed during implementation, the control wrench is not equal to the actuator wrench.

In Chapter 3.1 the quadrotor model has been presented in '+' configuration. The normalization factors have been calculated for this configuration. After the experiments, it has been found out that the experiment and the SITL simulation use the 'x' configuration. The configuration has an effect on the maximum roll and pitch torque, which are $\sqrt{2}$ times higher for the 'x' configuration. This error affects the effective values of the pitch/yaw controller parameters running offboard, but will not affect the stability directly.

6.1.5 Conclusion

Running the rotational damping on the onboard controller improves stability as it runs on a higher rate and has a lower delay than the ROS control loop. To improve stability and performance, Nagle's algorithm in the TCP transport layer should be disabled for each ROS subscriber to guarantee the actual control rate is as configured.

While the behavior of the controller in the SITL simulation compared to 20-sim showed in Figure 6.6 is very similar, it has been found that wrong assumptions have been made during implementation in Chapter 5. Most importantly, it has been found that the relation between the mixer input and wrench output is non-linear. A linear relation has been assumed in Chapter 5 in implementing wrench control using the attitude controller. As a consequence, the controller wrench is not equal to the actuated wrench, but only an approximation. The controller wrench being equal to the actuator wrench is a fundamental requirement for passivity. In general, wrench control by attitude messages is found to be incorrect.

The rotor inertial dynamics add low-pass behavior to the control loop which results in oscillations around the roll and pitch if the control gains are chosen too high. In the SITL simulations, the oscillations are stable. However, the oscillations reduce the robustness of the controller which may lead to unstable behavior when the controller is implemented on a real quad.

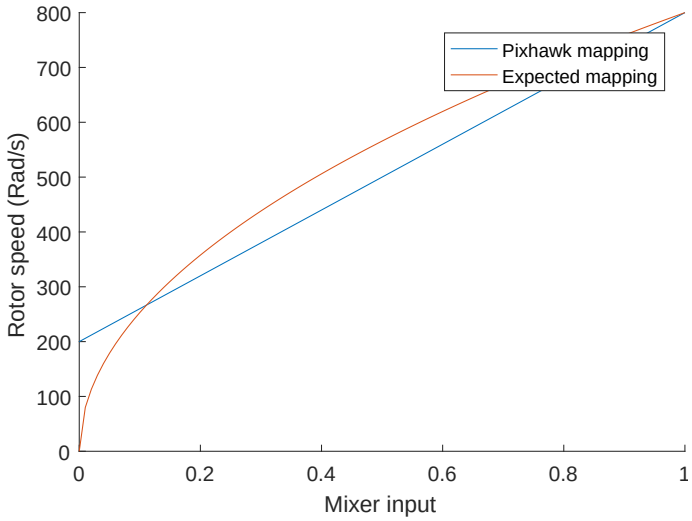


Figure 6.8: Expecited and actual SITL Pixhawk mixer input to rotor speed mapping for the thrust. A linear relation was expected by the author between the mixer input and ω^2, but the Pixhawk mapping relation is linear in ω and therefore non-linear in ω^2.

Figure 6.9: Quadrotor used during the experiments. The cable visible in the figure is the MAVLINK USB-interface and is connected between the Pixhawk and the laptop running ROS. During experiments, an additional cable was connected to the quadrotor to provide power, such that no onboard battery was needed.

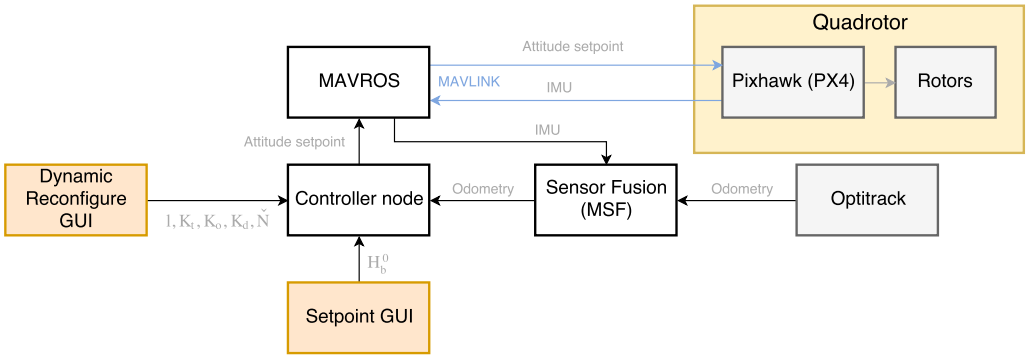


Figure 6.10: Overview of the experiment setup. The quadrotor setpoint is controlled via the 'Setpoint' and the controller parameters via the 'Dynamic Reconfigure' GUI. A sensor fusion algorithm ROS node fuses the IMU data from the Pixhawk and the Odometry data from OptiTrack.

6.2 Experimental validation

Experiments have been conducted to test the controller stability and performance on a real quadrotor running a Pixhawk flight controller. The size of the flying lab where the experiments have been conducted was limited (Figure 6.11). Only the controller stability and basic functionality have been tested, as only small control steps could be given in the small lab. In Subsection 6.2.1, an overview of the experiment setup is given. Next in Subsection 6.2.2, the results of the experiment are presented.

6.2.1 Setup

During the experiments, the stability of the controller have been tested. The experiment setup is similar to the SITL setup and shown in Figure 6.10. The SITL Pixhawk has been exchanged with a real Pixhawk (version 2, running PX4 firmware) and the quadrotor odometry (pose and twist) are estimated by a ROS node on the computer running a sensor fusion algorithm. The input are the IMU data provided by the Pixhawk and the odometry data provided by the OptiTrack motion capture system. By combining both, robustness and accuracy of the estimated odometry is improved. The setpoint is controlled manually by the user with the 'Setpoint GUI'.

Figure 6.11: Picture of the experiment setup. A power cable and usb cable are connected to the quadrotor. The computer visible on the mid-left of the figure runs the OptiTrack software and publishes the measured odometry over a wired network connection to an external laptop running ROS. The usb cable is also connected to the laptop running ROS to establish MAVLINK communication with the Pixhawk.

6.2.2 Controller stability

During the first experiment session, parameters have been chosen similar to those that gave good results in the SITL simulations in Section 6.1 (Figure 6.4). However, strong oscillations around the pitch and roll angle were present, making the quadrotor unstable.

Additional SITL simulations have been performed to try to improve the stability of the controller. When high damping rates were used in the SITL simulations, the damping became unstable caused by unmodeled loop dynamics (e.g. the rotor inertial dynamics). During the second experiment session, the parameters have been tuned such that the damping and spring constants were low. However, the controller was still unstable (Figure 6.12). As the spring and damper already had low gains, the unstable behavior was thought to be caused by the gravity compensation. In Section 3.2.3, gravity compensation in frame Ψ_i was proposed which adds a stabilizing torque to the quadrotor:

$$\tau = \sin(\alpha) * F_{\text{comp}} * l \tag{6.1}$$

Linearized around $\alpha = 0$, this stabilizing torque can be seen as a rotation spring with a stiffness of $F_{\text{comp}} * l$. In other words, using gravity compensation in frame Ψ_i, a rotational spring around the pitch and yaw has been added with a non-directly tunable stiffness of $F_{\text{comp}} * l$. During the experiment shown in Figure 6.12 this 'spring' became unstable.

To recover stability for the controller, the gravity compensation has been moved from Ψ_i to the body-fixed frame Ψ_a. Rotational spring constants k_{o_x} and k_{o_y} (which were set to zero during earlier simulations/experiments) have been set non-zero to provide a stabilizing torque instead. The parameters have been set to an arm length of $l = 0.5\,\text{m}$, damping of $b_{v_x}, b_{v_y}, b_{v_z} = 1.5\,\text{N/(m/s)}$, $b_{\omega_z} = 0.2\,\text{Nm/(rad/s)}$ and the spring constants of $k_{t_x}, k_{t_y}, k_{t_z} = 1.0\,\text{N m}^{-1}$, $k_{o_x}, k_{o_y} = 1.6\,\text{N m rad}^{-1}$, $k_{o_z} = 0.8\,\text{N m rad}^{-1}$. The normalization factors $\check{N}$ are shown in Table 5.2. Furthermore, the mass used to calculate the pre-compensation force is tuned to $m = 1.35\,\text{kg}$ to reduce the steady-state error.

During session 3, the quadrotor was stable. The first 30 s, only the setpoint z-position has been adjusted such that the quadrotor started hovering (Figure 6.13, 6.15). Small oscillations around the roll and pitch are present, but the quadrotor remained stable. The large steady-state error in the z-position can be explained by too little pre-compensation and the actuator force (thrust) being less than the controller force. As the spring force in the z-direction k_{t_z} has a value of

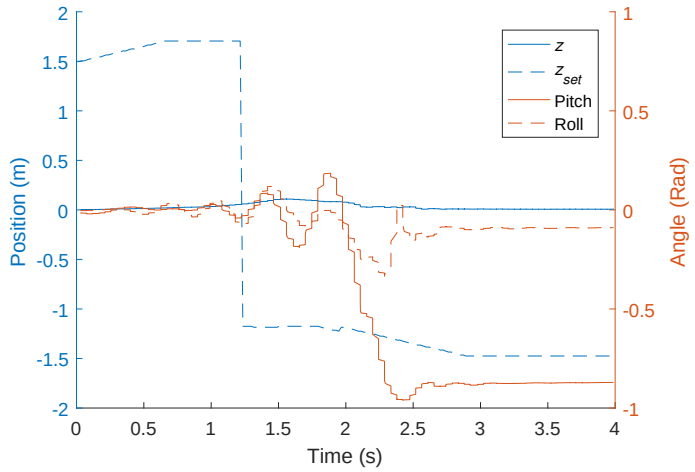


Figure 6.12: Experiment during session 2. The z-coordinate of the setpoint is slowly increased. Shortly after taking off, the quadrotor pitch and roll become unstable. Emergency cut-off triggered at $t \approx 1.8$ s. Spring and damper gains are low, but the gravity compensation in frame Ψ_i becomes unstable. Note that the thrust increases as the mixer prioritizes the torque setpoints over the thrust setpoint.

$1.0\,\mathrm{N\,m^{-1}}$, an error of 1 m results in a controller force of only 1 N. The weaker the spring, the larger the steady-state error. After 30 s, the z-position has been kept constant and the x- and y-position and the yaw angle of the setpoint have been adjusted. The positions and yaw angle are shown in Figure 6.14. The quadrotor follows the setpoint; steady-state errors can be explained by rotor inequalities, imperfect weight balancing on the quadrotor and external disturbances like the power- and USB-cable pulling on the quadrotor. The controller remained stable for the full experiment which lasted 300 s in total.

6.2.3 Conclusion

Experiments have been performed to examine the controller stability and performance on a real quadrotor. The controller loop delay and rotor inertial dynamics introduce a delay and low-pass behavior in the system. This, and the fact that the quadrotor has a relatively small inertia around the pitch and yaw, causes the quadrotor to become unstable easily if the gains are chosen too high. Furthermore, the stabilizing torque imposed by the gravity compensation in frame Ψ_i is found to have an unwanted effect on the stability and performance as it cannot be tuned. The gravity compensation should be done in the body-fixed frame Ψ_a. To prevent the swinging motion of the quadrotor, orientation spring constants k_{o_x} and k_{o_y} should be chosen non-zero. Using this configuration, stable behavior has been achieved. The quadrotor followed setpoint movements in the x-, y- and z-position and yaw angle. The controller behavior has not been compared to the SITL behavior, as only small control steps could be given to the quadrotor as it was flying in a confined space.

It has been shown that stability in the SITL simulation does not guarantee stability in a real-life experiment. In the SITL simulation, an idealized quadrotor model was used. Furthermore, parameter inaccuracies and external disturbances can explain the difference between the SITL and real-life experiments. Although there are differences between the SITL and real-life experiments, the SITL simulations have been proven to be very helpful for finding the cause of non-stable behavior during experiments.

As a consequence of the small inertia and relatively large loop delays and dynamics, low gains need to be used to guarantee stability. However, low values for the spring constants will increase the steady-state error. As wrench control with attitude messages is only an approximation and the wrench is sent to the actuators by feed-forward, there will always be an error between the controller wrench and actuator wrench. This error and external disturbances will

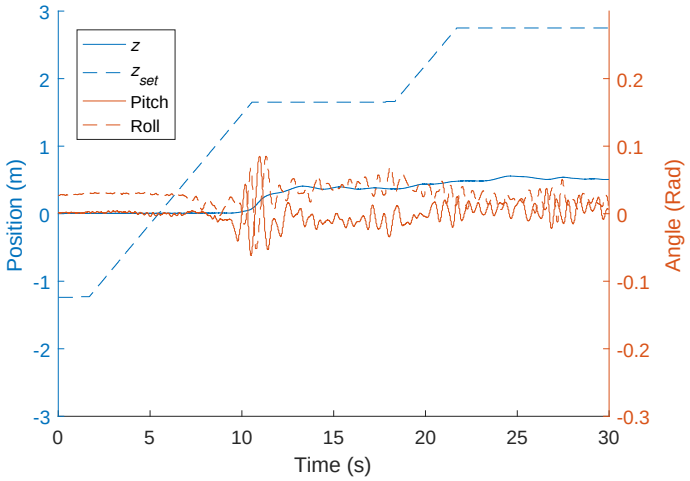


Figure 6.13: Experiment during session 3. The gravity compensation has been moved to frame Ψ_a and k_{o_x} and k_{o_y} are chosen non-zero. The z-coordinate of the setpoint is slowly increased such that the quadrotor starts hovering at $t \approx 10\,\text{s}$. The quadrotor has small oscillations in the roll and pitch but is stable. The steady-state error can be explained by not enough pre-compensation for the mass and/or the controller force not being equal to the actuator force.

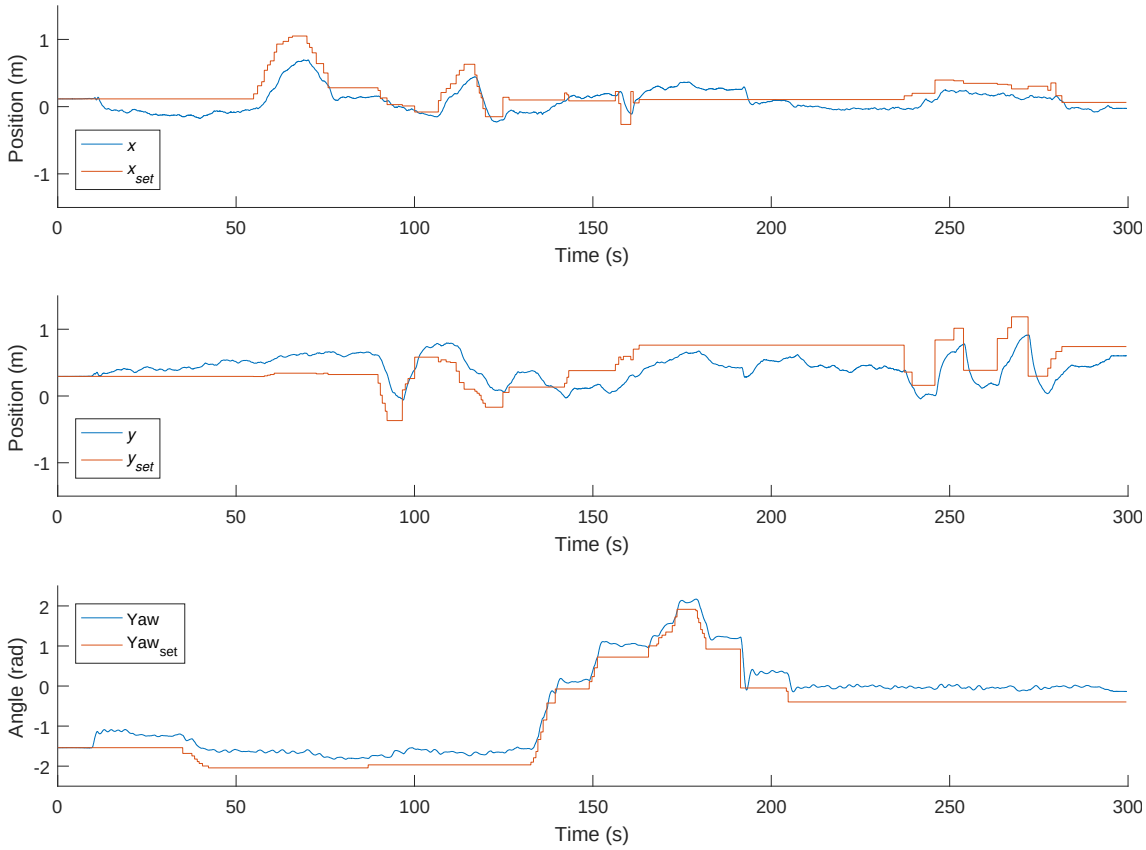


Figure 6.14: Setpoint and quadrotor x- and y-position and yaw angle during experiment session 3. The quadrotor follows the setpoint. Steady-state errors can be explained by rotor inequalities, imperfect weight balancing on the quadrotor itself and external disturbances like the cables pulling on the quadrotor.

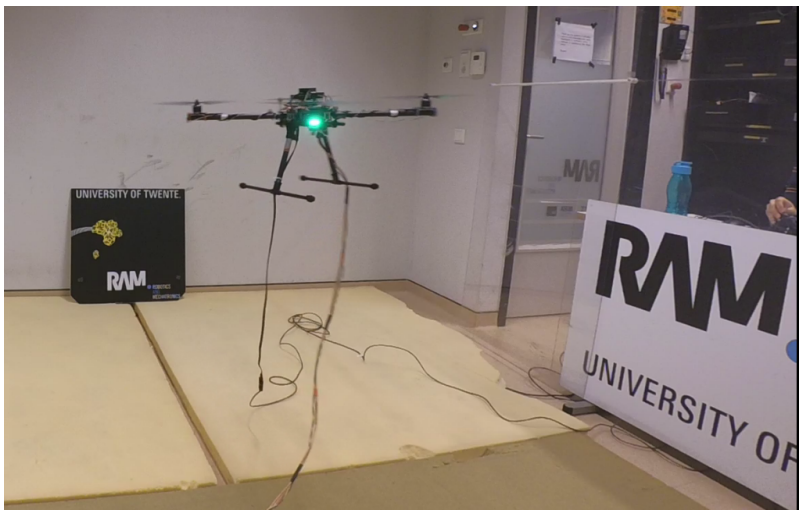


Figure 6.15: Quadrotor during experiment.

result in a steady-state error, as there is only a proportional spring (P) and derivative damper (D) and no integrator present in the controller loop. The steady-state error is dependent on the spring constants. By lowering the spring constants, the stability of the controller is increased, but the steady-state error will also increase. The particular application will dictate which of these two aspects is more important.

7 Conclusion

7.1 Conclusions

The following conclusions can be drawn:

The sliding behavior of the quadrotor in the non-actuated direction as a result of gravity is problematic for Passivity-Based Control (PBC):
It has been shown that for a projection to be passive, both the effort and flow should be affected. For the controller to be passive, a motion in the non-actuated direction should not enter the controller. This is problematic, as the gravity *does* have a component in the non-actuated direction if the quadrotor is tilted. The quadrotor will start 'sliding' in the non-actuated direction, and as this motion will not enter the passive controller, the sliding behavior will never stop. An energy tank that is filled by the supervisor has been proposed to equalize the motion entering the controller and the real quadrotor motion. What should be done when the energy tank is fully depleted is still an open question: allowing the energy tank to fully deplete is not a realistic option, as it would result in the sliding behavior and therefore an uncontrollable quadrotor. A possible solution is to switch to a completely different control law until the tank is refilled by the supervisor.

Unmodeled dynamics can easily destroy the passivity of a controller:
The rotor inertial dynamics have been ignored in the developed controller. The rotor inertial dynamics add a delay to the control loop which, strictly speaking, destroys passivity of the controller, as the control wrench and actuator wrench are not guaranteed equal. The motor inertial dynamics caused the quadrotor to become unstable when the controller parameter values were chosen too high (unstable behavior is clearly non-passive!). The rotor dynamics should be taken into account when implementing PBC on a quadrotor.

Currently available quadrotor hardware is not suited for interaction and energy-based control:
To be able to do energy-based control, the control wrench should be equal to the actuator wrench. Currently, the actuator wrench is only an approximation. Available motors, electronic speed controllers and flight controllers are not designed to control the wrench directly. The motors or speed controllers have no feedback, so the wrench is sent to the motors purely by feed-forward. Furthermore, flight controllers relate the mixer input to linear rotor speeds and therefore a non-linear wrench. As a consequence, a proportional controller on the attitude is not the same as a viscous/linear physical damper on the rotation. Furthermore, if an actuator reaches a limit, the mixer uses an optimization algorithm to determine the control outputs. Physics are lost in the signal-based flight controller.

The developed RCC controller is mainly suited for interaction control, not for position control in free flight.
For interaction control, energy and intuitive response of the controller are key. In the proposed controller, energy is explicitly taken into account and the behavior of the quadrotor can be controlled intuitively by changing the Remote Center of Compliance (RCC). However, due to its underactuated nature, the quadrotor is prone to motions in the non-actuated directions. This can result in a pendulum-like oscillation with frame Ψ_i as pivot point. This undesirable 'swinging' motion is slowly damped by the controller rotation spring and damping. Furthermore, low spring stiffnesses which are needed to get the controller stable, can result in large steady-state errors in the position. For these reasons, the developed controller is less suited for position control in free flight.

Since the controller implementation is distributed, communication should be real-time.
In this thesis, first steps have been taken to implement energy-based control with a Pixhawk and a laptop running ROS. The controller rotational damper has been implemented onboard

on the Pixhawk, and the rest of the controller damper and spring have been implemented in a ROS node. However, ROS is not designed for real-time application specifically. For example, Nagle's algorithm for the TCP transport layer is enabled by default which will result in a lower effective control rate. Nagle's algorithm should be disabled manually for each subscriber when using ROS for real-time applications.

A realistic simulation environment is indispensable for developing advanced control algorithms for quadrotors.
Quadrotors are inherently unstable and need to be stabilized by the control algorithm. The step from basic simulations (not taking the rotor dynamics, flight controller influence and control delays into account) to a real experiment is large. There are many elements which can influence the controller stability. Testing the experimental control algorithm in a more realistic simulation environment, like one including Software-in-the-loop (SITL), takes an extra step towards real-life experiments. The time-to-test is greatly reduced in comparison to real-life experiments, making debugging, adjusting and testing of the controller much more easy. Additionally, it will reduce the number of costly quadrotor crashes.

7.2 Future work

This section has been split in two separate subsections. In Subsection 7.2.1, future work that could be performed to improve the implementation of a energy-based control law on a quadrotor platform will be discussed. In Subsection 7.2.2, a research lookout will be given on RCC-based control introduced in this thesis.

7.2.1 Implementation

Quadrotor hardware currently available is not suited for energy-based control. It is difficult to actuate a physical wrench as there is no feedback available from the actuators and the flight controllers are designed for signal-based control. Currently, the only option is to estimate the wrench by feed-forward, which performance strongly depends on the accuracy of the actuation model and the flight controller mixer algorithm. Future work should focus on improving the accuracy of the actuation wrench. Implementing a custom mixer algorithm with signals that have a physical meaning could be the first step. The current rotors are velocity-controlled. Further research could focus on the accuracy of the aerodynamic map from rotor velocity to the wrench and the influence of obstacles in the proximity of the quadrotor on this map (which can be important during interaction). If the accuracy of the map is not sufficient for the application, feedback should be implemented. Further research could examine options like electrical power measurements as feedback (as explored in Bangura and Mahony (2017)) or fitting the quadrotor with force sensors (like explored in van Westerveld (2016)).

In this thesis, first steps have been taken to implement energy-based control with a Pixhawk and a laptop running ROS. However, ROS is not designed for real-time application specifically. ROS2 will have real-time support, and should as soon as it has been released supersede ROS in future work.

7.2.2 Research outlook

The experiments that have been performed in this thesis are limited. The controller has been tested in a small lab, such that only the stability and not the behavior of the controller could be verified as only small control steps could be commanded. In a larger lab, larger control steps could be given to more test the behavior and its dependency on the location of the RCC frame more accurately. Additionally, interaction experiments should be performed to test the performance of the controller during interaction.

The main motivation to implement energy-based control on a quadrotor is interaction. Future work should focus on the controller behavior during interaction and its dependency on the

RCC frame. In the current implementation, only the z-position of the RCC frame can be adjusted. The controller could be extended easily such that the RCC frame can be chosen freely. Depending on the application, RCC-based control could prove to be a powerful tool. When a quadrotor is equipped with an end-effector (e.g. a robotic arm), the RCC could be placed in the end-effector, such that the rotational and translation stiffnesses are decoupled in the contact point.

During experiments, it has been found that the effects of unmodeled dynamics on the passivity of the quadrotor are significant. In future research, the effect of especially the rotor dynamics should be further examined. For example, it should be investigated if the passivity can be recovered effectively by modeling the rotor inertial- and aerodynamics.

During this research, the supervisor controller has not been considered. Designing a supervisor controller that injects energy in the spring and energy tank is a topic for future research. Next to this, strategies have to be explored to overcome the sliding behavior in the non-actuated direction. One option would be to switch controllers when the energy tank is empty: the quadrotor could switch to a fully passive control law with springs and dampers on the fully-actuated rotations only. This law will counter-act the sliding behavior in the non-actuated direction, but it will still suffer from position drift. Another option would to overcome the underactuation by using tilting propellers.

A Twists and wrenches

The motion of a rigid body can be described by twists and wrenches. Using screw theory, twists and wrenches are a generalized, geometrical representation of respectively the velocity of and the force acting on the body. A short introduction will be given here.

A.1 Lie group and lie algebra

A general change of coordinates from Ψ_a to Ψ_b can be expressed with a homogeneous transformation matrix H:

$$H_a^b = \begin{pmatrix} R_a^b & p_a^b \\ 0_3{}^T & 1 \end{pmatrix} \in SE(3) \tag{A.1}$$

Where $R_a^b \in SO(3)$ is the rotation matrix and $p_a^b \in \mathbb{R}^3$ the distance between the origin of Ψ_a and Ψ_b. The result is a homogeneous transformation matrix in the special Euclidean group $SE(3)$. Associated with this lie group, a lie algebra $se(3)$ can be defined describing the twist T of the body. Taking respectively the left or right translations from the lie group to the lie algebra gives:

$$\widetilde{T}_a^{a,b} = H_b^a \dot{H}_a^b \tag{A.2a}$$

$$\widetilde{T}_a^{b,b} = \dot{H}_a^b H_b^a \tag{A.2b}$$

Where $\widetilde{T}_b^{c,a} \in se(3)$ is the twist of frame Ψ_a w.r.t. Ψ_b described in frame Ψ_c in tilde form. A twist is a six-dimensional vector and represent the generalized velocity of the body.

$$T = \begin{pmatrix} \omega \\ v \end{pmatrix} \tag{A.3}$$

Where ω represents the angular and v the linear velocity of the body.

A.2 Change of coordinates of a twist

To describe a Twist in another frame, a coordinate transformation can be performed. By substituting Equation A.2 and A.2b the transformation of a twist in tilde form is obtained:

$$\widetilde{T}_a^{a,b} = H_b^a \widetilde{T}_a^{b,b} H_a^b \tag{A.4}$$

which can be rewritten to the vector form:

$$T_a^{a,b} = Ad_{H_b^a} T_a^{b,b} \tag{A.5}$$

where $Ad_{H_b^a}$ is the Adjoint of the homogeneous matrix H_b^a:

$$Ad_{H_b^a} := \begin{pmatrix} R_b^a & 0 \\ \widetilde{p}_b^a R_b^a & R_b^a \end{pmatrix} \tag{A.6}$$

A.3 Wrenches

The twist is a generalized velocity. The dual of twist, the generalized force, is expressed as a wrench such that the power P is equal to the product:

$$P = TW \tag{A.7}$$

where the twist T and wrench W should be expressed in the same coordinate system. The twist is a vector, the wrench a co-vector:

$$\begin{pmatrix} \tau & f \end{pmatrix} \tag{A.8}$$

where τ and f represent the 3-dimensional torque and force respectively.

A.4 Coordinate transformation wrenches

The power calculated should not depend on the coordinate system so:

$$W^b T_b^{b,a} = W^a T_b^{a,a} \tag{A.9}$$

In Equation A.5 it is shown how a twist transforms when changing coordinate systems. $T_b^{b,a}$ in Equation A.9 can be expressed in $T_b^{a,a}$ using the Adjoint matrix:

$$T_b^{b,a} = Ad_{H_a^b} T_b^{a,a} \tag{A.10}$$

Substituting A.9 and A.10 gives:

$$W^b Ad_{H_a^b} T_b^{a,a} = W^a T_b^{a,a} \tag{A.11}$$

Which after taking the transpose on both sides results in the general equation for the transformation of a wrench:

$$(W^a)^T = Ad_{H_a^b}^T (W^b)^T \tag{A.12}$$

B Bond graphs

Graphically, port-hamiltonian systems can be visualized in bond graphs. Bond graphs make use of abstraction of domains to give a domain independent representation of dynamical systems. For example, mechanical velocity and electrical current can be further abstracted to a general *flow*. The dual mechanical force and electrical voltage can be abstracted to an *effort*. The product of the flow and effort gives power. The generalization of variables is depicted in Table B.1.

Table B.1: Generalization variables of the mechanical and electrical domain

Generalization	**Mechanical**	**Electrical**
Flow	Velocity	Current
Effort	Force	Voltage

B.1 Bonds

The basic elements in the dynamical system described in bond graphs have one or two *power* ports. The element is connected via the ports to the system through bonds (See Figure B.1). These bonds contain information of both the flow and effort of the element. The basic bond is shown in Figure B.1a. A multibond (Figure B.1b) represents a multidimensional bond. The direction of the half arrow of the bond shows the positive power direction. The vertical bar can be on either side of the bond, depending on the element and the configuration and represents the *causality* of the element. An element can either have a flow-in (equal to effort-out) or effort-in (equal to flow-out) causality. When the bar is at the side of the element, the causality is effort-in. When the bar is at the side of the junction, the causality of the element is flow-in. When the element has a flow-in causality, the effort of the element is calculated by the flow through the element. For effort-in, this is the other way around.

B.2 Junctions

Elements are connected to other elements via junctions. Fundamentally, elements can be connected to other elements in two different ways, either with a common flow or effort. For example, in the mechanical domain, this translates to a common velocity (elements placed in parallel) or a common force (elements placed in series) respectively. In the electrical domain, this translates to a common current (series) or voltage (parallel). Junctions define the configuration of the connection of the elements. The two junctions are shown in Figure B.2. As an example, three bonds are attached to the junction but no elements are shown. The common flow junction is shown in Figure B.2a. The constitutive equations of this 1-junction are:

$$f_1 = f_2 = f_3 = f_{...} \tag{B.1a}$$

$$e_1 + e_2 + e_3 + e_{...} = 0 \tag{B.1b}$$

(a) Bond **(b)** Multibond

Figure B.1: Half-arrow bond used in bond graphs to connect ports.

(a) 1-junction with 3 connected bonds **(b)** 0-junction with 3 connected bonds

Figure B.2: Junction structures.

The common effort junction is shown in Figure B.2b. The constitutive equations of this 0-junction are:

$$e_1 = e_2 = e_3 = e_{...} \tag{B.2a}$$

$$f_1 + f_2 + f_3 + f_{...} = 0 \tag{B.2b}$$

In Equation B.1b and B.2b, one could recognize Kirchhoff's voltage law and current law respectively. Note that at a 1-junction only a single element with a flow-out causality is allowed because the flow is equal for each element. For the 0-junction this is a single element with effort-out causality.

B.3 Sources

Sources in dynamical systems can be generalized to two types of sources, effort and flow sources. The mechanical and electrical analogies of these sources are presented in Table B.2. Intuitively, the causality of the sources are fixed.

Table B.2: Generalized sources

Source	**Mechanical**	**Electrical**
Effort (Se)	Force-fixed actuator	Voltage source
Flow (Sf)	Velocity-fixed actuator	Current source

B.4 One-port elements

Elements with a single power port and their mechanical and electrical analogy are the shown in Table B.3. The R-element calculates either the flow or effort (depending on the causality), using Ohms law:

$$\frac{e}{f} = R \tag{B.3}$$

The I-element has the following constitutive relation:

$$f_{\text{out}} = \frac{1}{m}\int e_{\text{in}}\,\mathrm{d}t \tag{B.4}$$

Normally, the flow is calculated as the integral of the effort. Effort-in is the so called *prefered* causality of the I-element. The C-element has the following constitutive relation:

$$e_{\text{out}} = \frac{1}{C}\int f_{in}\,\mathrm{d}t \tag{B.5}$$

where the effort is calculated as the integral of the flow when in preferred causality. As shown in the previous section, a 1-junction only allows a single element with flow-out causality and a 0-junction only a single element with effort-out causality. Connection of elements can result in a causality *conflict*. For example, when two I-elements are connected to a single 1-junction, only one of the I-elements can have the preferred flow-out causality. The other I-element will have a *differential* causality, where the effort is calculated by differentiating Equation B.4.

Table B.3: Generalized one-port elements in the mechanical and electrical domain

Element	**Mechanical**	**Electrical**
R-element	Friction	Resistor
I-element	Inertia	Inductor
C-element	Spring	Capacitor

B.5 Two-port elements

Generalized elements with two power ports are the transformer and the gyrator. The transformer scales the flow and effort with a transformation ratio r, such that the element is power conservative:

$$e_1 = r^T * e_2 \tag{B.6a}$$

$$f_2 = r * f_1 \tag{B.6b}$$

Examples of transformers in the mechanical domain are a gear box, a pulley system and a lever. In the electrical domain its analogy is just a transformer. A gyrator scales the flow from one of its ports to the effort from the other port and the other way around in a power conservative manner via the gyration ratio r:

$$e_1 = r^T * f_2 \tag{B.7a}$$

$$e_2 = r * f_1 \tag{B.7b}$$

Gyrators 'transform' a flow to an effort or the other way around. An example of a (cross-domain) gyrator is an ideal electrical motor. The rotational velocity (flow) of the motor depends on the voltage (effort) via a motor constant r. Note that the causality of both two-port elements is indifferent and determined by the configuration of the rest of the model.

B.6 Twists and wrenches in bond graphs

As shown in Appendix A, a twist is defined as a 6-dimensional vector:

$$T = \begin{pmatrix} \omega \\ v \end{pmatrix} \tag{B.8}$$

and a wrench as a 6-dimensional co-vector:

$$\begin{pmatrix} \tau & f \end{pmatrix} \tag{B.9}$$

and the product of the twist and wrench is the power. In bond graph terms, the twist and wrench are a 6-dimensional flow and effort respectively. A bond graph model describing the twists and wrenches of a body consists 6-dimensional elements and multibonds. Element parameters like the R, I, C and R are described by a 6x6 matrix.

C Quadrotor model

C.1 Newton-Euler equations

The quadrotor is modeled as a single rigid body. The Newton-Euler equations describe the motion of a rigid body as a function of the current state and the externally applied wrench. Expressed in the body-fixed frame Ψ_a, the Newton-Euler equation has the following form (Stramigioli and Bruyninckx, 2001):

$$\mathscr{I}^a \dot{T}_a^{a,0} = -ad_{T_a^{a,0}} \mathscr{I}^a T_a^{a,0} + (W^a|_{\text{ext}})^T \tag{C.1}$$

with $\mathscr{I}^a$ the principal inertia tensor, $T_a^{a,0}$ the body-fixed twist of the quadrotor w.r.t. the inertial frame and $W^a|_{\text{ext}}$ the externally applied body-fixed wrench.

$$-\mathscr{I}^a \dot{T}_a^{a,0} - ad_{T_a^{a,0}} \mathscr{I}^a T_a^{a,0} + (W^a|_{\text{ext}})^T = 0 \tag{C.2}$$

By bringing al elements of Equation C.1 to one side, the equation is rewritten to a sum of wrenches which equals zero (Equation C.2). In this equation, $-\mathscr{I}^a \dot{T}_a^{a,0}$, $-ad_{T_a^{a,0}} \mathscr{I}^a T_a^{a,0}$ and $(W^a|_{\text{ext}})^T$ are wrenches describing the inertia, rigid body fictitious wrench (e.g. gyroscopic effects) and external wrench respectively. Equation C.2 translates to a 1-junction in a bond graph, as shown in Figure C.1. The direction of the bonds are defined by the signs. The relations describing the I-element is:

$$W^a|_{\text{I}} = \mathscr{I}^a \dot{T}_a^{a,0} \tag{C.3}$$

which can be recognized as the differential form of the constitutive relation of a inertia. The preferred integral form is:

$$T_a^{a,0} = (\mathscr{I}^a)^{-1} \int W^a|_{\text{I}} \, \mathrm{d}t \tag{C.4}$$

The rigid body fictitious wrench is described by a gyrator with effort-out causality:

$$W^a|_{\text{gy}} = ad_{T_a^{a,0}} \mathscr{I}^a T_a^{a,0} \tag{C.5}$$

where $\mathscr{I}^a T_a^{a,0}$ is the momentum of the rigid body, which is equal to the state of the I-element and is provided as a signal from the I-element.

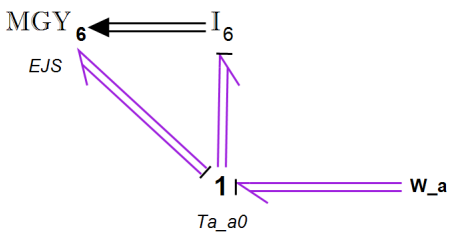


Figure C.1: Bond graph realization of the Newton-Euler equations of motion. The flow of the 1-junction is the body-fixed twist $T_a^{a,0}$. W_a is the external wrench, I_6 the 6D inertia and the MGY describes the fictitious wrench, calculated from the state of I_6.

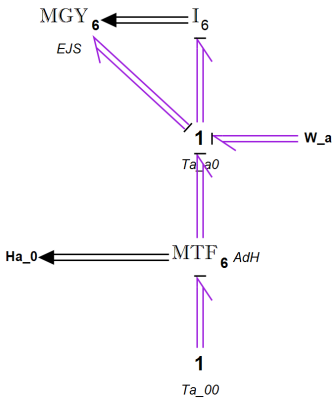


Figure C.2: Bond graph realization of a rigid body with transformation to world frame. The transformation of the twist and dual wrench expressed in the body-fixed frame Ψ_a to world frame Ψ_0 is performed by the MTF. H_a^0 required for the transformation is calculated by integrating the twist.

C.2 Transformation to world frame

The port variables (twist and dual wrench) of the 1-junction of the rigid body model of Figure C.1 are described in the body-fixed frame Ψ_a. A transformation transforms the port variables from Ψ_a to the world frame Ψ_0 (Appendix A):

$$T_a^{0,0} = Ad_{H_a^0} T_a^{a,0} \tag{C.6}$$

$$(W^a)^T = Ad_{H_a^0}^T (W^0)^T \tag{C.7}$$

Where H_a^0 is the homogeneous transformation matrix from frame Ψ_a to Ψ_0 and can be calculated by rewriting and integrating the definition of the twist $T_a^{0,0}$ (Appendix A):

$$\widetilde{T}_a^{0,0} = \dot{H}_a^0 H_0^a \tag{C.8}$$

$$H_a^0 = \int \widetilde{T}_a^{0,0} H_a^0 \mathrm{d}t + H_{a_{\text{init}}}^0 \tag{C.9}$$

The rigid body dynamics with the transformation to the world frame added is shown in Figure C.2. The MTF implements Equation C.9 to calculate H_a^0. H_a^0 is again used to perform the transformation of the twist and dual wrench.

C.3 Gravity

The gravitational wrench acts on the center of mass of the rigid body model and depends on the mass and the gravitational constant. A gravity frame can be defined in the center of mass of the rigid body which orientation is always equal to the world frame. In this gravity frame, the gravitational wrench is constant and equal to:

$$(W^g|_{\text{gravity}})^T = mass * \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ -9.81 \end{pmatrix} \tag{C.10}$$

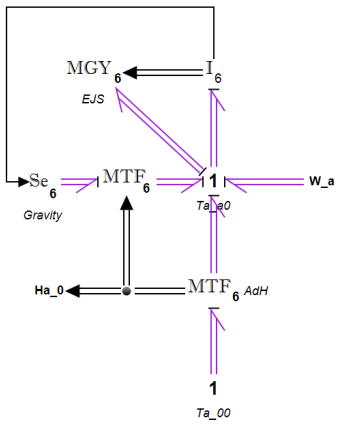


Figure C.3: Bond graph realization of a rigid body with transformation to world frame and gravity. The gravity wrench is expressed in the gravity frame Ψ_g. The wrench is transformed to frame Ψ_a with rotation matrix R_a^0.

To incorporate the gravity in the model, the wrench should be described in the body-fixed frame Ψ_a in stead of the local gravity frame Ψ_g. The wrench is transformed from the gravity frame to the body-fixed frame:

$$(W^a|_{\text{gravity}})^T = Ad_{H_a^g}^T (W^g|_{\text{gravity}})^T \tag{C.11}$$

Where

$$H_a^g = \begin{pmatrix} R_a^0 & 0 \\ 0 & 1 \end{pmatrix} \tag{C.12}$$

The final rigid body model including gravity is depicted in Figure C.3. The effort source is the gravity wrench source and is defined in frame Ψ_g. The input signal originating from the I-element caries the mass of the rigid body. The MTF element transforms the gravity wrench from the frame Ψ_g to Ψ_a with use of R_a^0 extracted from the H_a^0 signal.

Bibliography

Bangura, M. and R. Mahony (2017), Thrust control for multirotor aerial vehicles, **vol. 33**, no.2, pp. 390–405.

Breedveld, P. C. (1984), *Physical systems theory in terms of bond graphs*, Twente University.

Cheshire, S. (2005), TCP performance problems caused by interaction between Nagle's algorithm and delayed ACK, *Self-published online: http://www.stuartcheshire.org/papers/NagleDelayedAck*.

Dietrich, A., C. Ott and S. Stramigioli (2016), Passivation of Projection-Based Null Space Compliance Control Via Energy Tanks, **vol. 1**, no.1, pp. 184–191.

Fasse, E. D. and J. F. Broenink (1997), A spatial impedance controller for robotic manipulation, **vol. 13**, no.4, pp. 546–556.

Folkertsma, G. A. (2017), *Energy-based and biomimetic robotics*, Ph.D. thesis, University of Twente.

García-Canseco, E., D. Jeltsema, R. Ortega and J. M. Scherpen (2010), Power-based control of physical systems, **vol. 46**, no.1, pp. 127–132.

Hogan, N. (1984), Impedance control: An approach to manipulation, in *American Control Conference, 1984*, IEEE, pp. 304–313.

Ortega, R., A. J. Van Der Schaft, I. Mareels and B. Maschke (2001), Putting energy back in control, **vol. 21**, no.2, pp. 18–33.

Paynter, H. M. (1961), *Analysis and design of engineering systems*, MIT press.

van der Schaft, A., D. Jeltsema et al. (2014), Port-Hamiltonian systems theory: An introductory overview, **vol. 1**, no.2-3, pp. 173–378.

Sieberling, S., Q. Chu and J. Mulder (2010), Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction, **vol. 33**, no.6, pp. 1732–1742.

Stramigioli, S. (2001), Intrinsically passive control, *Modeling and IPC control of interactive mechanical systems - A coordinate-free approach*, pp. 125–145.

Stramigioli, S. (2015), Energy-aware robotics, in *Mathematical Control Theory I*, Springer, pp. 37–50.

Stramigioli, S. and H. Bruyninckx (2001), Geometry and screw theory for robotics, in *Proceedings IEEE International Conference on Robotics and Automation*, Citeseer.

Stramigioli, S. and V. Duindam (2001), Variable spatial springs for robot control applications, in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, volume 4, IEEE, pp. 1906–1911.

van Westerveld, J. (2016), Investigating the Usability of a Novel Type of Sensing for Aerial Vehicles.

Willems, J. C. (1972), Dissipative dynamical systems part I: General theory, **vol. 45**, no.5, pp. 321–351.

Yüksel, B., C. Secchi, H. H. Bülthoff and A. Franchi (2014), Reshaping the physical properties of a quadrotor through IDA-PBC and its application to aerial physical interaction, in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, IEEE, pp. 6258–6265.