

# Underactuated Control of Quadrotors with Collision Avoidance<sup>\*</sup>

Espen Oland Tom Stian Andersen Raymond Kristiansen

*Narvik University College, Department of Technology, 8505 Narvik, Norway (e-mail: {eol},{tsa},{rayk}@hin.no).*

---

**Abstract:** In this paper a saturated controller is derived that solves the translational control problem for underactuated quadrotors. The controller is applied to multiple quadrotors during a formation reconfiguration where the quadrotors move from an initial position to a desired position. The null-space-based behavioral control method is a popular method for avoiding collisions between mobile agents, and is augmented in this paper with saturation functions to enable feasible collision-free trajectories for the quadrotors during the reconfiguration. Simulations have been performed to validate the theoretical results and show the performance of the solution.

*Keywords:* Lyapunov stability, autonomous control, nonlinear control systems, obstacle avoidance, saturated control, tasks.

---

## 1. INTRODUCTION

The problem of controlling quadrotors has received much attention in the control community the last decade ranging from autonomous mapping, collision avoidance, formation flight and cooperative control (*cf.* Bouabdallah and Siegwart (2005), Grzonka et al. (2012), Guerrero et al. (2012), Guisser and Medromi (2009) and references therein). Flying with a formation of quadrotors have several distinct advantages. With several quadrotors, it is possible to obtain 3D measurements which will be more accurate than what a single quadrotor is able to obtain. Furthermore, pictures taken with multiple quadrotors can be stitched together providing a higher resolution than a single picture taken by one quadrotor. Even though there are many advantages, there are also several disadvantages where the main challenge is collision avoidance. With multiple quadrotors located in the same airspace some sort of supervisory method must be applied to avoid collisions. One of the most well known collision avoidance methods is potential fields presented by Khatib (1986) which creates an attractive potential around the desired position and repulsive potentials around obstacles. By following the negative gradient of the resulting field, an agent will be able to reach a desired position without colliding with any obstacles. But; the method suffers from several inherent limitations such as local minima which can trap an agent from reaching its destination, and the inability to pass through narrow gaps.

Another method that can be used for collision avoidance is the Null-Space-Based (NSB) behavioral control method presented by Antonelli et al. (2005). The method enables an agent to have multiple tasks, such as collision avoidance, move to target, maintain rigid formation and others. The basic idea is to prioritize the tasks into a hierarchy,

where higher level tasks subsumes lower level tasks. This means that if a task conflict with a higher level task, its conflicting velocity components will be projected onto the null-space and thereby removing its contribution from the total reference velocity vector. The NSB method has resulted in many papers with applications to mobile robots, ships and spacecraft (*cf.* Arrichiello et al. (2006), Antonelli et al. (2009), Oland et al. (2010), Schlanbusch et al. (2011) and Schlanbusch and Oland (2013)).

A quadrotor has four rotors used for both rotational and translational control, and can change its orientation by changing the rotation speed of the rotors. One of the main challenges with quadrotors is that they only have four actuators and six degrees of freedom. With fewer actuators than degrees of freedom this constitutes an underactuated control problem (*cf.* Reyhanoglu et al. (1999)), which makes controller design more challenging. Several different concepts have been proposed; such as feedback linearization, backstepping, sliding mode and PID control. Common for all schemes is that a coupling between the translational and rotational dynamics must be found in order to track a desired position vector. This can be done directly in the desired attitude, which is typical for guidance schemes such as way-point guidance; or it can be done as part of the controller design. In Cunha et al. (2009) the authors apply nested saturation functions based on Teel (1992) to control a quadrotor. Controlling the center of mass relative a desired trajectory, the authors backstep four times until the coupling between the rotation and translation is found.

Instead of controlling the center of mass, it is possible to shift the origin away from the center of mass to enable a coupling between the rotational and translational dynamics.

---

<sup>\*</sup> This work was supported by the Norwegian Research Council and is part of the Arctic Earth Observation project 195143/I60.

## 1.1 Contribution

In this paper a saturated controller is derived based on the coupling between the rotational and translational dynamics and enables a quadrotor to track a desired trajectory. The controller is analyzed using Lyapunov theory where the tracking error is shown to be uniformly globally asymptotically stable. The same controller is then applied to multiple quadrotors performing a position reconfiguration where collision avoidance is handled using the NSB method which is augmented with saturation functions to avoid large velocity commands.

This paper is organized as follows: In Section 3 the quadrotor modeling is presented and shows how to make the coupling between rotational and translational dynamics. The saturated controller is shown in Section 4 while the saturated NSB method is derived in Section 5, and in Section 6 a simulation of four quadrotors perform position reconfiguration is shown and the results are discussed.

## 2. PRELIMINARIES

### 2.1 Notation and reference Frames

The time derivative is denoted  $\dot{x} = \frac{dx}{dt}$ , and its second derivative is given as  $\ddot{x} = \frac{d^2x}{dt^2}$ . Superscripts denote the reference frame of a vector where  $b$  is the body frame, and  $n$  is the North-East-Down (NED) frame which is assumed to be inertial. The two frames are shown in Figure 1 where the  $\mathbf{x}^n$  axis points North,  $\mathbf{y}^n$  points East and  $\mathbf{z}^n$  points down towards the center of the Earth. The body frame has its  $\mathbf{x}^b$  axis pointing through rotor 1 as shown in Figure 2,  $\mathbf{y}^b$  points through rotor 3 while  $\mathbf{z}^b$  completes the right handed orthonormal reference frame. The rotation matrix from body frame to NED frame is denoted  $\mathbf{R}_b^n \in \mathbb{R}^{3 \times 3}$  which holds the properties that  $\mathbf{R}_b^n \mathbf{R}_n^b = \mathbf{I}$  where  $\mathbf{I}$  is the identity matrix such that  $(\mathbf{R}_b^n)^\top = \mathbf{R}_n^b$ . The angular velocity between two frames is denoted  $\boldsymbol{\omega}_{\alpha,\beta}^\gamma$  which represents the angular velocity of frame  $\beta$  relative  $\alpha$  referenced in frame  $\gamma$ . The time derivative of the rotation matrix is given as  $\dot{\mathbf{R}}_b^n = \mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_{n,b}^b)$  where  $\mathbf{S}(\boldsymbol{\omega}_{n,b}^b)$  is the cross-product operator and  $\boldsymbol{\omega}_{n,b}^b$  is the angular velocity of the body frame relative the NED frame referenced in the body frame. Given two vectors  $\mathbf{v}_1 = [v_1 \ v_2 \ v_3]^\top$ ,  $\mathbf{v}_2 \in \mathbb{R}^3$ , the cross product operator can be written as

$$\mathbf{S}(\mathbf{v}_1) = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \quad (1)$$

and where  $\mathbf{S}(\mathbf{v}_1)\mathbf{v}_2 = \mathbf{v}_1 \times \mathbf{v}_2$  and  $\mathbf{S}(\mathbf{v}_1)\mathbf{v}_1 = -\mathbf{S}(\mathbf{v}_1)\mathbf{v}_1$ .

## 3. MODELING

A quadrotor has four rotors that are used as actuation to perform both rotational and translational motion. As shown in Figure 2 the speed of the rotors can directly be controlled producing the desired forces and moments. The thrust from the rotors is found as (cf. Castillo et al. (2005))

$$\mathbf{f}_t^b = - \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ k_1\omega_1^2 + k_2\omega_2^2 + k_3\omega_3^2 + k_4\omega_4^2 \end{bmatrix} \quad (2)$$

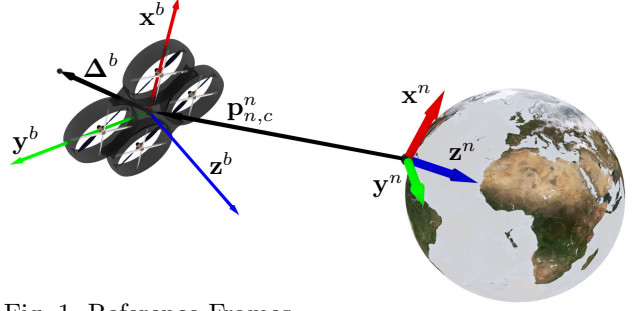


Fig. 1. Reference Frames

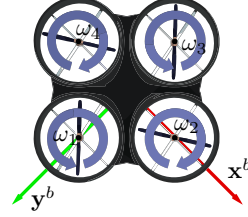


Fig. 2. Rotor Configuration

where  $k_1, k_2, k_3, k_4$  are positive constants and  $\omega_1, \omega_2, \omega_3, \omega_4$  are the speed of each rotor. The moments can be found related to the rotor speeds as

$$\boldsymbol{\tau}^b = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} (k_3\omega_3^2 - k_1\omega_1^2)l \\ (k_2\omega_2^2 - k_4\omega_4^2)l \\ \tau_1 + \tau_2 + \tau_3 + \tau_4 \end{bmatrix} \quad (3)$$

where  $\tau_1, \tau_2, \tau_3, \tau_4$  are the motor-moments of the rotors and  $l$  is the distance from the center of mass to the rotors.

By treating the NED frame as an inertial frame the rigid body dynamics of a quadrotor can be written as

$$\dot{\mathbf{p}}_{n,c}^n = \mathbf{R}_b^n \mathbf{v}^b \quad (4)$$

$$\dot{\mathbf{v}}^b = -\frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \mathbf{R}_b^n \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{v}^b - \mathbf{D} \mathbf{v}^b \quad (5)$$

$$\dot{\boldsymbol{\omega}}_{n,b}^b = \mathbf{J}^{-1} (-\mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{J} \boldsymbol{\omega}_{n,b}^b + \boldsymbol{\tau}^b) \quad (6)$$

$$\dot{\mathbf{R}}_b^n = \mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \quad (7)$$

where  $\mathbf{p}_{n,c}^n$  is the position vector from the origin of the NED frame to the center of mass of the quadrotor,  $\mathbf{v}^b$  is the velocity vector in body frame,  $m$  is the mass,  $g = 9.81 \text{ m/s}^2$  is the gravitational constant,  $\mathbf{D} = \text{diag}\{d_1, d_2, d_3\}$  is a positive definite damping matrix and  $\mathbf{J} = \text{diag}\{J_{xx}, J_{yy}, J_{zz}\}$  is the inertia matrix.

As illustrated in Figure 1, the origin can be shifted away from the center of mass giving the position vector in the NED frame as

$$\mathbf{p}_{n,o}^n = \mathbf{p}_{n,c}^n + \mathbf{R}_b^n \boldsymbol{\Delta}^b \quad (8)$$

where  $\mathbf{p}_{n,o}^n$  is the position vector between the NED frame and a point to be controlled and  $\boldsymbol{\Delta}^b := [0 \ 0 \ \Delta]^\top$  is a constant body-fixed displacement vector used to couple the translation and rotation. Differentiating (8), using (4) and (7), we obtain

$$\dot{\mathbf{p}}_{n,o}^n = \mathbf{R}_b^n (\mathbf{v}^b + \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \boldsymbol{\Delta}^b) \quad (9)$$

where the property  $\mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \boldsymbol{\Delta}^b = -\mathbf{S}(\boldsymbol{\Delta}^b) \boldsymbol{\omega}_{n,b}^b$  can be used to write (9) as

$$\dot{\mathbf{p}}_{n,o}^n = \mathbf{R}_b^n (\mathbf{v}^b - \mathbf{S}(\boldsymbol{\Delta}^b) \boldsymbol{\omega}_{n,b}^b). \quad (10)$$

Equation (10) can be differentiated once more, and by inserting (5) and (6) it becomes

$$\begin{aligned} \ddot{\mathbf{p}}_{n,o}^n = & \mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) (\mathbf{v}^b - \mathbf{S}(\boldsymbol{\Delta}^b) \boldsymbol{\omega}_{n,b}^b) \\ & + \mathbf{R}_b^n \left( -\frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ T \end{bmatrix} + \mathbf{R}_n^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{v}^b - \mathbf{D} \mathbf{v}^b \right. \\ & \left. - \mathbf{S}(\boldsymbol{\Delta}^b) \mathbf{J}^{-1} (-\mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{J} \boldsymbol{\omega}_{n,b}^b + \boldsymbol{\tau}^b) \right). \end{aligned} \quad (11)$$

Gathering the control signals, the acceleration vector can be written as

$$\begin{aligned} \ddot{\mathbf{p}}_{n,o}^n = & \mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) (\mathbf{v}^b - \mathbf{S}(\boldsymbol{\Delta}^b) \boldsymbol{\omega}_{n,b}^b) \\ & + \mathbf{R}_b^n (\mathbf{B} \mathbf{u} + \mathbf{R}_n^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{v}^b - \mathbf{D} \mathbf{v}^b) \\ & + \mathbf{S}(\boldsymbol{\Delta}^b) \mathbf{J}^{-1} \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{J} \boldsymbol{\omega}_{n,b}^b. \end{aligned} \quad (12)$$

where  $\mathbf{u} = [\tau_x \ \tau_y \ T]^\top$  and

$$\mathbf{B} = \begin{bmatrix} 0 & \frac{\Delta}{J_{yy}} & 0 \\ -\frac{\Delta}{J_{xx}} & 0 & 0 \\ 0 & 0 & -\frac{1}{m} \end{bmatrix}. \quad (13)$$

The acceleration now contains three states to be controlled and three actuator signals to control them.

*Remark 1.* No control objective has been placed on the yaw torque and it remains a free variable that can be used for an additional control objective.

#### 4. SATURATED CONTROLLER DESIGN

Let the tracking errors be defined as

$$\mathbf{e}_1 = \mathbf{p}_{n,o}^n - \mathbf{p}_r^n(t) \quad (14)$$

$$\mathbf{e}_2 = \dot{\mathbf{p}}_{n,o}^n - \dot{\mathbf{p}}_r^n(t) \quad (15)$$

where  $\mathbf{p}_r^n(t)$  is a reference trajectory that is to be tracked by the control law. The velocity vector can be written using (9) as  $\mathbf{v}^b = \mathbf{R}_b^n \mathbf{e}_2 + \mathbf{R}_n^b (\dot{\mathbf{p}}_r^n - \mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \boldsymbol{\Delta}^b)$  which can be used together by differentiating (14)-(15) to find the tracking error dynamics as

$$\dot{\mathbf{e}}_1 = \mathbf{e}_2 \quad (16)$$

$$\begin{aligned} \dot{\mathbf{e}}_2 = & \mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) (\mathbf{v}^b - \mathbf{S}(\boldsymbol{\Delta}^b) \boldsymbol{\omega}_{n,b}^b) \\ & + \mathbf{R}_b^n (\mathbf{B} \mathbf{u} + \mathbf{R}_n^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{v}^b - \mathbf{D} \mathbf{R}_n^b \mathbf{e}_2) \\ & - \mathbf{D} \mathbf{R}_n^b (\dot{\mathbf{p}}_r^n - \mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \boldsymbol{\Delta}^b) \\ & + \mathbf{S}(\boldsymbol{\Delta}^b) \mathbf{J}^{-1} \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{J} \boldsymbol{\omega}_{n,b}^b - \ddot{\mathbf{p}}_r^n(t) \end{aligned} \quad (17)$$

which is similar to a double integrator with dynamics.

Let a continuous saturation function be defined as  $\sigma(\mathbf{s}) = \tanh(\mathbf{s})$  which holds the property that  $\mathbf{s}^\top \sigma(\mathbf{s}) > 0 \ \forall \mathbf{s}$ . A Lyapunov Function Candidate can now be chosen as

$$V_1 = k_p \mathbf{e}_1^\top \sigma(\mathbf{e}_1) + \frac{1}{2} \mathbf{e}_2^\top \mathbf{e}_2 \quad (18)$$

where  $k_p$  is a positive gain, and where the Lyapunov Function Candidate is positive definite and radially unbounded. By differentiating (18) and inserting (16)-(17) the Lyapunov derivative becomes

$$\begin{aligned} \dot{V}_1 = & k_p \mathbf{e}_2^\top \sigma(\mathbf{e}_1) + k_p \mathbf{e}_1^\top \dot{\sigma}(\mathbf{e}_1) \mathbf{e}_2 \\ & + \mathbf{e}_2^\top (\mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) (\mathbf{v}^b - \mathbf{S}(\boldsymbol{\Delta}^b) \boldsymbol{\omega}_{n,b}^b) \\ & + \mathbf{R}_b^n (\mathbf{B} \mathbf{u} + \mathbf{R}_n^b \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{v}^b - \mathbf{D} \mathbf{R}_n^b \mathbf{e}_2) \\ & - \mathbf{D} \mathbf{R}_n^b (\dot{\mathbf{p}}_r^n - \mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \boldsymbol{\Delta}^b) \\ & + \mathbf{S}(\boldsymbol{\Delta}^b) \mathbf{J}^{-1} \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{J} \boldsymbol{\omega}_{n,b}^b - \ddot{\mathbf{p}}_r^n(t)) \end{aligned} \quad (19)$$

where  $\mathbf{e}_1 := [e_{1,x} \ e_{1,y} \ e_{1,z}]^\top$  and the derivative of the saturation function is found as

$$\dot{\sigma}(\mathbf{e}_1) = \mathbf{I} - \begin{bmatrix} \tanh^2(e_{1,x}) & 0 & 0 \\ 0 & \tanh^2(e_{1,y}) & 0 \\ 0 & 0 & \tanh^2(e_{1,z}) \end{bmatrix} \quad (20)$$

which is zero when it is in saturation. The control law can now be chosen as

$$\begin{aligned} \mathbf{u} = & \mathbf{B}^{-1} (-\mathbf{R}_b^n \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{v}^b + \mathbf{D} \mathbf{R}_n^b (\dot{\mathbf{p}}_r^n \\ & - \mathbf{R}_b^n \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \boldsymbol{\Delta}^b) - \mathbf{S}(\boldsymbol{\Delta}^b) \mathbf{J}^{-1} \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) \mathbf{J} \boldsymbol{\omega}_{n,b}^b \\ & + \mathbf{R}_n^b \dot{\mathbf{p}}_r^n(t) - \mathbf{S}(\boldsymbol{\omega}_{n,b}^b) (\mathbf{v}^b - \mathbf{S}(\boldsymbol{\Delta}^b) \boldsymbol{\omega}_{n,b}^b) \\ & + \mathbf{R}_n^b (-k_p \dot{\sigma}(\mathbf{e}_1) \mathbf{e}_1 - k_p \sigma(\mathbf{e}_1) - \mathbf{K}_d \sigma(\mathbf{e}_2))) \end{aligned} \quad (21)$$

where  $\mathbf{K}_d > \mathbf{0}$  is a diagonal gain matrix. The control law contains the term  $k_p \dot{\sigma}(\mathbf{e}_1) \mathbf{e}_1$  which is zero when the function is in saturation, meaning that all the tracking functions in the control law remain saturated. Inserting the control law (21) into (19) the Lyapunov derivative becomes

$$\dot{V}_1 = -\mathbf{e}_2^\top \mathbf{K}_d \sigma(\mathbf{e}_2) - \mathbf{e}_2^\top \mathbf{R}_b^n \mathbf{D} \mathbf{R}_n^b \mathbf{e}_2 \quad (22)$$

which is negative semidefinite. The matrix  $\mathbf{R}_b^n \mathbf{D} \mathbf{R}_n^b$  is a positive definite matrix that provides damping to the system. Since trace is similarity invariant, it follows that  $\text{Tr}(\mathbf{R}_b^n \mathbf{D} \mathbf{R}_n^b) = \text{Tr}(\mathbf{D})$  which contains the sum of the positive eigenvalues of the damping matrix, meaning that  $\mathbf{R}_b^n \mathbf{D} \mathbf{R}_n^b$  must be a positive definite matrix.

It is evident from the Lyapunov derivative (22) that the tracking error  $\mathbf{e}_2 \rightarrow \mathbf{0}$ , and to study the stability properties of  $\mathbf{e}_1$ , the control law (21) can be inserted into (17) resulting in the closed loop dynamics as

$$\dot{\mathbf{e}}_1 = \mathbf{e}_2 \quad (23)$$

$$\begin{aligned} \dot{\mathbf{e}}_2 = & -\mathbf{R}_b^n \mathbf{D} \mathbf{R}_n^b \mathbf{e}_2 - k_p \dot{\sigma}(\mathbf{e}_1) \mathbf{e}_1 - k_p \sigma(\mathbf{e}_1) \\ & - \mathbf{K}_d \sigma(\mathbf{e}_2). \end{aligned} \quad (24)$$

Since the reference trajectory is time-varying, the Matrosov theorem (*cf.* Hahn (1967)) can be used to prove stability of  $\mathbf{e}_1$ . Consider an auxiliary function  $W = \mathbf{e}_1^\top \mathbf{e}_2$  which can be differentiated and by inserting (23)-(24), it becomes

$$\begin{aligned} \dot{W} = & \mathbf{e}_2^\top \mathbf{e}_2 + \mathbf{e}_1^\top (-\mathbf{R}_b^n \mathbf{D} \mathbf{R}_n^b \mathbf{e}_2 - k_p \dot{\sigma}(\mathbf{e}_1) \mathbf{e}_1 \\ & - k_p \sigma(\mathbf{e}_1) - \mathbf{K}_d \sigma(\mathbf{e}_2)). \end{aligned} \quad (25)$$

Now consider the set  $\mathcal{N} = \{\mathbf{p}_{n,o}^n \in \mathbb{R}^3 \mid \mathbf{e}_2 = \mathbf{0}\}$ , and by evaluating the auxiliary function in this set, its derivative becomes

$$\dot{W} = -k_p \mathbf{e}_1^\top \dot{\sigma}(\mathbf{e}_1) \mathbf{e}_1 - k_p \mathbf{e}_1^\top \sigma(\mathbf{e}_1) \quad (26)$$

which is definitely nonzero. The Lyapunov function (18) is positive definite, decrescent and its derivative (22) is negative semidefinite and the derivative of the auxiliary function (26) is definitely nonzero in the set  $\mathcal{N}$ . Thus, all conditions of Theorem 55.3 in Hahn (1967) are met,

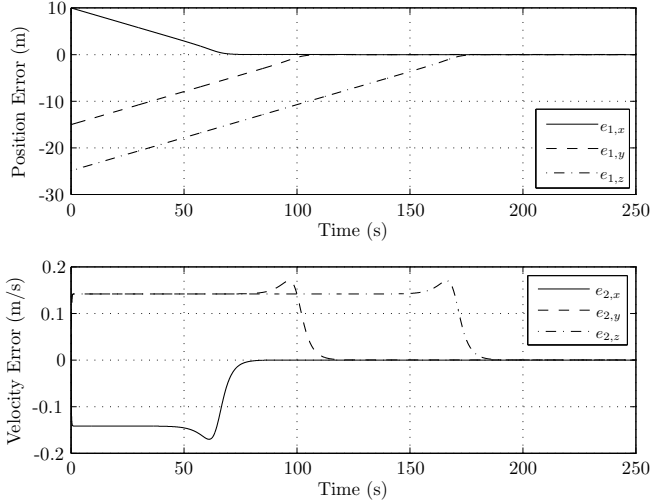


Fig. 3. Position and velocity tracking errors

and we can conclude that the origin  $(\mathbf{e}_1, \mathbf{e}_2) = (\mathbf{0}, \mathbf{0})$  is uniformly asymptotically stable (UAS). Furthermore, since the Lyapunov function (18) is radially unbounded it follows using standard Lyapunov arguments (*cf.* Khalil (2002)) that the origin  $(\mathbf{e}_1, \mathbf{e}_2) = (\mathbf{0}, \mathbf{0})$  is uniformly globally asymptotically stable (UGAS).

#### 4.1 Simulation of the Saturated Control Law

The attitude dynamics is used to solve the translational control problem. When  $\mathbf{e}_1$  and  $\mathbf{e}_2$  are in saturation, the trajectory tracking problem can be related to that of tracking a straight line, meaning that the attitude will converge to a constant value and the angular velocity will go to zero. Consider the scenario of tracking a circle where

$$\mathbf{p}_r^n(t) = [R \cos(\omega_r t) \quad R \sin(\omega_r t) \quad z_r]^\top \quad (27)$$

$$\dot{\mathbf{p}}_r^n(t) = [-R\omega_r \sin(\omega_r t) \quad R\omega_r \cos(\omega_r t) \quad 0]^\top \quad (28)$$

$$\ddot{\mathbf{p}}_r^n(t) = [-R\omega_r^2 \cos(\omega_r t) \quad -R\omega_r^2 \sin(\omega_r t) \quad 0]^\top \quad (29)$$

where  $R = 50$  m is the radius of the circle,  $\omega_r = 0.002$  rad/s is the reference angular velocity of the circle,  $t$  is the time,  $z_r = -25$  m is the reference altitude and the displacement vector is chosen as  $\Delta = 0.1$  m. The position and velocity errors are shown in Figure 3 where the velocity error goes into saturation and stays in saturation until the position error has gone to zero. The attitude and angular velocity are shown in Figure 4, where the attitude holds constant values while the velocity error is in saturation, while the angular velocity quickly converges to zero. The angular velocity has three small spikes when the position and velocity errors go to zero.

## 5. NULL-SPACE-BASED BEHAVIORAL CONTROL

Let us now consider the case of multiple quadrotors using the saturated controller (21) for translational control. With multiple quadrotors, there exist a possibility of collisions which in this paper is avoided using the NSB method. Based on the work by Antonelli et al. (2005) let a task variable be defined as a function of the position such that  $\xi_i = f(\mathbf{p}^n) \in \mathbb{R}^3$  where  $i = 1, 2$  and  $\mathbf{p}^n$  is a position vector. It maintains the differential relation as

$$\dot{\xi}_i = \frac{\partial f(\mathbf{p}^n)}{\partial \mathbf{p}^n} \dot{\mathbf{p}}^n := \mathbf{J}_i(\mathbf{p}^n) \dot{\mathbf{p}}^n \quad (30)$$

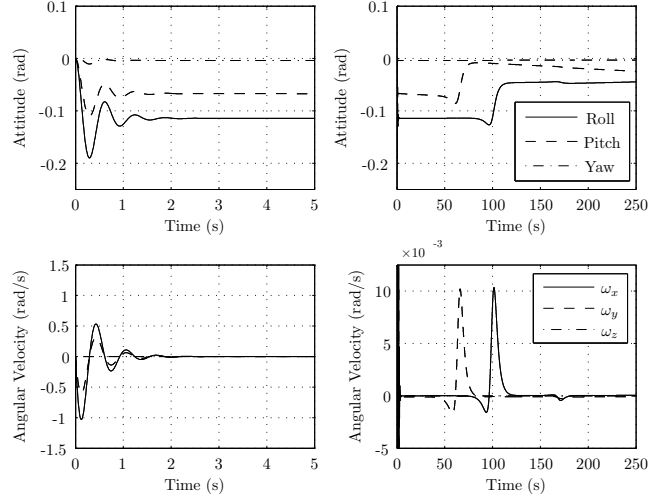


Fig. 4. Attitude and angular velocity

where  $\mathbf{J}_i$  is the configuration-dependent Jacobian matrix. This can be solved for the velocity using the Moore-Penrose pseudo-inverse as  $\dot{\mathbf{p}}^n = \mathbf{J}_i^\dagger \dot{\xi}_i$  where  $\mathbf{J}_i^\dagger = \mathbf{J}_i^\top (\mathbf{J}_i \mathbf{J}_i^\top)^{-1}$ . By defining a desired task,  $\xi_{i,d}$ , the least squares solution can be used to find a reference velocity that fulfills that task as

$$\dot{\mathbf{p}}_r^n = \mathbf{J}_i^\dagger \dot{\xi}_{i,d}. \quad (31)$$

As pointed out in Antonelli et al. (2005), if (31) is integrated in order to find the reference position, it will result in numerical drift. This is solved using a closed loop inverse kinematic algorithm as

$$\dot{\mathbf{p}}_r^n = \mathbf{J}_i^\dagger (\dot{\xi}_{i,d} + \Lambda \tilde{\xi}_i) \quad (32)$$

where  $\Lambda$  is a gain matrix and  $\tilde{\xi}_i = \xi_{i,d} - \xi_i$ . In analogy to (32) a single task velocity can now be computed as

$$\dot{\mathbf{p}}_1^n = \mathbf{J}_i^\dagger (\dot{\xi}_{i,d} + \Lambda_i \tilde{\xi}_i) \quad (33)$$

and multiple tasks can be added together. With two tasks, the reference velocity vector can be found as

$$\dot{\mathbf{p}}_r^n = \dot{\mathbf{p}}_1^n + (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1) \dot{\mathbf{p}}_2^n \quad (34)$$

which projects conflicting lower level velocity components onto the null-space and adds them together producing the total reference velocity vector.

#### 5.1 Saturated NSB Control

One major issue with the NSB method, is that when the initial position error is large, it will produce large reference velocities which may be hard or impossible to track for a given controller. To design a feasible reference velocity, let

$$\dot{\mathbf{p}}_r^n = \sigma_1(\dot{\mathbf{p}}_1^n) + (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1) \sigma_2(\dot{\mathbf{p}}_2^n) \quad (35)$$

and  $\sigma_i(\cdot)$ ,  $i = 1, 2$ , is a linear saturation function

$$\sigma_i(s) = \begin{cases} s & \text{if } |s| < \text{sign}(s) s_{max,i} \\ \text{sign}(s) s_{max,i} & \text{otherwise} \end{cases} \quad (36)$$

where  $s_{max,i}$  is the maximum output from the saturation function. Though the saturation function is defined for scalar functions, it can be applied on a component level to both vectors and matrices. To prove that the first task always is fulfilled, let a Lyapunov Function Candidate be defined as  $V_2 = \frac{1}{2} \tilde{\xi}_1^\top \tilde{\xi}_1$  which can be differentiated, and using that  $\dot{\xi}_1 = \mathbf{J}_1 \dot{\mathbf{p}}_r^n$  together with (35), it results in

$$\begin{aligned}\dot{V}_2 &= \tilde{\xi}_1^\top (\dot{\xi}_{1,d} - \mathbf{J}_1(\sigma_1(\dot{\mathbf{p}}_1^n) + (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1)\sigma_2(\dot{\mathbf{p}}_2^n))) \quad (37) \\ &= \tilde{\xi}_1^\top (\dot{\xi}_{1,d} - \mathbf{J}_1(\sigma_1(\mathbf{J}_1^\dagger (\dot{\xi}_{1,d} + \Lambda_1 \tilde{\xi}_1) \\ &\quad + (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1)\sigma_2(\dot{\mathbf{p}}_2^n))). \quad (38)\end{aligned}$$

Assuming that the two tasks conflict, it follows that  $\mathbf{J}_1(\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1) = \mathbf{0}$  removing the last term, and by designing the saturation function such that  $s_{max,1} > \max\{\|\mathbf{J}_1^\dagger\|, \|\dot{\xi}_{1,d}\|\}$  it follows that  $\dot{V}_2 = -\tilde{\xi}_1^\top \sigma_1(\Lambda_1 \tilde{\xi}_1)$  which is negative definite. By applying standard Lyapunov arguments (*cf.* Khalil (2002)) it follows that the origin ( $\tilde{\xi}_1 = \mathbf{0}$ ) is UGAS when the tasks conflict, meaning that the first task is always fulfilled. A Lyapunov Function Candidate for the second task can be defined as  $V_3 = \frac{1}{2} \tilde{\xi}_2^\top \tilde{\xi}_2$  and by differentiating it and using that  $\dot{\xi}_2 = \mathbf{J}_2 \dot{\mathbf{p}}_r^n$ , we obtain

$$\dot{V}_3 = \tilde{\xi}_2^\top (\dot{\xi}_{2,d} - \mathbf{J}_2(\sigma_1(\dot{\mathbf{p}}_1^n) + (\mathbf{I} - \mathbf{J}_1^\dagger \mathbf{J}_1)\sigma_2(\dot{\mathbf{p}}_2^n))). \quad (39)$$

When the two tasks are not conflicting, the Jacobian matrices hold the property that  $\mathbf{J}_2 \mathbf{J}_1^\dagger = \mathbf{0}$  which can be used to reduce (39) to

$$\dot{V}_3 = \tilde{\xi}_2^\top (\dot{\xi}_{2,d} - \mathbf{J}_2 \sigma_1(\dot{\mathbf{p}}_1^n) - \mathbf{J}_2 \sigma_2(\dot{\mathbf{p}}_2^n)) \quad (40)$$

$$= \tilde{\xi}_2^\top (\dot{\xi}_{2,d} - \mathbf{J}_2 \sigma_2(\mathbf{J}_2^\dagger (\dot{\xi}_{2,d} + \Lambda_2 \tilde{\xi}_2))) \quad (41)$$

and by designing  $s_{max,2} > \max\{\|\mathbf{J}_2^\dagger\|, \|\dot{\xi}_{2,d}\|\}$  the Lyapunov derivative becomes

$$\dot{V}_3 = \tilde{\xi}_2^\top (\dot{\xi}_{2,d} - \sigma_2(\dot{\xi}_{2,d} + \Lambda_2 \tilde{\xi}_2)) = -\tilde{\xi}_2^\top \sigma_2(\Lambda_2 \tilde{\xi}_2) \quad (42)$$

which is negative definite and has a UGAS equilibrium when the tasks are not conflicting. To summarize, if the tasks are conflicting, the quadrotor will change its reference velocity in such a way that the tasks no longer conflict; and then it will continue to complete its lower-level task.

## 5.2 Tasks

*Collision Avoidance* The most critical task is to avoid collision between the quadrotors. A protective sphere can be placed around each of the quadrotors, and if the relative distance between any of the quadrotors are less than a constant  $d$ , conflicting velocity component from lower level tasks will be projected onto the null-space. Let the position of the obstacle be defined as  $\mathbf{p}_o^n$  and the position of a quadrotor be defined as  $\mathbf{p}^n$ . Let the task variable be defined as  $\xi_1 = d - \|\mathbf{p}^n - \mathbf{p}_o^n\|$  and a desired task variable as  $\xi_{1,d} = d$ . The Jacobian is found as  $\mathbf{J}_1 = \hat{\mathbf{r}}^\top$  where the unit vector is given as

$$\hat{\mathbf{r}} = \frac{\mathbf{p}^n - \mathbf{p}_o^n}{\|\mathbf{p}^n - \mathbf{p}_o^n\|} \quad (43)$$

which is aligned with the direction between the quadrotor and the obstacle. Using (33), the velocity vector for the collision avoidance scheme becomes

$$\dot{\mathbf{p}}_1^n = \mathbf{J}_1^\dagger \lambda_1 (d - \|\mathbf{p}^n - \mathbf{p}_o^n\|). \quad (44)$$

Note that the task of keeping a relative distance away from obstacles is a scalar task, such that  $\mathbf{J}_1 \in \mathbb{R}^3$ , and the gain  $\lambda_1$  becomes scalar.

*Move to a Desired Position* The second task can be chosen to move to a desired position. The task variable becomes  $\xi_2 = \mathbf{p}^n$ , while the desired variable becomes

Table 1. Initial and Desired Position Vectors

Quadrotor	Initial Position (m)	Desired Position (m)
$Q_1$	$[-50 \ 20 \ 0]^\top$	$[20 \ 10 \ 20]^\top$
$Q_2$	$[-50 \ -30 \ 0]^\top$	$[40 \ 10 \ 20]^\top$
$Q_3$	$[-50 \ 20 \ 0]^\top$	$[60 \ 30 \ 20]^\top$
$Q_4$	$[-50 \ 50 \ 0]^\top$	$[80 \ 10 \ 20]^\top$

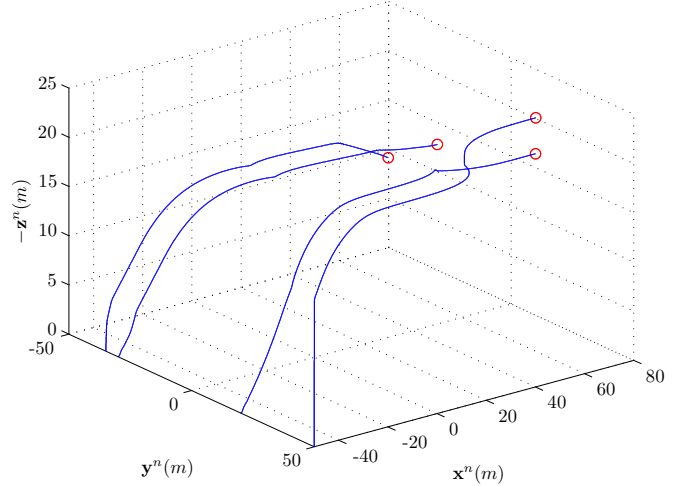


Fig. 5. Position Reconfiguration in East North Up Frame

$\xi_{2,d} = \mathbf{p}_d^n$ , where  $\mathbf{p}_d^n$  is a desired position. The Jacobian becomes  $\mathbf{J}_2 = \mathbf{I}$  such that the velocity vector becomes

$$\dot{\mathbf{p}}_2^n = \dot{\mathbf{p}}_d^n + \Lambda_2 (\mathbf{p}_d^n - \mathbf{p}^n) \quad (45)$$

which can be used for trajectory tracking. The saturated reference velocity is now found by inserting  $\mathbf{J}_1$ , (44) and (45) into (35).

## 6. SIMULATION

Consider four quadrotors, denoted  $Q_1, Q_2, Q_3, Q_4$  with initial and desired positions as shown in Table 1. The gains are chosen as  $\lambda_1 = 1$ ,  $\Lambda_2 = 0.2\mathbf{I}$ ,  $s_{max,1} = s_{max,2} = 2$ ,  $\Delta = 0.1$  m and the size of the protective sphere is set to  $d = 10$  m. As shown in Figure 5 all quadrotors are able to move to their desired positions. Quadrotors  $Q_1$  and  $Q_2$  begin with an initial relative distance of only  $5m$ , and as shown in Figure 6 they quickly move away to maintain a relative distance of  $10$  m or more. During the reconfiguration, the relative distance between any of the quadrotors never go below about  $10$  m except the initial case of  $Q_1$  and  $Q_2$ , and therefore no collisions occur.

As a comparison of the saturated NSB and the regular NSB method, consider a scenario of a quadrotor performing a position reconfiguration where the desired position is far away ( $1000$  m). Between the initial position and the final position there is an obstacle that must be avoided. The relative distance is shown in Figure 7, where the saturated NSB method is shown on the top, while the regular NSB method is shown at the bottom. The regular NSB method has a minimum relative distance of about  $3.5$  m due to high reference velocity, while the saturated NSB method has a minimum relative distance of about  $9.6$  m. Since velocity components are projected onto the null-space whenever the relative position is equal or less than  $d = 10$  m, it is natural that a very high relative velocity makes the

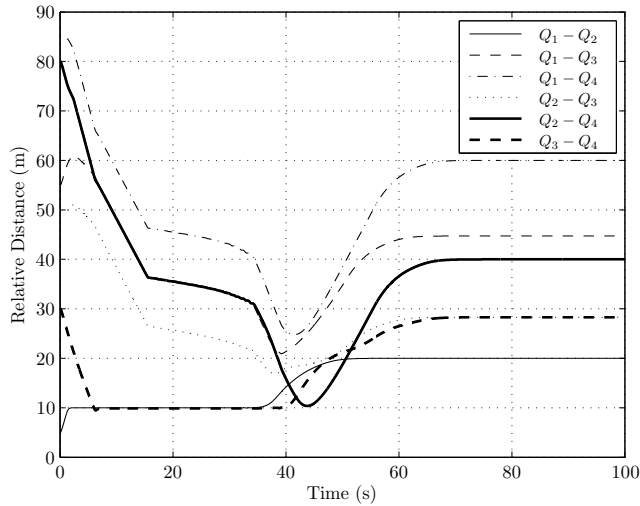


Fig. 6. Relative Distance between the Quadrotors

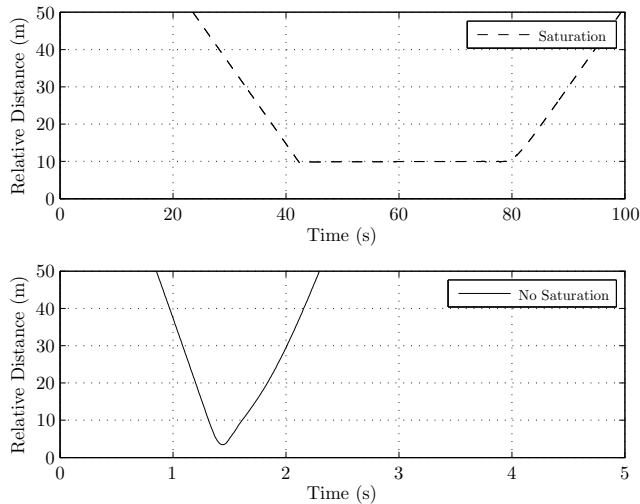


Fig. 7. Relative Distance with a Static Obstacle

quadrotor move close to the obstacle before the conflicting velocity components are removed. To be fair it is not a problem with the NSB method itself, but a problem of using relative distance as a measurement of when to switch on the collision avoidance method. The same problem exists for other collision avoidance methods, and can be remedied by using saturation functions as in this paper or by incorporating the relative velocity when checking for collisions. This has motivated research such as Schlanbusch and Oland (2013) where the relative velocity vector is used to increase the size of the protective sphere whenever an agent is on collision course with an obstacle, and maintains a minimum size whenever the agent is moving parallel or away from the obstacle.

## 7. CONCLUSION

In this paper we have derived a saturated controller for an underactuated quadrotor which is shown to be UGAS and validated by tracking a circle. With basis in the NSB behavioral control method, a saturated NSB method was derived and validated using a case of four quadrotors that performed a formation reconfiguration without collisions.

## REFERENCES

- Antonelli, G., Arrichiello, F., and Chiaverini, S. (2005). The Null-Space-Based Behavioral Control for Mobile Robots. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*.
- Antonelli, G., Arrichiello, F., and Chiaverini, S. (2009). Experiments of Formation Control With Multirobot Systems Using the Null-Space-Based Behavioral Control. *IEEE Transaction on Control Systems Technology*, Vol. 17, No. 5, 1173–1182.
- Arrichiello, F., Chiaverini, S., and Fossen, T. (2006). Formation Control of Underactuated Surface Vessels using the Null-Space-Based Behavioral Control. In *International Conference on Intelligent Robot and Systems*.
- Bouabdallah, S. and Siegwart, R. (2005). Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor. In *Proceedings of the IEEE International Conference on Robotics and Automation*.
- Castillo, P., Lozano, R., and Dzul, A. (2005). Stabilization of a Mini Rotorcraft with Four Rotors. *IEEE Control Systems Magazine*, Vol. 25, No. 6, 45–55.
- Cunha, R., Cabecinhas, D., and Silvestre, C. (2009). Nonlinear trajectory tracking control of a quadrotor vehicle. In *Proceedings of the 10th European Control Conference, Budapest, Hungary*.
- Grzonka, S., Grisetti, G., and Burgard, W. (2012). A Fully Autonomous Indoor Quadrotor. *IEEE Transactions on Robotics*, Vol. 28, No. 1, 90–100.
- Guerrero, J.A., Castillo, P., Salazar, S., and Lozano, R. (2012). Mini Rotorcraft Flight Formation Control Using Bounded Inputs. *Journal of Intelligent and Robotic Systems*, Vol 65, 175–186.
- Guisser, M. and Medromi, H. (2009). A High Gain Observer and Sliding Mode Controller for an Autonomous Quadrotor Helicopter. *International Journal of Intelligent Control and Systems*, Vol. 14, No. 3, 204–212.
- Hahn, W. (1967). *Stability of Motion*. Springer-Verlag Berlin Heidelberg New York.
- Khalil, H.K. (2002). *Nonlinear systems*. Prentice Hall, 3rd edition.
- Khatib, O. (1986). Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *The International Journal of Robotics Research*, Vol 5. No. 1, 90–98.
- Oland, E., Kristiansen, R., and Nicklasson, P.J. (2010). Spacecraft Formation Reconfiguration with Plume Avoidance. In *Proceedings of the IEEE Aerospace Conference, Big Sky, Montana*.
- Reyhanoglu, M., van der Schaft, A., McClamroch, N.H., and Kolmanovsky, I. (1999). Dynamics and control of a class of underactuated mechanical systems. *IEEE Transactions on Automatic Control*, Vol. 44, No. 9, 1663–1671.
- Schlanbusch, R., Kristiansen, R., and Nicklasson, P.J. (2011). Spacecraft formation reconfiguration with collision avoidance. *Automatica*, Vol. 47, No. 7, 1443–1449.
- Schlanbusch, R. and Oland, E. (2013). Spacecraft Formation Reconfiguration with Dynamic Collision Avoidance. In *Proceedings of the IEEE Aerospace Conference, Big Sky, Montana*.
- Teel, A.R. (1992). Global Stabilization and restricted tracking for multiple integrators with bounded controls. *Systems & Control Letters*, 18, 165–171.