Nonlinear State-Dependent Riccati Equation Control of a Quadrotor UAV

Holger Voos

***Abstract*— Small quadrotor UAVs represent a very interesting class of small flying robots because of their ability to fly in- and outdoor. Therefore, these vehicles have an enormous potential for near-area surveillance and exploration. However, especially indoor flight is a difficult task for vehicle control which has to stabilize the desired velocity vector and the required attitude of the quadrotor. This paper mainly describes the development of such a nonlinear vehicle control system based on state-dependent Riccati equations (SDRE). The controller is integrated in an overall mission system concept for UAVs.**

I. INTRODUCTION

Unmanned flying vehicles (UAVs) have gained increasing interest in recent years because of a wide area of possible military and commercial applications. A very promising class of UAVs comprises small vehicles that can be flown either in- and outdoor and offer the possibility for near-area surveillance and search-and-rescue. However, indoor flight comes up with some very challenging requirements in terms of size, weight and maneuverability of the vehicle that rule out most of the aircraft types, see [1] for a very good overview. One type of aircraft with a strong potential also for indoor flight is the rotorcraft and the so called quadrotor has been chosen by many researchers as the most promising vehicle, see e.g. [1], [2] and [3].

The quadrotor can be described as a vehicle with four propellers in a cross configuration. While the front and the rear motor rotate clockwise, the left and the right motor rotate counter-clockwise which nearly cancels gyroscopic effects and aerodynamic torques in trimmed flight. One additional advantage of the quadrotor compared to a conventional helicopter is the simplified rotor mechanics. By varying the speed of the single motors, the lift force can be changed and vertical and/or lateral motion can be created. Pitch movement is generated by a difference between the speed of the front and the rear motor while roll movement results from differences between the speed of the left and right rotor, respectively. Yaw rotation results from the difference in the counter-torque between each pair (front-rear and left-right) of rotors. The overall thrust is the sum of the thrusts generated by the four single rotors. In spite of the four actuators, the quadrotor is a dynamically unstable system that has to be stabilized by a suitable control system in order to operate autonomously.

There are some contributions in the literature that are concerned with control system design for quadrotor vehicles, see e.g. [1], [2] and [3] to mention only a few. Many of the proposed control systems are based on a linearized model and conventional PID- or state space control while other approaches apply sliding-mode or H_∞ control. In this paper we will take into account the nonlinear dynamics of the quadrotor and develop a controller based on state-dependent Riccati equations (SDRE). The basic idea of the SDRE-approach is proposed in [5] and has been applied to a number of control problems also in aerospace applications, see e.g. [5], [8], but not yet to the control of a small quadrotor UAV.

II. DYNAMIC MODEL OF THE QUADROTOR

The general dynamic model of a quadrotor has been presented in a number of papers and will not be discussed here in all details again. For further considerations of modelling, we refer to [1] and [3]. We consider an inertial frame and a body fixed frame whose origin is in the center of mass of the quadrotor as shown in Fig. 1.

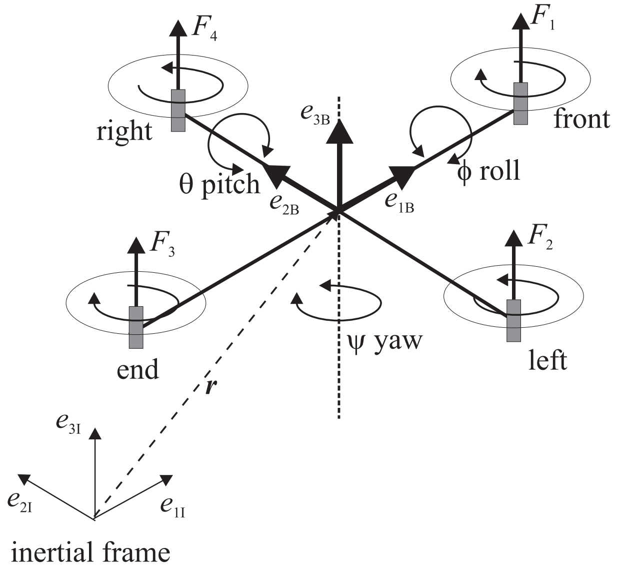

Fig. 1. Configuration, inertial and body fixed frame of the quadrotor.

The orientation of the quadrotor is given by the three Euler angles, namely yaw angle ψ, pitch angle θ and roll angle ϕ that together form the vector $\boldsymbol{\Omega}^T = (\phi, \theta, \psi)$. The position of the vehicle in the inertial frame is given by the vector $\boldsymbol{r}^T = (x, y, z)$. The transformation of vectors from the body fixed frame to the inertial frame is given by the rotation matrix $\boldsymbol{R}$ where c_θ for example denotes $\cos\theta$ and s_θ denotes

Prof. Dr.-Ing. H. Voos is with the Faculty of Electrical Engineering and Computer Science, University of Applied Sciences Ravensburg-Weingarten, D-88241 Weingarten, P.O.-Box 1261, Germany voos@hs-weingarten.de

$\sin\theta$:

$$\boldsymbol{R} = \begin{pmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi s_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{pmatrix} \quad (1)$$

Since the thrust force generated by rotor $i, i = 1, 2, 3, 4$ is $F_i = b \cdot \omega_i^2$ where b is the thrust factor and ω_i is the speed of rotor i, we obtain a first set of differential equations that describe the acceleration of the quadrotor:

$$\ddot{\boldsymbol{r}} = -g \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} + \boldsymbol{R} \cdot b/m \sum_{i=1}^{4} \omega_i^2 \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (2)$$

With the inertia matrix $\boldsymbol{I}$ (which is a diagonal matrix with the inertias I_x, I_y and I_z on the main diagonal), the rotor inertia J_R and the vector $\boldsymbol{\tau}$ that describes the torque applied to the vehicle's body we obtain a second set of differential equations:

$$\boldsymbol{I}\ddot{\boldsymbol{\Omega}} = -\dot{\boldsymbol{\Omega}} \times \boldsymbol{I}\dot{\boldsymbol{\Omega}} - \sum_{i=1}^{4} J_R(\dot{\boldsymbol{\Omega}} \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix})\omega_i + \boldsymbol{\tau} \quad (3)$$

The vector $\boldsymbol{\tau}$ is defined as

$$\boldsymbol{\tau} = \begin{pmatrix} lb(\omega_4^2 - \omega_2^2) \\ lb(\omega_3^2 - \omega_1^2) \\ d(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \end{pmatrix} \quad (4)$$

with the drag factor d and the length l of the lever. With a renaming of the inputs as

$$\begin{aligned} u_1 &= b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \\ u_2 &= b(\omega_4^2 - \omega_2^2) \\ u_3 &= b(\omega_3^2 - \omega_1^2) \\ u_4 &= d(\omega_2^2 + \omega_4^2 - \omega_1^2 - \omega_3^2) \end{aligned} \quad (5)$$

and the definition of a new variable

$$\omega_d = \omega_2 + \omega_4 - \omega_1 - \omega_3 \quad (6)$$

evaluation of (2) and (3) yields the overall dynamic model in the following form

$$\begin{aligned} \ddot{x} &= (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) \cdot u_1/m \\ \ddot{y} &= (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) \cdot u_1/m \\ \ddot{z} &= -g + (\cos\phi \cos\theta) \cdot u_1/m \\ \ddot{\phi} &= \dot{\theta}\dot{\psi}(\frac{I_y - I_z}{I_x}) - \frac{J_R}{I_x}\dot{\theta}\omega_d + \frac{l}{I_x}u_2 \\ \ddot{\theta} &= \dot{\phi}\dot{\psi}(\frac{I_z - I_x}{I_y}) + \frac{J_R}{I_y}\dot{\phi}\omega_d + \frac{l}{I_y}u_3 \\ \ddot{\psi} &= \dot{\phi}\dot{\theta}(\frac{I_x - I_y}{I_z}) + \frac{l}{I_z}u_4 \end{aligned} \quad (7)$$

This model can be rewritten in state-space form $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ where $\boldsymbol{u}^T = (u_1, u_2, u_3, u_4)$ is the vector of input variables given in (5) and $\boldsymbol{x} \in \mathbb{R}^{12}$ is the vector of state variables given as follows:

$$\boldsymbol{x}^T = (x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}) \quad (8)$$

From (7) and (8) we obtain

$$\dot{\boldsymbol{x}} = \begin{pmatrix} x_2 \\ (\cos x_7 \sin x_9 \cos x_{11} + \sin x_7 \sin x_{11}) \cdot u_1/m \\ x_4 \\ (\cos x_7 \sin x_9 \sin x_{11} - \sin x_7 \cos x_{11}) \cdot u_1/m \\ x_6 \\ -g + (\cos x_7 \cos x_9) \cdot u_1/m \\ x_8 \\ x_{12}x_{10}I_1 - \frac{J_R}{I_x}x_{10}\omega_d + \frac{l}{I_x}u_2 \\ x_{10} \\ x_{12}x_8I_2 + \frac{J_R}{I_y}x_8\omega_d + \frac{l}{I_y}u_3 \\ x_{12} \\ x_{10}x_8I_3 + \frac{l}{I_z}u_4 \end{pmatrix} \quad (9)$$

with the abbreviations $I_1 = (I_y - I_z)/I_x$, $I_2 = (I_z - I_x)/I_y$ and $I_3 = (I_x - I_y)/I_z$.

It becomes obvious that the state space model can be decomposed into one subset of differential equations that describes the dynamics of the attitude (i.e. the angles) and one subset that describes the translation of the UAV. From (9) we obtain the first subset of differential equations, called submodel M_1 as

$$\begin{pmatrix} \dot{x}_8 \\ \dot{x}_{10} \\ \dot{x}_{12} \end{pmatrix} = \begin{pmatrix} x_{12}x_{10}I_1 - \frac{J_R}{I_x}x_{10}\omega_d + \frac{l}{I_x}u_2 \\ x_{12}x_8I_2 + \frac{J_R}{I_y}x_8\omega_d + \frac{l}{I_y}u_3 \\ x_{10}x_8I_3 + \frac{l}{I_z}u_4 \end{pmatrix} \quad (10)$$

The input variables of that first submodel are the variables u_2, u_3 and u_4. The variable ω_d is looked upon as a measurable disturbance here. From the angular rates as the output of M_1 the angles are obtained by pure integration. The three angles (or state variables x_7, x_9 and x_{11}) are the inputs of the next submodel M_2 which is given by

$$\begin{pmatrix} \dot{x}_2 \\ \dot{x}_4 \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} (\cos x_7 \sin x_9 \cos x_{11} + \sin x_7 \sin x_{11})\frac{u_1}{m} \\ (\cos x_7 \sin x_9 \sin x_{11} - \sin x_7 \cos x_{11})\frac{u_1}{m} \\ -g + (\cos x_7 \cos x_9)\frac{u_1}{m} \end{pmatrix} \quad (11)$$

Herein, the controllable input is the variable u_1. The resulting structure of the quadrotor model is shown in Fig. 2.

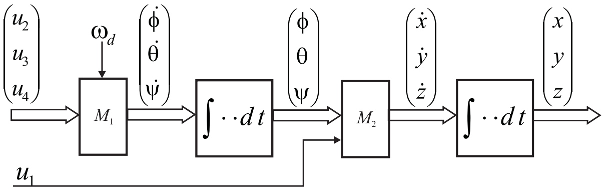

Fig. 2. Decomposed structure of the quadrotor model.

III. OVERALL UAV CONTROL SYSTEM

When considering present available UAV systems (also including small size UAVs), most of them have a similar system structure. In general, a UAV system comprises a control station with man-machine interface and human operator, a bidirectional data link and the unmanned vehicle or UAV itself, see also [4]. In most applications, the vehicle is remotely controlled by the human operator at the control station. In this case, the data link is not only used for data transmission from the vehicle to the control station but also for the transmission of the operator's control commands.

From a control engineering point of view, a UAV system contains two main control loops [4]. The first main control loop is the vehicle control loop. Even if a UAV is under remote control, the human operator is most often not able to overtake all tasks of an aircraft pilot because of limited situation awareness and low reaction times. Therefore, the vehicle control system is necessary for UAVs of all dimensions regardless of the UAV's task or mission. The second main loop is the mission control loop that comprises the stabilized vehicle as a platform for mission related sensors and actuators and the mission control system. The functionality of the mission control system might be distributed between on-board functions and functions performed by the human operator at the control station. Therefore this control loop is at least partially closed via data link. The mission control system commands desired values to the vehicle control loop, uses the mission related sensors to collect environmental information and commands the available mission related actuators.

Even if this paper mainly deals with the development of the vehicle control system of the quadrotor UAV, also the mission control system and the available sensors must be considered in order to define the structure of the vehicle control. Our UAV system is equipped with an inertial measurement unit (IMU) and a GPS system. The IMU is used together with a Kalman Filter to measure the angular rates, the angles and the current velocities. The remote pilot or the onboard mission system commands a desired velocity vector to the vehicle control in order to guide the UAV on a desired flight path. Direct position control as proposed in many papers (see e.g. [2], [3]) is most often not necessary for vehicle guidance and position measurement or estimation is most often not accurate enough for direct feedback control. The default command from the mission system (or in cases where the remote pilot stops acting) is the zero velocity vector, i.e. the quadrotor UAV should hover at the current position. The GPS is mainly used by the mission system for supervision of the flight path and the calculation of the desired velocity vector. The overall UAV control structure is shown in Fig. 3. In this paper the main focus is on the vehicle control system as described in the following sections.

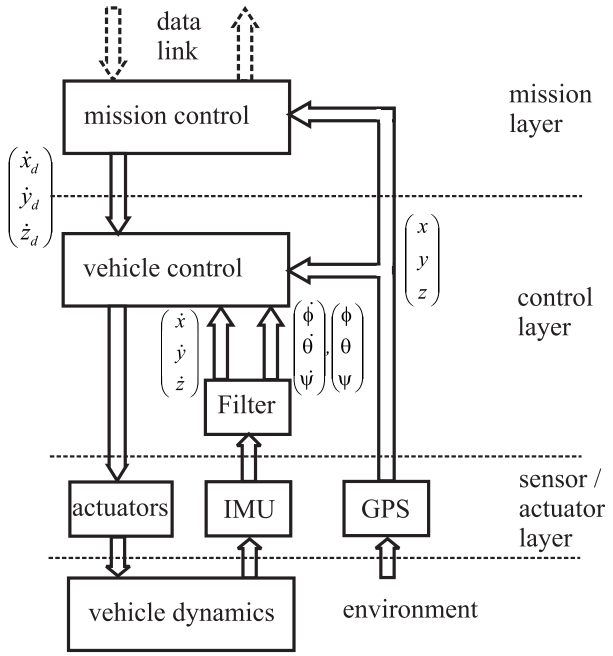
data
link
mission control
mission
layer
$\begin{pmatrix}\dot{x}_d\\ \dot{y}_d\\ \dot{z}_d\end{pmatrix}$
vehicle control
$\begin{pmatrix}x\\ y\\ z\end{pmatrix}$
control
layer
$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{pmatrix}$
$\begin{pmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{pmatrix}, \begin{pmatrix}\phi\\ \theta\\ \psi\end{pmatrix}$
Filter
sensor /
actuator
layer
actuators
IMU
GPS
vehicle dynamics
environment

Fig. 3. Overall UAV control structure.

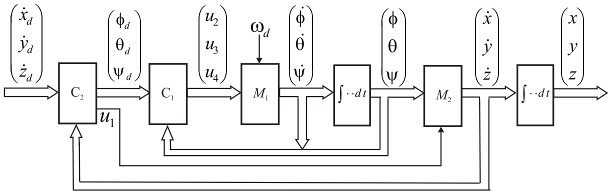
$\begin{pmatrix}\dot{x}_d\\ \dot{y}_d\\ \dot{z}_d\end{pmatrix}$
C₂
$\begin{pmatrix}\phi_d\\ \theta_d\\ \psi_d\end{pmatrix}$
u_1
C₁
$\begin{pmatrix}u_2\\ u_3\\ u_4\end{pmatrix}$
ω_d
M_1
$\begin{pmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{pmatrix}$
$\int\cdot\cdot dt$
$\begin{pmatrix}\phi\\ \theta\\ \psi\end{pmatrix}$
M_2
$\begin{pmatrix}\dot{x}\\ \dot{y}\\ \dot{z}\end{pmatrix}$
$\int\cdot\cdot dt$
$\begin{pmatrix}x\\ y\\ z\end{pmatrix}$

Fig. 4. Nested structure of the UAV vehicle control.

IV. VEHICLE CONTROLLER DESIGN

The decomposed model structure as shown in Fig. 2 already suggests a nested structure for vehicle control. In order to achieve and maintain a desired velocity vector, first the necessary attitude of the UAV has to be stabilized. Therefore, we propose a decomposition of the control system in an outer-loop velocity control and an inner-loop attitude control. In this structure, the inner attitude control loop has to be much faster than the outer loop and stabilizes the desired angles that are commanded by the outer loop. This nested structure is shown in Fig. 4. First we consider the inner control loop with controller C_1, the attitude control loop, that has to stabilize the desired roll, pitch and yaw angle, i.e. the desired vector $\boldsymbol{\Omega}_d^T = (\phi_d, \theta_d, \psi_d) = (x_{7,d}, x_{9,d}, x_{11,d})$. The corresponding dynamic model comprises the last six equations of the state space model (8) which is a series of the nonlinear submodel M_1 and an integrator. Then we derive the outer-loop controller C_2 to stabilize a desired velocity vector.

A. State-dependent Riccati Equation Control

The state-dependent Riccati equation (SDRE) control was initially derived by Cloutier, see [5] for an overview. The basic idea was motivated by linear quadratic regulation and introduces a factorization of a nonlinear system in a way that it becomes linear at any fixed state

$$\dot{\boldsymbol{x}} = \boldsymbol{A}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} \qquad (12)$$

where the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ both depends on the current state variables. The controllability issues of such methods are discussed in [5]. Control gains at any state $\boldsymbol{x}$ can be calculated using standard linear optimal control theory, i.e. choosing that control that minimizes the cost function

$$J = 0.5 \int_{t_0}^{\infty} \boldsymbol{x}^T \boldsymbol{Q}(\boldsymbol{x})\boldsymbol{x} + \boldsymbol{u}^T \boldsymbol{R}(\boldsymbol{x})\boldsymbol{u} dt \qquad (13)$$

where $\boldsymbol{Q}(\boldsymbol{x})$ penalizes the state and $\boldsymbol{R}(\boldsymbol{x})$ penalizes control effort. By solving the algebraic Riccati equation

$$\boldsymbol{A}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{A} + \boldsymbol{Q} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P} = \boldsymbol{0} \qquad (14)$$

we obtain the matrix $\boldsymbol{P}(\boldsymbol{x})$ and the control gains become

$$\boldsymbol{u} = -\boldsymbol{K}(\boldsymbol{x})\boldsymbol{x} = -\boldsymbol{R}(\boldsymbol{x})^{-1}\boldsymbol{B}^T(\boldsymbol{x})\boldsymbol{P}(\boldsymbol{x})\boldsymbol{x} \qquad (15)$$

In general, this technique requires that the algebraic Riccati equation must be solved at every state and therefore also the control gains have to be recalculated at every state. This seems to be computationally complex, but [8] developed some real-time methods that can be implemented on an embedded microcontroller. In addition, stability of the SDRE approach was shown in [6], [7].

B. Attitude Control using SDRE

We apply the SDRE method to the attitude control problem. The vector of state variables for that problem is given by $\boldsymbol{x}_I^T = (x_7, x_8, x_9, x_{10}, x_{11}, x_{12})$ while the vector of input variables is $\boldsymbol{u}_I^T = (u_2, u_3, u_4)$ and one state-dependent model can be obtained from (9) by factorization as

$$\dot{\boldsymbol{x}}_I = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{12}I_1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & x_{12}I_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & x_8 I_3 & 0 & 0 \end{pmatrix} \cdot \boldsymbol{x}_I$$
$$+ \begin{pmatrix} 0 & 0 & 0 \\ \frac{l}{I_x} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \frac{l}{I_y} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{l}{I_z} \end{pmatrix} \cdot \boldsymbol{u}_I + \begin{pmatrix} 0 \\ -\frac{J_R}{I_x}x_{10} \\ 0 \\ \frac{J_R}{I_y}x_8 \\ 0 \\ 0 \end{pmatrix} \cdot \omega_d \qquad (16)$$

Please note that this factorization is not unique as also discussed in [5]. In (16) only the matrix $\boldsymbol{A}(\boldsymbol{x}_I)$ is state-dependent while the matrix $\boldsymbol{B}$ is a constant matrix.

Using (16), the control gain matrix $\boldsymbol{K}(\boldsymbol{x}_I)$ can be calculated, while the overall control input $\boldsymbol{u}_I$ must also take into account the measurable disturbance ω_d as well as the fact that a desired state $\boldsymbol{x}_{I,d}$ given by the outer velocity control loop must be stabilized. The measurable disturbance can be compensated using a compensation

$$\boldsymbol{u}_c = -\left(\boldsymbol{B}^T\boldsymbol{B}\right)^{-1}\boldsymbol{B} \cdot \begin{pmatrix} 0 \\ -\frac{J_R}{I_x}x_{10} \\ 0 \\ \frac{J_R}{I_y}x_8 \\ 0 \\ 0 \end{pmatrix} \cdot \omega_d \qquad (17)$$

The stabilization of a desired state which is not zero can be guaranteed by a pre-filter matrix $\boldsymbol{M}(\boldsymbol{x}_I)$ assuming that the control gain matrix $\boldsymbol{K}(\boldsymbol{x}_I)$ is already determined:

$$\boldsymbol{M}(\boldsymbol{x}_I) = \mathrm{pinv}\big((\boldsymbol{B}\boldsymbol{K}(\boldsymbol{x}_I) - \boldsymbol{A}(\boldsymbol{x}_I))^{-1}\boldsymbol{B}\big) \qquad (18)$$

where pinv() denotes the pseudo-inverse of a non-quadratic matrix. The overall attitude control law, i.e. the controller C_1 can then be summarized as follows:

$$\boldsymbol{u}_I = -\boldsymbol{K}(\boldsymbol{x}_I)\boldsymbol{x}_I + \boldsymbol{M}(\boldsymbol{x}_I)\boldsymbol{x}_{I,d} + \boldsymbol{u}_c \qquad (19)$$

C. Velocity Control

If the inner-loop attitude control is sufficiently fast, we can assume that a desired value of the roll, pitch and yaw angle is achieved very fast with respect to the outer velocity control loop. Therefore the closed inner control loop can approximately be considered as a static block that just transfers the desired values of roll, pitch and yaw angle to the next model M_2, resulting in the structure given in Fig. 5.

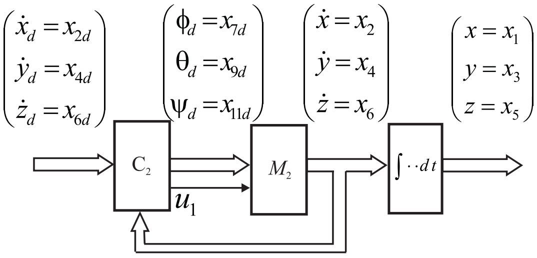

Fig. 5. Structure of the outer (velocity) control loop.

According to (11), we can describe model M_2 by the following set of nonlinear differential equations:

$$\begin{aligned} \dot{x}_2 &= (\cos x_{7d}\sin x_{9d}\cos x_{11d} + \sin x_{7d}\sin x_{11d}) \cdot u_1/m \\ \dot{x}_4 &= (\cos x_{7d}\sin x_{9d}\sin x_{11d} - \sin x_{7d}\cos x_{11d}) \cdot u_1/m \\ \dot{x}_6 &= \cos x_{7d}\cos x_{9d} \cdot u_1/m - g \end{aligned} \qquad (20)$$

where all x_{7d}, x_{9d}, x_{11d} and u_1 are input variables. If we assume that the angles, i.e. the input variables x_{7d}, x_{9d}, x_{11d} are sufficiently small, the trigonometric functions can be replaced by $\sin\alpha \approx \alpha$ and $\cos\alpha \approx 1$. In addition, we assume the existence of three independent ”artificial” input variables $\tilde{u}_1, \tilde{u}_2, \tilde{u}_3$ that depend on the other four input variables, which yields

$$\begin{aligned} \dot{x}_2 &\approx (x_{9d} + x_{7d}\cdot x_{11d}) \cdot u_1/m = \tilde{u}_1 \\ \dot{x}_4 &\approx (x_{9d}\cdot x_{11d} - x_{7d}) \cdot u_1/m = \tilde{u}_2 \\ \dot{x}_6 &\approx u_1/m - g = \tilde{u}_3 \end{aligned} \qquad (21)$$

Concerning the new input variables, the control task is very simple since it comprises the control of three independent systems of first order which might be solved by a pure proportional controller, respectively:

$$\begin{aligned} \tilde{u}_1 &= k_1 \cdot (x_{2d} - x_2) \\ \tilde{u}_2 &= k_2 \cdot (x_{4d} - x_4) \\ \tilde{u}_3 &= k_3 \cdot (x_{6d} - x_6) \end{aligned} \qquad (22)$$

Herein the controller parameters k_1, k_2 and k_3 could be chosen in a way that the outer loop is sufficiently fast but not too fast with respect to the inner loop attitude control.

Since $\tilde{u}_1, \tilde{u}_2, \tilde{u}_3$ are the "artificial" input variables, the real input variables x_{7d}, x_{9d}, x_{11d} and u_1 must be calculated with the help of (21). Here we have three equations with four unknowns that gives us the possibility to choose $x_{11d} = \psi_d = 0$. That means that the quadrotor is steered without yaw rotation which is not necessary for achieving a desired velocity vector. Thus (21) can be rewritten as

$$\begin{aligned} x_{9d} \cdot u_1/m &= \tilde{u}_1 \\ -x_{7d} \cdot u_1/m &= \tilde{u}_2 \\ u_1/m - g &= \tilde{u}_3 \end{aligned} \tag{23}$$

The set of equations (23) can be solved to obtain the desired input variables:

$$\begin{aligned} u_1 &= m \cdot (\tilde{u}_3 + g) \\ x_{7d} &= \frac{-\tilde{u}_2}{\tilde{u}_3 + g} \\ x_{9d} &= \frac{\tilde{u}_1}{\tilde{u}_3 + g} \\ x_{11d} &= 0 \end{aligned} \tag{24}$$

where $\tilde{u}_1, \tilde{u}_2, \tilde{u}_3$ are calculated using (22).

V. SIMULATION RESULTS

The quadrotor model and the control algorithms derived as explained is implemented and simulated in Matlab/Simulink. For that purpose, the parameters of a real quadrotor are identified and inserted in the simulation model. The quadrotor is modelled using the nonlinear equations as given in (9). The parameter of the outer velocity control loop are chosen as $k_1 = k_2 = k_3 = 1$, obtained from experiments.

First a pure attitude control task is simulated. Here we start with a deviation of the three angles $\phi = 10°, \theta = 10°$ and $\psi = 10°$. The desired state which has to be achieved by the control action is the hovering state where all angles and all velocities are zero. The simulation result is shown in Fig. 6 where the time plot of all angles is presented. It becomes obvious that after a short transition phase all angles are stabilized at the required value of zero. During that compensation of the initial disturbances of the angles, the quadrotor changes the position until a new hovering position is reached and stabilized. This movement at the beginning cannot be avoided since there is also an initial horizontal acceleration because of the deviation of the attitude. The movement of the quadrotor during attitude control is shown in Fig. 7.

In a next simulation, inner-loop attitude control and outer-loop velocity control is applied. Again we start with an initial deviation of the three angles $\phi = 10°, \theta = 10°$ and $\psi = 10°$. The task now is to achieve and stabilize a velocity vector in pure x-direction with $\dot{x}_d = 1$ m/sec. The time plot of all velocities is presented in Fig. 8 while the position in three dimensions is shown in Fig. 9. Again, the desired state is achieved after a short transition phase and the quadrotor is moving with constant velocity in pure x-direction. During that constant flight the angles are also kept constant and the required attitude for that desired movement is maintained.

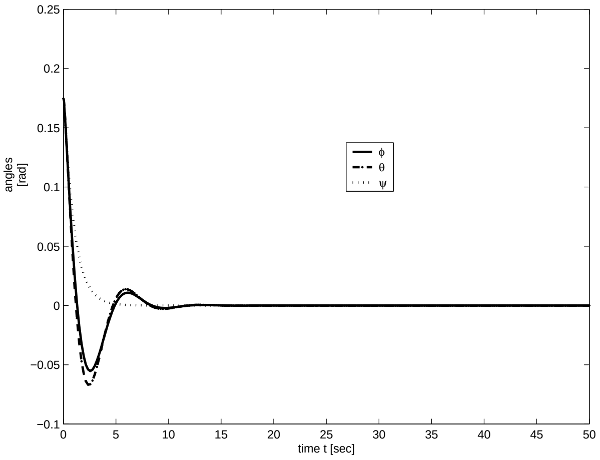

Fig. 6. Time plot of the angles during attitude control.

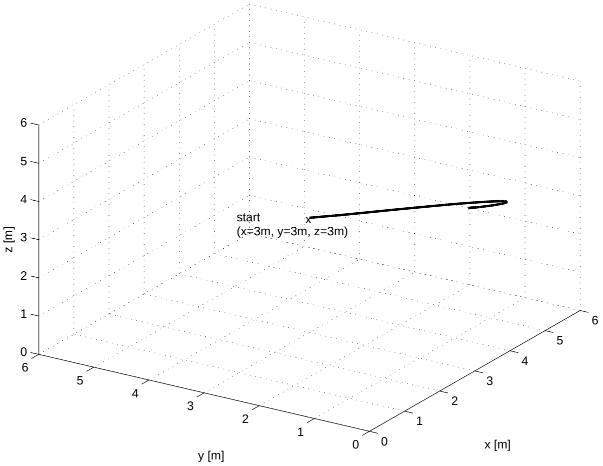

Fig. 7. Plot of position during attitude control.

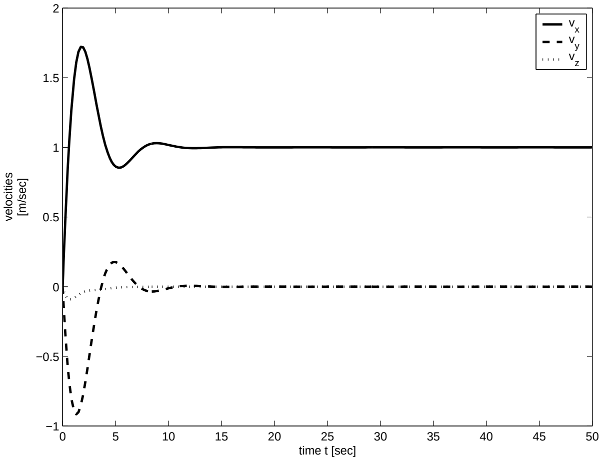

Fig. 8. Time plot of velocities during velocity control.

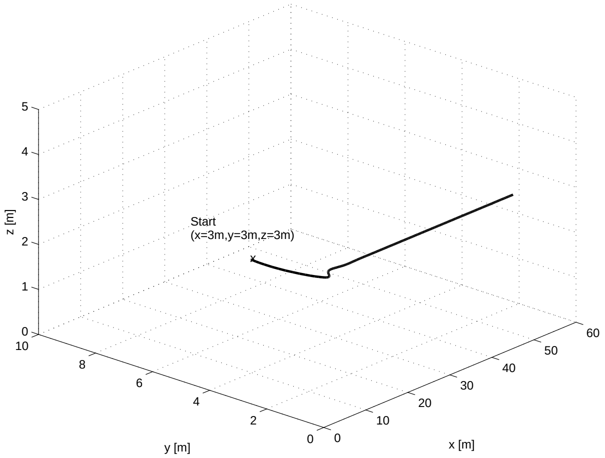

Fig. 9. Plot of position during velocity control.

VI. CONCLUSIONS AND FUTURE WORKS

This paper presents a vehicle control system for a small quadrotor UAV based on the state-dependent Riccati equation controller (SDRE). Both an inner-loop attitude controller and an outer-loop velocity controller are developed and embedded in an overall mission control concept for UAVs. The dynamic model of the quadrotor is derived and implemented in a Matlab/Simulink simulation model. With the help of that simulation, the nonlinear vehicle control system is tested and its efficiency demonstrated. In our ongoing work this control concept will be implemented in an embedded system in the real quadrotor and the overall mission system will be realized.

REFERENCES

[1] S. Bouabdallah, P. Murrieri, R. Siegwart, “Design and Control of an Indoor Micro Quadrotor”, *in Proc. of the Int. Conf. on Robotics and Automation ICRA’2004*, New Orleans, USA, 2004.

[2] A. Tayebi, S. McGilvray, Attitude stabilization of a four-rotor aerial robot, *in Proc. of 43rd IEEE Conf. on Decision and Control*, Atlantis, Paradise Island, Bahamas, 2004.

[3] P. Castillo, A. Dzul, R. Lozano, Real-time stabilization and tracking of a four-rotor mini rotorcraft, *IEEE Trans. on Control Systems Technology*, VOL.12, No. 4, July 2004, pp. 510 - 516.

[4] H. Voos, Autonomous Systems Approach to UAVs, *in Proc. of the 18th Bristol International Conference on Unmanned Air Vehicle Systems*, Bristol, UK, 2003.

[5] J.R. Cloutier, “State-Dependent Riccati Equation Techniques: An Overview”, *Proc. of the 1997 American Control Conference*, June 1997, Albuquerque, NM.

[6] Y. Zhang, S. Agrawal, P. Hemanshu, M. Piovoso, Optimal Control using State Dependent Riccati Equation (SDRE) for a Flexible Cable Transporter System with Arbitrarily Varying Lengths, *Proc. of the 2005 IEEE Conference on Control Applications*, Toronto, Canada, August 2005, pp. 1063 - 1068.

[7] C. Willard, B. Randal, “Ensuring Stability of State-dependent Riccati Equation Controllers Via Satisficing”, *Proc. of the 41st IEEE Conference on Decision and Control*, Las Vegas, Nevada USA, Dec. 2002, pp. 2645-2650.

[8] P.K. Menon, T. Lam, L.S. Crawford, V.H. Cheng, “Real-time Computational Methods for SDRE Nonlinear Control of Missiles”, *Proc. of the 2002 American Control Conference*, May 2002, Anchorage, AK.