Description of Algorithms for Simulation of the Autonomous Model Helicopter MARVIN

Carsten Deeg

17th November 2003

Technische Universität Berlin
Real-Time Systems & Robotics
`http://pdv.cs.tu-berlin.de/MARVIN/`

1. Introduction

Controlling a helicopter requires specialized algorithms, which permanently have to stabilize the vehicle, as it is inherently instable. To be able to test these algorithms "offline", before the real helicopter is used, a software simulator is needed. The simulator described in this paper is designed especially for the autonomous helicopter MARVIN (**M**ulti-purpose **A**erial **R**obot **V**ehicle with **I**ntelligent **N**avigation) contructed at TU Berlin. It can be used to validate changes to the flight controller as well as changes to the software of the IMU (see below).

2. Basic Concept

2.1. Application Flow

The whole simulation combines several components of the MARVIN system:

Controller This is the part of software, which makes MARVIN fly.

IMU The **I**nertial **M**easurement **U**nit and the on-board GPS are used by the IMU software to tell the controller where the helicopter is and how it is oriented. For simulation the IMU software is used and the hardware sensors are emulated (see below).

Sensors Sensors produce the input for the IMU software. For simulation they are emulated including their problems and incorrect measurements.

Simulator This is the part, which simulates all the physics of a flying helicopter. In the following text the word *simulator* denotes this software part.

For the first two components (controller and IMU) the same code is used as for real flights.

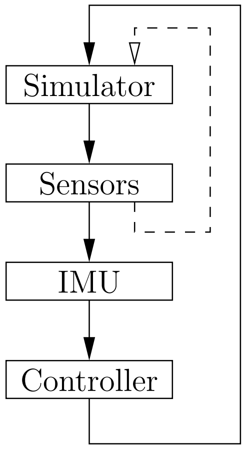

Figure 1: Application flow

During the simulation the output of the simulator is used to produce the input for the IMU software (see figure 1). Special characteristics of the sensors are simulated. Based on these "measurements" the IMU software computes position, orientation etc. Then the controller uses the output of the IMU software to control the helicopter. The calculated servo values are returned to the simulator to produce the results for the next time step.

Since the controller is designed for a fixed time step of $0.05s$, this cannot be changed for simulation. To improve accuracy of simulation, it is possible to run the simulator more frequently than the controller. In figure 1, this is indicated by the dashed line. See section 4.1 for details.

2.2. Helicopter State

The state of the helicopter describes the location, velocity and all the other values that change over time and which are needed for calculating future states. The following table shows all these information, stored in a struct called `SimState`:

$\vec{s} = (\; x \;\; y \;\; z \;)^T$	Position in BCS (see below)
$\vec{v} = (\; v_x \;\; v_y \;\; v_z \;)^T$	Velocity in BCS
$\vec{a} = (\; a_x \;\; a_y \;\; a_z \;)^T$	Acceleration in BCS (for sensors only)
$\vec{\varphi} = (\; \varphi_x \;\; \varphi_y \;\; \varphi_z \;)^T$	Orientation in yaw-pitch-roll
$\vec{\omega} = (\; \omega_x \;\; \omega_y \;\; \omega_z \;)^T$	Angular velocity in HCS (see below)
ω_r	Angular velocity of main rotor (positive for clockwise)
p_c,p_x,p_y,p_t,th	Servo positions for main rotor (collective pitch p_c, cyclic pitch p_x,p_y), tail rotor (pitch p_t), throttle for engine

There are two different coordinate systems used for the values:

BCS This is the Base Coordinate System. It is fixed in the environment (x-axis towards north, y-axis west, z-axis up).

HCS The Helicopter Coordinate System changes its point of origin and its orientation with the movement of the helicopter (x-axis forward, y-axis left, z-axis up).

All values are stored in SI base units with the exception of servo values, which are given in proprietary counts of servo steps.

3. Physics of Flight

This section gives an overview of the physical laws, needed to understand and simulate a flying helicopter.

3.1. Basics

3.1.1. Translation

To simulate the movement of the helicopter along the axes of the BCS, the simple and well known equation of Newton is used:

$$\vec{F} = m\ddot{\vec{s}} \tag{1}$$

m represents the mass of the helicopter and $\vec{F}$ is the resulting force of all simulated effects.

3.1.2. Rotation

Equations for rotational movements are more complex than for simple translation. First there is the corresponding one to equation 1:

$$\vec{M} = \underline{J_H} \cdot \ddot{\vec{\varphi}} \tag{2}$$

$\vec{M}$ is the torque and $\underline{J_H}$ the tensor of the moments of inertia:

$$\underline{J_H} = \begin{pmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{pmatrix}$$

For simulation of simple rotational movements such as the rotational speed of the main rotor according to the torque of the engine, equation 2 is sufficient. But in order to alter rotor axis' direction we have to deal with the effects of gyros such as precession. To fulfill all of these effects we start the derivation with the law of conservation of angular momentum ($\vec{L}$):

$$\begin{aligned} \vec{M} &= \dot{\vec{L}} \\ &= \frac{d}{dt}(\vec{L}_M + \vec{L}_T + \vec{L}_H) \end{aligned} \tag{3}$$

The components of $\vec{L}$ are the momentum of the main rotor ($\vec{L}_M$), the tail rotor ($\vec{L}_T$) and the rotating helicopter body ($\vec{L}_H$). To break down equation 3 we have to have a look at the separate components.

Figure 2 illustrates the dependencies for $\vec{L}_M$ ($\vec{L}_T$, respectively):

$$\dot{\vec{L}}_M = \vec{\omega} \times \vec{L}_M = J_M \, \vec{\omega} \times \vec{\omega}_M \tag{4}$$

$$\dot{\vec{L}}_T = \vec{\omega} \times \vec{L}_T = J_T \, \vec{\omega} \times \vec{\omega}_T \tag{5}$$

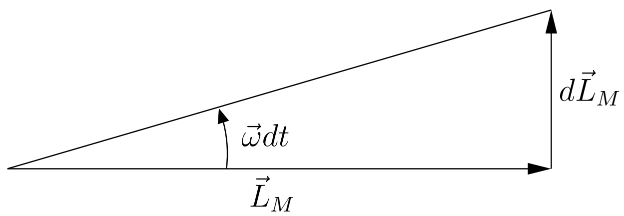

Figure 2: Momentum of main rotor

The complete equation for the whole helicopter combines both rotors:

$$\vec{M} = J_M \, \vec{\omega} \times \vec{\omega}_M + J_T \, \vec{\omega} \times \vec{\omega}_T + \underline{J_H} \cdot \frac{d}{dt}\vec{\omega} \tag{6}$$

3.2. Forces and Torques caused by a Rotor

To be able to calculate the forces and torques, it is necessary to have a look at the rotor blades. There are always two forces when a blade is moved through air: lift and drag. Lift is the component, that makes any air vehicle fly, drag is the force, that slows down the movement due to air friction.

While blades of rotors are moved through air, the forces change with different positions during one cycle as long as there is wind or the helicopter moves. To keep simulation simple, one mean value should be used. One cycle of our main rotor takes approximately $0.05s$, which has to be divided by the number of blades (2) to get the effective frequency of $40Hz$ (the tail rotor has 3 blades and runs at a higher rotary speed). Our controller works at a frequency of $20Hz$, so that it "sees" just a mean value of the forces and torques.

3.2.1. Lift

The following equation describes the mean value of the lifting force of the whole rotor. The blades range from radius R_1 to R_2. Figure 3 shows the definitions for integration over the area of the main rotor.

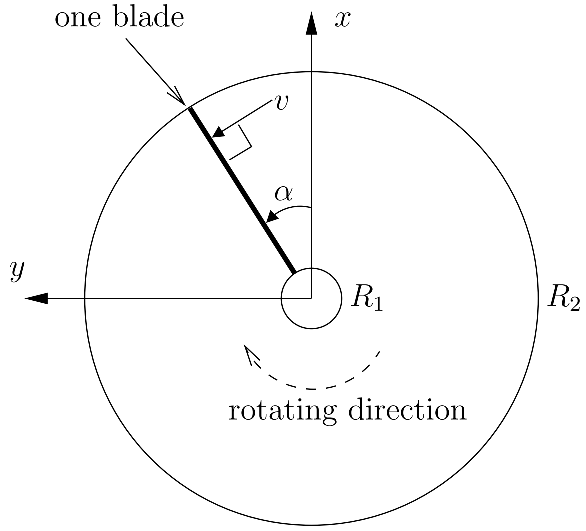

Figure 3: Definitions for integration

$$F_L = \int_0^{2\pi} \int_{R_1}^{R_2} F_L'(\alpha, r) r \, dr \, d\alpha \tag{7}$$

F_L depends on several values such as the pitch of the blades, angular speed and velocity of air with components v_{ax} and v_{ay}. According to several references [1, 2], lift is given by (C_L coefficient of drag, ρ density of air, A area of blade, v component of relative speed of air within the rotor plane orthogonal to the blade)

$$dF_L = \frac{1}{2} C \rho v^2 dA \tag{8}$$

ω_M and the velocity air $\vec{v}_a$ are combined to the speed of air relative and orthogonal to the blade:

$$v = -v_{ax} \sin(\alpha) + v_{ay} \cos(\alpha) - \omega_M r$$

This equation for v is valid as long as $|\vec{v}_a| < |\omega_M r|$. Otherwise the direction of airflow at this radius is wrong. But since v is used squared, a changing sign does not affect the result. For the inner radius of $R_1 = 0.1m$ this means that in simulation wind must not be faster than $12m/s$ at $\omega_M \approx -120rad/s$ ($34m/s$ for tail rotor $R_1 = 0.05m$, $\omega_T \approx -680rad/s$).

How does C depend on the angle of attack, given by three values of pitch? First we assume here pitch to be equal to angle of attack [1]. According to [1, 2] for the normal range of flight (i.e. no stall) we assume C to be proportional to any value of pitch (see figure 4). Collective pitch p_c and cyclic pitch p_x, p_y have to be combined. p_x is defined to produce a positive rotation around x-axis. According to theory of spinning objects (see section 3.1.2) this means mainly a positive torque around y-axis. p_y analogical produces a negative torque around x-axis:

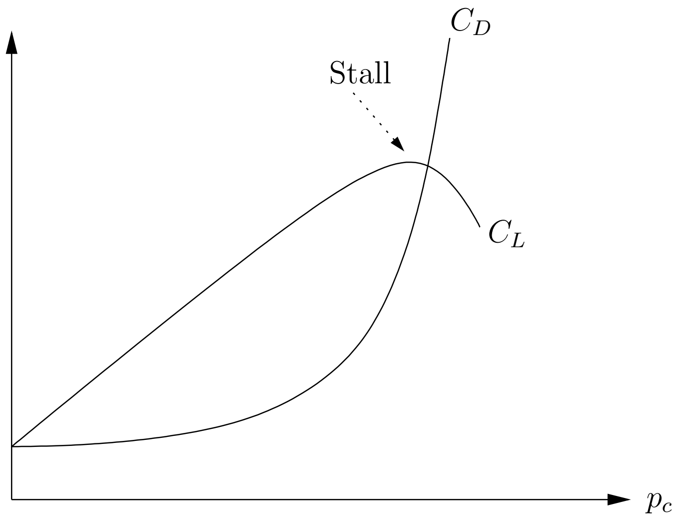

Figure 4: Coefficients of lift and drag

$$C \sim p_c - \cos(\alpha) p_x - \sin(\alpha) p_y$$

C is not zero, if there is no pitch, because the blades are airfoils. Combining all constants in equation 8 the complete relationship looks as follows with new constants C_1 and C_2:

$$F'_L(\alpha, r) = (C_1 + C_2(p_c - \cos(\alpha)p_x - \sin(\alpha)p_y))(-v_{ax} \sin(\alpha) + v_{ay} \cos(\alpha) - \omega_M r)^2 \tag{9}$$

C_1 and C_2 are specified by measurements of real flying tests with the helicopter (see section 4). To include the effect of rare air in great altitudes these constants can be modified by applying the well known barometric equation (density ρ_0 and pressure p_0 at $z = 0$, g gravity):

$$\rho(z) = \rho_0 \exp\left(-\frac{\rho_0}{p_0} g z\right) \tag{10}$$

Solving of equation 7 is no problem.

$$\begin{aligned} F_L &= \frac{\pi}{2}(C_1 + C_2 p_c)(\omega_M^2(R_2^4 - R_1^4) + (v_{ax}^2 + v_{ay}^2)(R_2^2 - R_1^2)) \\ &- \frac{2\pi}{3} C_2 \omega_M(-p_x v_{ay} + p_y v_{ax})(R_2^3 - R_1^3) \end{aligned} \tag{11}$$

Equation 11 contains cyclic pitch. Since the tail rotor does not have any cyclic pitch, the equation can be reduced in this case.

3.2.2. Torques

There are several possible sources of torque on the helicopter. First there is the tail rotor, which has to compensate the torque of the engine on the main rotor. Another source is the cyclic pitch p_x, p_y. But even wind can cause torques, when it is blowing over the rotor.

To calculate these torques, produced by one rotor due to wind and rotation, F'_L of equation 9 is used. This force has to be integrated as follows:

$$M_x = \int_0^{2\pi} \int_{R_1}^{R_2} F'_L(\alpha, r) r^2 \sin(\alpha)\, dr\, d\alpha \tag{12}$$

$$M_y = \int_0^{2\pi} \int_{R_1}^{R_2} -F'_L(\alpha, r) r^2 \cos(\alpha)\, dr\, d\alpha \tag{13}$$

These equations can be solved as well:

$$\begin{aligned} M_x &= \frac{\pi}{2}(C_1 + C_2 p_c)\omega_M v_{ax}(R_2^4 - R_1^4) \\ &+ \frac{\pi}{6} C_2 p_x v_{ax} v_{ay}(R_2^3 - R_1^3) \\ &- \frac{\pi}{60} C_2 p_y (12\omega_M^2(R_2^5 - R_1^5) + (15 v_{ax}^2 + 5 v_{ay}^2)(R_2^3 - R_1^3)) \end{aligned} \tag{14}$$

$$\begin{aligned} M_y &= \frac{\pi}{2}(C_1 + C_2 p_c)\omega_M v_{ay}(R_2^4 - R_1^4) \\ &- \frac{\pi}{6} C_2 p_y v_{ax} v_{ay}(R_2^3 - R_1^3) \\ &+ \frac{\pi}{60} C_2 p_x (12\omega_M^2(R_2^5 - R_1^5) + (5 v_{ax}^2 + 15 v_{ay}^2)(R_2^3 - R_1^3)) \end{aligned} \tag{15}$$

3.2.3. Drag

Dealing with rotors, the drag has to be transformed into a torque. This means that we have to use another slightly different integral than in equations 7,12 and 13:

$$M_D = \int_0^{2\pi} \int_{R_1}^{R_2} F'_D(\alpha, r) r^2 \, dr \, d\alpha \tag{16}$$

Drag can be represented by an equation similar to 8:

$$dF_D = \frac{1}{2} K \rho v^2 dA \tag{17}$$

K is a constant similar to C. But as seen in figure 4, K is not proportional to the pitch. In [1] it is suggested to assume that K is proportional to the square of the pitch:

$$K \sim (p_c - \cos(\alpha)p_x - \sin(\alpha)p_y)^2$$

At no pitch there is still drag, so with two new constants K_1 and K_2 we have a new equation for F'_D, similar to 9:

$$F'_D(\alpha, r) = (K_1 + K_2(p_c - \cos(\alpha)p_x - \sin(\alpha)p_y)^2)(-v_{ax}\sin(\alpha) + v_{ay}\cos(\alpha) - \omega_M r)^2 \tag{18}$$

Integral solved:

$$\begin{aligned} M_D &= \pi(K_1 + K_2 p_c^2)\left(\frac{2}{5}\omega_M^2(R_2^5 - R_1^5) + \frac{1}{3}(v_{ax}^2 + v_{ay}^2)(R_2^3 - R_1^3)\right) \\ &+ \pi K_2(p_x^2 + p_y^2)\left(\frac{1}{5}\omega_M^2(R_2^5 - R_1^5) + \frac{1}{4}(v_{ax}^2 + v_{ay}^2)(R_2^3 - R_1^3)\right) \\ &- \pi K_2\left(\omega_M p_c(-p_x v_{ay} + p_y v_{ax})(R_2^4 - R_1^4) + \frac{1}{6}(p_x v_{ax} + p_y v_{ay})^2(R_2^3 - R_1^3)\right) \end{aligned} \tag{19}$$

4. Implementation

4.1. Numerical Integration (Runge-Kutta)

According to equations 1 and 2, forces and torques cause accelerations. To get the according velocity and position, these accelerations have to be integrated. It is not possible to solve the equations symbolically, because there is no known function. All the parameters such as pitch and wind will change, depending on the actions of the controller. The only possibility is to integrate numerically.

A very efficient method for numerical integration is the algorithm called Runge-Kutta. In [?] it is described as “best compromise between programming effort, computational time and numeric accuracy”. To integrate a given differential equation $\frac{d}{dt}\vec{x} = \vec{f}(\vec{x})$, the algorithm tries to estimate a representative gradient for the current time step. This is done using the following equations with a time step of Δt:

$$\begin{aligned}
\vec{x}_{i+1} &= \vec{x}_i + \frac{1}{6}(\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4) + \mathcal{O}(\Delta t^4) \\
\vec{k}_1 &= \Delta t\, \vec{f}(\vec{x}_i) \\
\vec{k}_2 &= \Delta t\, \vec{f}\left(\vec{x}_i + \frac{\vec{k}_1}{2}\right) \\
\vec{k}_3 &= \Delta t\, \vec{f}\left(\vec{x}_i + \frac{\vec{k}_2}{2}\right) \\
\vec{k}_4 &= \Delta t\, \vec{f}(\vec{x}_i + \vec{k}_3)
\end{aligned}$$

The remaining error of numerical integration is represented by $\mathcal{O}(\Delta t^4)$, Runge-Kutta is a fourth order method.

The smaller you choose Δt the better the accuracy of simulation becomes. As the controller is called at a fixed frequency of $20Hz$, this is the upper limit for Δt. To increase accuracy the simulator can be called more frequently than the controller. But besides longer durations for calculation there is another problem. Because of smaller Δt, rounding errors will be increased, too. To avoid this, one could use higher resolution data types than `double`. But since the accuracy produced by Runge-Kutta algorithm is very good, there is no need to call the simulator more often than the controller. The best results are reached by calling both controller and simulator at the same frequency.

Using Runge-Kutta for integration, the differential equations for the helicopter can be reduced to five very simple but still coupled equations:

$$\omega_r = \int \dot{\omega}_r dt \quad \overrightarrow{s} = \int \overrightarrow{v} dt \quad \overrightarrow{v} = \int \overrightarrow{a} dt \quad \overrightarrow{\varphi} = \int \dot{\overrightarrow{\varphi}} dt \quad \overrightarrow{\omega} = \int \dot{\overrightarrow{\omega}} dt \qquad (20)$$

Note that not $\vec{\omega}$ is used for $\vec{\varphi}$, because it uses a different coordinate system. The following section describes the calculation of $\dot{\omega}_r$, $\overrightarrow{v}$, $\overrightarrow{a}$, $\dot{\overrightarrow{\varphi}}$ and $\dot{\overrightarrow{\omega}}$. These values are used by the explained Runge-Kutta algorithm to compute ω_r, $\overrightarrow{s}$, $\overrightarrow{v}$, $\overrightarrow{\varphi}$ and $\overrightarrow{\omega}$ of the next time step.

4.2. Simulation of Flight

In section 3 the individual physical equations are described. Now they will be put together to build the simulator.

4.2.1. Servos

First lets have a look at the servos. They need time to reach the value given by the controller. To model this behavior is very difficult, as servos contain their own small controller to move to the given position. The best approach found so far is quite simple: The remaining difference is decreased exponentially (p_{cs} target position).

$$p_c(t + \Delta t) = p_c(t) + (p_{cs} - p_c(t))2.5\Delta t$$

In this example the collective pitch p_c is used, but every servo is simulated this way. The constant 2.5 has been chosen by experiments.

4.2.2. Wind

In most of the equations the velocity of air is needed, which consists of two components: wind ($\vec{v_w}$) and the movement of the helicopter ($\vec{v}$). The air movement has to be calculated in HCS, using the matrix for rotation from BCS to HCS ${}_H\underline{R}^B$:

$$\vec{v_a} = {}_H\underline{R}^B \cdot (\vec{v_w} - \vec{v}) \tag{21}$$

4.2.3. Rotor

This section deals with the rotational speed of the rotors. Both main rotor (ω_M) and tail rotor (ω_T) are connected to the engine (ω_e) via a fixed gear transmission ratio, so that their rotational speed is given by ω_r and a coefficient:

$$\begin{aligned} \omega_M &= -\omega_r \\ \omega_T &= -\omega_r n_T \\ \omega_e &= \omega_r n_e \end{aligned}$$

There are four torques on the rotors and engine. Two because of drag of the rotors (M_M, M_T), friction in gear (M_g) and the accelerating torque of the engine (M_e). For the first two, equation 19 is used:

$$\begin{aligned} M_M &= (K_{M1} + K_{M2}p_c^2)\left(\frac{2}{5}\omega_r^2(R_{M2}^5 - R_{M1}^5) + \frac{1}{3}(v_{ax}^2 + v_{ay}^2)(R_{M2}^3 - R_{M1}^3)\right) \qquad (22) \\ &+ K_{M2}(p_x^2 + p_y^2)\left(\frac{1}{5}\omega_r^2(R_{M2}^5 - R_{M1}^5) + \frac{1}{4}(v_{ax}^2 + v_{ay}^2)(R_{M2}^3 - R_{M1}^3)\right) \\ &+ K_{M2}\left(\omega_r p_c(-p_x v_{ay} + p_y v_{ax})(R_{M2}^4 - R_{M1}^4) - \frac{1}{6}(p_x v_{ax} + p_y v_{ay})^2(R_{M2}^3 - R_{M1}^3)\right) \\ M_T &= (K_{T1} + K_{T2}p_t^2)\left(\frac{2}{5}\omega_r^2 n_T^2(R_{T2}^5 - R_{T1}^5) + \frac{1}{3}(v_{ax}^2 + v_{az}^2)(R_{T2}^3 - R_{T1}^3)\right) \qquad (23) \end{aligned}$$

The values of all the R can be simply determined by measuring the size of the blades of the helicopter. Determining the other constants (K_{M1}...) is not so simple. In each equation there are two unknown constants K_1 and K_2. So we need two equations to solve them. One good choice is using the operating point of hovering. To get another equation, you can choose another hovering situation at a different angular speed ω_r.

To determine M_M for an operating point, we need several information about this point. The tail rotor (lever arm d_T) has to compensate the torque of the engine, which compensates the torque M_M. On the other hand the force of the tail rotor F_{TA} (index A denotes operating point), which pushes the helicopter to the side, has to be compensated by the main rotor by pulling to the other side. This force can be determined by using the appropriate slope φ_{xA}. Finally we need to know the force of lift of the main rotor F_{zA}. Since it has to keep the helicopter in the air, this force has to compensate exactly the gravity. This leads to the following definitions for an operating point:

$$\begin{aligned} F_{zA} &= \cos(\varphi_{xA})mg \qquad (24) \\ F_{TA} &= \sin(\varphi_{xA})mg \\ M_{MA} &= F_{zA}d_T \end{aligned}$$

M_T cannot be determined this way. It is much smaller, so the measurements are too imprecise. This means the effect of M_T may be unessential for simulation. But nevertheless it is possible to include even the effect of the tail rotor, if the constants can be identified.

To receive K_{M1} and K_{M2}, equation 22 (23, respectively) has to be solved for two operating points (similar solutions for tail rotor K_{Ti}):

$$\begin{aligned}
K_{M1} &= K_{M0}(-M_{MA_1}\omega_{rA_2}^2 p_{cA_2}^2 + M_{MA_2}\omega_{rA_1}^2 p_{cA_1}^2) \\
K_{M2} &= K_{M0}(M_{MA_1}\omega_{rA_2}^2 - M_{MA_2}\omega_{rA_1}^2) \\
K_{M0} &= \frac{5}{2}\frac{\rho(z)}{\rho_0}\frac{1}{\omega_{rA_1}^2\omega_{rA_2}^2(p_{cA_1}^2 - p_{cA_2}^2)(R_{M2}^5 - R_{M1}^5)}
\end{aligned}$$

The torque inside the gear is estimated to be proportional to angular speed:

$$M_g = \omega_r \frac{M_{gA}}{\omega_{rA}} \tag{25}$$

The torque produced by the engine is approximated very simple. Only reduced power due to rare air is simulated:

$$M_e = \frac{th}{th_A}(M_{MA} + M_{TA}n_T + M_{gA})\frac{\rho(z)}{\rho_0} \tag{26}$$

Using equations 22 through 26, the acceleration of angular speed $\dot{\omega_r}$ can be calculated with known moments of inertia of the rotors (J_M, J_T) and the gear (J_g):

$$\dot{\omega_r} = \frac{M_e - M_M - M_T n_T - M_g}{J_M + J_T n_T + J_g} \tag{27}$$

4.2.4. Translation

To calculate the accelerations of the helicopter, all the forces are needed. They are composed of lift according to equation 11 and friction due to movement of the helicopter body. Equation 17 shows how to get the force because of friction. But a problem for helicopters is that they produce most of the wind themselves. Nevertheless this approach is used in this simulator. All constants in equation 17 are merged to only one (f) per direction. The values are estimated from flight experiments.

The forces produced by both rotors (F_M and F_T) are used to calculate the resulting forces on the helicopter:

$$\begin{aligned}
F_T &= f_T(v_{ay} + \omega_z d_T)\,|v_{ay} + \omega_z d_T| \qquad (28) \\
&+ \frac{1}{2}(C_{T1} + C_{T2}p_t)(\omega_r^2 n_T^2 (R_{T2}^4 - R_{T1}^4) + (v_{ax}^2 + v_{az}^2)(R_{T2}^2 - R_{T1}^2)) \\
F_M &= \frac{1}{2}(C_{M1} + C_{M2}p_c)(\omega_r^2(R_{M2}^4 - R_{M1}^4) + (v_{ax}^2 + v_{ay}^2)(R_{M2}^2 - R_{M1}^2)) \qquad (29)
\end{aligned}$$

$$
\begin{aligned}
&+ \quad \frac{2}{3}C_{M2}\omega_r(-p_x v_{ay} + p_y v_{ax})(R_{M2}^3 - R_{M1}^3) \\
F_x &= f_x v_{ax}\,|v_{ax}| \qquad (30)\\
F_y &= f_y v_{ay}\,|v_{ay}| + F_T \qquad (31)\\
F_z &= f_z v_{az}\,|v_{az}| + F_M \qquad (32)
\end{aligned}
$$

Here again new constants have to be defined, which is possible using the definitions of the operating point of equations 24 (similar solutions for tail rotor C_{Ti}):

$$
\begin{aligned}
C_{M1} &= C_{M0}(-F_{MA_1}\omega_{rA_2}^2 p_{cA_2} + F_{MA_2}\omega_{rA_1}^2 p_{cA_1}) \\
C_{M2} &= C_{M0}(F_{MA_1}\omega_{rA_2}^2 - F_{MA_2}\omega_{rA_1}^2) \\
C_{M0} &= \frac{\rho(z)}{\rho_0}\frac{2}{\omega_{rA_1}^2\omega_{rA_2}^2(p_{cA_1} - p_{cA_2})(R_{M2}^4 - R_{M1}^4)}
\end{aligned}
$$

Now the resulting accelerations of these forces have to be converted to BCS. (In equation 21 the transformation to HCS was needed). Finally the acceleration of gravity has to be removed from the result:

$$
\vec{a} = {}_B\underline{R}^H \cdot \frac{1}{m}\begin{pmatrix} F_x \\ F_y \\ F_z \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \qquad (33)
$$

4.2.5. Rotation

Here we need the several torques that affect the body of the helicopter. Equations 14 and 15 are used, together with torques of the engine according to 26 and the effect of the tail rotor (28).

The torque of the engine is applied to both main and tail rotor. Each rotor consumes one portion for the drag. This is directly represented by the appropriate torque M_M (M_T). The remaining portion is used to increase (or decrease, if it is negative) the angular speed. Using equation 2, it can be expressed as $\dot{\omega}_r J_M$.

Another reason of torque is given by the location of the center of gravity of the helicopter. The main rotor pulls along the rotor axis. Any distance between the center of gravity and the main rotor axis (m_x, m_y) causes additional torque.

$$
\begin{aligned}
M_x = &- \frac{1}{2}(C_{M1} + C_{M2}p_c)\omega_r v_{ax}(R_{M2}^4 - R_{M1}^4) \qquad (34)\\
&+ \frac{1}{6}C_{M2}p_x v_{ax} v_{ay}(R_{M2}^3 - R_{M1}^3) \\
&- \frac{1}{60}C_{M2}p_y(12\omega_r^2(R_{M2}^5 - R_{M1}^5) + (15v_{ax}^2 + 5v_{ay}^2)(R_{M2}^3 - R_{M1}^3)) \\
&- \frac{1}{2}(C_{T1} + C_{T2}p_t)\omega_r n_T v_{ax}(R_{T2}^4 - R_{T1}^4) \\
&- F_M m_y
\end{aligned}
$$

$$
\begin{aligned}
M_y = & - \frac{1}{2}(C_{M1} + C_{M2}p_c)\omega_r v_{ay}(R_{M2}^4 - R_{M1}^4) & (35)\\
& - \frac{1}{6}C_{M2}p_y v_{ax} v_{ay}(R_{M2}^3 - R_{M1}^3) \\
& + \frac{1}{60}C_{M2}p_x(12\omega_r^2(R_{M2}^5 - R_{M1}^5) + (5v_{ax}^2 + 15v_{ay}^2)(R_{M2}^3 - R_{M1}^3)) \\
& + M_T + \dot{\omega}_r J_T n_T \\
& + F_M m_x \\
M_z = & + M_M + \dot{\omega}_r J_M & (36)\\
& - F_T d_T \\
& - \frac{1}{2}(C_{T1} + C_{T2}p_t)\omega_r n_T v_{az}(R_{T2}^4 - R_{T1}^4)
\end{aligned}
$$

These torques M_x, M_y, M_z have to be used to calculate the angular accelerations. This is done by applying equation 6.

$$
\begin{aligned}
\dot{\omega_x} &= J_x^{-1}(M_x + J_M\omega_y\omega_r - J_T n_T \omega_z \omega_r) & (37)\\
\dot{\omega_y} &= J_y^{-1}(M_y - J_M\omega_x\omega_r) \\
\dot{\omega_z} &= J_z^{-1}(M_z + J_T n_T \omega_x \omega_r)
\end{aligned}
$$

As mentioned for equations 20, we have to convert $\vec{\omega}$ to BCS in order to apply it to $\vec{\varphi}$. Since we use yaw-pitch-roll angles, ω_x does not have to be converted. ω_y has to be rotated on the x-axis and ω_z on x- and y-axis, which is denoted as $\underline{R}_x(\alpha)$ for a rotation on x-axis of α:

$$
\dot{\vec{\varphi}} = \begin{pmatrix} \omega_x \\ 0 \\ 0 \end{pmatrix} + \underline{R}_x(\varphi_x) \cdot \begin{pmatrix} 0 \\ \omega_y \\ 0 \end{pmatrix} + \underline{R}_y(\varphi_y) \cdot \underline{R}_x(\varphi_x) \cdot \begin{pmatrix} 0 \\ 0 \\ \omega_z \end{pmatrix} \tag{38}
$$

4.3. Simulation of Sensors

For the complete simulation of MARVIN, the IMU algorithms should be used. So the values of all connected sensors have to be simulated as well. After each step of flight simulation, the results are passed to the simulation of sensors.

There are errors in every measurement. To simulate them, firstly the amount was estimated from log-files of real flights (see appendix B). This error is used to compute a standard deviation at simulation time. Since the computer only knows uniformly distributed random numbers ($r(a,b)$ is a distribution between a and b), we have to build a standard deviation ourselves. This can be done using this definition [3]:

$$
N(0,1) = \lim_{n\to\infty} \left(\frac{\sum_{i=1}^{n} r(0,1) - \frac{n}{2}}{\sqrt{\frac{n}{12}}} \right)
$$

A good choice for implementation is $n = 12$. The distribution of values is quite good and computation is simple.

For simulation of the compass it is possible to choose a declination and inclination. This is the difference of magnetic north vector and geographic horizontal vector. Inclination is positive, if magnetic north is below horizon. Declination is positive if magnetic north is west of geographic north.

5. Conclusion

The presented simulator for MARVIN is still under development but it is already capable of verifying the basic controller and IMU software. A controller, which is capable of flying the real helicopter will work in simulation, too. On the other hand, a controller that fails in simulation will not be able to control the real helicopter successfully.

Furthermore some correlations of flight characteristics to certain circumstances such as wind, rare air or bad sensors can simply be tested.

Really exact results are not reached, since there are too many simplifications and assumptions. For example in section 3.2.1 we assume pitch to be equal to angle of attack. This is not really true, because any rotor causes air movement [1]. This effect is nowhere considered in this simulator. In section 3.2.1 there is explained, why wind should not be faster than $12m/s$ during simulation (relative to the helicopter). During normal flight this is correct. But while the rotors are accelerating to reach there operating point, this rule is not valid.

Ground effects are not simulated. Servos are simulated very roughly.

Another important point is that we do not know all constants precisely. Many of them are just estimated (see appendix A).

This simulator can simulate many of interesting effects but one should not expect them to show the whole truth.

A. Helicopter Constants

Identifier	Value	Comment
m	$11kg$	mass of whole helicopter
m_x	$-0.019m$	position of center of gravity in HCS
m_y	$-0.013m$	position of center of gravity in HCS
J_x	$0.6kgm^2$	moment of inertia of helicopter body (estimated)
J_y	$1kgm^2$	(estimated)
J_z	$1kgm^2$	(estimated)
J_g	$0.1kgm^2$	engine and gear (estimated)
J_M	$0.184kgm^2$	main rotor
J_T	$0.00029kgm^2$	tail rotor
R_{M1}	$0.1m$	inner edge of main rotor blades
R_{M2}	$0.92m$	outer edge
R_{T1}	$0.05m$	inner edge of tail rotor blades
R_{T2}	$0.17m$	outer edge
d_T	$1.045m$	distance of tail rotor to main rotor axis
n_T	$\frac{85}{15}$	tail rotor is faster than main rotor
n_e	$\frac{64}{36}n_T$	engine is faster than rotors
f_x	$0.3\frac{kg}{ms^2}$	friction without rotors ($f \hat{=} c_w A \frac{\rho}{2}$) (estimated)
f_y	$0.3\frac{kg}{ms^2}$	friction without rotors and tail (estimated)
f_z	$0.2\frac{kg}{ms^2}$	friction without rotors (estimated)
f_T	$0.15\frac{kg}{ms^2}$	friction of tail for y-axis (estimated)
ω_{rA}	$1150rpm \hat{=} 120\frac{rad}{s}$	main rotor speed for hovering in operating point
φ_{xA}	$0.092rad$	slope
M_{gA}	$0.7Nm$	torque due to friction in gear (estimated)
p_{cA}	$1030 - p_{c0}$	collective pitch of main rotor
p_{tA}	$335 - p_{t0}$	pitch of tail rotor
th_A	$830 - th_0$	throttle
p_{c0}	0	value of servo at zero pitch
p_{t0}	540	zero pitch for tail rotor (less is more)
th_0	100	idle throttle (no torque on rotor)

B. Sensor Constants

Sensor	Measurements per second	Standard deviation
GPS position	5	$0.01m$
GPS velocity	5	$0.03\frac{m}{s}$
GPS antenna position in HCS: $-925mm$		
Ultrasonic altitude	2.5	$0.05m$
Ultrasonic only works reliable between $0.41m$ and $4.5m$.		
Acceleration	20	$0.059\frac{m}{s^2}$
Rotational velocity	20	$0.021\frac{rad}{s}$
Magnetic field	20	$700nT \widehat{=} 1.5\%$
Main rotor velocity	once a cycle	$120\mu s$

References

[1] John S. Decker. See how it flies. http://www.monmouth.com/ jsd/how/, 2001.

[2] R. Lux. Manuskript ”Aerodynamik”. TU Dresden - Institut für Luftfahrt, 2002.

[3] Detlef Steinhausen. Simulationstechniken -Das Buch-. http://www.fh-muenster.de/FB9/person/steinha/buch/default.shtm, 1993.