Abstract

This paper presents a generalized Multirotor Aerial Vehicle (MAV) modeling framework which includes rigid body dynamics, gyroscopic effect and motor dynamics. We illustrate how this model can be used to derive any MAV platform constructed with an arbitrary number of rotors by using the quadrotor case as an example. Based on this result, we design the first Modelica-based MAV simulator. We validate the proposed design by using a simple altitude and attitude stabilization control system through a Modelica simulation setup.

Keywords: Multirotor Aerial Vehicle, Modeling, Modelica

1 Introduction

Technological advancements in recent years, including the miniaturization in battery, sensor and actuation technologies, as well as the availability of low cost high performance computing boards have enabled the genesis of intelligent autonomous flying machines. The most popular class of these machines are the so-called Multirotor Aerial Vehicles (MAVs) which represent motorized rotorcrafts that have favourable dynamical properties and can achieve small geometries. MAVs and especially the quadrotor configuration are now the de facto standard research platforms for aerial robotics with many potential applications including search and rescue in indoor and outdoor environments (Tomie et al., 2012), precision agriculture (Zhang and Kovacs, 2012), aerial construction (Lindsey et al., 2011; Willmann et al., 2012), inspection and maintenance (Mellinger et al., 2011; Jimenez-Cano et al., 2013), environmental monitoring (Alexis et al., 2009), exploration and mapping (Fraundorfer et al., 2012), aerial transportation (Michael et al., 2011; Mellinger et al., 2013) and swarming (Kushleyev et al., 2013).

Due to this growing interest, there have emerged multiple MAV simulation platforms mainly in MATLAB and ROS with notable examples being (Bresciani, 2008) and (Furrer et al., 2016), respectively. Both provide simulation for MAV dynamics (with the former covering only the quadrotor case) and sensors, and the latter having a less user-friendly interface via pure code and configuration. To the best of our knowledge, there are no existing MAV simulation platforms within the Modelica community.

Our paper gives a simple way of deriving a proper dynamical model for a MAV constructed with an arbitrary number of rotors by using a generalized MAV model. Based on this paradigm, we also present a Modelica simulator that can be used for multirotor aerial vehicles. To the best of the authors’ knowledge, this is the first Modelica-based MAV simulator available within the Modelica community.

The remainder of the paper is organized as follows. Section 2 describes how generalized MAV dynamics can be derived and how an appropriate dynamical model can be extracted for a quadrotor-based MAV. In Section 4, we describe necessary classes to design the Modelica-based simulator for MAVs, while in Section 4, we validate the results throughout a simple altitude and attitude stabilization control system. Concluding remarks are presented in Section 5.

2 MAV dynamics

A large number of papers address MAV modeling putting the focus mostly on the quadrotor case. Noteworthy classical contributions include (Altug et al., 2002), (Hamel et al., 2002), (Pounds et al., 2002) and (Bouabdallah et al., 2004a). More recent examples of very detailed quadrotor and octocopter modeling are presented in (Bangura and Mahony, 2012) and (Osmic et al., 2016), respectively. To the best of our knowledge, one of the most complete work regarding MAVs can be found in (Mahony et al., 2012), where the authors have derived MAV dynamics, included advanced state estimation, control and motion planning algorithms and therefore provided full system autonomy.

In this section, we will describe the dynamical model of the quadrotor, which is frequently considered to be the standard research platform for MAVs due to its simple construction and purposeful functionality. We use the results and nomenclature from (Osmic et al., 2016) and show that only minor changes are necessary to apply the final octocopter model presented in (Osmic et al., 2016) to any MAV, including also the quadrotor case.

2.1 MAV rigid body dynamics

In order to model the dynamics of any mobile robot it is common to define two frames of reference. A body fixed
frame \( \{o\} \) is attached to the robots center of mass and all sensory data is measured with respect to this frame, while a ground fixed frame \( \{g\} \) is used to define workspace goals in an intuitive and user-friendly manner. The body fixed and ground fixed frame represent right-handed Cartesian coordinate systems and are usually referred to as the local and global coordinate system, respectively.

Workspace goals can be defined in terms of global position coordinates \( x, y \) and \( z \) and orientation coordinates \( \phi, \theta \) and \( \psi \) (see Fig. 1), where positive directions of \( \phi, \theta \) and \( \psi \) are chosen according to the right-hand rule. Therefore, the position vector \( \mathbf{x} = [x \ y \ z]^T \) and the orientation vector \( \mathbf{\Psi} = [\phi \ \theta \ \psi]^T \) can completely determine the vehicle’s location in the workspace. As shown in Fig. 2, the local coordinates are described by the linear velocities \( u, v \) and \( w \) and the angular velocities \( P, Q, R \). The positive directions of the angular velocities \( P, Q \) and \( R \) are also chosen according to the right-hand rule and therefore coincide with the positive directions of \( \phi, \theta \) and \( \psi \). Both linear and angular velocity coordinates can also be expressed in compact vector form as \( \mathbf{v} = [u \ v \ w]^T \) and \( \mathbf{P} = [P \ Q \ R]^T \), respectively.

Forces and torques which act on a MAV are shown in Fig. 3. The thrust \( T \) is a force that acts towards the positive direction of the \( Z \) axis of the local coordinate system \( \{o\} \), while the force \( G \) represents the gravitational force acting towards the negative direction of the \( Z_B \) axis of the global coordinate system \( \{g\} \). \( \tau_v, \tau_\theta, \text{ and } \tau_\psi \) represent the torques that move the vehicle around the \( X, Y \) and \( Z \) axes of the local coordinate system, respectively, and can be compactly denoted as \( \mathbf{\tau} = [\tau_v \ \tau_\theta \ \tau_\psi]^T \). Their positive direction is also chosen to coincide with the positive directions of the angular velocities \( P, Q \) and \( R \).

We can now describe the rigid body dynamics of any MAV in accordance to the results presented in (Osmic et al., 2016). The kinematic model of the linear motion is given as

\[
\dot{\mathbf{x}} = \mathbf{R}(\phi, \theta, \psi) \mathbf{v},
\]

where \( \mathbf{R}(\phi, \theta, \psi) \) is the total rotation matrix which for the ZYX Euler convention has the form

\[
\mathbf{R}(\phi, \theta, \psi) = \mathbf{R}(Z, \psi) \mathbf{R}(Y, \theta) \mathbf{R}(X, \phi)
\]

and the elementary rotation matrices \( \mathbf{R}(Z, \psi), \mathbf{R}(Y, \theta) \) and \( \mathbf{R}(X, \phi) \) are defined as

\[
\mathbf{R}(X, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix}, \quad \mathbf{R}(Y, \theta) = \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix}, \quad \mathbf{R}(Z, \psi) = \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}
\]

The kinematic model of the angulator motion can be described by

\[
\mathbf{\Psi} = \mathbf{R}_A^{-1}(\phi, \theta, \psi) \mathbf{P},
\]

where the matrix \( \mathbf{R}_A^{-1}(\phi, \theta, \psi) \) for the ZYX Euler convention is

\[
\mathbf{R}_A^{-1}(\phi, \theta, \psi) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\theta}{c_\phi} & \frac{c_\theta}{c_\phi} \end{bmatrix}
\]
The dynamic model of the linear motion can be represented by the following equation
\[
\dot{v} = \begin{bmatrix} 0 \\ 0 \\ \frac{f}{m_0} \end{bmatrix} + g \begin{bmatrix} s_\theta \\ -s_\phi c_\theta \\ -c_\phi c_\theta \end{bmatrix} - Sv, \tag{8}
\]
where \(m_0\) is the total mass of the MAV and the matrix \(S\) is formed as
\[
S = \begin{bmatrix} 0 & -R & Q \\ R & 0 & -P \\ -Q & P & 0 \end{bmatrix}. \tag{9}
\]
Finally, the dynamic model of the angular motion can be caught with
\[
\dot{P} = J^{-1}(\tau - SP), \tag{10}
\]
where \(J\) is a \(3 \times 3\) matrix representing the inertia tensor of the MAV.

### 2.2 Quadrotor modeling

To tailor the previously derived MAV model to the quadrotor case we need to derive the inertia tensor \(J\), and define the thrust \(T\) and the torque vector \(\tau\). Since all of these quantities depend on the MAV’s geometry, we consider a quadrotor case shown in Fig. 4 along with its simplified geometry illustrated in Fig. 5, where the length of the four arms is \(l\), a hardware support plate is modeled as a solid sphere of mass \(M\) having a radius \(r\), and the four motors constructed with fixed pitch propellers are modeled as particles with mass \(m\).

The axes of the local coordinate system, as shown in Fig. 4, represent principal axes of inertia, where the inertia tensor matrix has the diagonal form
\[
J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \tag{11}
\]
and \(I_{xx}, I_{yy}, I_{zz}\) being the moments of inertia around the \(X\), \(Y\) and \(Z\) axes of the local coordinate system, respectively. These components can be derived via the Huygens-Steiner theorem (Morin, 2008) as
\[
I_{xx} = I_{yy} = \frac{2Mr^2}{5} + 2ml^2 \tag{12}
\]
and
\[ I_{zz} = \frac{2Ml^2}{5} + 4ml^2. \quad (13) \]

In order to derive the thrust \( T \) and the \( \tau_x \) and \( \tau_y \) components of the torque vector \( \tau \), we will consider the rotor forces acting on the quadrotor system as depicted in Fig. 6. Thus \( T, \tau_x \) and \( \tau_y \) are given as follows
\[ T = F_1 + F_2 + F_3 + F_4, \quad (14) \]
\[ \tau_x = l(F_1 - F_3), \quad (15) \]
\[ \tau_y = l(F_4 - F_2). \quad (16) \]

In accordance to the work presented in (Mahony et al., 2012), the rotor forces \( F_i \) \((i = \overline{1,4})\) can be approximated as
\[ F_i = b\Omega_i^2 \quad (i = \overline{1,4}), \quad (17) \]
where \( b \left[ \frac{Nm^2}{rad^2} \right] \) is the rotor thrust constant and \( \Omega_i \left[ \frac{rad}{s} \right] \) is the angular velocity of the \( i \)-th rotor. Combining eqs. (14), (15), (16) and (17) yields
\[ T = b \left( \Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2 \right), \quad (18) \]
\[ \tau_x = bl \left( \Omega_1^2 - \Omega_3^2 \right) \quad (19) \]
and
\[ \tau_y = bl \left( \Omega_4^2 - \Omega_2^2 \right). \quad (20) \]

The torque \( \tau_z \) is a consequence of Newton’s third law and can be formed as
\[ \tau_z = -M_1 + M_2 - M_3 + M_4, \quad (21) \]
where \( M_i \) \((i = \overline{1,4})\) is the counter induced torque of the \( i \)-th rotor. According to (Mahony et al., 2012) the counter torque can approximated as
\[ M_i = d\Omega_i^2 \quad (i = \overline{1,4}), \quad (22) \]
where \( d \left[ \frac{Nm^2}{rad^2} \right] \) is the rotor drag constant. Combining equations (21) and (22) yields
\[ \tau_z = d \left( -\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2 \right). \quad (23) \]

Finally, we can represent the system actuation via matrix equation
\[ \begin{bmatrix} T \\ \tau \end{bmatrix} \quad (24) \]
where \( A \) is the actuation matrix
\[ A = \begin{bmatrix} b & b & b & b \\ bl & 0 & -bl & 0 \\ 0 & -bl & 0 & bl \\ -d & d & -d & d \end{bmatrix}, \quad (25) \]
and \( \Omega_s \) is the squared rotor velocity vector defined as
\[ \Omega_s = \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix}. \quad (26) \]

It is evident from this result that any MAV can be modelled by choosing the appropriate inertia tensor \( J \) and actuation matrix \( A \) as parameters, and picking the squared rotor velocity vector \( \Omega_s \) of the right size as a system input. For any MAV constructed with \( n \geq 4 \) rotors, the actuation matrix has the dimension \( 4 \times n \) and the squared rotor velocity vector \( \Omega_s \) has the length \( n \).

Moreover, we can include the gyroscopic effect in the dynamic model of the angular motion given by eq. (10) as
\[ \dot{P} = J^{-1} \left( \tau - SJP - S \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right), \quad (27) \]
where \( I_{zxm} \) is the rotor moment of inertia and \( W_g \) is the gyroscopic term given as
\[ W_g = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \quad (28) \]
for the quadrotor case. In order to generalize the gyroscopic term for any MAV configuration, it is more appropriate to choose the rotor velocity vector \( \Omega \)
\[ \Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} \quad (29) \]
as system input and express the gyroscopic term as
\[ W_g = \text{sign}(A_z)\Omega, \quad (30) \]
where \( A_z \) is the fourth row of the actuation matrix \( A \), while the squared rotor velocity vector \( \Omega_s \) can easily be computed by calculating the element-wise square of the vector \( \Omega \).
2.3 Motor dynamics

Each rotor of a quadrotor MAV is driven by a DC motor. Therefore, in order to obtain a precise MAV model, it is inevitable to include the motor dynamics to address effects like motor response time, saturation and power consumption. In accordance to the work presented in (Osmic et al., 2016), a simplified model can be used for this purpose which is given by

\[ I_{z,m} \dot{\Omega}_i + \frac{K_m K_r}{R} \Omega_i = \frac{K_m}{R} v_i - \tau_i, \quad i = 1, 4, \]  
(31)

where \( K_m \left[ \frac{Nm}{A} \right] \) is the mechanical motor constant, \( K_r \left[ \frac{Vs}{rad} \right] \) being the electrical motor constant, \( R \) denotes the armature resistance, \( v_i \) is the armature voltage, with \( \tau_i \) being the load torque of the \( i \)-th motor. The load torque is the aerodynamic drag which can be computed as

\[ \tau_i = d\Omega_i^2, \quad i = 1, 4. \]  
(32)

The input voltage of each motor is saturated by the following box constraint

\[ 0 \leq v_i \leq v_{\text{max}}, \quad i = 1, 4 \]  
(33)

where \( v_{\text{max}} \) is the maximum armature voltage, and consequently the angular velocity of each rotor is also box constrained by

\[ 0 \leq \Omega_i \leq \Omega_{\text{max}}, \quad i = 1, 4, \]  
(34)

where \( \Omega_{\text{max}} \) is the maximum angular velocity which can be easily computed from the stationary state of the motor dynamic model given by (31).

Finally, the rotor moment of inertia \( I_{z,m} \) can be approximately calculated as

\[ I_{z,m} = \frac{m_p l_p^2}{12}, \]  
(35)

where \( m_p \) is the mass and \( l_p \) being the length of the rotor.

3 Modelica design

In order to provide a greater end-user utilization, we designed the following Modelica blocks / classes:

- MavBase
- MavSimple
- MavFull

The MaveBase block, as shown in Fig. 7, is the simplest and it models the rigid body dynamics including the gyroscopic effect covered with eqs. (1), (6), (8) and (27). Its inputs are the generalized forces acting on the system and the outputs are the global coordinates of the system and its derivations.

The MavSimple block, as shown in Fig. 8, extends the MavBase block with the actuation model given by eq. (24) with the input being the angular velocity vector \( \Omega \) and the outputs being the global coordinates of the system and its derivations.

Finally, the MavFull block, as shown in figure 9, provides the greatest level of detail. It extends the MavSimple block and adds the motor dynamics (31) to the model. The block input is the motor voltage vector \( v \) with the outputs being the global coordinates of the system and its derivations, as well as the angular velocity vector \( \Omega \). The angular velocity vector as system output is necessary to provide motor level control possibilities.

The parameters of the blocks are given in Table 1, 2 and 3, and their default values match the AscTec Pelican quadrotor (AscTec, 2016).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R )</td>
<td>0.1107</td>
<td>( \Omega )</td>
<td>Resistance</td>
</tr>
<tr>
<td>( K_m )</td>
<td>0.01</td>
<td>( [\frac{Nm}{A}] )</td>
<td>Motor size constant</td>
</tr>
<tr>
<td>( K_r )</td>
<td>0.01</td>
<td>( [\frac{Vs}{rad}] )</td>
<td>Motor velocity constant</td>
</tr>
<tr>
<td>( v_{\text{max}} )</td>
<td>11.1</td>
<td>( V )</td>
<td>Maximum voltage</td>
</tr>
<tr>
<td>( \Omega_0 )</td>
<td>569.3572</td>
<td>( [\frac{rad}{sec}] )</td>
<td>Initial angular velocity</td>
</tr>
</tbody>
</table>
Table 2. MavBase block parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_o )</td>
<td>1.32 kg</td>
<td></td>
<td>MAV total mass</td>
</tr>
</tbody>
</table>
| \( J \)   | \[
\begin{bmatrix}
0.0128 & 0 & 0 \\
0 & 0.0128 & 0 \\
0 & 0 & 0.0239
\end{bmatrix}
\] | kgm²   | Inertia tensor    |
| \( I_{zim} \) | \(4.3011 \times 10^{-5}\) kgm² | Rotor moment of inertia |

Table 3. MavSimple block parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n )</td>
<td>4</td>
<td></td>
<td>Input size</td>
</tr>
<tr>
<td>( b )</td>
<td>(9.9865 \times 10^{-6})</td>
<td>N/m² rad²</td>
<td>Aerodynamic thrust constant</td>
</tr>
<tr>
<td>( d )</td>
<td>(1.5978 \times 10^{-7})</td>
<td>N/m² rad²</td>
<td>Aerodynamic drag constant</td>
</tr>
</tbody>
</table>
| \( A \)   | \[
\begin{bmatrix}
b & b & b & b \\
0.211 \cdot b & 0 & -0.211 \cdot b & 0 \\
0 & -0.211 \cdot b & 0 & 0.211 \cdot b \\
-d & d & -d & d
\end{bmatrix}
\] |             | Actuation matrix |

Figure 10. Altitude and attitude control simulation results
4 Simulation results

A simple altitude and attitude control system was designed in order to validate the designed classes. Altitude and attitude control simulation results are presented in Fig. 10. We notice that the system states have been stabilized within 1 second and that only very minor overshoots are present in the altitude $z$ and the pitch $\theta$.

The simulation example shows that the control results are very satisfactory, in particular the system suffers only a minor loss in altitude during the challenging reference orientation maneuver, which can be considered excellent control behaviour. Additionally, the simulation results are very similar to those obtained in (Bouabdallah et al., 2004b) and (Osmic et al., 2016) which suggests that the model derivation in this paper is correct.

5 Conclusion

This paper described how a generalized MAV modeling framework can be used to obtain any MAV model. A quadrotor based MAV was presented as an example, and the final model was formed by using its rigid body dynamics, the gyroscopic effect that influences the vehicles motion, and appropriate motor dynamics. To model the dynamics of any given MAV platform, it was shown that is only required to choose adequate parameter values, which correspond to the vehicle of interest, and inject them into the generalized MAV model.

Based on the presented generalized MAV model derivation, we have designed the following Modelica classes: MavBase, MavSimple and MavFull. MavBase represents the rigid body dynamics of the MAV including the gyroscopic effect. MavSimple extends the MavBase class and adds system actuation, while MavFull extends MavSimple with motor dynamics. These classes can be used to simulate the dynamic behaviour of any MAV within Modelica to any required level of detail, and thus providing similar functionalities as the Gazebo simulator RotorS (Furrer et al., 2016) which is frequently used for this purposes, but with a more user friendly interface.

Finally, we have validated the designed Modelica simulator through a simple altitude and attitude stabilization control system. Namely, we have obtained very similar control results like those currently present in the state of the art, which suggests that the generalized model derived and the MAV simulator designed in this paper are correct.

References


