Computationally Fast Dynamical Model of a SATCOM Antenna Suitable for Extensive Optimization Tasks

J. Najman*, M. Bastl, M. Appel and R. Grepl
MECHLAB, ISMMB, FME, Brno University of Technology, Czech Republic

The manuscript was received on 10 July 2018 and was accepted after revision for publication on 17 December 2018.

Abstract:
Development of new SATCOM (Satellite Communications) antennas and their control systems is a complex problem, in which strict requirements for mechanical design, drives, sensors and the overall speed and accuracy of the control algorithm must be met. Therefore, a simulation model is often used at different stages of development, which greatly accelerates the process of designing and optimizing the whole system. The computationally most demanding part of the simulation is the dynamical model of antenna. This article proposes an alternative approach for dynamical model creation, and the results are compared to a model created by using a universal multibody simulation environment. It is shown that the proposed new approach gives nearly the same-quality results and it is several times computationally faster.

Keywords:
mobile SATCOM, antenna pointing, dynamical model, simulation, MATLAB, Simulink

1. Introduction
Ensuring a reliable communication is an important task, not only in the civilian sector, but especially in military applications [1]. Communication antenna placement on a moving vehicle (land or sea), along with strict requirements for satellite tracking accuracy [2], puts high demands on the antenna’s mechatronic system – mechanical design, drives, sensors, data fusion and control algorithms [3, 4].

An appropriate choice of sensors, control system topology, and data processing is not a solved problem and technical development and research are still ongoing [5].

This article deals with the design of a simplified antenna dynamics model for designing and dimensioning of the mechanical construction of a real antenna system. The model also serves for sensors and control system testing with the use of Model-Based
Design [6, 7]. Testing of control [8] and signal processing algorithms, integrated Inertial Measurement Units [9-12], other MEMS (MicroElectroMechanical Systems) sensors outputs, as well as the optimization calculations of mechanical design, can be greatly accelerated by executing simulations on the antenna and sensors model. The essential requirements for the created antenna system model are sufficient accuracy, minimum parameters, and especially low computational complexity.

The results described in this article are based on the collaborative research effort of Mechatronics laboratory at Brno University of Technology and PROFEN Communication Technologies, Inc. [13].

1.1. System Description
To successfully lock onto a satellite, it is necessary to ensure the movement of the antenna in two axes – azimuth (yaw) and elevation (pitch). However, in such configuration, there may be problems with a mechanical singularity in the area where the tracked satellite is directly above the antenna. Therefore, it is preferable to add a third axis – cross-level (Fig. 1), which provides an additional degree of freedom to the system and solves the problem of mechanical singularities.

In practice, the individual axes are usually implemented using BLDC (Brushless DC motor) drives and belt transmission. The system must be mechanically well balanced and designed with very low passive resistance in order to use as small drives as possible.

To achieve the required pointing accuracy of the antenna (pointing error of less than 0.2° [2]), the system should have a minimal backlash. In addition, it is necessary to select sufficiently precise and fast sensors for antenna orientation measurement. Typically, a combination of encoders (relative or absolute), compass, GPS, IMU (Inertial Measurement Unit) or AHRS (Attitude Heading Reference System) is used.

![Fig. 1 Visualization of 3 DOF (Degrees of Freedom) antenna (1 – azimuth axis, 2 – cross-level axis, 3 – elevation axis)](image)

1.2. Objectives
Since the development of more advanced control algorithms requires many tests during which a real system cannot be used, the computational demands of the simulation model used are one of the main aspects affecting the development speed.

The standard approach is the use of the theory for Multi-Body-System dynamics [14], leading to the following equation:
\[ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + b(\dot{q}) + g(q) = u. \]  

where \( M \) is the inertial matrix, \( C \) gives the Coriolis and centrifugal force terms, \( b \) is related to dissipative forces, \( g \) includes gravity terms, \( u \) is the vector of actuator torques.

This equation contains a cross-coupling that is not significant in our case and the derivation of individual matrices can be non-trivial. The use of complex multibody simulation tools can be advantageous because they automatically assemble all the necessary differential equations for the specified mechanism. However, a disadvantage is the unnecessarily large complexity of the resulting dynamical equations and the number of parameters used that are not needed for a simple mechanism such as this.

The goal of this paper is to create a simplified 3× SISO (single-input and single-output) antenna dynamical model that will be computationally faster than the multibody model and will produce the same results at the same time.

2. Methods

2.1. Physical Model in Simulink/SimMechanics

A model that includes full nonlinear antenna dynamics can be created in Simscape Multibody™. An advantage over modelling with manually assembled differential equations is the design speed – the user defines only the geometry and inertial properties, and the dynamical model is generated automatically. This model (Fig. 2) is used as a reference, both in terms of accuracy and computational difficulty. The figure shows a complete forward-dynamics model for three axes.

![Physical model in Simulink/SimMechanics](image)

*Fig. 2 Physical model in Simulink/SimMechanics used as a reference*
For further demonstration purposes, we set the model parameters according to Tab. 1.

Tab. 1 Parameters used for Simulink model

<table>
<thead>
<tr>
<th>Global parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>T [N m]</td>
<td>10</td>
</tr>
<tr>
<td>b [N ms]</td>
<td>0.2</td>
</tr>
<tr>
<td>torque_input [N m] (same for all axes)</td>
<td>11⋅sin(0.4πt)</td>
</tr>
<tr>
<td>gravity vector [m s(^{-2})]</td>
<td>[0 −9.81 0]</td>
</tr>
<tr>
<td>Analysis mode</td>
<td>Forward dynamics</td>
</tr>
<tr>
<td>Solver step [s]</td>
<td>0.001</td>
</tr>
<tr>
<td>Solver type</td>
<td>ode4 (Runge-Kutta)</td>
</tr>
</tbody>
</table>

Azimuth axis

| Revolute – Axis of Action                  | [0 1 0] (reference CS = World) |
| Body – Mass [kg]                          | 1                                 |
| Body – Inertia [kg m\(^2\)]              | eye(3)                            |
| Body – Position [m]                       | CG [0 0 0]  
|                                            | CS1 [0 0 0]  
|                                            | CS2 [0 0 0 4 0]                      |

CrossLevel axis

| Revolute – Axis of Action                  | [1 0 0] (reference CS = World) |
| Body – Mass [kg]                          | 1                                 |
| Body – Inertia [kg m\(^2\)]              | eye(3)                            |
| Body – Position [m]                       | CG [0 0 0]  
|                                            | CS1 [0 0 0]  
|                                            | CS2 [0 0 0]                      |

Elevation axis

| Revolute – Axis of Action                  | [0 0 1] (reference CS = World) |
| Body – Mass [kg]                          | 1                                 |
| Body – Inertia [kg m\(^2\)]              | eye(3)                            |
| Body – Position [m]                       | CG [0 0 0]  
|                                            | CS1 [0 0 0]  
|                                            | CS2 [0 0 0]                      |

2.2. Overview of the Simplified 3× SISO Model and SISO Dynamics

The design of the SISO model is based on the assumption that the cross-coupling effect will be minimal and thus the dynamics and kinematics of the antenna can be separated. The dynamics will then be simulated by three independent SISO models and the kinematics will be computed afterward (Fig. 3).

A dynamic model of each drive axis of the antenna must include the effects of a BLDC motor (with a gearbox) and belt transmission, bearings, and the mass of the entire kinematic chain further away from the joint.

The whole complex dynamics can be simplified into the form of:

\[
J_i \ddot{q}_i + b_i \dot{q}_i + T_i \text{sign}(\dot{q}_i) = \tau_i, \quad i = 1...3, \tag{2}
\]

where \(q\) is the generalized coordinate (in this case, the angle of rotation), \(J\) is the moment of inertia, \(b\) is the viscous friction, \(T\) is the dry (Coulomb) friction, \(\tau\) is the control torque.
and $i$ is the axis number (azimuth, elevation, cross-level). All of these parameters are always related to one of these three pivot axes.

2.3. Forward Kinematics

We follow the movement of the antenna in the Cartesian coordinate system; we are interested in the angles of roll $\phi$, pitch $\vartheta$, and yaw $\psi$. Therefore, the goal of fkine is to calculate the angles $\phi$, $\vartheta$ and $\psi$ with the known $q_1$, $q_2$, $q_3$. This kinematics can easily be deduced using the standard notation of Denavith-Hartenberg (DH) parameters, so we have to define antenna coordinate systems and then perform the coordinate offset (Fig. 4, Tab. 2).
Tab. 2 Denavit-Hartenberg (DH) parameters for the definition of forward kinematics

<table>
<thead>
<tr>
<th>joint</th>
<th>a</th>
<th>d</th>
<th>(\alpha) [rad]</th>
<th>offset [rad]</th>
</tr>
</thead>
<tbody>
<tr>
<td>azimuth axis (q_1)</td>
<td>0</td>
<td>(d_1)</td>
<td>(\pi/4)</td>
<td>(\pi/2)</td>
</tr>
<tr>
<td>cross-elevation axis (q_2)</td>
<td>0</td>
<td>0</td>
<td>(-\pi/2)</td>
<td>(\pi/2)</td>
</tr>
<tr>
<td>elevation axis (q_3)</td>
<td>0</td>
<td>0</td>
<td>(\pi/2)</td>
<td>(-\pi/4)</td>
</tr>
</tbody>
</table>

Based on DH parameters, we calculate partial transformation matrices:

\[
T_{i-1,i'} = 
\begin{bmatrix}
\cos q_i & -\sin q_i & 0 & 0 \\
\sin q_i & \cos q_i & 0 & 0 \\
0 & 0 & 1 & d_i \\
0 & 0 & 0 & 1
\end{bmatrix},
\quad \alpha_i = 
\begin{bmatrix}
1 & 0 & 0 & a_i \\
0 & \cos \alpha_i & -\sin \alpha_i & 0 \\
0 & \sin \alpha_i & \cos \alpha_i & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]

The complete transformation matrix of the antenna is calculated as:

\[
T_A = f(q) = \prod_{i=1}^{3} T_{i-1,i} T_{i',i}.
\]

The angles \(\phi, \theta, \psi\) are then calculated from the transformation matrix as follows:

\[
\phi = \arctan \frac{t_{21}}{t_{11}}, \quad \theta = \arctan \frac{-t_{31}}{\sqrt{t_{32}^2 + t_{33}^2}}, \quad \psi = \arctan \frac{t_{32}}{t_{33}}.
\]

During implementation, the function atan2 is used to solve the problem of dividing by zero.

The direct kinematic model for speeds is formulated with the knowledge of the matrix structure (2) as:

\[
\dot{T}_{i-1,i'} = \dot{T}_{i-1,i} T_{i',i} = T_{i-1,i} D_q q_i T_{i',i},
\]

where \(D_q\) is the constant differential operator:

\[
D_q = 
\begin{bmatrix}
0 & -1 & 0 & 0 \\
1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{bmatrix}.
\]

Angular speeds \(\dot{\phi}, \dot{\theta}, \dot{\psi}\) are determined from the speed matrix:

\[
V_A = T_A^{-1} \dot{T}_A.
\]

This is how we formulated direct kinematic models for position and speed:

\[
[\phi_A, \theta_A, \psi_A] = f(T_A) = f(q),
\]

\[
[\dot{\phi}_A, \dot{\theta}_A, \dot{\psi}_A] = f(T_A, \dot{T}_A) = f(q, \dot{q}).
\]

2.4. Disturbance and Sensor Modelling

The total transformation matrix \(T_A\) derived in the previous chapter shows the transformation of the coordinate system to the base of the antenna. This relation could therefore
be used when the base of the antenna is stationary to the global coordinate system – for example, a ground static station.

In the case of an antenna on a moving vehicle (ship, car, ...), it is necessary to add a transformation matrix expressing the movement of the antenna base to the global coordinate system $\mathbf{T}_w$. The input of this matrix is the movement of the vehicle that we perceive as a fault and is denoted as $\mathbf{w}$.

$$\begin{bmatrix} \phi_w, \dot{\phi}_w, \psi_w \end{bmatrix} = f(\mathbf{T}_w) = f(\mathbf{w}) , \quad (11)$$

$$\begin{bmatrix} \dot{\phi}_w, \ddot{\phi}_w, \dot{\psi}_w \end{bmatrix} = f(\mathbf{T}_w, \dot{\mathbf{T}}_w) = f(\mathbf{w}, \dot{\mathbf{w}}) . \quad (12)$$

The influence of disturbance on the orientation of the end effector is obtained using the Eq. (13):

$$\mathbf{T} = \mathbf{T}_w \mathbf{T}_A . \quad (13)$$

The content of the transform matrix $\mathbf{T}_w$ is dependent on how the disturbance entering the system is simulated.

The last part of the system is modelling of sensors measuring angles, angular velocities or other quantities. Sensor models are again highly dependent on their location, type, and parameters such as accuracy, drift, signal-to-noise ratio, etc.

These parts of the system are next to the models of forward dynamics and kinematics of the antenna itself and thus will not affect the content of this article.

3. Results

The resulting model, the individual parts of which were presented in the previous text, is implemented in Simulink (Fig. 6).

![Fig. 6 Resulting SISO Model implemented in Simulink](image-url)
The comparison will be made at the level of the joint coordinates (azimuth, cross-level, elevation), which is the equivalent comparison in RPY coordinates. The accuracy of the SISO model compared to the reference model in SimMechanics (Fig. 2) is compared in the simulation of the harmonic input of the moments on the individual axes.

The model parameters (global) for the simulation are the same as in Tab. 2 with one exception in the inertia parameter for each axis. Because of the different approach to modelling these axes, their inertia must also account for the rest of the bodies, connected to them.

The results show a relatively significant influence of dry friction (Figs 7, 8) and very good agreement of both models. The time consumption was tested for both models by repeatedly running the simulation with the same parameters using MATLAB script with time measurement. Tab. 3 contains a comparison of the mean values. The computational demands of the SISO model are about 7 times smaller (Tab. 3).

<table>
<thead>
<tr>
<th>model</th>
<th>computational time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference SimMechanics</td>
<td>1.53</td>
</tr>
<tr>
<td>simplified SISO</td>
<td>0.22</td>
</tr>
</tbody>
</table>

Fig. 7 Comparison of velocity calculated using reference model (SimMechanics) and the simplified SISO model

4. Conclusion
A comparison with the model in SimMechanics clearly shows that the simulation results of both models are almost the same (Figs 7, 8). On the other hand, the computational demands of the newly proposed simplified model are significantly lower (Tab. 3).
Fig. 8 Comparison of positions (azimuth, cross-level, and elevation axis) calculated using the reference model (SimMechanics) and the simplified SISO model.

The created model allows the use of extensive optimization algorithms where various structures and parameters of both sensor algorithms and control algorithms can be compared. An example of such a complex optimization and algorithm selection is a comparison of a complementary filter for data fusion from gyroscopes and accelerometers with the more demanding the Kalman filter.

Acknowledgment

This work is an output of project NETME CENTRE PLUS (LO1202) created with financial support from the Ministry of Education, Youth and Sports under the “National Sustainability Programme I”.

References


