AVOIDING GIMBAL LOCK IN A TRAJECTORY SIMULATION

David Hosier

July 2018

U.S. ARMY ARMAMENT RESEARCH, DEVELOPMENT AND ENGINEERING CENTER
Munitions Engineering Technology Center
Picatinny Arsenal, New Jersey
The views, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

The citation in this report of the names of commercial firms or commercially available products or services does not constitute official endorsement by or approval of the U.S. Government.

Destroy by any means possible to prevent disclosure of contents or reconstruction of the document. Do not return to the originator.
AVOIDING GIMBAL LOCK IN A TRAJECTORY SIMULATION

A physics-based trajectory simulation can experience the phenomenon of gimbal lock when the flight path angle of the projectile approaches vertical. Methods of gimbal lock avoidance have been developed in the past, but often rely on complicated mathematical constructs. A simple working solution for gimbal lock avoidance is discussed in this report, largely for use in three-degree-of-freedom trajectory simulations, to aid in the maneuverability calculations for guided projectiles.
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>Analysis</td>
<td>1</td>
</tr>
<tr>
<td>Coordinate Systems</td>
<td>1</td>
</tr>
<tr>
<td>Rotation Matrices</td>
<td>2</td>
</tr>
<tr>
<td>Trajectory Simulations</td>
<td>4</td>
</tr>
<tr>
<td>Gimbal Lock</td>
<td>4</td>
</tr>
<tr>
<td>Test Case</td>
<td>5</td>
</tr>
<tr>
<td>Discussion</td>
<td>7</td>
</tr>
<tr>
<td>Conclusions</td>
<td>7</td>
</tr>
<tr>
<td>References</td>
<td>9</td>
</tr>
<tr>
<td>Distribution List</td>
<td>11</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

The author wishes to express his gratitude to Dr. Thomas Recchia and Yin Chen for their patience and insightfulness in the discussions that led to this report.
INTRODUCTION

A trajectory simulation can encounter gimbal lock under specific flight conditions, leading to incomplete or incorrect trajectory information. The background topics of coordinate systems and rotation matrices will be discussed in this report, and a method of gimbal lock avoidance will be introduced.

ANALYSIS

Coordinate Systems

A coordinate system is a concept that describes dimensional space. In three-dimensional (3D) space, a coordinate system has three axes: x, y, and z. It is often helpful if these axes follow the right-hand rule, which is a common mnemonic, stating that using a person's right hand, the right thumb points along the z-axis, and the curl of the fingers represents a motion from the x-axis to the y-axis. This can be represented mathematically with the vector cross product:

\[ \mathbf{x} = \mathbf{y} \times \mathbf{z} \]  \hspace{1cm} (1)

\[ \mathbf{y} = \mathbf{z} \times \mathbf{x} \]  \hspace{1cm} (2)

\[ \mathbf{z} = \mathbf{x} \times \mathbf{y} \]  \hspace{1cm} (3)

A common coordinate system reference can be defined by combining two of the four cardinal directions with either “up” or “down,” as in north (x-axis), up (y-axis), and east (z-axis). In fact, there are 24 ways of defining a coordinate system in this manner, eight of the most common of which are shown in table 1. For clarity, it is often desirable to choose a coordinate system where the yaw angle, when viewed from above, follows the positive clockwise sense of a compass. Other times, it can be more intuitive to have a pitch angle which, when viewed from the side, is positive in the upward direction. No matter which coordinate system is chosen, as long as the conventions are followed properly, the math works out the same.

<table>
<thead>
<tr>
<th>X-axis</th>
<th>Y-axis</th>
<th>Z-axis</th>
<th>Abbreviation</th>
<th>Pitch</th>
<th>Yaw</th>
</tr>
</thead>
<tbody>
<tr>
<td>North</td>
<td>Up</td>
<td>East</td>
<td>NUE</td>
<td>Up</td>
<td>Counterclockwise</td>
</tr>
<tr>
<td>North</td>
<td>West</td>
<td>Up</td>
<td>NWU</td>
<td>Down</td>
<td>Counterclockwise</td>
</tr>
<tr>
<td>North</td>
<td>Down</td>
<td>West</td>
<td>NDW</td>
<td>Down</td>
<td>Clockwise</td>
</tr>
<tr>
<td>North</td>
<td>East</td>
<td>Down</td>
<td>NED</td>
<td>Up</td>
<td>Clockwise</td>
</tr>
<tr>
<td>East</td>
<td>Up</td>
<td>South</td>
<td>EUS</td>
<td>Up</td>
<td>Counterclockwise</td>
</tr>
<tr>
<td>East</td>
<td>North</td>
<td>Up</td>
<td>ENU</td>
<td>Down</td>
<td>Counterclockwise</td>
</tr>
<tr>
<td>East</td>
<td>Down</td>
<td>West</td>
<td>EDW</td>
<td>Down</td>
<td>Clockwise</td>
</tr>
<tr>
<td>East</td>
<td>South</td>
<td>Down</td>
<td>ESD</td>
<td>Up</td>
<td>Clockwise</td>
</tr>
</tbody>
</table>

A vector in a 3D coordinate system can be described by angular rotations performed in a specific order about the three coordinate system axes. It is often useful in aerospace applications to refer to these rotation angles (or Euler angles) as roll, pitch, and yaw. The roll axis goes from the back to the front of an aircraft, pitch goes from side-to-side, and yaw goes from top to bottom.
positive direction for each axis may be reversed in some cases according to the right-hand rule. In certain coordinate system representations, pitch is about the y-axis and yaw is about the z-axis. In other coordinate systems, this relationship is reversed.

For the NUE coordinate system, pitch is about the z-axis, and yaw is about the y-axis. For a vector \([x, y, z]\) in NUE, the rotation angles about each of the axes are defined by equations 4, 5, and 6. Note that the angle about the pitch axis is defined differently than the other angles due to the order in which the rotation matrices are multiplied together.

\[
\begin{align*}
\theta_x &= \tan^{-1}\left( \frac{z}{y} \right) \\
\theta_y &= \tan^{-1}\left( \frac{-z}{x} \right) \\
\theta_z &= \tan^{-1}\left( \frac{y}{\sqrt{x^2 + z^2}} \right)
\end{align*}
\]

Rotation Matrices

Rotating a vector from one coordinate system to a different coordinate system is done via a rotation matrix. To rotate a 3D vector from one coordinate system to another, a series of three rotation matrices are multiplied in a specific order to create one final rotation matrix. The order of these rotations, according to a common convention used in many aerospace applications, is first yaw, then pitch, and lastly roll. The individual rotation matrices about the three coordinate system axes (not necessarily roll, pitch, and yaw) are:

\[
R_x = \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos \theta_x & -\sin \theta_x \\
0 & \sin \theta_x & \cos \theta_x
\end{bmatrix}
\]

\[
R_y = \begin{bmatrix}
\cos \theta_y & 0 & \sin \theta_y \\
0 & 1 & 0 \\
-\sin \theta_y & 0 & \cos \theta_y
\end{bmatrix}
\]

\[
R_z = \begin{bmatrix}
\cos \theta_z & -\sin \theta_z & 0 \\
\sin \theta_z & \cos \theta_z & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

A slightly different, but mathematically identical, way to create the individual rotation matrices is by using trigonometric definitions in place of trigonometric functions. For the NUE coordinate system, equations 10, 11, and 12 are listed. Note that these equations differ depending on the coordinate system; therefore, this method is somewhat suboptimal.
Another method of calculating the total rotation matrix is with a direction cosine matrix. This method requires additional information, namely a reference vector with constant orientation, such as gravity ($\vec{g}$). For the NUE coordinate system, the cross product between the vector $[x, y, z]$ and gravity is calculated to identify the side-to-side axis. A second cross product identifies the down-to-up axis. A final concatenation operation combines these axes into a matrix:

$$
\hat{i} = \frac{[x, y, z]}{\sqrt{x^2 + y^2 + z^2}}
$$

$$
\hat{g} = [0, -9.8, 0]
$$

$$
\hat{g} = \frac{\hat{g}}{|\hat{g}|}
$$

$$
\hat{k} = \hat{g} \times \hat{i}
$$

$$
\hat{j} = \hat{k} \times \hat{i}
$$

$$
R_{total} = \begin{bmatrix}
\hat{i}^T \\
\hat{j} \\
\hat{k}
\end{bmatrix}
$$

A final, slightly less intuitive, method to rotate a vector in 3D space is by using quaternions. Each single-axis rotation is defined by a quaternion based on the angle about that axis. Quaternions
use their own unique functions for operations such as multiplication and inversion, so to simplify things, the Matlab code for a three-axis rotation in NUE is shown in equation 22.

\[
Q_x = [\cos \frac{\theta_x}{2}, \sin \frac{\theta_x}{2}, 0, 0] 
\]

\[
Q_y = [\cos \frac{\theta_y}{2}, 0, \sin \frac{\theta_y}{2}, 0] 
\]

\[
Q_z = [\cos \frac{\theta_z}{2}, 0, 0, \sin \frac{\theta_z}{2}] 
\]

\[
\text{quatrotate}(\text{quatinv}(\text{quatmultiply}(\text{quatmultiply}(Q_x, Q_z), Q_y)), [x, y, z]) 
\]

**Trajectory Simulations**

A trajectory simulation is a physics-based mathematical model that propagates an object’s physical state forward in time. The coordinate system defines the directions in which the equations of motion operate. For a three-degree-of-freedom (3-DOF) point mass trajectory simulation, the force of gravity acts in the earth frame while the forces of drag and lift operate in the body frame. The total force can be calculated in either the earth frame or the body frame as follows, noting that the “T” superscript refers to the matrix transpose:

\[
F_{\text{earth}} = R_{\text{roll}}R_{\text{pitch}}R_{\text{yaw}}(F_{\text{drag}} + F_{\text{lift}}) + F_{\text{gravity}} 
\]

\[
F_{\text{body}} = F_{\text{drag}} + F_{\text{lift}} + (R_{\text{roll}}R_{\text{pitch}}R_{\text{yaw}})^TF_{\text{gravity}} 
\]

A six-degree-of-freedom (6-DOF) trajectory simulation uses the same coordinate system convention with the addition of rotational moments that act about a center of gravity. One major difference between a 6-DOF and a 3-DOF is that a 6-DOF obtains Euler angles from the angular rates that are calculated at each time step while a 3-DOF obtains Euler angles from the velocity vector, with the assumption that the projectile body axis will always be aligned to the velocity vector. This assumption allows the 3-DOF to calculate a rotation matrix directly from the velocity vector, which is a helpful simplification of the physics but one that can cause the problem discussed in the next section.

**Gimbal Lock**

An issue that can arise during the formation of a rotation matrix is called gimbal lock (ref. 1), referring to the loss of one degree of freedom in a three-axis gimbal when two gimbals rotate about the same axis. This issue occurs mainly in 3-DOFs when the flight path angle approaches ±90 deg in the presence of a force in the body frame normal to the velocity vector, such as a lift force. In simple terms, the projectile is unable to determine which direction is up and which direction is to the side because of a discontinuity in the mathematical functions that make up any of the variations of rotation matrix. Put another way, when the projectile is pointed down, which way is up? Each formulation of rotation matrix mentioned previously experiences gimbal lock.
There are at least three commonly-suggested solutions to the problem of gimbal lock in a trajectory simulation. One is to use quaternions. This solution works well in a 6-DOF, but quaternions are unable to solve gimbal lock in a 3-DOF because the quaternion rotation matrix is formed using Euler angles, which is where the discontinuity arises in the first place (ref. 2).

A second method is to use a series of rules when setting up the rotation matrix. For example, the angle about the y-axis can be defined differently depending on whether the x-value of the velocity vector is positive or negative. The problem with this method is that there is a tendency for a rule to solve the problem in one test case while exacerbating the problem in a different test case. Plus, one rule tends to lead to another rule, which leads to another rule, etc. The complication of this method makes it largely untenable.

A third method is to switch coordinate systems at or near the trigonometric discontinuities. For example, the simulation can use the NUE coordinate system until the angle about the z-axis approaches -90, then switch to down-north-east. This method has been shown to work in a 6-DOF (ref. 3), but the author was unable to achieve a working solution in a 3-DOF.

Finally, the actual working method of avoiding gimbal lock is to form a direction cosine matrix using a position pointing vector instead of gravity. The pointing vector is simply the projectile’s current position subtracted from a target’s position. For scenarios that apply a lift force in a constant direction, the target should be a position in space in the direction of desired travel, preferably at a distance that is unattainably far away in order to avoid causing the velocity vector to be coincident with the position pointing vector. Equations 25 through 30 are nearly identical to the gravity-based direction cosine method, with some slight changes to the order of operations.

\[
\hat{i} = \frac{[x, y, z]}{\sqrt{x^2 + y^2 + z^2}} \quad (25)
\]

\[
\vec{r} = \vec{r}_{target} - \vec{r}_{projectile} \quad (26)
\]

\[
\vec{k} = \hat{i} \times \vec{r} \quad (27)
\]

\[
\hat{k} = \frac{\vec{k}}{|\vec{k}|} \quad (28)
\]

\[
\hat{j} = \hat{k} \times \hat{i} \quad (29)
\]

\[
R_{total} = \begin{bmatrix} \hat{i} & \hat{j} & \hat{k} \end{bmatrix}^T \quad (30)
\]

Test Case

A simple test case was set up to demonstrate the concept. An object with a mass of 10 kg and a diameter of 0.1 m, with an initial speed of 250 m/s at a flight path angle of 80 deg and a roll angle of zero, was subjected to a constant gravitational acceleration of 9.8 m/s² in air at a constant density of 1.225 kg/m³. The object had a drag coefficient of 0.3 and a lift coefficient of 0.9, and the lift force was applied in the downward direction starting at 20 sec in flight. A very simple trajectory simulation was run to calculate the ground impact point of the object. One simulation was run with a standard rotation matrix, and a second simulation was run using the direction cosine matrix based on a target position. Figure 1 shows the position of the object, specifically as it approaches vertical and...
either experiences gimbal lock or avoids it. Figure 2 shows the flight path angle of the object as it either gets locked into the angular discontinuity around -90 deg or simply passes through it.

![Graph showing height versus range of two trajectory calculation methods]

**Figure 1**
Height versus range of two trajectory calculation methods showing the appearance and the avoidance of gimbal lock
Figure 2
Flight path angle versus time showing one trajectory approaching and getting locked at -90 deg while another trajectory avoids it

Discussion

A key application of this technology is in the early development of highly-maneuverable projectiles. It is often desirable, in light of cost and time constraints, to quantify a projectile’s maneuverability in a realistic simulation environment without the complication and potential aerodynamic instability of a 6-DOF. The use of a 3-DOF reduces the amount of aerodynamic information required up front, and the method of gimbal lock avoidance discussed in this report can produce useful trajectory estimates for projectiles whose maneuverability enables the reversal of forward velocity.

CONCLUSIONS

An investigation into coordinate systems and rotation matrices yielded some elucidation as to the source of gimbal lock in trajectory simulations. A variety of solutions were attempted, finally settling on the direction cosine matrix formed with a target position vector. Its effectiveness was demonstrated with a simple three-degree-of-freedom trajectory simulation.
REFERENCES


DISTRIBUTION LIST

U.S. Army ARDEC
ATTN: RDAR-EIK
   RDAR-MEF-A, J. Grau
   W. Toledo
   E. Vazquez
   S. Chung
   T. Recchia
   Y. Chen
   D. Hosier (3)
   RDAR-MEF-P, C. Stout
Picatinny Arsenal, NJ  07806-5000

Defense Technical Information Center (DTIC)
ATTN: Accessions Division
8725 John J. Kingman Road, Ste. 0944
Fort Belvoir, VA  22060-6218
REVIEW AND APPROVAL OF ARDEC REPORTS

THIS IS A:

[X] TECHNICAL REPORT  [ ] SPECIAL REPORT

[ ] MEMORANDUM REPORT  [ ] ARMAMENT GRADUATE SCHOOL REPORT

FUNDING SOURCE
[e.g., TEX3; 6.1 (LIR, FTAS); 6.2; 8.3. PM funded EMD; PM funded Production/ESIP; Other (please identify)]

Avoiding Gimbal Lock in a Trajectory Simulation

Title

Project:

David Hosier

Author/Project Engineer

Report number/Date received (to be completed by LCSD)

1576 94

Extension Building

RDAR-MEF-A

Author's Office Symbol

PART 1. Must be signed before the report can be edited.

a. The draft copy of this report has been reviewed for technical accuracy and is approved for editing.

b. Use Distribution Statement A, B, C, D, E, or F for the reason checked on the continuation of this form. Reason:

1. If Statement A is selected, the report will be released to the National Technical Information Service (NTIS) for sale to the general public. Only unclassified reports whose distribution is not limited or controlled in any way are released to NTIS.

2. If Statement B, C, D, E, or F is selected, the report will be released to the Defense Technical Information Center (DTIC) which will limit distribution according to the conditions indicated in the statement.

c. The distribution list for this report has been reviewed for accuracy and completeness.

John C. Grau

Division Chief (Date)

PART 2. To be signed either when draft report is submitted or after review of reproduction copy.

This report is approved for publication.

John C. Grau

Division Chief (Date)

Andrew Pskowski

RDAR-GS (Date)

LCSD 49 (1 Sept 16)

Supersedes SMCAR Form 49, 20 Dec 06