**Coordinate Transformations for Unsteady Frames**

Johnathan Clouse[[1]](#footnote-1)

University of Colorado, Boulder, CO, 80309

A Frenet frame is a useful coordinate system to describe motion with respect to the motion of a particle. A system utilizing a ground station, mothership, and Micro Aerial Vehicle (MAV) is explored. MAV motion is determined by coordinate transformation to provide position, velocity, and acceleration with respect to the ground station. Error between direct numerical differentiation of the position and rotational kinematics is investigated.

# Nomenclature

= tangent vector of Frenet frame

= binormal vector of Frenet frame

= normal vector of Frenet frame

= position vector of Micro Aerial Vehicle (MAV) expressed in the ground station (GS) Cartesian frame

= velocity vector of MAV expressed in the GS Cartesian frame

= acceleration vector of MAV expressed in the GS Cartesian frame

*t* = time of measurement

*G* = fundamental metric tensor between two frames

**=** rotation rate between two frames

= angular acceleration between two frames

= MAV quantities expressed in the mothership (MS) Frenet frame

= MS quantities expressed in the GS Cartesian frame

# Introduction

T

HE Frenet frame is a useful set of coordinates centered on a particle. The basis vectors describe the motion of the particle, showing the instantaneous direction of motion and the instantaneous radius of curvature of the path1. This project focuses on the use of the Frenet frame on the paths of an unmanned aerial vehicle (UAV) mothership (MS) and Micro Aerial Vehicles (MAVs) deployed by the MS to act as sensory equipment1. The MAV motion with respect to the MS are a known quantity. Sensory data is then relayed from the MS to a ground station (GS), so coordinate transformations must be applied to obtain the MAV motion with respect to a Cartesian frame centered on the GS.

The Frenet frame for a particle on a path is obtained by defining the tangent vector as

One can see that the tangent vector is solely in the direction of motion. The binormal vector is defined as

Thus, the binormal vector is normal to both the direction of motion and the particle acceleration. It is important to note that this vector cannot be defined when the velocity and acceleration are collinear. Finally, the normal vector is found to complete the three-dimensional coordinate frame by

This normal vector is normal to the curve of the path, and so it is useful to find the normal component of a particle’s acceleration.

To find the Frenet basis vectors for a given instant, the first and second derivatives of the position vector must be known. An analytic expression for the position vector will easily yield an analytic expression for these derivatives, but it is not the case when only time-stamped position data are provided. In the latter case, one can forward-differentiate the *n*th data point by

This method will reduce the amount of data points by one for each derivative taken. For example, given five position vectors in time, one will end up with four velocity vectors and three acceleration vectors.

# Simulation of Known Analytical Path

Consider the path of the MS, with respect to the GS and expressed in a Cartesian frame1:

And the path of an MAV in the MS Frenet frame1:

Units are not given, so generic distance units (DU) and time units (TU) are used in the plots.

The derivatives required to obtain the Frenet bases are obtained by numerical forward differentiation, rather than analytically, to allow the script to handle discrete data from another source. For this analytical case, 1000 time-points are used. Figure 1 and Figure 2 show the positions of the mothership and MAV, with their Frenet basis vectors.

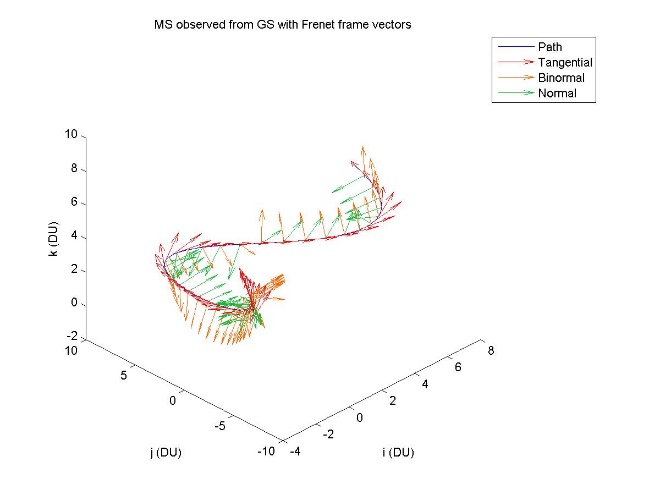


Figure :Mothership position wrt GS Cartesian, with its Frenet basis vectors

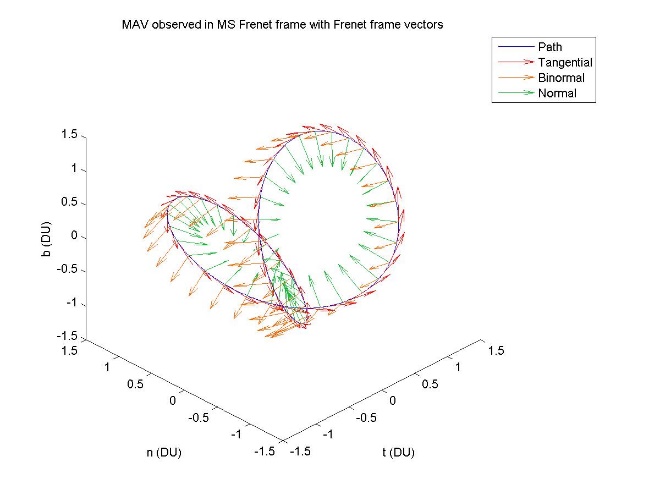


Figure : MAV position wrt MS Frenet frame, with its Frenet basis vectors

The mothership’s motion experiences an inflection in its path curvature, causing the normal and binormal Frenet vectors to change substantially. The MAV experiences a path with no inflections, so the change in the Frenet vectors is continuous.

The mothership’s speed and accelerations with respect to the GS Cartesian frame is shown in Figure 3 and Figure 4 below:

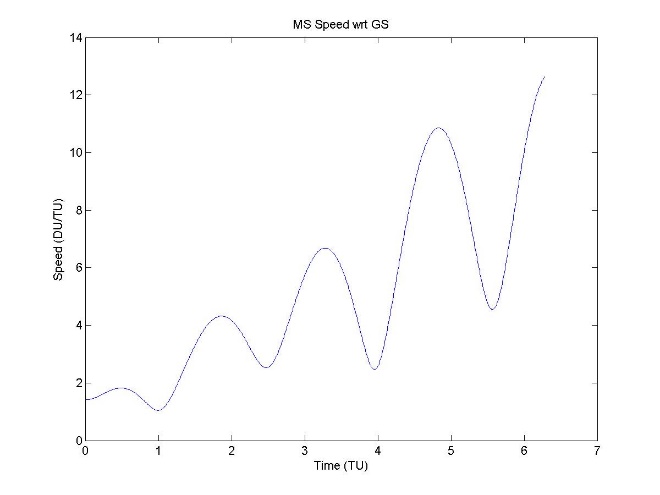


Figure : MS speed wrt GS

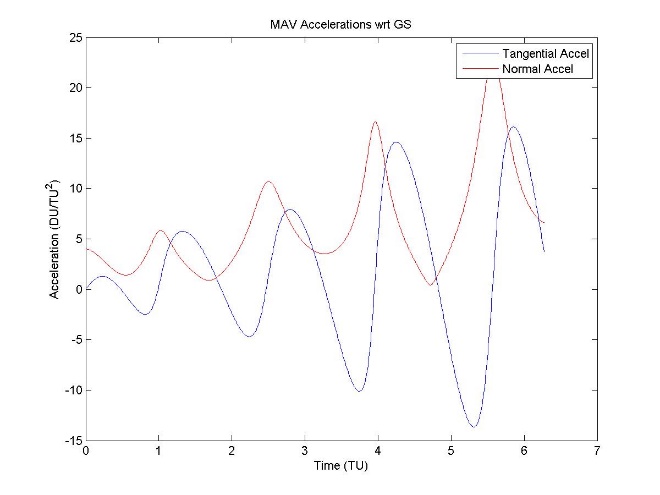


Figure : MS normal and tangential accelerations wrt GS

The MAV speed and accelerations with respect to the MS Frenet frame is shown in Figure 5 and Figure 6.

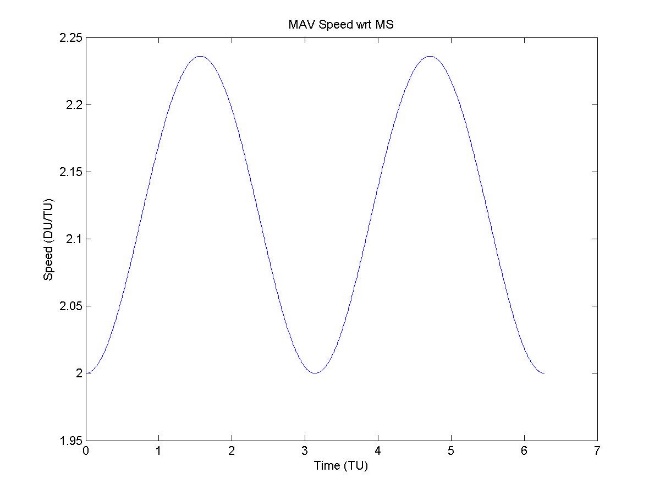


Figure : MAV speed wrt MS

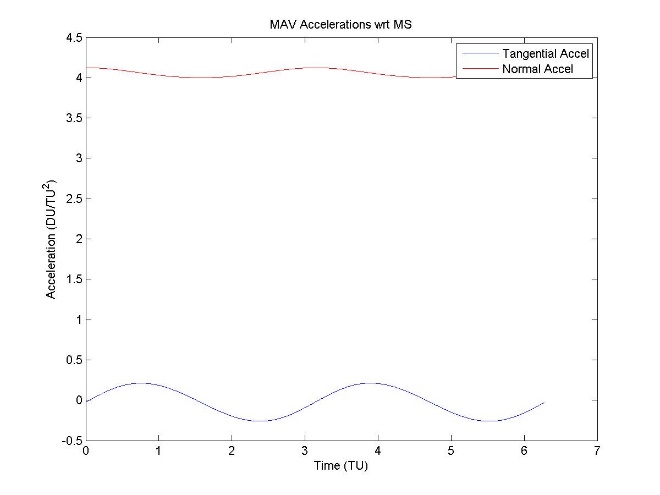


Figure : MAV accelerations wrt MS

The fundamental metric tensor between the GS Cartesian and MS Frenet frames is

(7)

where the barred vectors are for*i*=1,2,3 respectively and the Cartesian vectors are for *j*=1,2,3 respectively. The position of the MAV in the GS coordinate frame is then represented by

The resultant MAV positions are shown in Figure 7. There is a large change in MAV position in the vicinity of the MS inflection point.

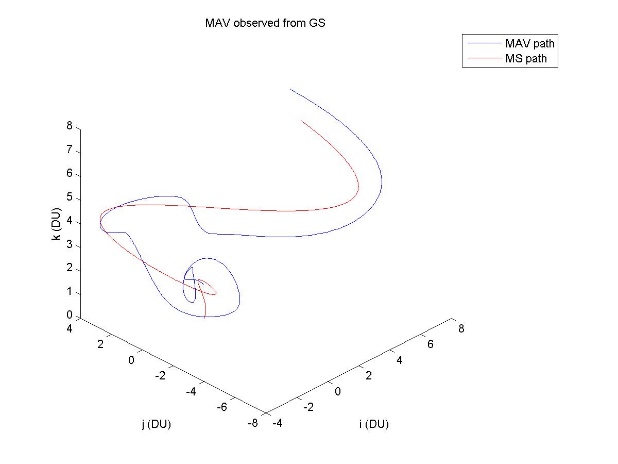
The MAV speed and acceleration with respect to the ground station frame are found by finding the rotational rate of the MS Frenet frame as seen by the 

Figure : MAV position in GS Cartesian frame

GS and using relative motion equations. The rotational rate of one frame with respect to another can be found by2

Between the GS Cartesian and MS Frenet frames, and the Euler angles are found by using the fundamental metric tensor2

The velocity in the GS Cartesian frame is calculated by

The acceleration in the GS Cartesian frame is

The angular velocity is the time-derivative of , which is

The application of this kinematic method can be seen in Figure 8 and Figure 9 for the speed calculation in both the rotating coordinate method and direct numerical differentiation of ***r***. Figure 10 shows the normal and tangential accelerations of the MAV in the GS Cartesian frame as calculated from the above kinematic equations; and Figure 11 shows the error between the kinematic solution and direct numerical differentiation of ***r***.

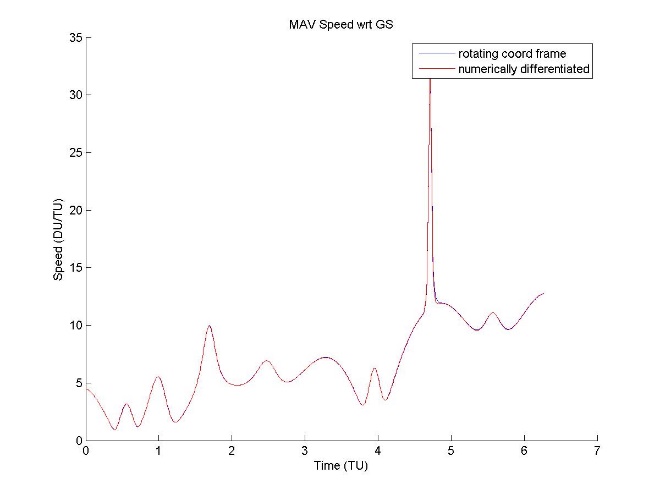


Figure :MAV speed in GS Cartesian frame

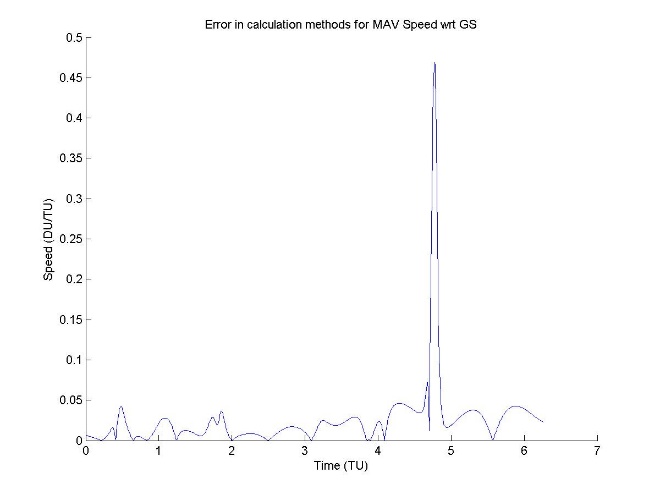


Figure : Error between methods for MAV speed wrt GS

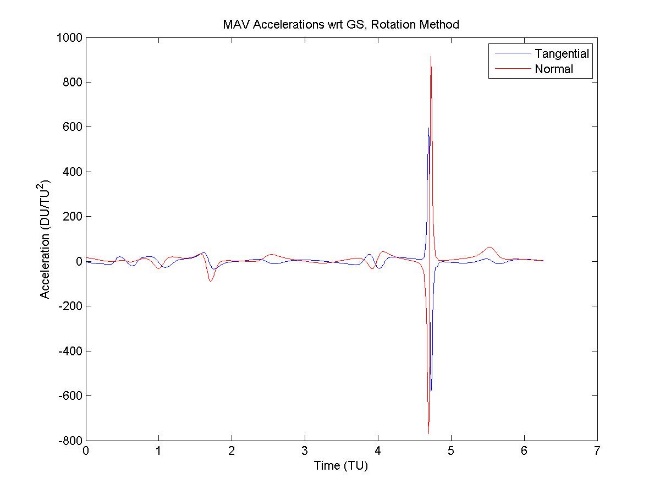


Figure :MAV accelerations in the GS Cartesian frame

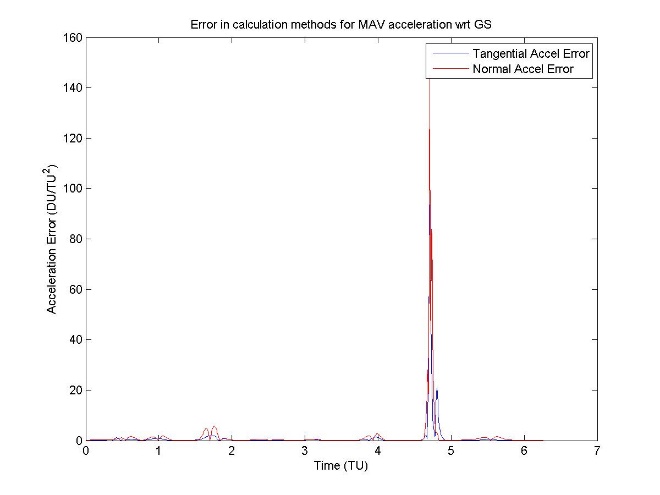


Figure : Error between methods for MAV accelerations.

From the above acceleration plots, it is clear that the inflection point of the MS affects the acceleration of the MAV viewed by the GS. The fast change in the mothership’s normal and binormal Frenet vectors require such a change in the MAV’s position as well, causing the spike in acceleration. The error between the two methods in acceleration calculation are also affected by this discontinuity.

# Discrete Data for MAV Position

Now, a data file with MAV positions in the MS Frenet frame is used to provide MAV position data; MS position is the same as the previous section. The data file lists 100 evenly-spaced data points for the same timescale as the last section. Figure 12 shows the MAV position in the MS Frenet frame, with its own Frenet vectors. Figure 13 shows the MAV position in the GS Cartesian frame. The speed and accelerations of the MAV can be seen in Figure 14 and Figure 15.

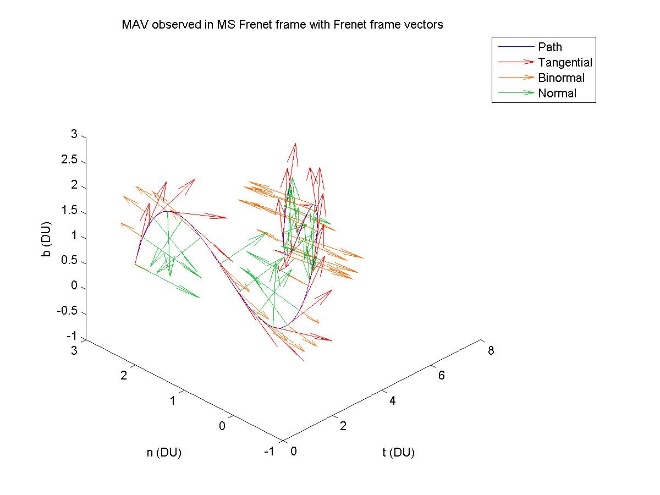


Figure : MAV position in MS Frenet frame, with Frenet vectors

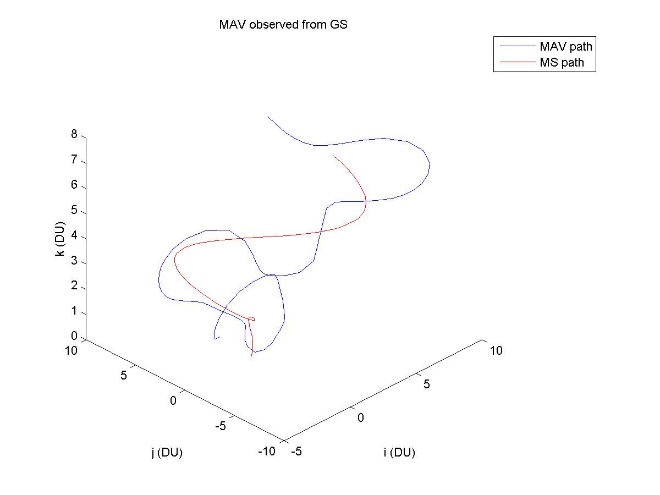


Figure : MAV position in GS Cartesian frame



Figure : MAV speed wrt MS

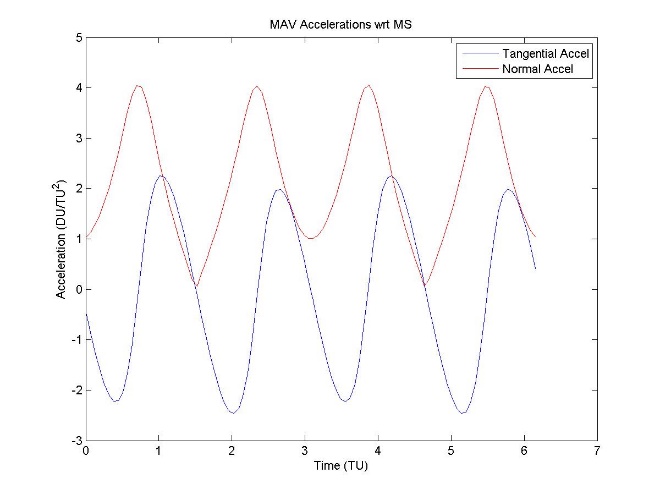


Figure : MAV accelerations wrt MS

The curvature of the MAV’s path in the MS Frenet frame experiences inflections in its own path this time. The MAV position in the GS Cartesian frame again experiences a large position change in the vicinity of the MS inflection.

The speed of the MAV with respect to the GS is found kinematically and numerically, and the results are displayed in Figure 16. The error between the two methods’ results are shown in Figure 17. The kinematically-computed accelerations of the MAV with respect to the GS are seen in Figure 18, and the numerically-differentiated result is shown in Figure 19. Figure 20 shows the error between the acceleration computations.

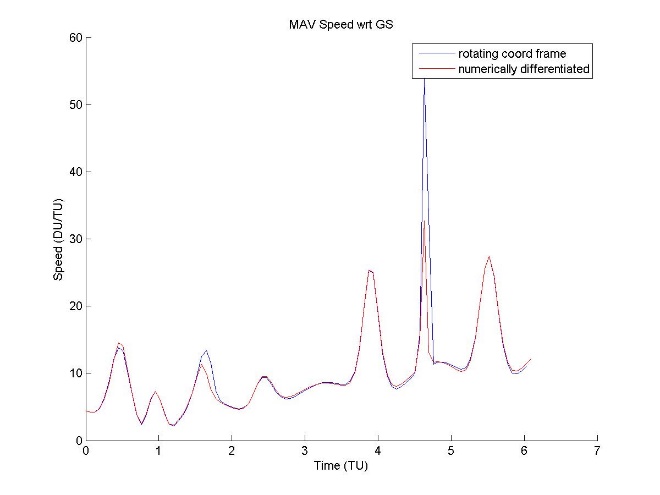


Figure : MAV speed wrt GS

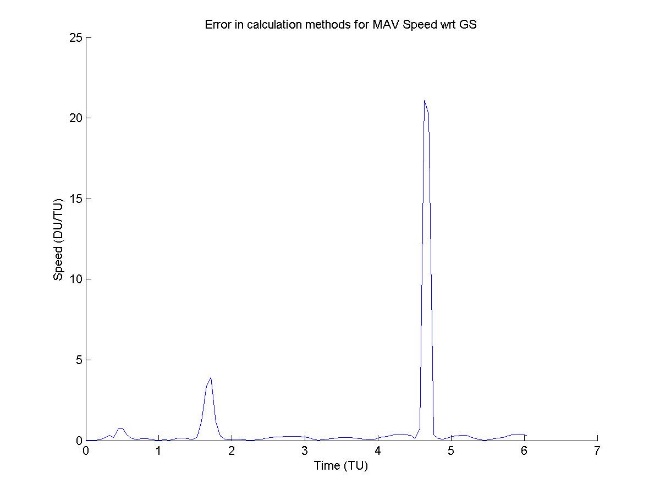


Figure : Error between methods for MAV speed wrt GS

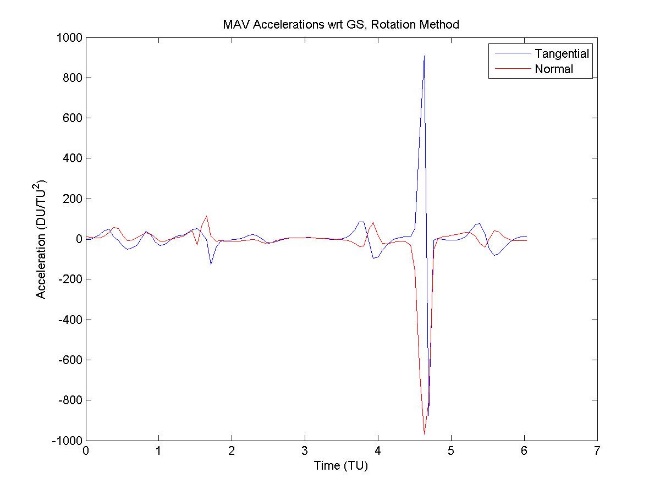


Figure : MAV accelerations wrt GS Cartesian frame, rotation method

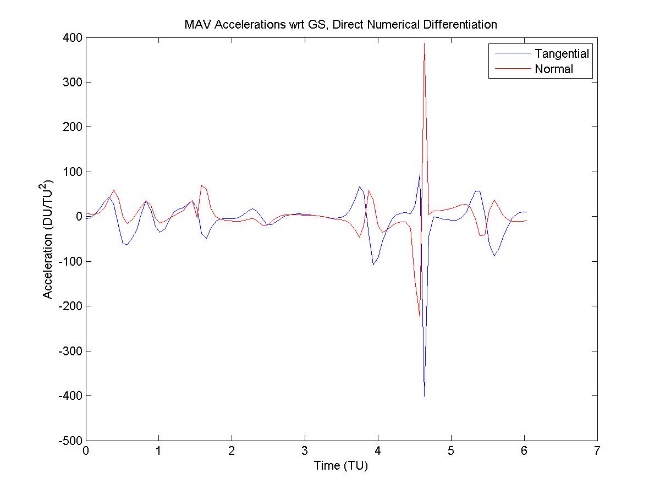


Figure : MAV accelerations wrt GS Cartesian frame, direct numerical differentiation

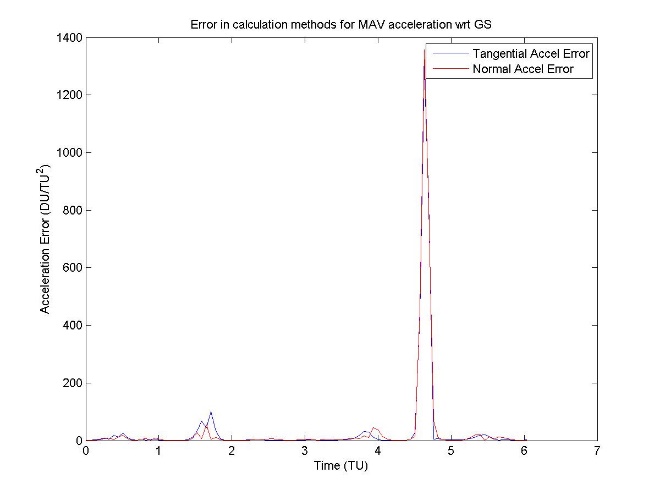


Figure : Error between methods for MAV accelerations.

The error between the kinematic and numerical differentiation methods does not track as well as before. This is due in part to the larger time step between the provided positions of the MAV, giving the numerical differentiation less granularity. The MS inflection point exacerbates the spike in the accelerations and the error calculation in the region. In fact, the accelerations are shown to increase in opposite directions between the two methods.

# Conclusion

The Frenet frame is a useful way to describe things with respect to a particle in motions, but there are caveats to its usage. First, the frame is undefined if the velocity and acceleration are collinear. One may work around this by fixing the basis vectors when the velocity and acceleration are nearly collinear, but care should be take that this does not cause different problems. Second, motion of a vehicle maintaining some position in the Frenet of another particle is at the mercy of the second particle’s motion. Inflections in the path could cause unsafe accelerations in the vehicle, or cause it to diverge from the desired path as it corrects its motion.

While a MAV may experience the aforementioned issues, an orbital mothership may be a better-suited use for the Frenet frame. Since two-body motion results in a continuously curved path, no inflections would be experienced and a secondary vehicle’s motion could easily be described.

Direct numerical differentiation of the MAV with respect to the ground station tracked well with rotational kinematics solution, but not when there were large changes in acceleration. Directly sensing the Euler angles would improve the kinematic solution’s performance. A lower timestep between the data points would also help drive the error down. However, the kinematic solution was computationally costly to solve, compared to the direct differentiation.

# References

1Hussein, M., “Project: Coordinate Transformations for Unsteady Frames,” Fall 2016.

2Hussein, M., “Project Supplement: Rotating Coordinate Frames: Euler Angles,” Fall 2016.

# Appendix

MATLAB code for the project follows. A significant amount of space was taken by the plot generation and saving.

%% ASEN 5227 Project

% John Clouse

%% Initialize

clear all

close all

hw\_pub.figWidth = 1120; % pixels

hw\_pub.figHeight = 840; % pixels

hw\_pub.figPosn = [0, 0, hw\_pub.figWidth, hw\_pub.figHeight];

scenarios = {'Part 1', 'Part 2'};

% Some color definitions

orange = [1.0,0.4,0.0];

lgreen = [20 187 51] ./ 255;

%% Loop through the analytical and discreet scenarios

% The mothership calculations are only done on the first part

% The big thing to remember here is that only forward differentiation is

% used, so each differentiation has one fewer element than the diff'ed

% quantity.

for scenario = scenarios

% get time for the analysis.

if strcmp(scenario, 'Part 1')

num\_pts = 1000;

time = linspace(0,2\*pi,num\_pts);

else

mav\_data = dlmread('mav\_data\_txt.txt');

time = mav\_data(:,4)';

num\_pts = length(time);

end

t\_diffs = (time(2:end) - time(1:end-1));

R = [time.\*cos(time); time.\*sin(2\*time); time];

V = [(cos(time)-time.\*sin(time)); ...

(sin(2\*time)+2\*time.\*cos(2\*time));...

1\*ones(1,num\_pts)];

A = [(-2\*sin(time)-time.\*cos(time));...

(4\*cos(2\*time)-4\*time.\*sin(2\*time));...

zeros(1,num\_pts)];

if strcmp(scenario, 'Part 1')

r\_bar = [cos(time); sin(2\*time); cos(2\*time)]; %Frenet

v\_bar = [-sin(time); 2\*cos(2\*time); -2\*sin(2\*time)];

a\_bar = [-cos(time); -4\*sin(2\*time); -4\*cos(2\*time)];

else

r\_bar = mav\_data(:,1:3)';

end

% Forward differentiation

% Mothership velocity

V\_forward = forward\_diff(R,time);

figure('Position', hw\_pub.figPosn);

hold on

plot(abs(V(1,1:end-1) - V\_forward(1,:)));

plot(abs(V(2,1:end-1) - V\_forward(2,:)),'r');

plot(abs(V(3,1:end-1) - V\_forward(3,:)),'k');

title('MS Velocity: Error Between Analytical and Numerical Solution')

% Mothership acceleration

A\_forward = forward\_diff(V\_forward,time);

figure('Position', hw\_pub.figPosn);

hold on

plot(abs(A(1,1:end-2) - A\_forward(1,:)));

plot(abs(A(2,1:end-2) - A\_forward(2,:)),'r');

plot(abs(A(3,1:end-2) - A\_forward(3,:)),'k');

title('MS Acceleration: Error Between Analytical and Numerical Solution')

v\_bar\_forward = forward\_diff(r\_bar,time);

a\_bar\_forward = forward\_diff(v\_bar\_forward,time);

if strcmp(scenario, 'Part 1')

% MAV velocity

figure('Position', hw\_pub.figPosn);

hold on

plot(abs(v\_bar(1,1:end-1) - v\_bar\_forward(1,:)));

plot(abs(v\_bar(2,1:end-1) - v\_bar\_forward(2,:)),'r');

plot(abs(v\_bar(3,1:end-1) - v\_bar\_forward(3,:)),'k');

title('MAV Velocity: Error Between Analytical and Numerical Solution')

% MAV acceleration

figure('Position', hw\_pub.figPosn);

hold on

plot(abs(a\_bar(1,1:end-2) - a\_bar\_forward(1,:)));

plot(abs(a\_bar(2,1:end-2) - a\_bar\_forward(2,:)),'r');

plot(abs(a\_bar(3,1:end-2) - a\_bar\_forward(3,:)),'k');

title(['MAV Acceleration:'...

'Error Between Analytical and Numerical Solution'])

end

% Mothership's local Frenet frame

t = zeros(3,num\_pts);

b = zeros(3,num\_pts);

n = zeros(3,num\_pts);

for ii = 1:num\_pts-2

% Analytically differentiated

% t(:,ii) = V(:,ii)/norm(V(:,ii));

% b(:,ii) = cross(V(:,ii),A(:,ii))/norm(cross(V(:,ii),A(:,ii)));

% n(:,ii) = cross(b(:,ii),t(:,ii));

% Numerically differentiated

t(:,ii) = V\_forward(:,ii)/norm(V\_forward(:,ii));

b(:,ii) = cross(V\_forward(:,ii),A\_forward(:,ii))...

/norm(cross(V\_forward(:,ii),A\_forward(:,ii)));

n(:,ii) = cross(b(:,ii),t(:,ii));

end

% plot the MS path

if strcmp(scenario, 'Part 1')

figure('Position', hw\_pub.figPosn);

plot3(R(1,:),R(2,:),R(3,:));

xlabel('i (DU)'); ylabel('j (DU)'); zlabel('k (DU)')

hold on

plot\_idx = [1:20:num\_pts num\_pts];

quiver3(R(1,plot\_idx),R(2,plot\_idx),R(3,plot\_idx),...

t(1,plot\_idx),t(2,plot\_idx),t(3,plot\_idx),'r');

quiver3(R(1,plot\_idx),R(2,plot\_idx),R(3,plot\_idx),...

b(1,plot\_idx),b(2,plot\_idx),b(3,plot\_idx),'color',orange);

quiver3(R(1,plot\_idx),R(2,plot\_idx),R(3,plot\_idx),...

n(1,plot\_idx),n(2,plot\_idx),n(3,plot\_idx),'color',lgreen);

xlabel('i (DU)'); ylabel('j (DU)'); zlabel('k (DU)')

title('MS observed from GS with Frenet frame vectors')

view([1,0,0])

saveas(gcf, ['Figures\' 'MS\_JK'],'jpg')

view([0,0,1])

saveas(gcf, ['Figures\' 'MS\_IJ'],'jpg')

view([-1,-1,1])

legend({'Path','Tangential','Binormal','Normal'});

saveas(gcf, ['Figures\' 'MS\_ISO'],'jpg')

end

% MAV's Frenet frame expressed in the MS Frenet frame

t\_MAV = zeros(3,num\_pts);

b\_MAV = zeros(3,num\_pts);

n\_MAV = zeros(3,num\_pts);

for ii = 1:num\_pts-2 % missing 2 points at end from 2 forward diffs

t\_MAV(:,ii) = v\_bar\_forward(:,ii)/norm(v\_bar\_forward(:,ii));

b\_MAV(:,ii) = cross(v\_bar\_forward(:,ii),a\_bar\_forward(:,ii))...

/norm(cross(v\_bar\_forward(:,ii),a\_bar\_forward(:,ii)));

n\_MAV(:,ii) = cross(b\_MAV(:,ii),t\_MAV(:,ii));

end

% Plot MAV path

if strcmp(scenario, 'Part 1')

plot\_idx = [1:20:num\_pts num\_pts];

else

plot\_idx=[1:4:num\_pts num\_pts];

end

figure('Position', hw\_pub.figPosn);

plot3(r\_bar(1,:),r\_bar(2,:),r\_bar(3,:));

xlabel('i (DU)'); ylabel('j (DU)'); zlabel('k (DU)')

hold on

quiver3(r\_bar(1,plot\_idx),r\_bar(2,plot\_idx),r\_bar(3,plot\_idx),...

t\_MAV(1,plot\_idx),t\_MAV(2,plot\_idx),t\_MAV(3,plot\_idx),'r');

quiver3(r\_bar(1,plot\_idx),r\_bar(2,plot\_idx),r\_bar(3,plot\_idx),...

b\_MAV(1,plot\_idx),b\_MAV(2,plot\_idx),b\_MAV(3,plot\_idx),'color',orange);

quiver3(r\_bar(1,plot\_idx),r\_bar(2,plot\_idx),r\_bar(3,plot\_idx),...

n\_MAV(1,plot\_idx),n\_MAV(2,plot\_idx),n\_MAV(3,plot\_idx),'color',lgreen);

xlabel('t (DU)'); ylabel('n (DU)'); zlabel('b (DU)')

title('MAV observed in MS Frenet frame with Frenet frame vectors')

view([1,0,0])

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_')) '\_MAV\_NB'],'jpg')

view([0,0,1])

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_')) '\_MAV\_TN'],'jpg')

view([-1,-1,1])

legend({'Path','Tangential','Binormal','Normal'});

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_')) '\_MAV\_ISO'],'jpg')

if strcmp(scenario, 'Part 1')

% Mothership normal, tangent accels wrt GS

MS\_accel\_tangent = zeros(1,length(A\_forward));

MS\_accel\_normal = zeros(1,length(A\_forward));

MS\_accel\_bi = zeros(1,length(A\_forward));

for ii = 1:length(A\_forward)

MS\_accel\_tangent(ii) = dot(t(:,ii),A\_forward(:,ii));

MS\_accel\_normal(ii) = dot(n(:,ii),A\_forward(:,ii));

MS\_accel\_bi(ii) = dot(b(:,ii),A\_forward(:,ii));

end

% figure('Position', hw\_pub.figPosn);

% plot(MS\_accel\_tangent)

% title('MS Tangential Acceleration wrt GS')

% figure('Position', hw\_pub.figPosn);

% plot(MS\_accel\_normal)

% title('MS Normal Acceleration wrt GS')

% figure('Position', hw\_pub.figPosn);

% plot(MS\_accel\_bi)

% title('MS Binormal Acceleration wrt GS')

% Plot the accelerations, speed of MS

figure('Position', hw\_pub.figPosn);

plot(time(1:end-2),MS\_accel\_tangent)

hold on

plot(time(1:end-2),MS\_accel\_normal,'r')

title('MAV Accelerations wrt GS')

xlabel('Time (TU)'); ylabel('Acceleration (DU/TU^2)')

legend('Tangential Accel', 'Normal Accel')

saveas(gcf, ['Figures\' 'MS\_Accels'],'jpg')

figure('Position', hw\_pub.figPosn);

plot(time(1:end-1),sqrt(sum(V\_forward.^2,1)))

title('MS Speed wrt GS')

xlabel('Time (TU)'); ylabel('Speed (DU/TU)')

saveas(gcf, ['Figures\' 'MS\_Speed'],'jpg')

end

% MAV normal, tangent accels wrt MS

MAV\_accel\_tangent = zeros(1,length(a\_bar\_forward));

MAV\_accel\_normal = zeros(1,length(a\_bar\_forward));

MAV\_accel\_bi = zeros(1,length(a\_bar\_forward));

for ii = 1:length(a\_bar\_forward)

MAV\_accel\_tangent(ii) = dot(t\_MAV(:,ii),a\_bar\_forward(:,ii));

MAV\_accel\_normal(ii) = dot(n\_MAV(:,ii),a\_bar\_forward(:,ii));

MAV\_accel\_bi(ii) = dot(b\_MAV(:,ii),a\_bar\_forward(:,ii));

end

% figure('Position', hw\_pub.figPosn);

% plot(MAV\_accel\_tangent)

% title('MAV Tangential Acceleration wrt MS')

% figure('Position', hw\_pub.figPosn);

% plot(MAV\_accel\_normal)

% title('MAV Normal Acceleration wrt MS')

% figure('Position', hw\_pub.figPosn);

% plot(MAV\_accel\_bi)

% title('MAV Binormal Acceleration wrt MS')

% Plot the accelerations, speed of MAV

figure('Position', hw\_pub.figPosn);

plot(time(1:end-2),MAV\_accel\_tangent)

hold on

plot(time(1:end-2),MAV\_accel\_normal,'r')

title('MAV Accelerations wrt MS')

xlabel('Time (TU)'); ylabel('Acceleration (DU/TU^2)')

legend('Tangential Accel', 'Normal Accel')

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_')) '\_MAV\_Accels'],'jpg')

figure('Position', hw\_pub.figPosn);

plot(time(1:end-1),sqrt(sum(v\_bar\_forward.^2,1)))

title('MAV Speed wrt MS')

xlabel('Time (TU)'); ylabel('Speed (DU/TU)')

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_')) '\_MAV\_Speed'],'jpg')

% MAV wrt GS

% The euler angles are found from G

% These are plugged into the w/alpha calcs

% nans are used to init so that the zeros don't drag down the plots

r = nan(3,length(t));

v = nan(3,length(t));

a = nan(3,length(t));

phi = zeros(1,length(t));

theta = zeros(1,length(t));

psi = zeros(1,length(t));

phi\_dot = zeros(1,length(t));

theta\_dot = zeros(1,length(t));

psi\_dot = zeros(1,length(t));

phi\_dotdot = zeros(1,length(t));

theta\_dotdot = zeros(1,length(t));

psi\_dotdot = zeros(1,length(t));

w\_MS\_Frenet\_wrt\_GS = zeros(3,length(t));

w\_MS\_Frenet\_wrt\_body = zeros(3,length(t));

alpha\_MS\_Frenet\_wrt\_body = zeros(3,length(t));

alpha\_MS\_Frenet\_wrt\_GS = zeros(3,length(t));

% MAV's Frenet frame in GS cartesian

t\_MAV\_GS = zeros(3,num\_pts);

b\_MAV\_GS = zeros(3,num\_pts);

n\_MAV\_GS = zeros(3,num\_pts);

for ii = 1:length(t) - 2 % due to forward diff

rot = [t(:,ii)';n(:,ii)';b(:,ii)'];

% Orthogonal transformation, so the inverse is the transpose.

G\_MS2GS = rot';

r(:,ii) = R(:,ii)+G\_MS2GS\*r\_bar(:,ii);

phi(ii) = atan2(G\_MS2GS(1,3),-G\_MS2GS(2,3));

theta(ii) = atan2(sqrt(1-G\_MS2GS(3,3)^2),G\_MS2GS(3,3));

psi(ii) = atan2(G\_MS2GS(3,1),G\_MS2GS(3,2));

% Unroll the angles so we have continuous derivatives

if phi(ii) < 0

phi(ii) = phi(ii) + 2\*pi;

end

if theta(ii) < 0

theta(ii) = theta(ii) + 2\*pi;

end

if psi(ii) < 0

psi(ii) = psi(ii) + 2\*pi;

end

if ii >= 2 && abs(phi(ii)-phi(ii-1)) > pi

phi(ii) = phi(ii) + 2\*pi;

end

if ii >= 2 && abs(theta(ii)-theta(ii-1)) > pi

theta(ii) = theta(ii) + 2\*pi;

end

if ii >= 2 && abs(psi(ii)-psi(ii-1)) > pi

psi(ii) = psi(ii) + 2\*pi;

end

end

% Euler angle derivs

for ii = 1:length(t) - 1 % due to forward diff

phi\_dot(ii) = (phi(ii+1)-phi(ii))/t\_diffs(ii);

theta\_dot(ii) = (theta(ii+1)-theta(ii))/t\_diffs(ii);

psi\_dot(ii) = (psi(ii+1)-psi(ii))/t\_diffs(ii);

end

% Euler double-derivs + body rates/accel

for ii = 1:length(t) - 4 % due to forward diff

rot = [t(:,ii)';n(:,ii)';b(:,ii)'];

% Orthogonal transformation, so the inverse is the transpose.

G\_MS2GS = rot';

% Euler angle double-derivs

phi\_dotdot(ii) = (phi\_dot(ii+1)-phi\_dot(ii))/t\_diffs(ii);

theta\_dotdot(ii) = (theta\_dot(ii+1)-theta\_dot(ii))/t\_diffs(ii);

psi\_dotdot(ii) = (psi\_dot(ii+1)-psi\_dot(ii))/t\_diffs(ii);

% MS frame rot rate in a couple systems

w\_MS\_Frenet\_wrt\_GS(1,ii) = psi\_dot(ii)\*sin(theta(ii))\*sin(phi(ii)) ...

+ theta\_dot(ii)\*cos(phi(ii));

w\_MS\_Frenet\_wrt\_GS(2,ii) = -psi\_dot(ii)\*sin(theta(ii))\*cos(phi(ii)) ...

+ theta\_dot(ii)\*sin(phi(ii));

w\_MS\_Frenet\_wrt\_GS(3,ii) = psi\_dot(ii)\*cos(theta(ii))+phi\_dot(ii);

w\_MS\_Frenet\_wrt\_body(1,ii) = phi\_dot(ii)\*sin(theta(ii))\*sin(psi(ii))...

+theta\_dot(ii)\*cos(psi(ii));

w\_MS\_Frenet\_wrt\_body(2,ii) = phi\_dot(ii)\*sin(theta(ii))\*cos(psi(ii))...

-theta\_dot(ii)\*sin(psi(ii));

w\_MS\_Frenet\_wrt\_body(3,ii) = phi\_dot(ii)\*cos(theta(ii))+psi\_dot(ii);

% The MAV velocity in GS calculation

v(:,ii) = V(:,ii) + G\_MS2GS\*(v\_bar\_forward(:,ii) ...

+ cross(w\_MS\_Frenet\_wrt\_body(:,ii),r\_bar(:,ii)));

% MS frame rot rate-rate in a couple systems

alpha\_MS\_Frenet\_wrt\_body(1,ii) = ...

phi\_dotdot(ii)\*sin(theta(ii))\*sin(psi(ii)) ...

+ phi\_dot(ii)\*theta\_dot(ii)\*cos(theta(ii))\*sin(psi(ii))...

+ phi\_dot(ii)\*psi\_dot(ii)\*sin(theta(ii))\*cos(psi(ii))...

+ theta\_dotdot(ii)\*cos(psi(ii)) ...

- theta\_dot(ii)\*psi\_dot(ii)\*sin(psi(ii));

alpha\_MS\_Frenet\_wrt\_body(2,ii) = ...

phi\_dotdot(ii)\*sin(theta(ii))\*cos(psi(ii)) ...

+ phi\_dot(ii)\*theta\_dot(ii)\*cos(theta(ii))\*cos(psi(ii))...

- phi\_dot(ii)\*psi\_dot(ii)\*sin(theta(ii))\*sin(psi(ii))...

- theta\_dotdot(ii)\*sin(psi(ii)) ...

- theta\_dot(ii)\*psi\_dot(ii)\*cos(psi(ii));

alpha\_MS\_Frenet\_wrt\_body(3,ii) = ...

phi\_dotdot(ii)\*cos(theta(ii)) ...

- phi\_dot(ii)\*theta\_dot(ii)\*sin(theta(ii))...

+ psi\_dotdot(ii);

alpha\_MS\_Frenet\_wrt\_GS(1,ii) = ...

psi\_dotdot(ii)\*sin(theta(ii))\*sin(phi(ii)) ...

+ psi\_dot(ii)\*theta\_dot(ii)\*cos(theta(ii))\*sin(phi(ii))...

+ psi\_dot(ii)\*phi\_dot(ii)\*sin(theta(ii))\*cos(phi(ii))...

+ theta\_dotdot(ii)\*cos(phi(ii)) ...

- theta\_dot(ii)\*phi\_dot(ii)\*sin(phi(ii));

alpha\_MS\_Frenet\_wrt\_GS(2,ii) = ...

-psi\_dotdot(ii)\*sin(theta(ii))\*cos(phi(ii)) ...

- psi\_dot(ii)\*theta\_dot(ii)\*cos(theta(ii))\*cos(phi(ii))...

+ psi\_dot(ii)\*phi\_dot(ii)\*sin(theta(ii))\*sin(phi(ii))...

+ theta\_dotdot(ii)\*sin(phi(ii)) ...

+ theta\_dot(ii)\*phi\_dot(ii)\*cos(phi(ii));

alpha\_MS\_Frenet\_wrt\_GS(3,ii) = ...

psi\_dotdot(ii)\*cos(theta(ii)) ...

- psi\_dot(ii)\*theta\_dot(ii)\*sin(theta(ii))...

+ phi\_dotdot(ii);

% a(:,ii) = A(:,ii) + G\_MS2GS\*(a\_bar\_forward(:,ii) ...

% + cross(alpha\_MS\_Frenet\_wrt\_body(:,ii),r\_bar(:,ii))...

% + 2\*cross(w\_MS\_Frenet\_wrt\_body(:,ii),v\_bar\_forward(:,ii))...

% + cross(w\_MS\_Frenet\_wrt\_body(:,ii),...

% cross(w\_MS\_Frenet\_wrt\_body(:,ii),r\_bar(:,ii))));

% The MAV accel in GS

a(:,ii) = A(:,ii) + G\_MS2GS\*a\_bar\_forward(:,ii) ...

+ cross(alpha\_MS\_Frenet\_wrt\_GS(:,ii),G\_MS2GS\*r\_bar(:,ii))...

+ 2\*cross(w\_MS\_Frenet\_wrt\_GS(:,ii),G\_MS2GS\*v\_bar\_forward(:,ii))...

+ cross(w\_MS\_Frenet\_wrt\_GS(:,ii),...

cross(w\_MS\_Frenet\_wrt\_GS(:,ii),G\_MS2GS\*r\_bar(:,ii)));

% MAV's Frenet frame in GS cartesian

t\_MAV\_GS(:,ii) = v(:,ii)/norm(v(:,ii));

b\_MAV\_GS(:,ii) = cross(v(:,ii),a(:,ii))...

/norm(cross(v(:,ii),a(:,ii)));

n\_MAV\_GS(:,ii) = cross(b\_MAV(:,ii),t\_MAV(:,ii));

end

% Plot the MAV position in GS Cartesian

figure('Position', hw\_pub.figPosn)

plot3(r(1,:),r(2,:),r(3,:));

hold on

plot3(R(1,:),R(2,:),R(3,:),'r');

xlabel('i (DU)'); ylabel('j (DU)'); zlabel('k (DU)')

title('MAV observed from GS')

view([1,0,0])

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_')) '\_MAV\_JK'],'jpg')

view([0,0,1])

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_')) '\_MAV\_IJ'],'jpg')

view([-1,-1,1])

legend({'MAV path','MS path'});

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_'))...

'\_MAV\_GS\_ISO'],'jpg')

% Plot the velocity component err

v\_forward = forward\_diff(r,time);

figure('Position', hw\_pub.figPosn)

hold on

plot(time(1:end-1),abs(v(1,1:end-1) - v\_forward(1,:)));

plot(time(1:end-1),abs(v(2,1:end-1) - v\_forward(2,:)),'r');

plot(time(1:end-1),abs(v(3,1:end-1) - v\_forward(3,:)),'k');

title(['MAV Velocity wrt GS: Error Between Euler-Angle-Rotation and '...

'Numerical Differentiation Solution'])

legend('v\_x error', 'v\_y error', 'v\_z error');

xlabel('Time'); ylabel('Velocity Error');

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_'))...

'\_MAV\_vel\_err'],'jpg')

% Plot the MAV speed wrt GS

figure('Position', hw\_pub.figPosn)

hold on

rotated\_speed = sqrt(sum(v.\*v,1));

nd\_speed = sqrt(sum(v\_forward.\*v\_forward,1));

plot(time(1:length(rotated\_speed)),rotated\_speed);

plot(time(1:length(v\_forward)),nd\_speed,'r');

title('MAV Speed wrt GS')

xlabel('Time (TU)'); ylabel('Speed (DU/TU)');

legend('rotating coord frame', 'numerically differentiated')

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_'))...

'\_MAV\_GS\_speed'],'jpg')

% Plot the MAV speed err between the two methods

figure('Position', hw\_pub.figPosn)

hold on

plot(time(1:length(nd\_speed)),...

abs(nd\_speed-rotated\_speed(1:length(nd\_speed))));

% plot(time(1:length(v\_forward)),sqrt(sum(v\_forward.\*v\_forward,1)),'r');

title('Error in calculation methods for MAV Speed wrt GS')

xlabel('Time (TU)'); ylabel('Speed (DU/TU)');

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_'))...

'\_MAV\_GS\_speed\_err'],'jpg')

% Compute the MAV Frenet frame in GS Cartesian

a\_forward = forward\_diff(v\_forward,time(1:end-1));

t\_MAV\_GS\_ND = zeros(3,length(a\_forward));

b\_MAV\_GS\_ND = zeros(3,length(a\_forward));

n\_MAV\_GS\_ND = zeros(3,length(a\_forward));

for ii = 1:length(a\_forward)

t\_MAV\_GS\_ND(:,ii) = v\_forward(:,ii)/norm(v\_forward(:,ii));

b\_MAV\_GS\_ND(:,ii) = cross(v\_forward(:,ii),a\_forward(:,ii))...

/norm(cross(v\_forward(:,ii),a\_forward(:,ii)));

n\_MAV\_GS\_ND(:,ii) = cross(b\_MAV(:,ii),t\_MAV(:,ii));

end

% Plot the kinematic-method results

figure('Position', hw\_pub.figPosn)

plot(time,sum(t\_MAV\_GS.\*a,1))

hold on

plot(time,sum(n\_MAV\_GS.\*a,1),'r')

% plot(time,sum(b\_MAV\_GS.\*a,1))

legend('Tangential','Normal')

xlabel('Time (TU)');ylabel('Acceleration (DU/TU^2)');

title('MAV Accelerations wrt GS, Rotation Method')

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_'))...

'\_MAV\_rot\_accel'],'jpg')

% Plot the ND results

figure('Position', hw\_pub.figPosn)

plot(time(1:end-2),sum(t\_MAV\_GS\_ND.\*a\_forward,1))

hold on

plot(time(1:end-2),sum(n\_MAV\_GS\_ND.\*a\_forward,1),'r')

legend('Tangential','Normal')

xlabel('Time (TU)');ylabel('Acceleration (DU/TU^2)');

title('MAV Accelerations wrt GS, Direct Numerical Differentiation')

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_'))...

'\_MAV\_nd\_accel'],'jpg')

% Plot the error between the two methods

figure('Position', hw\_pub.figPosn)

tan\_diff = sum(t\_MAV\_GS.\*a,1);

tan\_diff = abs(tan\_diff(1:end-2)-sum(t\_MAV\_GS\_ND.\*a\_forward,1));

normal\_diff = sum(n\_MAV\_GS.\*a,1);

normal\_diff = abs(normal\_diff(1:end-2)-sum(n\_MAV\_GS\_ND.\*a\_forward,1));

plot(time(1:length(tan\_diff)),tan\_diff)

hold on

plot(time(1:length(normal\_diff)),normal\_diff,'r')

title('Error in calculation methods for MAV acceleration wrt GS')

xlabel('Time (TU)');ylabel('Acceleration Error (DU/TU^2)');

legend('Tangential Accel Error', 'Normal Accel Error');

saveas(gcf, ['Figures\' char(strrep(scenario,' ','\_'))...

'\_MAV\_rot\_nd\_accel\_err'],'jpg')

end

1. Graduate Student, Aerospace Engineering Sciences, 1111 Engineering Drive, Boulder, CO, 80309-0429 [↑](#footnote-ref-1)