An Algorithm for Roll Maneuver Execution with Resolved Maneuver for Satellite Launch Vehicles

Ashwin Chandar O.R
Scientist, Advanced Systems Laboratory
DRDO, Min. of Defence
Hyderabad, India
orashwin.asl@gmail.com

Reshma Gopal M.
Scientist, Advanced Systems Laboratory
DRDO, Min. of Defence
Hyderabad, India
reshmagopal.asl@gmail.com

doi: https://doi.org/10.37285/bsp.SACAD2025.11

***Abstract*—This paper presents a new algorithm and methodology to execute a desired attitude maneuver in roll without execution of roll maneuver. Although the execution of conventional roll maneuver has its own merits, it involves a complex trajectory maneuvers specifically when the requirement of maneuver angle is large. This will also call for additional control requirements in terms of thruster firing, aerodynamic control execution etc. In most of the times, there is a time limit for completion of roll maneuver, which again demands a very high roll rate. This is not preferable at times specifically during the atmospheric phase of flight. The maximum orientation possible may be limited due to the Control impulse requirements. In order to avoid such limtations and have the flexibility to launch in any orientation different than the intended orientation, a Resolved maneuver algorithm is designed by resolving the Roll angle into pitch and yaw planes which can take care of any launcher roll orientation and which facilitates efficient roll control along wth accurate injection. The new algorithm will ensure that the vehicle follows the intended flight path without orientation of the vehicle in the desired attitude. With this algorithm launch vehicles will be able to position itself in the desired trajectory plane without actually orienting the vehicle, which otherwise would have required to perform roll correction in-flight to align itself to reach the intended injection point.**

***Index Terms*—Roll maneuver, Launch vehicle orientation, Resolve Maneuver**

I. Introduction

The **roll maneuver** in a launch vehicle refers to the intentional rotation of the vehicle about its **longitudinal axis** (also called the *roll axis*, typically the X-axis in body-fixed coordinates). This maneuver is crucial during the initial phase of the flight trajectory and is commonly executed shortly after liftoff. The roll maneuver serves several key purposes:

1) **Alignment with Desired Trajectory Plane**
The roll maneuver helps align the launch vehicle with the intended orbital or flight plane. This is essential for accurate orbital insertion or ballistic path adherence.
2) **Simplification of Guidance Control**
By aligning the vehicle's control surfaces or thrusters, roll maneuvers simplify the implementation of guidance laws, reducing complexity in pitch and yaw control loops.
3) **Structural Load Optimization**
Performing a roll maneuver allows the vehicle to orient itself to distribute aerodynamic and structural loads more effectively, thereby minimizing stresses and enhancing structural integrity.
4) **Proper Orientation of Payload**
Roll maneuvers ensure the correct orientation of the payload at the time of deployment, which is critical for satellite missions and interplanetary probes.
5) **Ground Station Communication Alignment**
Adjusting the vehicles roll can align communication antennas or sensors toward ground stations, improving signal quality and data transmission reliability.
6) **Solar Panel or Sensor Orientation**
In spacecraft, roll maneuvers are used to point solar panels or instruments toward the sun or Earth, ensuring maximum power generation or optimal data acquisition.
7) **Minimization of Cross-Coupling Effects**
By aligning the control axes correctly, roll maneuvers help decouple the control actions (e.g., pitch, yaw), improving overall stability and control precision.

The roll maneuver is executed using one or more of the following control mechanisms:

- **Gimbaled engines** that can vector thrust to generate roll torque.
- **Reaction control thrusters (RCS)** in upper stages or vacuum phases.
- **Aerodynamic surfaces** (e.g., fins) during early ascent in the lower atmosphere.

The vehicle's **Guidance, Navigation, and Control (GNC)** system computes the required roll angle and issues commands to initiate and terminate the maneuver. The roll rate and final angle are closely monitored using on-board sensors such as gyroscopes and accelerometers.

The roll maneuver is typically performed:

- Within the first few seconds after lift-off.
- Before or simultaneously with pitch and yaw maneuvers.
- At low altitudes to establish the correct orientation for the gravity turn.

Consider a launch vehicle aiming for a polar orbit but launching from a site that faces eastward by default. A roll maneuver of 90° may be required to reorient the pitch axis toward the north-south direction, enabling the vehicle to follow the correct polar trajectory after the pitch maneuver. In summary, the roll maneuver is a critical part of the launch vehicle's ascent profile. It ensures the correct alignment with the mission trajectory and facilitates efficient control and structural stability throughout the flight.

A typical axis system of launch vehicle and a satellite is shown in Fig. 1 and Fig. 2

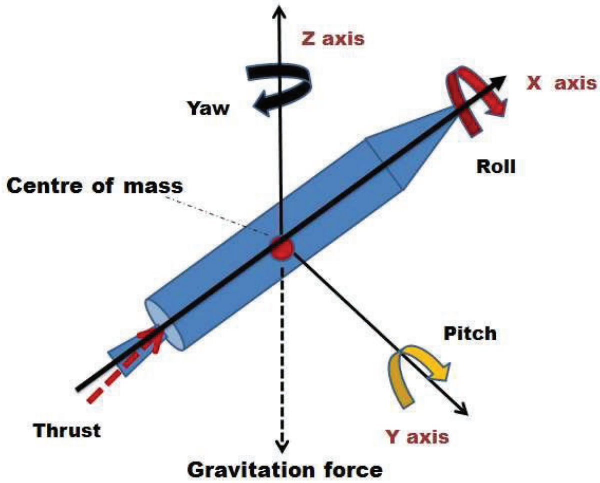

Fig. 1: *Axis System of a typical LV*

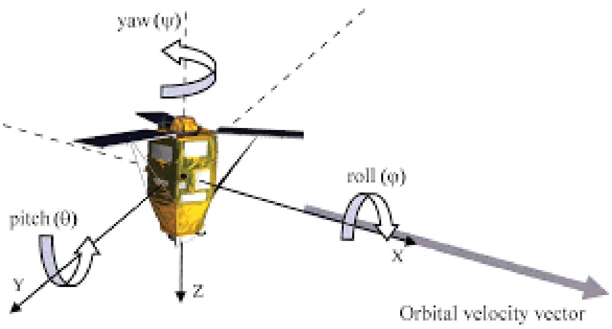

Fig. 2: *Axis System of a typical Satellite*

II. A brief description of Axis Transformation Methods

Axis transformation methods are essential in engineering and physics for converting vectors and coordinate representations from one reference frame to another. These methods are especially important in aerospace, robotics, and computer graphics where multiple coordinate systems are involved.

1. Rotation Matrix

A rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$ is an orthogonal matrix with determinant 1 that transforms a vector $\mathbf{v}$ from one frame to another:

$$\mathbf{v}_{\text{new}} = \mathbf{R}\,\mathbf{v}_{\text{old}}$$

Properties:

$$\mathbf{R}^T = \mathbf{R}^{-1}, \quad \det(\mathbf{R}) = 1$$

2. Direction Cosine Matrix (DCM)

The direction cosine matrix is a special form of rotation matrix whose elements are the cosines of angles between coordinate axes:

$$C_{ij} = \cos(\theta_{ij})$$

where θ_{ij} is the angle between the i-th axis of one frame and the j-th axis of another.

3. Euler Angles

Euler angles represent a sequence of three rotations about specified axes (e.g., ZYX). The total rotation matrix is the product of three basic rotation matrices:

$$\mathbf{R} = \mathbf{R}_z(\psi)\mathbf{R}_y(\theta)\mathbf{R}_x(\phi)$$

Euler angles are intuitive but can suffer from singularities (gimbal lock).

4. Quaternions

Quaternions provide a compact and numerically stable method for representing rotations. A unit quaternion $q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$ can represent a rotation about an arbitrary axis $\hat{n} = [n_x, n_y, n_z]^T$ by angle θ:

$$q = \cos\left(\frac{\theta}{2}\right) + \hat{n}\sin\left(\frac{\theta}{2}\right)$$

5. Homogeneous Transformation Matrix

A homogeneous transformation combines rotation and translation into a single 4×4 matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ 0 & 1 \end{bmatrix}$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ is the rotation matrix and $\mathbf{d} \in \mathbb{R}^3$ is the translation vector.

III. Quaternions for Axis transformation during Roll Maneuver

Quaternions are a number system that extends complex numbers and are widely used in three-dimensional rotations, especially in computer graphics, robotics, and aerospace applications.

A quaternion is expressed in the form:

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$$

where $a, b, c, d \in \mathbb{R}$ and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are the fundamental quaternion units, satisfying the following multiplication rules:

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$$

$$\mathbf{ij} = \mathbf{k}, \quad \mathbf{jk} = \mathbf{i}, \quad \mathbf{ki} = \mathbf{j}$$

$$\mathbf{ji} = -\mathbf{k}, \quad \mathbf{kj} = -\mathbf{i}, \quad \mathbf{ik} = -\mathbf{j}$$

Quaternions can be used to represent rotations in 3D space without the singularities (gimbal lock) that occur with Euler angles. A unit quaternion satisfies:

$$\|q\| = \sqrt{a^2 + b^2 + c^2 + d^2} = 1$$

Given a vector $\vec{v} \in \mathbb{R}^3$, it can be represented as a pure quaternion $0 + v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$. Rotation of this vector by a unit quaternion q is given by:

$$\vec{v}_{\text{rotated}} = q\vec{v}q^{-1}$$

Quaternions provide a robust and efficient way to perform rotations in three-dimensional space. A quaternion is defined as:

$$q = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k}$$

or in vector form:

$$q = \begin{bmatrix} q_0 \\ \mathbf{q}_v \end{bmatrix} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix}$$

where q_0 is the scalar part and $\mathbf{q}_v = [q_1, q_2, q_3]^T$ is the vector part.

A unit quaternion satisfies:

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1$$

1) Rotation of a Vector: Given a vector $\mathbf{v} \in \mathbb{R}^3$, represented as a pure quaternion:

$$\mathbf{v}_q = 0 + v_x\mathbf{i} + v_y\mathbf{j} + v_z\mathbf{k}$$

the rotated vector $\mathbf{v}'$ is computed using quaternion multiplication:

$$\mathbf{v}' = q\,\mathbf{v}_q\,q^{-1}$$

where q^{-1} is the inverse (or conjugate, for unit quaternions) of q.

2) Quaternion to Rotation Matrix: To convert a unit quaternion $q = [q_0, q_1, q_2, q_3]^T$ to a rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$, use:

$$\mathbf{R} = \begin{bmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & 1-2(q_1^2+q_3^2) & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & 1-2(q_1^2+q_2^2) \end{bmatrix}$$

This matrix can then be used for frame transformations:

$$\mathbf{v}_{\text{new}} = \mathbf{R}\,\mathbf{v}_{\text{old}}$$

3) Advantages of Quaternions for Axis Transformation:

- No gimbal lock (unlike Euler angles)
- Compact representation (4 values vs 9 in rotation matrix)
- Efficient for interpolation (e.g., SLERP)
- Stable numerical behavior

IV. Algorithm for Roll Maneuver Execution in Launch Vehicle

Algorithm 1 Roll Maneuver Algorithm : Implementation and Execution

1: **Pre-Launch Inputs:** Obtain initial launch azimuth (ψ_L), computed azimuth at injection (ψ_T) from orbital parameters
2: Obtain the initial attitude angles θ_{init}, ψ_{init} and ϕ_{init}
3: Form the initial quaternion q_{init} from the initial attitude angles.
4: **Initialize:** Desired roll maneuver rate as ω_{RM}
5: Wait for liftoff and initialize Lift-Off time (t_0) and expected Injection time (t_{inj})
6: Confirm stable ascent
7: **Post Lift-off Computations:** Compute the desired roll maneuver angle ϕ_r as
$\phi_r = \psi_T - \psi_L$
8: Compute roll maneuver quaternion (q_{RM}) with incremental roll command as $q_{RM} = quat(\omega_{RM} * \Delta t)$
9: Compute expected roll maneuver time (t_{RM})
10: **On-Board Guidance Execution:**Compute guidance command quaternion q_d from on-board pitch and yaw guidance algorithm in terms of quaternions as $q_d = quat(\theta_d) * quat(\psi_d)$
11: Execute pitch and yaw guidance commands q_d from on-board guidance
12: Compute attitude error (q_e) as $q_e = q(t)^{-1} * q_d$ and command the autopilot with the attitude error q_e
13: **Roll Maneuver Execution:**Initiate roll maneuver execution after ($t > t_0 + t_{RMstart}$)
14: Compute the present roll orientation ϕ_a from quaternion data
15: Compute roll error $\Delta\phi = \phi_r - \phi_a(t)$
16: **while** $\|\Delta\phi\| < \epsilon$ **do**
17:     Update pitch and yaw guidance command with roll maneuver command and form the guidance command quaternion as $q_{dRM} = q_d * q_{RM}$
18:     Measure present attitude $q(t)$ from INS data
19:     Compute attitude error (q_e) as $q_e = (q(t))^-1 * q_{dRM}$
20:     Command autopilot with attitude error q_e
21:     Apply control torque T_r using RCS or TVC to achieve ω_r
22:     Update system dynamics and attitude
23:     Update present roll orientation ϕ_a from quaternion data
24:     Update roll error $\Delta\phi = \phi_r - \phi_a(t)$
25: **end while**
26: Terminate Roll Maneuver algorithm and re-initialize q_{RM} as $(1.0, 0.0, 0.0, 0.0)^T$
27: Continue with Pitch and yaw Guidance commands
28: **Output:** Vehicle aligned with desired roll orientation within the given tolerance limits at the end of Roll maneuver execution

V. Algorithm for Roll Maneuver Execution in Launch Vehicle

Algorithm 2 Proposed Roll Maneuver Algorithm with Resolved Maneuver

1: **Pre-Launch Inputs:** Obtain initial launch azimuth (ψ_L), computed azimuth at injection (ψ_T) from orbital parameters
2: Obtain the initial attitude angles θ_{init}, ψ_{init} and ϕ_{init}
3: Form the initial quaternion q_{init} from the initial attitude angles.
4: **Initialize:** Desired roll maneuver rate as ω_{RM}
5: Wait for liftoff and initialize Lift-Off time (t_0) and expected Injection time (t_{inj})
6: Confirm stable ascent
7: **Post Lift-off Computations:** Compute the desired roll maneuver angle ϕ_r as
$\phi_r = \psi_T - \psi_L$
8: **On-Board Guidance Execution:** Update pitch and yaw guidance commands θ_{dN} and ψ_{dN} respectively with required roll maneuver angle as $\theta_{dN} = \theta_d * cos(\phi_r) + \psi_d * sin(\phi_r)$ and $\psi_{dN} = \psi_d * cos(\phi_r) - \theta_d * sin(\phi_r)$
9: Compute guidance command quaternion q_{dN} from on-board pitch and yaw guidance algorithm in terms of quaternions as $q_{dN} = quat(\theta_{dN}) * quat(\psi_{dN})$
10: Execute pitch and yaw guidance commands q_{dN} from onboard guidance
11: Compute attitude error (q_e) as $q_e = q(t)^{-1} * q_{dN}$ and command the autopilot with the attitude error q_e
12: Command autopilot with attitude error q_e
13: Apply control torque T_r using RCS or TVC to achieve ω_r
14: Update system dynamics and attitude
15: Continue with pitch and yaw Guidance commands
16: **Output:** Vehicle will be flying in the desired azimuth orientation although the vehicle axis is not aligned with desired roll orientation

VI. Merits of Proposed algorithm

The proposed algorithm doesnot require a separate hardware for roll correction. This algorithm can tolerate roll hardware failure, provided it is able to track the roll orientation from INS. In the conventional roll maneuver algorithm, the failure in roll control during a mission can lead to critical anomalies in vehicle dynamics, guidance, and mission success. The major consequences are:

1) **Trajectory Deviation**
Inability to perform roll alignment may prevent the vehicle from achieving the correct trajectory or orbital plane, resulting in mission failure.
2) **Loss of Attitude Stability**
Without roll control, the coupling between roll, pitch, and yaw dynamics can induce uncontrollable rotations, potentially leading to tumbling or structural failure.
3) **Incorrect Payload Orientation**
Failure in roll control may result in payload deployment with incorrect attitude, rendering satellites or probes ineffective.
4) **Increased Aerodynamic Loads**
Asymmetric aerodynamic forces due to uncommanded roll can impose excessive loads, risking structural integrity.
5) **Reduced Communication and Navigation Accuracy**
Misalignment of on-board antennas and sensors affects signal quality and position estimation, compromising mission control.
6) **Guidance System Instability**
Roll misalignment can lead to incorrect control inputs in guidance algorithms, causing control loop divergence or oscillation.

VII. Simulation Model

The 6- DOF simulation model essentially comprised of the dynamic model of the vehicle which is governed by the set of equations of motion, and the other models, which are essential for the simulation are the Propulsion model, Atmospheric model, gravity model, wind model, aerodynamic model etc. serve as inputs to the system. The time varying moment of inertia and centre of gravity data is also available to the simulation. The navigation, guidance and control algorithms are integrated in the simulation taking into consideration of the non-linear actuator dynamics. The 6-DOF model is designed in such a way that the flight scenario is exactly replicated in the simulation. The information flow in a 6-DOF model digital simulation is given in

A. Kinematics Model

The simulation model assumes that the vehicle is a rigid body with *six degrees of freedom*(6-DOF). The six degrees of freedom consists of (1) three translations and (2) three rotations, along and about the vehicle axes. The translations being (U,V,W) and the rotations being (P,Q,R). In compact form, the translation and rotation of a rigid body may be expressed mathematically by the following equations.
Translational Equations

$$\Sigma F_x = \frac{d}{dt}(mU) \tag{1}$$
$$\Sigma F_y = \frac{d}{dt}(mV) \tag{2}$$
$$\Sigma F_z = \frac{d}{dt}(mW) \tag{3}$$

Rotational Equations

$$\Sigma M_x = \frac{d}{dt}(rXmU) \tag{4}$$
$$\Sigma M_y = \frac{d}{dt}(rXmV) \tag{5}$$
$$\Sigma M_z = \frac{d}{dt}(rXmW) \tag{6}$$

The aerodynamic forces and moments are assumed to be functions of Mach Number(M) and Angle of Attack (α). The linear velocity of the launch vehicle $\underline{V}$ can be broken up into components U,V,W. The moment of the momentum may be written in terms of moments of inertia and products of inertia and angular velocities as follows:

$$\Sigma M_x = \frac{d}{dt}(PI_x - QI_{xy} - RI_{xz}) \tag{7}$$
$$\Sigma M_y = \frac{d}{dt}(QI_y - RI_{yz} - PI_{xy}) \tag{8}$$
$$\Sigma M_z = \frac{d}{dt}(RI_z - PI_{xz} - QI_{yz}) \tag{9}$$

The above moment equations can be expanded as

$$\Sigma M_x = \dot{P}I_x + P\dot{I}_x - \dot{Q}I_{xy} - Q\dot{I}_{xy} - \dot{R}I_{xz} - R\dot{I}_{xz} \quad (10)$$

$$\Sigma M_y = \dot{Q}I_y + Q\dot{I}_y - \dot{R}I_{yz} - R\dot{I}_{yz} - \dot{P}I_{xy} - P\dot{I}_{xy} \quad (11)$$

$$\Sigma M_z = \dot{R}I_z + R\dot{I}_z - \dot{P}I_{xz} - P\dot{I}_{xz} - \dot{Q}I_{yz} - Q\dot{I}_{yz} \quad (12)$$

The above moment equations can be written in Eulerian axes, which is a moving axis system with a right hand system of orthogonal coordinate axes whose orientation is fixed with respect to the vehicle. The velocities along these axes are identical to those measured on-board.

$$\Sigma F_x = m(\dot{U} + QW - RV) \quad (13)$$
$$\Sigma F_y = m(\dot{V} + RU - PW) \quad (14)$$
$$\Sigma F_z = m(\dot{W} + PV - QU) \quad (15)$$

$$\Sigma M_x = \dot{P}I_x - \dot{Q}I_{xy} - \dot{R}I_{xz} + QR(I_z - I_y) - PQI_{xz} - Q^2 I_{zy} + R^2 I_{yz} + PRI_{xy} \quad (16)$$

$$\Sigma M_y = \dot{Q}I_y - \dot{R}I_{yz} - \dot{R}I_{xz} + PR(I_x - I_z) - QRI_{xy} - R^2 I_{xz} + P^2 I_{xz} + PQI_{xz} \quad (17)$$

$$\Sigma M_z = \dot{R}I_x - \dot{P}I_{xy} + PQ(I_y - I_x) - PRI_{yz} - P^2 I_{xy} + Q^2 I_{xy} + QRI_{xz} \quad (18)$$

The vehicle configuration is symmetric in xy and xz plane and consequently $I_{yz} = I_{xy} = I_{xz} = 0$. Also, $I_y = I_z$. Hence the resultant simplified equations are

$$\Sigma F_x = m(\dot{U} + QW - RV) \quad (19)$$
$$\Sigma F_y = m(\dot{V} + RU - PW) \quad (20)$$
$$\Sigma F_z = m(\dot{W} + PV - QU) \quad (21)$$
$$\Sigma M_x = \dot{P}I_x \quad (22)$$
$$\Sigma M_y = \dot{Q}I_y + PR(I_x - I_z) \quad (23)$$
$$\Sigma M_z = \dot{R}I_x + PQ(I_y - I_x) \quad (24)$$

B. Results

The performance of the vehicle under implementation of both the Roll maneuver and Resolved maneuver algorithms is simulated by the 6 DoF model and the flight trajectory parameters are analyzed for satisfactory performance of the vehicle under all possible flight conditions. The vehicle performance under various initial orientation conditions are shown here below.

1) *Case1:*

- Initial Attitude Orientation (ψ_L) : $90°$
- Intended Attitude Orientation(ψ_T)= $90°$
- Roll Maneuver Angle (ϕ_{RM}) = $0°$

The comparison of the vehicle parameters in case of both the algorithms is shown here.

The lateral accelerations is shown in Fig.[3], body orientation in terms of quaternions is shown in Fig.[4], Guidance demands in Y and Z planes is shown in Fig.[5] and the Ground Trace is shown in Fig.[6].

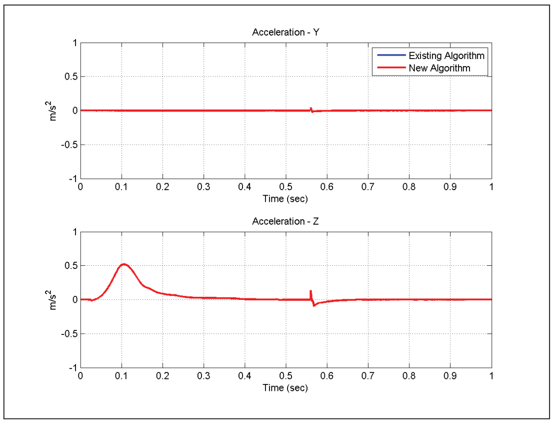

Fig. 3: *Case1: Lateral Acceleration*

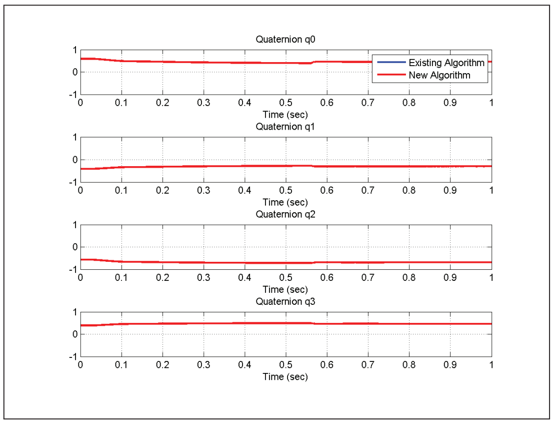

Fig. 4: *Case1: Vehicle Orientation*

All these parameters show that in case of no Roll maneuver requirement, the new algorithm behaves exactly same as the existing algorithm. There is no difference in performance of the new algorithm, which serves as a validation.

2) *Case2:*

- Initial Attitude Orientation (ψ_L) : $90°$
- Intended Attitude Orientation(ψ_T)= $45°$
- Roll Maneuver Angle (ϕ_{RM}) = $-45°$

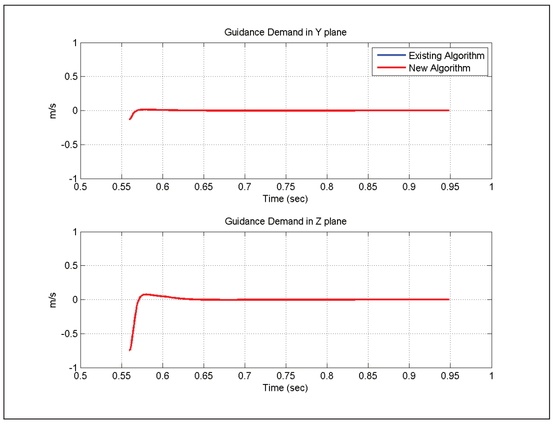

Fig. 5: *Case1: Guidance Demands*

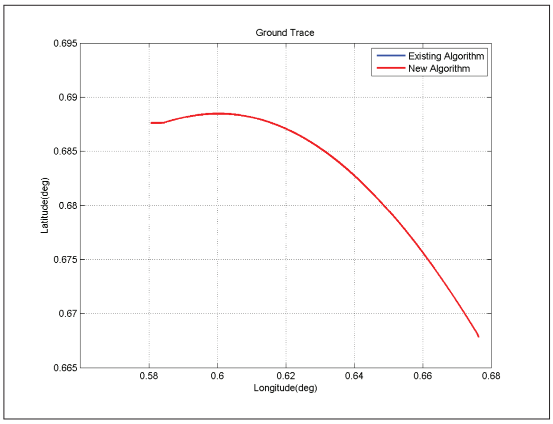

Fig. 6: *Case1: Ground Trace*

The lateral accelerations is shown in Fig.[7], body orientation in terms of quaternions is shown in Fig.[8], Guidance demands in Y and Z planes is shown in Fig.[9], and the Ground Trace is shown in Fig.[10].

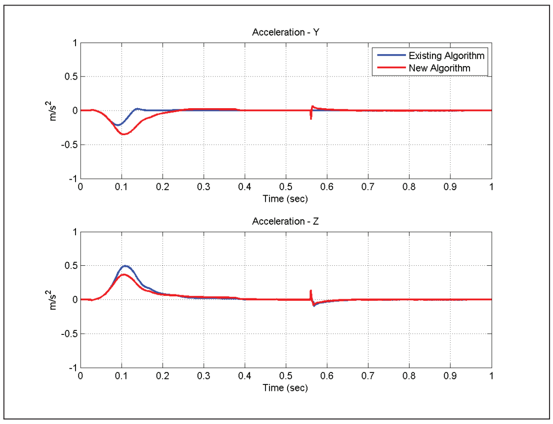

Fig. 7: *Case2: Lateral Acceleration*

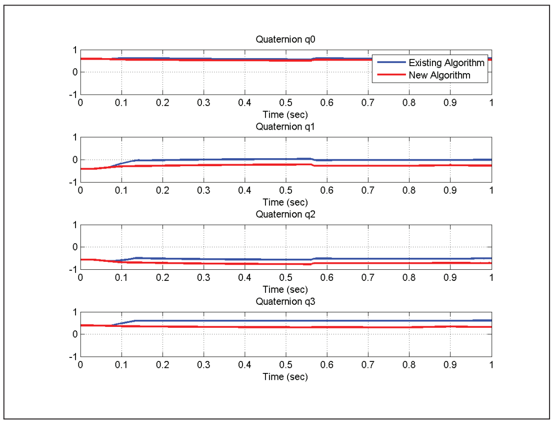

Fig. 8: *Case2: Vehicle Orientation*

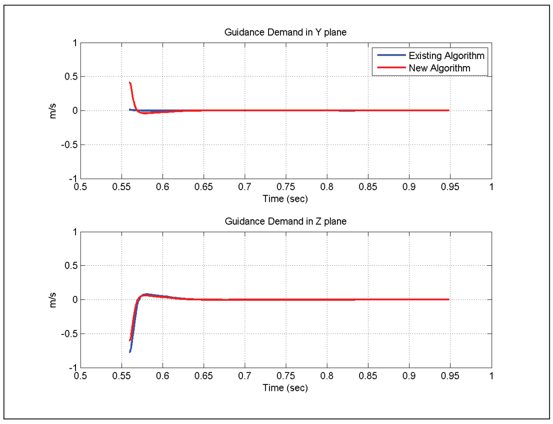

Fig. 9: *Case2: Guidance Demands*

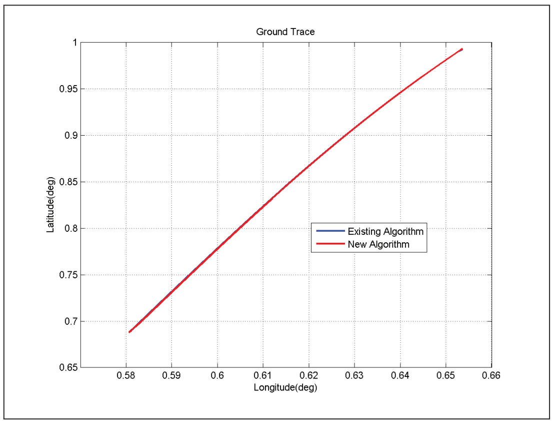

Fig. 10: *Case2: Ground Trace*

It can be seen from the results that by resolving the desired angle, the flight intended orientation is achieved at the end of the flight.

3) Case3:

- Initial Attitude Orientation (ψ_L) : 90°
- Intended Attitude Orientation(ψ_T)= 135°

- Roll Maneuver Angle (ϕ_{RM}) = 45°

The lateral accelerations is shown in Fig.[11], body orientation in terms of quaternions is shown in Fig.[12], Guidance demands in Y and Z planes is shown in Fig.[13], and the Ground Trace is shown in Fig.[14].

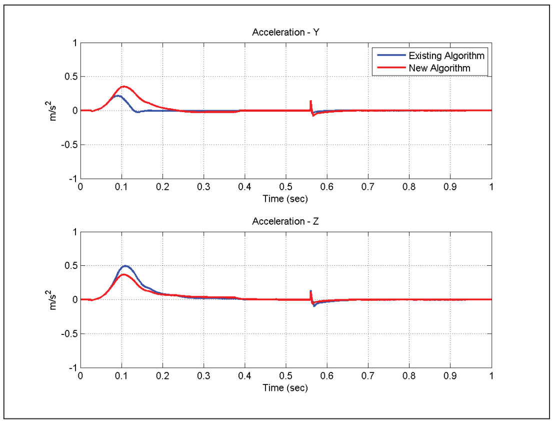

Fig. 11: *Case3: Lateral Acceleration*

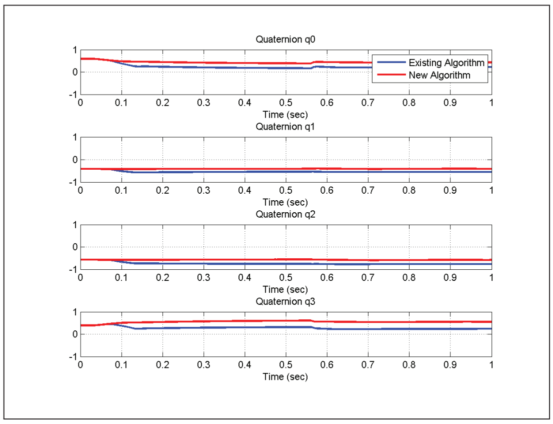

Fig. 12: *Case3: Vehicle Orientation*

It can be seen from the results that by resolving the desired angle, the flight intended orientation is achieved at the end of the flight.

4) Case4:

- Initial Attitude Orientation (ψ_L) : 120°
- Intended Attitude Orientation(ψ_T)= 300°
- Roll Maneuver Angle (ϕ_{RM}) = 180°

The lateral accelerations is shown in Fig.[15], body orientation in terms of quaternions is shown in Fig.[16], Guidance demands in Y and Z planes is shown in Fig.[17], and the Ground Trace is shown in Fig.[18].

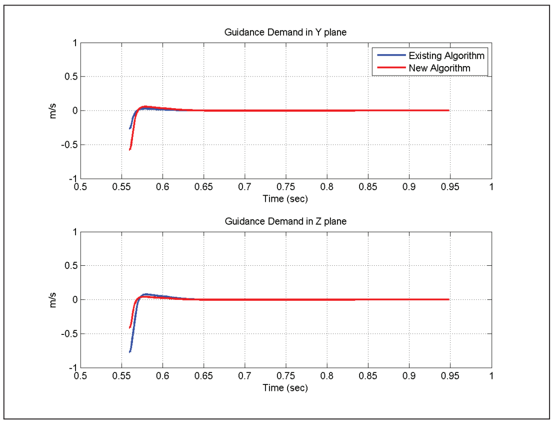

Fig. 13: *Case3: Guidance Demands*

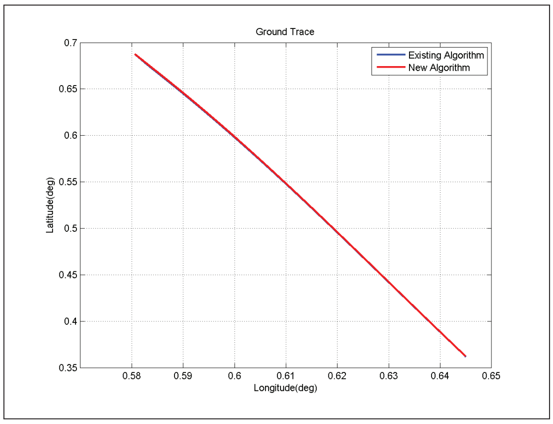

Fig. 14: *Case3: Ground Trace*

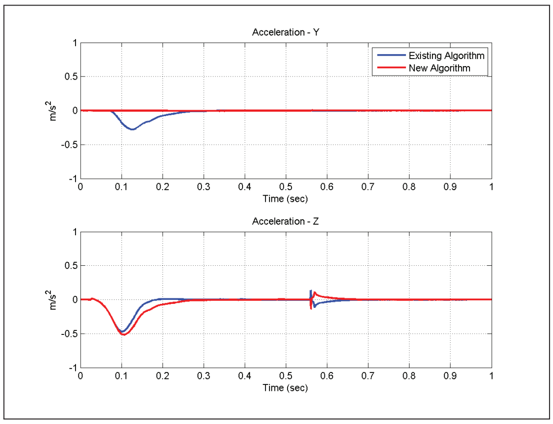

Fig. 15: *Case4: Lateral Acceleration*

It can be seen from the results that by resolving the desired angle, the flight intended orientation is achieved at the end of the flight.

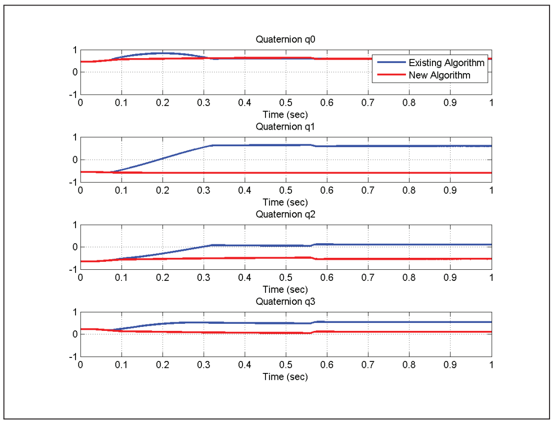

Fig. 16: *Case4: Vehicle Orientation*

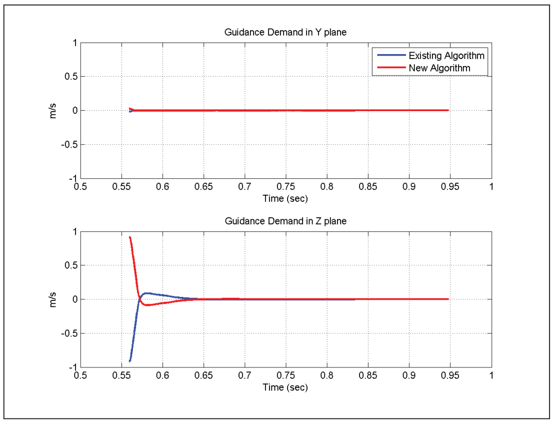

Fig. 17: *Case4: Guidance Demands*

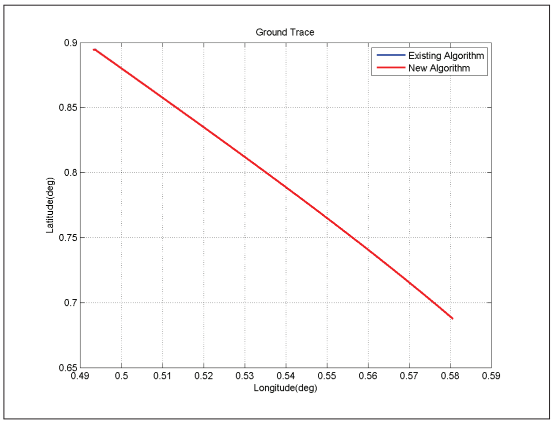

Fig. 18: *Case4: Ground Trace*

VIII. Conclusion

From the results, it can be seen that the trajectory parameters are met and the injection is achieved irrespective of the missile orientation azimuth. It has been verified in all quadrants for positive and negative maneuver angles.

References

[1] 1. Wertz, J.R., " *Spacecraft Attitude Determination and Control"*, Boston, Hingham, MA, 1978 (Reprint 1985)

[2] Frank J. Regan, Satya M. Anadakrishnan, " *Dynamics of Atmospheric Re-Entry"*, AIAA Education Series, Washington, 1993

[3] M.V. Cook, " *Flight Dynamics Principles"*, John Wiley and Sons Inc., Newyork, 1997.

[4] N.V. Kadam, " *Practical Design of Flight Control Systems for Launch Vehicles and Missiles"*, 2005.

[5] Matthew C Vandyke and Christopher D Hall, " *Decentralized Coordinated Attitude Control within a formation of spacecraft"*, Journal of Guidance, Control and Dynamics, Vol.29, No.5, Sep-Oct 2006.

[6] S.K. Srivastava, V.J. Modi, " *Satellite Attitude Dynamics and Control in the presence of Environmental Torques - A Brief Survey"'*, Journal of Guidance, Vol.6, No.6, Nov-Dec 1983.

[7] L. Muhlfelder, " *Evolution of an Attitude Control System for Body Stabilized Communication Spacecraft"*, Journal of Guidance, Vol.9, No.1.

[8] Wie, H. Weiss, A. Arapostathis, " *Quaternion Feedback Regulator for Spacecraft Eigenaxis Rotations"'*, Journal of Guidance,Vol.12, No.3, May-June 1989.

[9] Benjamin O. Lange, et.al., " *Control Synthesis for Spinning Aerospace vehicles"*, journal of Spacecraft, Vol.4, No.2 Feb. 1967.

[10] Ernest P. Todosiev, " *Periodic Attitude Control of a slowly spinning spacecraft"'*, Journal of Spacecraft, Vol.10, No.11.