State-Space Modeling of a Rocket for Optimal Control System Design

Aliyu Bhar Kisabo and Aliyu Funmilayo Adebimpe

Additional information is available at the end of the chapter

Abstract

This chapter is the first of two others that will follow (a three-chapter series). Here we present the derivation of the mathematical model for a rocket's autopilots in state space. The basic equations defining the airframe dynamics of a typical six degrees of freedom (6DoFs) are nonlinear and *coupled*. Separation of these nonlinear coupled dynamics is presented in this chapter to isolate the *lateral* dynamics from the *longitudinal* dynamics. Also, the need to determine aerodynamic coefficients and their derivative components is brought to light here. This is the crux of the equation. Methods of obtaining such coefficients and their derivatives in a sequential form are also put forward. After the aerodynamic coefficients and their derivatives are obtained, the next step is to trim and linearize the decoupled nonlinear 6DoFs. In a novel way, we presented the linearization of the decoupled 6DoF equations in a generalized form. This should provide a lucid and easy way to implement trim and linearization in a computer program. The longitudinal model of a rocket presented in this chapter will serve as the main mathematical model in two other chapters that follow in this book.

Keywords: rocket, six degrees of freedom (6DoFs), state space, trimming, linearization

1. Introduction

Over the years, a number of authors [1–4] have considered modeling rocket/missile dynamics for atmospheric flights. In the majority of the published work on these mathematical models, trimming and locally linearization are done without detailed explanation to the variables in the decoupled airframe dynamics. As such, the easy-to-write computer programs to facilitate this process numerically have been impeded.

With the advent of fast processors and numerical software like MATLAB®, Maple, Python, etc., it is now possible to take a complex nonlinear 6DoF equation like that of a rocket and run a program that can trim and linearize it with ease. This has made the field of control system design to grow at an exponential rate [5].

It is a known fact that the mathematical models are developed with their use in mind. This means before delving into the realization of a model, one should be well informed of the purpose for which that mathematical model will serve [6]. Especially, in the field of control system design, a mathematical model in transfer function might not be ideal for *optimal control* design. However, problem-solving environments (PSEs) like MATLAB®/Simulink come with built-in functions capable of transforming, say state-space model, to transfer functions. One should bear in mind that this is not without a "cost."

2. Mathematical model

A nontrivial part of any control problem is modeling the process. The objective is to obtain the simplest mathematical description that adequately predicts the response of the physical system to all inputs. For a rigid dynamic body, its motion can be described in translational, rotational, and angular inclinations at all times.

2.1. Translational motion

An accelerometer is often used to measure force on a dynamic body. For a rocket in motion, these forces [7] are represented as given in (1). Actually, this is a measure of the specific *force*, i.e., the nongravitational force per unit mass in x,y,z-directions, respectively. The specific force (also called the *g-force* or *mass-specific force*) has units of acceleration or ms^{-1}. So, it is not actually a force at all but a type of acceleration:

$$\dot{u} = \frac{F_{A_{x_b}} + F_{P_{x_b}} + F_{g_{x_b}}}{m} - (qw - rv),\, m/s^2$$
$$\dot{v} = \frac{F_{A_{y_b}} + F_{P_{y_b}} + F_{g_{y_b}}}{m} - (ru - pw),\, m/s^2 \tag{1}$$
$$\dot{w} = \frac{F_{A_{z_b}} + F_{P_{z_b}} + F_{g_{z_b}}}{m} - (pv - qu),\, m/s^2$$

where

$F_{A_{xb}}, F_{A_{yb}}, F_{A_{zb}}$ = components of aerodynamic force vector F_A expressed in the body.

coordinate system, N.

$F_{g_{xb}}, F_{g_{yb}}, F_{g_{zb}}$ = components of gravitational force vector F_g expressed in body coordinate system, N.

$F_{p_{xb}}, F_{p_{yb}}, F_{p_{zb}}$ = components of thrust vector F_p expressed in the body coordinate system, N.

m = instantaneous rocket mass, kg.

p, q, r = components of angular rate vector ω expressed in body coordinate system (roll, pitch, and yaw, respectively), rad/s.

u, v, w = components of absolute linear velocity vector V_x expressed in the body coordinate system, m/s.

$\dot{u}, \dot{v}, \dot{w}$ = components of linear acceleration expressed in body coordinate system, m/s^2.

The aerodynamic forces have the following components:

$$
\begin{aligned}
F_{A_{x_b}} &= -0.5\rho V_M^2 C_A S \\
F_{A_{y_b}} &= 0.5\rho V_M^2 C_{N_y} S \\
F_{A_{z_b}} &= 0.5\rho V_M^2 C_{N_z} S
\end{aligned}
\tag{2}
$$

where

C_A = aerodynamic axial force coefficient, dimensionless.

C_N = aerodynamic normal force coefficient, dimensionless.

C_{Ny} = coefficient corresponding to component of normal force on yb-axis

$$
C_{N_y} = C_N \frac{-v}{\sqrt{v^2 + w^2}}, \text{ dimensionless}
$$

C_{Nz} = coefficient corresponding to component of normal force on zb-axis

$$
C_{N_z} = C_N \frac{1}{\sqrt{v^2 + w^2}}, \text{ dimensionless}
$$

S = aerodynamic reference area, m^2.

V_M = magnitude of velocity vector of the center of mass of the rocket, m/s.

ρ = atmospheric density, kg/m^3.

The propulsive forces in (1), as depicted in (**Figure 1**), are computed [8] as follows:

$$
\begin{aligned}
F_{p_{x_b}} &= F_p \cos\gamma_2 \cos\gamma_1, N \\
F_{p_{y_b}} &= F_p \cos\gamma_2 \sin\gamma_1, N \\
F_{p_{z_b}} &= -F_p \sin\gamma_2, N
\end{aligned}
\tag{3}
$$

where

F_p = magnitude of total instantaneous thrust force vector, N.

$F_{pxb}, F_{pyb}, F_{pzb}$ = components of thrust vector F_p expressed in body coordinate system, N.

γ_1 = angle measured from xb-axis to projection of thrust vector F_p on $x_b\, y_b$-plane, rad (deg).
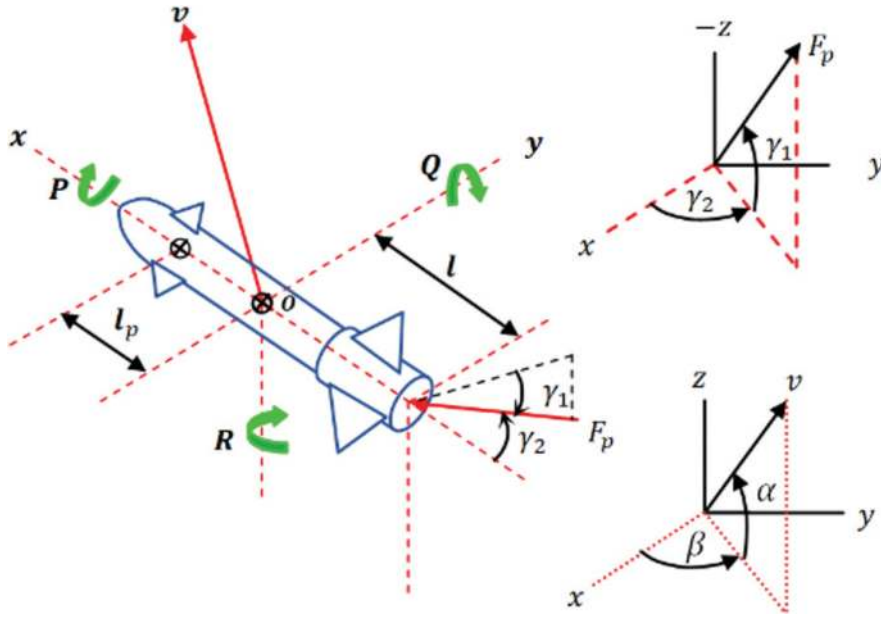
Figure 1. Propulsion force from the nozzle of a rocket.

γ_2 = angle measured from projection of thrust vector F_p on x_b y_b-plane to the thrust vector F_p, rad.

l_p = the distance from the aerodynamic center to center of mass, m.

l = the distance from the center of mass to nozzle, m.

If the rocket was designed for thrust vector control via gimbaling of the nozzle, F_p will be computed as given in (3). Here we assume that F_p is acting along the line of symmetry of the rocket because the nozzle is fixed (fin control). Hence, the magnitude of the thrust force F_p is calculated by

$$F_p = F_{pref} + \left(p_{ref} - p_a \right) A_e, N \tag{4}$$

where

A_e = rocket nozzle exit area, m^2.

F_{pref} = reference thrust force, N.

P_a = ambient atmospheric pressure, Pa.

P_{ref} = reference ambient pressure, Pa.

The gravitational forces in (1) are computed as follows:

$$\begin{aligned} F_{g_{x_e}} &= 0, N \\ F_{g_{y_e}} &= 0, N \\ F_{g_{z_e}} &= mg, N \end{aligned} \tag{5}$$

where

F_{gxe}, F_{gye}, F_{gze} = components of gravitational force vector F_g expressed in earth coordinate system, N.

g = acceleration due to gravity, m/s^2.

m = instantaneous mass of rocket, kg.

The dependence of the acceleration due to gravity on the altitude of the rocket is given by

$$g = g_0 \left[\frac{R_e^2}{(R_e + h)^2} \right], m/s^2 \tag{6}$$

where

g = acceleration due to gravity, m/s^2.

g_0 = acceleration due to gravity at earth surface (nominally 9.8 m/s^2), m/s^2.

h = altitude above sea level, m.

R_e = radius of the earth, m.

The gravitational force expressed in body coordinates is computed by multiplying (5) by matrix (7):

$$[T_{e/b}] = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta c\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix}, \text{dimensionless} \tag{7}$$

where

c = cosine function $(c\theta = cos\ \theta)$, dimensionless.

s = $sine$ function $(s\theta = sin\ \theta)$, dimensionless.

θ = the Euler angle of rotation in elevation (pitch) of body frame relative to earth frame, rad (deg).

φ = the Euler angle of rotation in roll of body frame relative to earth frame, rad (deg).

ψ = the Euler angle of rotation in azimuth (heading) of body frame relative to earth frame, rad (deg).

$[T_{e/b}]$ = transformation matrix from earth to body.

A vector v expressed in the body coordinate system can be transformed to the earth coordinate system by the matrix equation

$$v_e = [T_{e/b}] v_b \tag{8}$$

Hence, considering (5) we can write the following:

$$\begin{bmatrix} F_{g_{x_b}} \\ F_{g_{y_b}} \\ F_{g_{z_b}} \end{bmatrix} = \begin{bmatrix} T_{b/e} \end{bmatrix} \begin{bmatrix} F_{g_{x_e}} \\ F_{g_{y_e}} \\ F_{g_{z_e}} \end{bmatrix}, N \tag{9}$$

The terms F_{gxb}, F_{gyb}, and F_{gzb} are the components of the gravitational force substituted into (1). The **mass** in (1) is given below:

$$m = m_0 - \frac{1}{I_{sp}} \int_0^t F_{p_{ref}} dt, \, kg \tag{10}$$

where

F_{pref} = reference thrust force, N.

I_{sp} = specific impulse of propellant, Ns/kg.

m_0 = rocket mass at time zero (i.e., at the time of launch), kg.

t = simulated time, s.

2.2. Rotational motion

A gyroscope or gyro is a device that measures the angular acceleration or rotational motion of a dynamic body. On a rocket, this rotational motion can be described as

$$\dot{p} = \frac{L_A + L_p - qr(I_z - I_y)}{I_x}, \, rad/s^2 \left(\deg/s^2 \right)$$

$$\dot{q} = \frac{M_A + M_p - rp(I_x - I_z)}{I_y}, \, rad/s^2 \left(\deg/s^2 \right) \tag{11}$$

$$\dot{r} = \frac{N_A + N_p - pq(I_y - I_x)}{I_z}, \, rad/s^2 \left(\deg/s^2 \right)$$

where

L_A, M_A, N_A = components of aerodynamic moment vector M_A expressed in body coordinate system (roll, pitch, and yaw, respectively), Nm.

L_p, M_p, N_p = components of propulsion moment vector M_p expressed in body coordinate system (roll, pitch, and yaw, respectively), Nm.

I_x, I_y, I_z = components of inertia (diagonal elements of inertia matrix when products of inertia are zero), kg-m^2.

p, q, r (P, Q, R) = components of angular rate vector ω expressed in body coordinate system (roll, pitch, and yaw, respectively), rad/s (deg/s).

$\dot{p}, \dot{q}, \dot{r}$ = components of angular acceleration $\dot{\omega}$ expressed in body coordinate (roll, pitch, and yaw, respectively), rad/s^2 (deg/s^2).

$$L_A = 0.5\rho V_M^2 C_l Sd, \, Nm$$
$$M_A = 0.5\rho V_M^2 C_m Sd, \, Nm \tag{12}$$
$$N_A = 0.5\rho V_M^2 C_n Sd, \, Nm$$

where

C_l = aerodynamic roll moment coefficient about center of mass, dimensionless.

C_m = aerodynamic pitch moment coefficient about center of mass, dimensionless.

C_n = aerodynamic yaw moment coefficient about center of mass, dimensionless.

d = aerodynamic reference length of body, m.

The aerodynamic moment coefficients are of the form

$$C_l = C_{l_\delta}\delta_r + \frac{d}{2V_M}\left(C_{l_p}p\right)$$
$$C_m = C_{m_{ref}} - C_{N_z}\frac{x_{cm} - x_{ref}}{d} + \frac{d}{2V_M}\left(C_{m_q} + C_{m_\alpha}\right)q \quad \text{dimensionless} \tag{13}$$
$$C_n = C_{n_{ref}} + C_{N_y}\frac{x_{cm} - x_{ref}}{d} + \frac{d}{2V_M}\left(C_{n_r} + C_{n_\beta}\right)r$$

where

C_{lp} = roll damping derivative relative to roll rate p, rad^{-1} (deg^{-1}).

$C_{l\delta}$ = slope of curve formed by roll moment coefficient C_1 versus control surface deflection, rad^{-1}(deg^{-1}).

C_{mref} = pitching moment coefficient about reference moment station, dimensionless.

C_{mq} = pitch damping derivatives relative to pitch rate q, rad^{-1}(deg^{-1}).

$C_{m_{\dot{\alpha}}}$ = pitch damping derivative relative to angle of attack rate $\dot{\alpha}$ (slope of curve formed by C_α verses α), rad^{-1} (deg^{-1}).

C_{Ny} = coefficient corresponding to component of normal force on yb-axis, dimensionless.

C_{Nz} = coefficient corresponding to component of normal force on zb-axis, dimensionless.

C_{n_r} = yaw damping derivative relative to yaw rate $\dot{r}$, rad^{-1}(deg^{-1}).

C_{nref} = yawing moment coefficient about reference moment station, dimensionless.

$C_{n_{\dot{\beta}}}$ = yaw damping derivative relative to angle of sideslip rate $\dot{\beta}$, rad-l (deg-1).

x_{cm} = instantaneous distance from rocket nose to center of mass, m.

x_{ref} = distance from rocket nose to reference moment station, m.

δ_r = effective control surface deflection causing rolling moment, rad (deg).

Considering that the rocket we intend to control is via fin deflection, fins on the rocket will be designated as shown in **Figure 2**.

Hence, the following moment coefficients are also given as

$$\begin{aligned}
C_{m_{ref}} &= C_{m_\alpha}\alpha + C_{m_\delta}\delta_\eta \\
C_{n_{ref}} &= C_{n_\beta}\beta + C_{n_\delta}\delta_\zeta
\end{aligned} \text{, dimensionless} \tag{14}$$

where

C_{mref} = pitching moment coefficient about reference moment station (this is the static value normally measured in the wind tunnel.), dimensionless.

C_{m_α} = slope of curve formed by pitch moment coefficient. C_m versus angle of attack, α rad^{-1} (deg^{-1}) slope of tune formed.

C_{m_δ} = slope of tune formed by pitch moment coefficient C_m versus control surface deflection for pitch, δ_p rad^{-1} (deg^{-1}).

C_{n_β} = slope of curve formed by yawing moment coefficient C_n versus angle of sideslip β, rad^{-1} (deg^{-1}).

C_{n_δ} = slope of curve formed by yaw moment coefficient C_n versus effective control surface deflection for yaw, δ_y rad^{-1} (deg^{-1}).

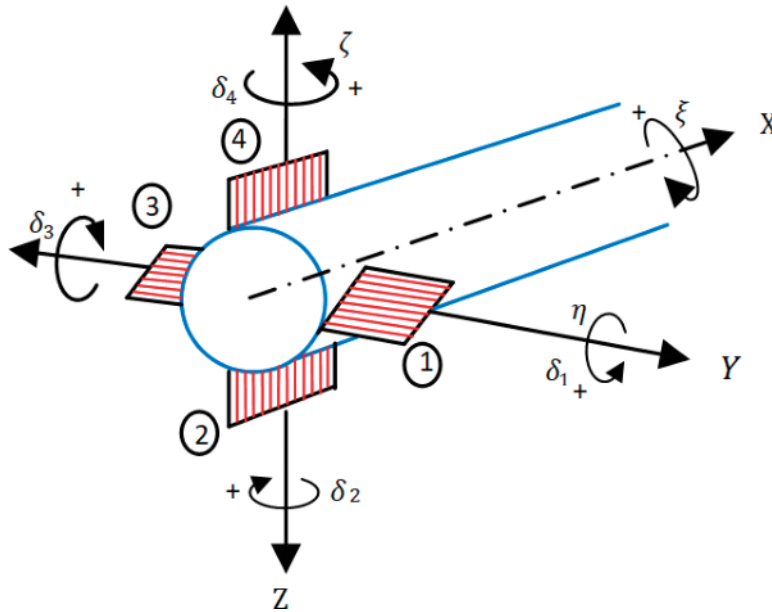**Figure 2.** Fin control and designation for control.

α = angle of attack, rad (deg).

β = angle of sideslip, rad (deg).

$\delta_\eta = \delta_1 = \delta_3$ effective control surface deflection causing pitching moment, rad (deg).

$\delta_\zeta = \delta_4 = \delta_2$ = effective control surface deflection causing yawing moment, rad (deg).

The angle of attack, angle of sideslip, and roll angle required for the realization of the aerodynamic coefficients are

$$\alpha = Tan^{-1}\left(\frac{w}{u}\right) \text{or } \alpha_t = \tan^{-1}\left[\sqrt{\left(\frac{v^2 + w^2}{u}\right)}\right]$$

$$\beta = \sin^{-1}\left(\frac{v}{V_M}\right) \qquad\qquad , \text{rad (deg)} \qquad\qquad (15)$$

$$\phi = \tan^{-1}\left(\frac{v}{w}\right)$$

where $\varnothing$ is aerodynamic roll angle, rad (deg), and α_t is the total angle of attack.

Table 1 gives a list of the aerodynamic coefficients that must be obtained for every rocket design before a model can be realized. There exist numerical and semi-numerical means of obtaining such coefficients. Examples of software that can do semiempirical computation of such coefficients and their derivatives are Missile Digital DATCOM® [9] and Flexible Structures Simulator (FSS) [10]. Finally, a wind tunnel test is expected to validate and update all coefficients and their derivatives before the system engineer delves in the control design.

The third and final component to fully describe the motion of a rocket is its angular inclination or attitude. We chose the Euler angles to describe the attitude of the rocket.

2.3. The Euler angles

Missile attitude is required for a number of simulation functions including the calculation of angle of attack, seeker gimbal angles, fuze look angles, and warhead spray pattern. In simulations with six degrees of freedom, the missile attitude is calculated directly by integrating the set of equations that define the Euler angle rates, i.e.,

$$\dot{\phi} = p + (q\sin\phi + r\cos\phi)\tan\theta, rad/s(\deg/s)$$

$$\dot{\theta} = q\cos\phi - r\sin\phi, rad/s(\deg/s) \qquad\qquad (16)$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta}, rad/s(\deg/s)$$

where

θ = the Euler angle rotation in elevation (pitch angle), rad (deg).

ϕ = the Euler angle rotation in roll (roll angle), rad (deg).

$\dot{\psi}, \dot{\theta}, \dot{\phi}$ = rates of change of the Euler angles in heading, pitch, and roll, respectively, rad/s (deg/s).

P, q, r = components of angular rate vector w expressed in body coordinate system (roll, pitch, and yaw, respectively), rad/s.

The three heading angle of $\dot{\psi}, \dot{\theta}, \dot{\phi}$ can be measured directly using a magnetometer. Note also that the Euler angles in (16) can all be derived by integrating gyroscopic measurements. As such we might not need an instrument that will measure it directly. Nevertheless, a magnetometer can be added to the instrumentation on board to measure heading.

S/N	Variable	Description
1	C_N	Normal force coefficient (body axis)
2	C_L	Lift coefficient (wind axis)
3	C_M	Pitching moment coefficient (body axis)
4	X_{cp}	Center of pressure in calibers from moment reference center
5	C_A	Axial force coefficient (body axis)
6	C_D	Drag coefficient (wind axis)
7	C_Y	Side force coefficient (body axis)
8	C_n	Yawing moment coefficient (body axis)
9	C_l	Rolling moment coefficient (body axis)
10	$C_{N\alpha}$	Normal force coefficient derivative with angle of attack
11	$C_{M\alpha}$	Pitching moment coefficient derivative with angle of attack
12	$C_{Y\beta}$	Side force coefficient derivative with sideslip angle
13	$C_{n\beta}$	Yawing moment coefficient derivative with sideslip angle (body axis)
14	$C_{l\beta}$	Rolling moment coefficient derivative with sideslip angle (body axis)
15	C_{Mq}	Pitching moment coefficient derivative with pitch rate
16	C_{Nq}	Normal force coefficient derivative with pitch rate
17	C_{Aq}	Axial force coefficient derivative with pitch rate
18	$C_{M\dot{\alpha}}$	Pitching moment derivative with rate of change of angle of attack
19	$C_{N\dot{\alpha}}$	Normal force derivative with rate of change of angle of attack
20	C_{lp}	Rolling moment coefficient derivative with roll rate
21	C_{np}	Yawing moment coefficient derivative with roll rate
22	C_{Yp}	Side force coefficient derivative with roll rate
23	C_{lr}	Rolling moment coefficient derivative with yaw rate
24	C_{nr}	Yawing moment coefficient derivative with yaw rate
25	C_{Yr}	Side force coefficient derivative with yaw rate

Table 1. Aerodynamic coefficients as a function of angle of attack.

Combining (1), (11), and (16) gives a complete six-degree-of-freedom equation of motion for a rocket in flight as shown in **Figure 3**. This could be written together as given in (17). In today's modern aerospace industry, a single device like the MPU6050, a MEM-based integrated chip, can be used to give numerical values for state variables of (17) on any dynamic body. For control system design, the rocket system as described in (17) needs to be separated into the two planes (decouple); these are called the lateral (*la*) and longitudinal (*lo*) dynamic equations of motion:

$$\dot{u} = \frac{F_{A_{x_b}} + F_{P_{x_b}} + F_{g_{x_b}}}{m} - (qw - rv), m/s^2$$

$$\dot{v} = \frac{F_{A_{y_b}} + F_{P_{y_b}} + F_{g_{y_b}}}{m} - (ru - pw), m/s^2$$

$$\dot{w} = \frac{F_{A_{z_b}} + F_{P_{z_b}} + F_{g_{z_b}}}{m} - (pv - qu), m/s^2$$

$$\dot{p} = \frac{L_A + L_p - qr(I_z - I_y)}{I_x}, rad/s^2 \left(deg/s^2\right)$$

$$\dot{q} = \frac{M_A + M_p - rp(I_x - I_z)}{I_y}, rad/s^2 \left(deg/s^2\right) \qquad (17)$$

$$\dot{r} = \frac{N_A + N_p - pq(I_y - I_x)}{I_z}, rad/s^2 \left(deg/s^2\right)$$

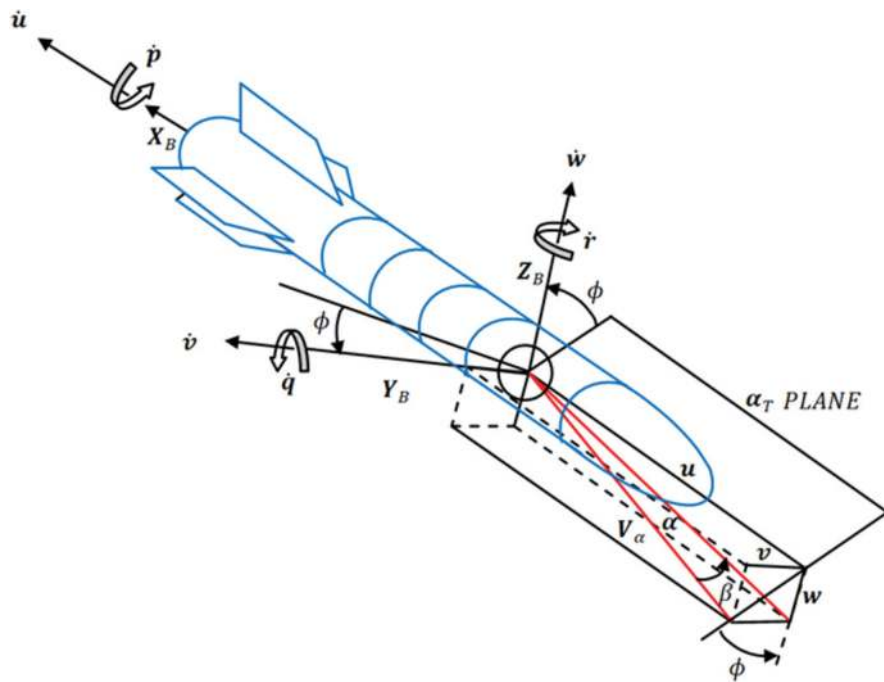**Figure 3.** Six-degree-of-freedom motion of a rocket.

$$\dot{\phi} = p + (q\sin\phi + r\cos\phi)\tan\theta, \, rad/s(\text{deg}/s)$$

$$\dot{\theta} = q\cos\phi - r\sin\phi, \, rad/s(\text{deg}/s)$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta}, \, rad/s(\text{deg}/s)$$

Since the control system we intend to design is in a family of linear controllers and (17) is a nonlinear system of differential equations, trimming and linearization must be done.

2.4. Trimming

To trim or find equilibrium values requires a good knowledge of advance computational techniques. A trim point, also known as an equilibrium point, is a point in the parameter space of a dynamic system at which the system is in a steady state. The trim problem for a rocket can be described as finding a set of suitable input values to satisfy a set of conditions. Hence, a trim point involves setting of its controls that causes the rocket to fly straight and level in the longitudinal plane. The suitable input values are the control surface deflections, the thrust setting, and the rocket attitude [11]. The set of conditions are the rocket's accelerations. The variables associated with the trim problems can be divided into three categories:

- Objective variables

- Control variables

- Flight condition variables

The objective variables need to be driven toward the specified values, often zero (i.e., steady flight with zero sideslip). The objective parameters are combined in the objective vectors o_{la} (state vectors) and o_{lo}, for the lateral and longitudinal missile dynamics as

$$o_{la} = \begin{bmatrix} \dot{v} & \dot{p} & \dot{r} & \beta \end{bmatrix}^T. \tag{18}$$

$$o_{lo} = \begin{bmatrix} \dot{u} & \dot{w} & \dot{q} & \alpha \end{bmatrix}^T. \tag{19}$$

The sideslip angle is also included, since for most cases, there are multiple solutions to the trim problem, each with a different sideslip angle. In the desired solution, the sideslip angle should be zero. In that case, the drag is at a minimum. The control parameters are adjusted in order to drive the objective parameters to their specified values. Together, they form the control vectors c_{la} and c_{lo}, described in (22). The control variables (input variables) describe the trimmed pilot control and the aircraft attitude:

$$c_{la} = \begin{bmatrix} \delta_a & \delta_r & \phi & \psi \end{bmatrix}^T, \tag{20}$$

$$c_{lo} = \begin{bmatrix} \delta_e & \tau & \theta \end{bmatrix}^T, \tag{21}$$

Finally, the 12 states of the 6DoF equation of motion must be initialized with the initial state conditions. In MATLAB®, the *trim* command is used to find equilibrium points. The object of

trimming is to bring the forces and moments acting on the rocket into a state of equilibrium. That is the condition when the axial, normal, and side forces and the roll, pitch, and yaw moments are all zero. Mathematically, trimming combines implicit and explicit Jacobian approaches. The Jacobian trim approach is the preferred method for rigid rocket. The Jacobian method is robust, since trim convergence is likely to occur even with rough estimates of the Jacobian and a rough first guess. Note that in general, any optimization routine could be used to solve the trim problem, as long as it is robust enough.

The Jacobian approach is based on the assumption that the change of the objective vector is linearly related to the change in the control input, which is shown in (22):

$$o_{i+1} - o_i = J_i[c_{i+1} - c_i]. \tag{22}$$

In (22), J_i is the Jacobian matrix evaluated near control input c_i. Its entries are first-order partial derivatives and represent the effect of changes in each control input on acceleration.

Note that changes in lateral plane induce changes in the longitudinal plane and vice versa; thus, we can write (23) the Jacobian for the lateral dynamics and (24) for the longitudinal or pitch dynamics:

$$J_i = \frac{\partial o}{\partial c}\bigg|_{c_{i(la)}} = \begin{bmatrix} \frac{\partial \dot{u}}{\partial \delta_a} & \frac{\partial \dot{u}}{\partial \delta_r} & \frac{\partial \dot{u}}{\partial \phi} & \frac{\partial \dot{u}}{\partial \psi} \\ \frac{\partial \dot{v}}{\partial \delta_a} & \frac{\partial \dot{v}}{\partial \delta_r} & \frac{\partial \dot{v}}{\partial \phi} & \frac{\partial \dot{v}}{\partial \psi} \\ \frac{\partial \dot{w}}{\partial \delta_a} & \frac{\partial \dot{w}}{\partial \delta_r} & \frac{\partial \dot{w}}{\partial \phi} & \frac{\partial \dot{w}}{\partial \psi} \\ \frac{\partial \dot{p}}{\partial \delta_a} & \frac{\partial \dot{p}}{\partial \delta_r} & \frac{\partial \dot{p}}{\partial \phi} & \frac{\partial \dot{p}}{\partial \psi} \\ \frac{\partial \dot{q}}{\partial \delta_a} & \frac{\partial \dot{q}}{\partial \delta_r} & \frac{\partial \dot{q}}{\partial \phi} & \frac{\partial \dot{q}}{\partial \psi} \\ \frac{\partial \dot{r}}{\partial \delta_a} & \frac{\partial \dot{r}}{\partial \delta_r} & \frac{\partial \dot{r}}{\partial \phi} & \frac{\partial \dot{r}}{\partial \psi} \\ \frac{\partial \beta}{\partial \delta_a} & \frac{\partial \beta}{\partial \delta_r} & \frac{\partial \beta}{\partial \phi} & \frac{\partial \beta}{\partial \psi} \end{bmatrix}_{c_i}. \tag{23}$$

$$J_i = \frac{\partial o}{\partial c}\bigg|_{c_{i(lo)}} = \begin{bmatrix} \frac{\partial \dot{u}}{\partial \delta_e} & \frac{\partial \dot{u}}{\partial \tau} & \frac{\partial \dot{u}}{\partial \theta} \\ \frac{\partial \dot{v}}{\partial \delta_e} & \frac{\partial \dot{v}}{\partial \tau} & \frac{\partial \dot{v}}{\partial \theta} \\ \frac{\partial \dot{w}}{\partial \delta_e} & \frac{\partial \dot{w}}{\partial \tau} & \frac{\partial \dot{w}}{\partial \theta} \\ \frac{\partial \dot{p}}{\partial \delta_e} & \frac{\partial \dot{p}}{\partial \tau} & \frac{\partial \dot{p}}{\partial \theta} \\ \frac{\partial \dot{q}}{\partial \delta_e} & \frac{\partial \dot{q}}{\partial \tau} & \frac{\partial \dot{q}}{\partial \theta} \\ \frac{\partial \dot{r}}{\partial \delta_e} & \frac{\partial \dot{r}}{\partial \tau} & \frac{\partial \dot{r}}{\partial \theta} \\ \frac{\partial \beta}{\partial \delta_e} & \frac{\partial \beta}{\partial \tau} & \frac{\partial \beta}{\partial \theta} \end{bmatrix}_{c_i}. \tag{24}$$

If the rocket is assumed to be at equilibrium, or trim condition, then the equations of motion can be linearized, and the 6DoF equation of motion can be resolved into their lateral and longitudinal states.

2.5. Linearization

The system of first-order nonlinear differential equations of the rocket as presented by (17) is said to be in state variable form if its mathematical model is described by a system of n first-order differential equations and an algebraic output equation as [12]:

$$
\begin{aligned}
\dot{x}_1 &= f_1(x_1, \ldots, x_n, u) \\
\dot{x}_2 &= f_2(x_1, \ldots, x_n, u) \\
&\ldots \\
\dot{x}_n &= f_n(x_1, \ldots, x_n, u) \\
y &= h(x_1, \ldots, x_n, u)
\end{aligned}
\tag{25}
$$

The column vector $x = [x_1, \ldots x_n]^{\mathrm{T}}$ is called the *state* of the system. The scalars u and y are called the control input and the system output, respectively, denoting

$$
f(x, u) =
\begin{bmatrix}
f_1(x_1, \ldots, x_n, u) \\
f_2(x_1, \ldots, x_n, u) \\
\vdots \\
f_n(x_1, \ldots, x_n, u)
\end{bmatrix}.
\tag{26}
$$

Concisely, (26) is written as

$$
\begin{aligned}
\dot{x} &= f(x, u), \\
y &= h(x, u).
\end{aligned}
\tag{27}
$$

where f and h are nonlinear functions of x and u; then, we say that the system is nonlinear. To linearize (26), we desire it to become

$$
\begin{aligned}
\dot{x} &= Ax + Bu \\
y &= Cx + Du.
\end{aligned}
\tag{28}
$$

where A is $n \times n$, B is $n \times 1$, C is $1 \times n$, and D is all scalar in MATLAB®/Simulink.

One reason for approximating the nonlinear system (26) by a linear model of the form (28) is that, by so doing, one can apply rather simple and systematic linear control design techniques. Given the nonlinear system (26) and an equilibrium or trimmed points $x^* = [x_1^* \ldots x_n^*]^T$ obtained when $u = u^*$, noting that $\Delta x = x = x^*$, we define a coordinate transformation as follows:

$$
\Delta x =
\begin{bmatrix}
\Delta x_1 \\
\vdots \\
\Delta x_n
\end{bmatrix}
=
\begin{bmatrix}
x_1 - x_1^* \\
\vdots \\
x_n - x_n^*
\end{bmatrix}.
$$

Further, denoting $\Delta u = u - u^*$, $\Delta y = y - h(x^*, u^*)$. The new coordinates Δx, Δu, and Δy represent the variations of x, u, and y from their equilibrium values. You have to think of these as a

new state, new control input, and new output, respectively. The linearization of (26) is at x in which the equilibrium or trim [12] state is given by

$$\Delta \dot{x} = A\Delta x + B\Delta u$$
$$\Delta y = C\Delta x + D\Delta u,$$
(29)

where

$$A = \left[\frac{\partial f}{\partial u}\right]_{x^*, u^*} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1}(x_1^*, \ldots, x_n^*, u^*) & \cdots & \dfrac{\partial f_1}{\partial x_n}(x_1^*, \ldots, x_n^*, u^*) \\ \cdots & \cdots & \cdots \\ \dfrac{\partial f_n}{\partial x_1}(x_1^*, \ldots, x_n^*, u^*) & \cdots & \dfrac{\partial f_n}{\partial x_n}(x_1^*, \ldots, x_n^*, u^*) \end{bmatrix},$$

$$B = \left[\frac{\partial f}{\partial u}\right]_{x^*, u^*} = \begin{bmatrix} \dfrac{\partial f_1}{\partial u}(x_1^*, \ldots, x_n^*, u^*) \\ \vdots \\ \dfrac{\partial f_n}{\partial u}(x_1^*, \ldots, x_n^*, u^*) \end{bmatrix},$$

$$C = \left[\frac{\partial h}{\partial x}\right]_{x^*, u^*} = \begin{bmatrix} \dfrac{\partial h}{\partial x_1}(x_1^*, \ldots, x_n^*, u^*) & \cdots & \dfrac{\partial h}{\partial x_n}(x_1^*, \ldots, x_n^*, u^*) \end{bmatrix}, D = \left[\frac{\partial h}{\partial u}\right]_{x^*, u^*}.$$

Note that the matrices A, B, C, and D have constant coefficients in that all partial derivatives are evaluated at the numerical values $(x_1^*, \ldots, x_n^*, u^*)$.

2.6. Lateral dynamics of a rocket

Equations of motion in the lateral plane are described by (30). Note that (30) comprises of one of the force equations (F_y), one of the momentum equations (M_y), and two of the Euler angles from the 6DoF equations (decoupled from (17)):

$$\dot{v} = \frac{F_{A_{y_b}} + F_{P_{y_b}} + F_{g_{y_b}}}{m} - (ru - pw), m/s^2$$

$$\dot{p} = \frac{L_A + L_p - qr(I_z - I_y)}{I_x}, rad/s^2 \left(deg/s^2\right)$$

$$\dot{r} = \frac{N_A + N_p - pq(I_y - I_x)}{I_z}, rad/s^2 \left(deg/s^2\right)$$
(30)

$$\dot{\phi} = p + (q\sin\phi + r\cos\phi)\tan\theta, rad/s(deg/s)$$

$$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta}, rad/s(deg/s)$$

For a completely computed aerodynamic coefficients and their derivatives, (30) can be expressed in state-space form [13] as

$$
\begin{bmatrix} \dot{v} \\ \dot{p} \\ \dot{r} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} y_v & y_p & y_r & y_\phi & y_\psi \\ l_v & l_p & l_r & l_\phi & l_\psi \\ n_v & n_p & n_r & n_\phi & n_\psi \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ r \\ \phi \\ \psi \end{bmatrix} + \begin{bmatrix} y_\xi & y_\zeta \\ l_\xi & l_\zeta \\ n_\xi & n_\zeta \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \xi \\ \zeta \end{bmatrix}.
\tag{31}
$$

All the variables in *A* matrix of (31) are the lateral dimensionless aerodynamic stability derivatives with respect to the system state vectors. The variables in *B* matrix of (31) are the lateral dimensionless aerodynamic control derivatives with respect to the designated control surfaces.

2.7. Longitudinal dynamics of a rocket

The longitudinal dynamics in motion is also called the pitch plane. Equations describing the motion of a rocket in this plane can be described as given in (32). Note that (32) comprises two of the force equations (F_x and F_z), two of the momentum equations (M_x and M_z), and two of the Euler angles from the 6DoF equations as given in (17):

$$
\dot{u} = \frac{F_{A_{x_b}} + F_{P_{x_b}} + F_{g_{x_b}}}{m} - (qw - rv), m/s^2
$$

$$
\dot{w} = \frac{F_{A_{z_b}} + F_{P_{z_b}} + F_{g_{z_b}}}{m} - (pv - qu), m/s^2
$$

$$
\dot{q} = \frac{M_A + M_p - rp(I_x - I_z)}{I_y}, rad/s^2 \left(deg/s^2\right)
\tag{32}
$$

$$
\dot{\theta} = q \cos \phi - r \sin \phi, rad/s(deg/s)
$$

Just as with (31), (32) can also be re-expressed in state space as

$$
\begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} x_u & x_w & x_q & x_\theta \\ z_u & z_w & z_q & z_\theta \\ m_u & m_w & m_q & m_\theta \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} x_\eta & x_\tau \\ z_\eta & z_\tau \\ m_\eta & m_\tau \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \eta \\ \tau \end{bmatrix}
\tag{33}
$$

In MATLAB® the *linmod* [14] command is used to invoke linearization. The assumption made for decoupling the linear model is that the cross-coupling effects between the two modes are negligible. These assumptions are:

- The rocket is designed with conventional control surfaces that do not give significant cross-coupling control between the lateral and longitudinal modes.

- The rocket is symmetrical about the *xz* plane in which the inertia cross-coupling in (*xy* and *xz* planes) result to cross-coupling between the lateral and longitudinal modes is minimum.

It can be shown that a typical trimmed and linearized model of the pitch plane (longitudinal dynamics) for a rocket [15] is given as presented in (34). Notice that compared to (33), the velocity in x-direction is not included in (34). This is basically due to the fact that in this pitch plane, translational motion for a rocket is predominantly in the z-direction (velocity w):

$$
\begin{bmatrix} \dot{\theta} \\ \dot{q} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 14.7805 & 0 & 0.01958 \\ -100.858 & 1 & -0.1256 \end{bmatrix} \begin{bmatrix} \theta \\ q \\ w \end{bmatrix} + \begin{bmatrix} 0 \\ 3.4858 \\ 20.42 \end{bmatrix} \delta_\eta + \begin{bmatrix} 0 \\ 14.7805 \\ -94.8557 \end{bmatrix} \alpha_w
$$

$$
y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ q \\ w \end{bmatrix}
$$

(34)

where

$$
A = \begin{bmatrix} 0 & 1 & 0 \\ 14.7805 & 0 & 0.01958 \\ -100.858 & 1 & -0.1256 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 3.4858 \\ 20.42 \end{bmatrix}, G = \begin{bmatrix} 0 \\ 14.7805 \\ -94.8557 \end{bmatrix} C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, D = 0.
$$

3. Discussion of result

From (34), it can be seen that a three state variable models have been realized in state space. This implies that modern observer like the Kalman filter can be designed to estimate and predict the trajectory of such rocket dynamics. This mathematical model also can be used to design all the control algorithms that fall in the class of modern (optimal theory) control. Particularly, this model is important in the realization of the longitudinal autopilot system of a rocket.

4. Conclusion

Mathematical models are the bedrock of almost all scientific activities. Here we were able to define the nonlinear airframe dynamics of a rocket in 6DoF. We went further to decouple the 6DoF equations of motion for the rocket and presented forms in which the decoupled 6DoF is easily trimmed and linearized with a computer program like MATLAB®. The process presented here can be repeated for any size of the rocket and aircrafts/unmanned aerial vehicle (UAV). Note that if the aerospace vehicle being model is not a rocket, and a state-space model is needed, all the procedures outlined in this chapter will be the same. The only changes that will be accommodated will come from the numerical values of the aerodynamic coefficients and their derivatives.

Author details

Aliyu Bhar Kisabo* and Aliyu Funmilayo Adebimpe

*Address all correspondence to: aliyukisabo@yahoo.com

Centre for Space Transport and Propulsion (CSTP), Epe, Lagos-State, Nigeria

References

[1] Papp I. Missile mathematical model and system design. Atlantic Association for Research in the Mathematical Sciences. 2017;**16**(1):29-35

[2] Farhan AF. State space model for autopilot design of aerospace vehicles. In: Weapons Systems Division. Defence Science and Technology Organisation, DSTO-TR-1990; 2007

[3] Chelaru T-V, Barbu C. Mathematical model for sounding rockets, using attitude and rotation angles. International Journal of Applied Mathematics and Informatics. 2009; **3**(2):35-44

[4] Belega B-A. A mathematical model for computing the trajectory of rockets in a resistant medium taking into account the Earth's rotation. The system differential equations of the rocket trajectory. In: International Conference of Scientific Paper AFASES 2012; 2012

[5] Siouris GM. Missile Guidance and Control Systems. New York: Springer-Verlag, Inc; 2004. ISBN: 0-387-00726-1

[6] Blakelock JH. Automatic Control of Aircraft and Missiles. New York: John Wiley & Sons, Inc; 1965

[7] Missile Flight Simulation Part One Surface-to-Air Missiles, MIL-HDBK-1211(MI) 17 July 1995. In: Military Handbook. 1995

[8] Tewari A. Advance Control of Aircraft, Spacecraft and Rockets. United Kingdom: John Wiley & Sons, Ltd; 2011. ISBN 9781119971207

[9] MISSILE DATCOM User's Manual—2011 Revision. Christopher Rosema, Joshua Doyle, Lamar Auman, and Mark Underwood-U.S. Army Aviation & Missile Research, Development and Engineering Centre. William B. Blake-Control Design and Analysis Branch Control Sciences Division. AFRL-RB-WP-TR-2011-3071 (Supersedes 2008 Revision, AFRL-RB-WP-TR-2009-3015)

[10] User Manual. Software Package for Flexible Structures Simulation (FSS). International Institute for Advanced Aerospace Technologies of St. Petersburg State University of Aerospace Instrumentation. Performed according to the Contract No. I-011-2012; 2013

[11] Yang WY et al. Applied Numerical Methods Using MATLAB. Hoboken, New Jersey: John Wiley & Sons, Inc; 2005. ISBN 0-471-69833-4

[12] White RE. Computational Mathematics: Models, Methods, and Analysis with MATLAB and MPI. New York: Chaoman & Hall/CRC; 2004. ISBN: 1-58488-364-2

[13] Cook MV. Flight Dynamics Principles: A Linear Approach to Aircraft Stability and Control. 2nd ed. Burlington, USA: Butterworth-Heinemann. 2007. ISBN: 978-0-7506-6927-6

[14] User's Guide. Control System ToolboxTM. Revised for Version 9.7 (Release 2014a). The MathWorks, Inc; 2014

[15] Kisabo AB. Expendable Launch Vehicle Flight Control-Design & Simulation with Matlab/Simulink. Germany: Lambert Academic Publishing (LAP). ISBN: 978-3-8443-2729-8; 2011