Nonlinear Output Feedback Control and Simulations for launch vehicles

University of Rome Sapienza

Department of Computer, Control and Management Engineering Antonio Ruberti (DIAG)

Phd in: AUTOMATIC CONTROL, BIOENGINEERING AND OPERATIONS RESEARCH ABRO

Andrea Ilgrande
Matricola 1563861

Supervisors
Prof. Stefano Battilotti
Prof. Salvatore Monaco

34th cycle

TABLE OF CONTENTS

1. Introduction

Launchers are vehicles that have the objective of bringing payloads in orbit. Their story and evolution is tightly linked to rocket technology development.

The design of a launch vehicle is a very challenging task, since it requires the combination of a wide variety of disciplines (aerospace, electronics, software, control etc.).

In this thesis the focus is put on the control design for space launchers, so on the algorithms that have to command the actuators in order to keep the desired trajectory and thus lead to the mission success. The control design is a part of the discipline that takes the name of "Guidance, Navigation and Control" (GNC).

For the control engineer the GNC domain is divided in:

- Define the reference for the control system → guidance
- Provide measurements to the control system → navigation
- Compute the desired commands for the actuators in order to follow the reference trajectory → control

The launch vehicle is an autonomous system and thus the control design plays a crucial role for the mission success.
The flight Software, which runs on the On Board Computer, cyclically executes the GNC (Guidance Navigation and Control) algorithms to control the launcher from lift-off up to payloads release.

The control design for a launcher vehicle presents several difficulties due to the complexity of the system. Some of the reasons that discourage from performing and implementing complicated control laws are:

- The intrinsic instability of the vehicle
- The presence of several disturbances (e.g. aerodynamic loads and elastic modes)

- The high variation of the structure (time varying properties)
- The algorithms computational time.

In the next chapters will be recapped a compact set of equations (see [1], [3], [4]) that allows to model the launcher dynamics and will be presented the design of nonlinear feedback output regulation in absence of full state information.

1.1. Guidance

Guidance is the discipline that has as scope the generation of the trajectory to follow in order to achieve the mission destination. Specifically, the guidance system provides to the control system the reference inputs needed to track the optimal trajectory.
Guidance can either be open loop or can adjust the reference based on the current launcher conditions (closed loop).
Commonly three different phases are considered: atmospheric flight, exoatmospheric flight, and coast flight.

1.1.1. Atmospheric Flight

In this phase the objective is to minimize the aerodynamic load and heating to avoid severe damages to the vehicle. The atmospheric forces are a function of the angle of attack of the trajectory must be designed to minimize the angle of attack due to high dynamic pressure ($Q^*\alpha$ constraint). The trajectory is computed offline (before the flight) and provided as input to the attitude steering commands to the vehicle.

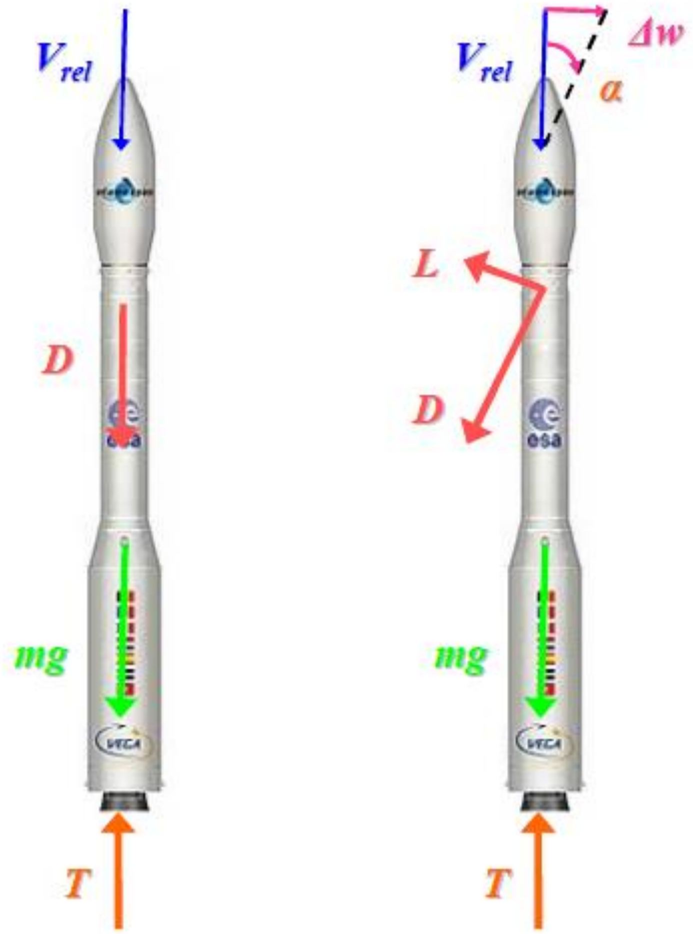

FIG. 1 – Aerodynamic forces acting on launch vehicle (see [1])

In this figure it can be seen how a small disturbance that impacts in a variation of the angle of attack causes the generation of an aerodynamic moment (lift force "L") that leads to an unstable launcher behavior.

Another phenomenon that has to be taken into account carefully is the elastic behavior of the structure. The flexible dynamics introduces frequencies that impacts the control frequency domain. Typically, in the control design are included specific filters in order to remove the bending frequency modes which otherwise could easily impacts the expected control behavior designed considering the rigid body motion. The elastic modes also impact the measurements done by the inertial platform device.

The time varying behavior due to the high amount of propellant that is consumed, especially in this phase, implies a huge variation in the structural properties of the vehicle (mass, inertia tensor) and due to the aerodynamic coefficients it causes a lot of difficulties in designing controllers and in verifying the stability of the vehicle under the action of the control laws. The absence of time varying metrics for stability leads to the need of a huge number of simulations for testing the behavior of the controller in different mission conditions.

1.1.2. Exoatmospheric Flight

The exoatmospheric manuevers requires though computationally expensive optimization solutions for the two-point boundary value problem. Due to the real time constraints of the on-board algorithms, complex on-line optimization problems are discouraged.

1.2. Navigation

The navigation algorithms compute the estimation of the vehicle position and velocity (linear and angular) from measurements provided by an aninertial measurement unit (IMU), which is composed by and electronic unit and a set of accelerometers and gyroscopes.

Basically, provided an initial position and velocity, the IMU integrates accelerations in three orthogonal directions to derive velocity. This result is then integrated to determine a new position as a function of time.

1.3. Control

The control system is devoted to command the actuators to achieve the desired trajectory and attitude based on the guidance computations.

Since the launcher operates in different environments, from sea level to outside of the atmosphere, the control task is very challenging in order to guarantee the stability of the vehicle in all these situations considering the different type of external perturbations and all the mass and structural variations that the launcher faces during the flight.

The main control actuator is called TVC ("Thrust Vector Control"), which consist in two electromechanical actuators that move the nozzle to produce specific torques to correct the launcher attitude.

Due to the axial symmetry of the launcher structure, the rigid body motion is typically divided in the pitch-yaw plane and roll plane. The TVC is devoted to keep the desired attitude in terms of pitch and yaw while for the roll control separated thrusters are commonly used. The roll dynamics is mostly stable during the flight and specific correction are enough to keep it minimal when necessary and continue to consider the dynamics decoupled.

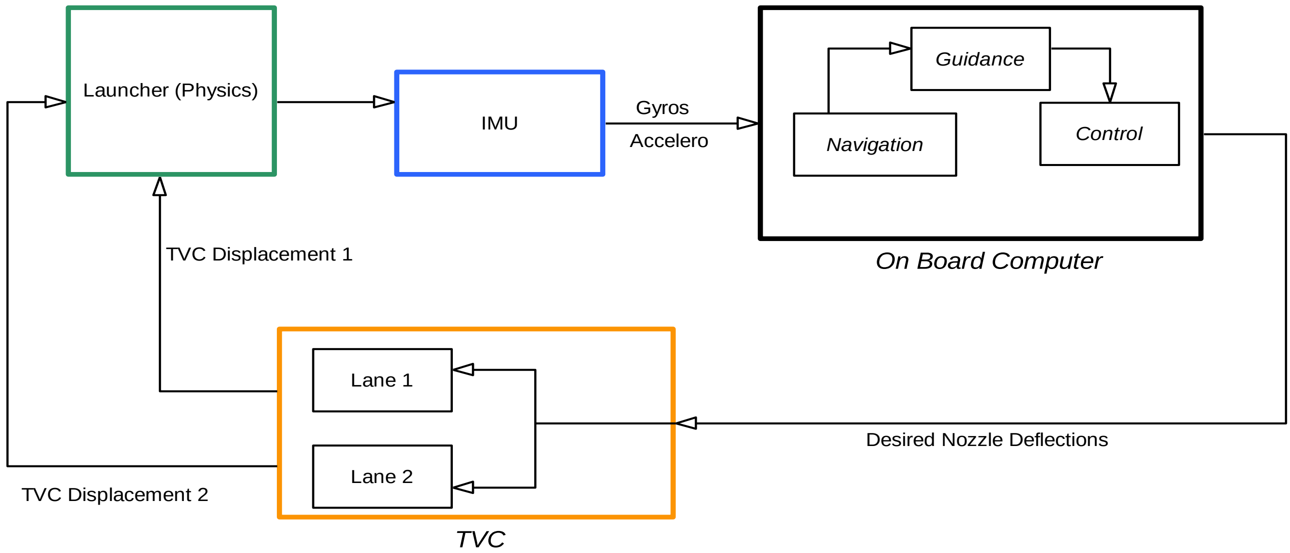

FIG. 2 – Common GNC scheme for launcher

2. Modeling

The modeling of a Launch vehicle is a very challenging task. The classical derivation of the equations of motion applies to general aerospace vehicles. What is crucial for the modeling is the characterization of the different types of environmental conditions that "disturbs" the classical rigid body dynamics and kinematics formulation. The modeling aspect is crucial for control system design, but also the choice of a proper model approximation for the controller tuning. This because the number of states and effects considered can make unfeasible the approach of designing a controller starting from the full nonlinear equations. Depending on the flight scenario and the mission time, different aspects become either more or less negligible. Knowing this is fundamental for the success of the controller.
In this chapter will be derived the equations of motions that can be used for analysis and control design of a launch vehicle.

2.1. Reference frame

In order to model properly the launch vehicle it is necessary to define some reference frames. Typically, several coordinates are used, from standard ECI (Earth Centered Inertial) and Body frame for rigid body and kinematics description, to orbit, trajectory and wind reference frames.

2.1.1. Common Reference frames

As stated before the reference frame definition is fundamental in order to characterize the motion of bodies in space. Below some of the commonly used frames for aerospace applications will be described.

- Inertial reference frame

  An inertial frame is not rotating with respect to the "fixed stars". In this frame the Newton's law of inertia are valid.

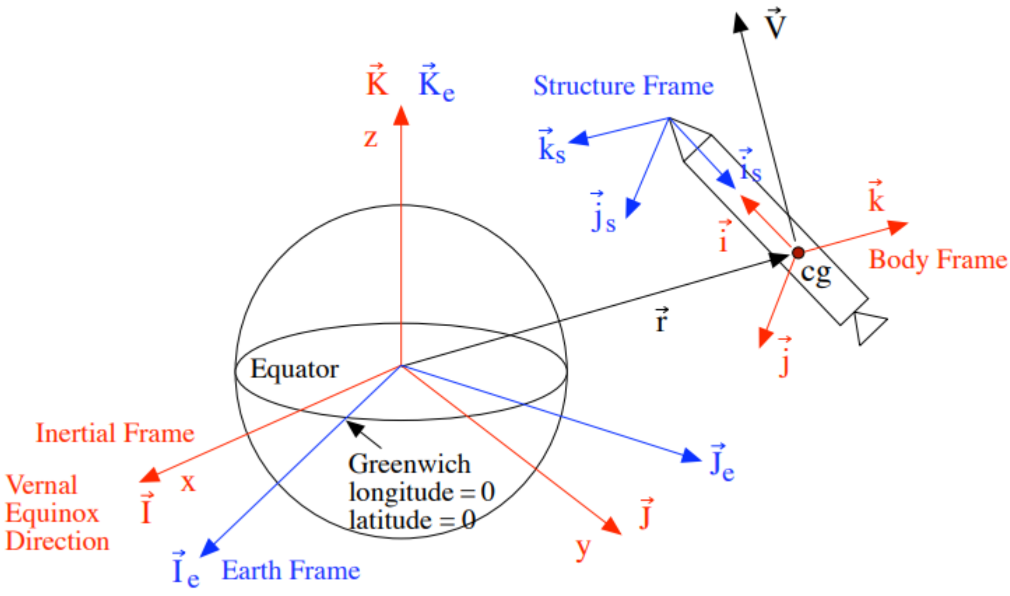

FIG. 3 – ECI reference frame (see [33]).

The frame has its origin in the earth center of mass.

The direction of I axis is aligned with the Earth's mean equator and equinox at 12:00 terrestrial time on 1st January 2000.

The direction of K axis is aligned with the earth's spin axis at 12:00 terrestrial time on 1st January 2000.

The direction of J is defined such that it completes the set of unit vectors to define a right-handed frame, and is rotated to the east about the celestial equator.

- <u>Body reference frame</u>

  Attached to the vehicle's body. This frame is mostly used for control systems design and when it is necessary to describe rigid body motion with respect to an inertial frame of reference.

- <u>Wind-axes frame</u>

During the atmospheric flight phase the launcher is subject to aerodynamic forces due to the wind action over its structure. It is useful to express these forces in the wind-axes frame.

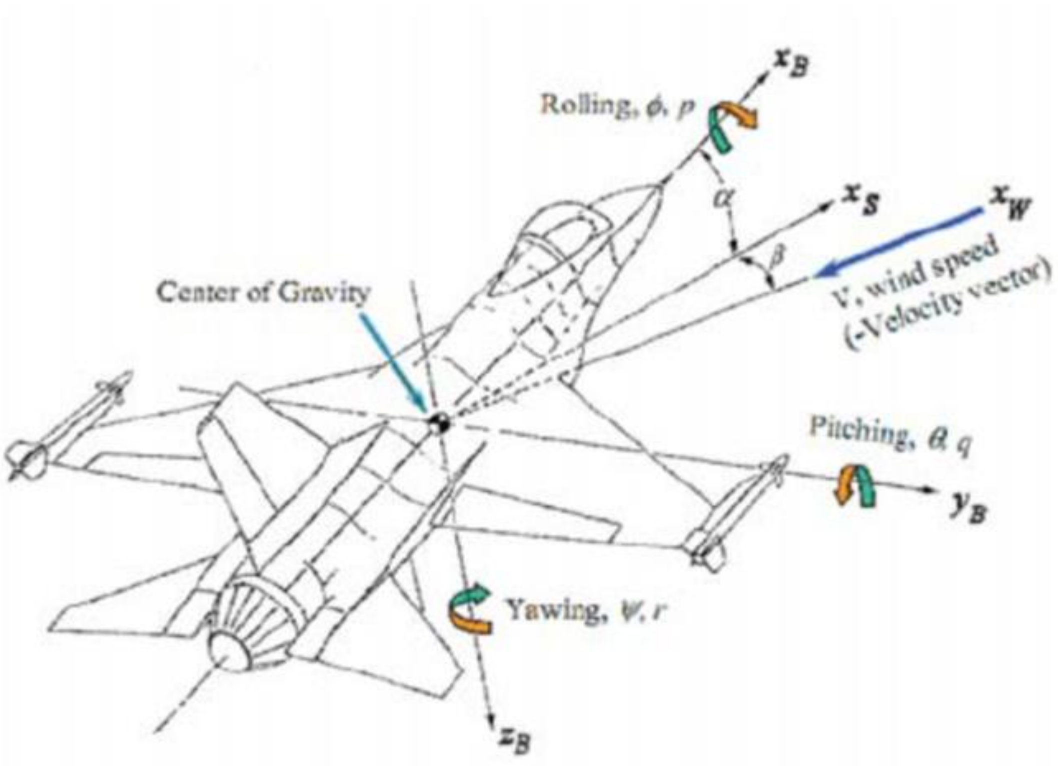

FIG. 4 – Example of Reference frame definition for aerospace application (see [20])

2.1.2. Euler Angles

Three parameters are sufficient to describe the orientation of a rigid body in space.
The most common and easy way to define the orientation of a rigid body with respect to an inertial reference frame are the Euler angles.
Firstly the concept of rotation matrix must be introduced. A rotation matrix is used to describe rotations of an element in in the two or three dimensional space. An elementary counterclockwise rotation of a vector in the bi-dimensional space is described by the following matrix.

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

The following equation expresses the relation between the original and the new rotated vector

$$\boldsymbol{v}' = R(\theta)\boldsymbol{v}$$

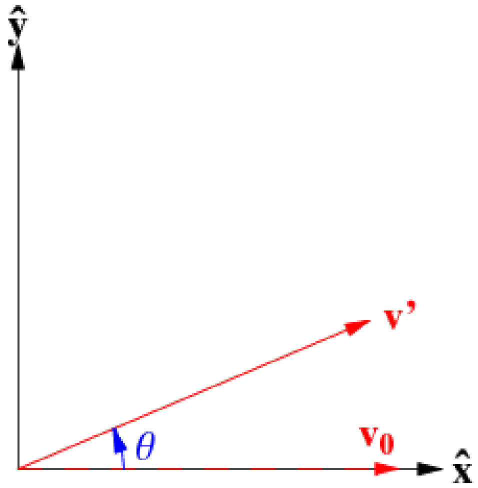

FIG. 5 – The body-fixed reference frame attached to the spacecraft.

For the 3 dimensional space three rotation matrices are defined; namely around the x-axis, the y-axis and the z-axis.

$$R(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\,(\theta_x) & sin\,(\theta_x) \\ 0 & -sin(\theta_x) & cos\,(\theta_x) \end{bmatrix}$$

$$R(\theta_y) = \begin{bmatrix} cos\,(\theta_y) & 0 & -sin(\theta_y) \\ 0 & 1 & 0 \\ sin(\theta_y) & 0 & cos\,(\theta_y) \end{bmatrix}$$

$$R(\theta_z) = \begin{bmatrix} cos\ (\theta_z) & sin\ (\theta_z) & 0 \\ -sin(\theta_z) & cos\ (\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotations matrices are referred also as direction cosine matrices, because their elements are the cosines of the unsigned angles between the body-fixed axes and the world axes.

The Euler rotation theorem states that every rotation can be obtained by means of three elementary rotations. Therefore, by fixing a set of rotation on each axis, it is possible to define every rotation from one frame to another.

Consider the rotation around the z-axis then around the y-axis and finally the x-axis, the resulting rotation matrix is given by:

$$R(\theta_x, \theta_y, \theta_z) = R(\theta_x)R(\theta_y)R(\theta_z)$$

$$= \begin{bmatrix} c\ (\theta_y)c\ (\theta_z) & c\ (\theta_y)s(\theta_z) & -s(\theta_y) \\ c(\theta_z)\, s(\theta_x)\, s(\theta_y) - c(\theta_x)\, s(\theta_z) & s(\theta_z)\, s(\theta_x)\, s(\theta_y) + c(\theta_x)\, c(\theta_z) & s(\theta_x)\, c(\theta_y) \\ c(\theta_x)\, c(\theta_z)\, s(\theta_y) + s(\theta_x)\, s(\theta_z) & c(\theta_x)\, s(\theta_z)\, s(\theta_y) - s(\theta_x)\, c(\theta_z) & c(\theta_x)\, c(\theta_y) \end{bmatrix}$$

This combination is named "3-2-1" (one of the Euler angles sets) and is widely used for aerospace applications.

These three angles are commonly named as roll, pitch and yaw.

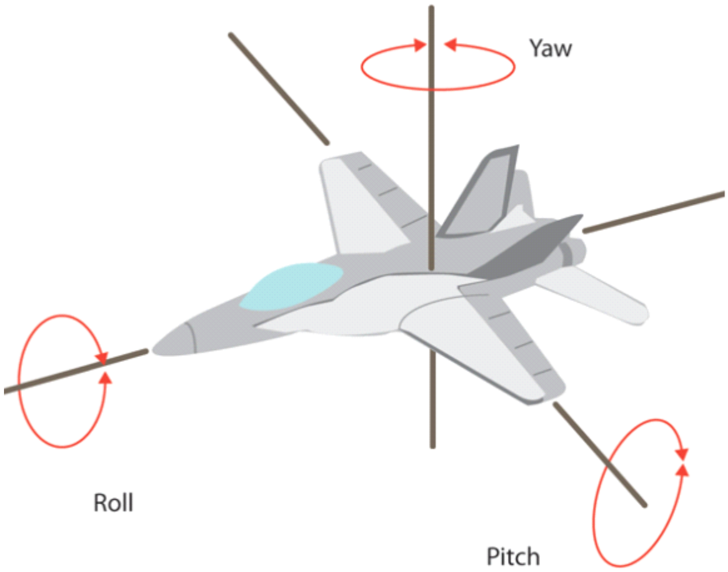

FIG. 6 – Roll, pitch and yaw angle (www.machinedesign.com)

There exist many combinations of Euler angles, each defining a different rotation matrix. Therefore, it is always necessary to define which set will be used to describe the system.

The angular velocity vector can be computed from the Euler angles rate, thus introducing the attitude kinematics equation for the vehicle:

$$\vec{\omega}_B = \dot{\varphi}\vec{\imath_B} + \dot{\theta}\vec{\jmath'_B} + \dot{\psi}\vec{k_B} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\varphi & s\varphi c\theta \\ 0 & -s\varphi & c\varphi c\theta \end{bmatrix} \cdot \begin{bmatrix} \dot{\varphi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

2.2. Launch Vehicle Equations

The equations of motion for a launch vehicle, considering the relative motion between particles and time varying properties due to fuel sloshing, gas flowing, flexible characteristics and engine rotation, are quite tough.

There are many books and famous papers that well describe this complex system, either for generic aerospace vehicles or more precisely for rocket launchers (See [3], [4], [22], [23], [24]).

The position of a mass element can be expressed, in the inertial reference frame, as follows:

$$\vec{\boldsymbol{R}}_{\boldsymbol{i}} = \vec{\boldsymbol{R}}_{\boldsymbol{0}} + \vec{\boldsymbol{r}}_{\boldsymbol{i}}$$

With $\vec{\boldsymbol{R}}_{\boldsymbol{0}}$ as the position of the body fixed frame origin in the inertial frame and $\vec{\boldsymbol{r}}_{\boldsymbol{i}}$ the position of the mass element in the body frame.

Differentiating the previous equation it yields:

$$\dot{\vec{\boldsymbol{R}}}_{\boldsymbol{\iota}} = \dot{\vec{\boldsymbol{R}}}_{\boldsymbol{0}} + \dot{\vec{\boldsymbol{r}}}_{\boldsymbol{\iota}} + \vec{\boldsymbol{\omega}}\, \boldsymbol{x}\, \vec{\boldsymbol{r}}_{\boldsymbol{\iota}}$$

where $\dot{\vec{R}}_0$ is the velocity of the body fixed-axis frame origin in the body reference frame, $\dot{\vec{R}}_\iota$the velocity of the mass element in the inertial reference frame and $\dot{\vec{r}}_\iota$ the velocity of the mass element in the body reference frame.

The expression of the derivative of $\vec{\boldsymbol{r}}_{\boldsymbol{i}}$ is due to the fact that this vector is not expressed in an inertial frame. Thus, its derivation is equal to the one of the vector modulus plus the derivation of the frame (by introducing the rotation velocity of the body frame with respect to the fixed inertial frame).

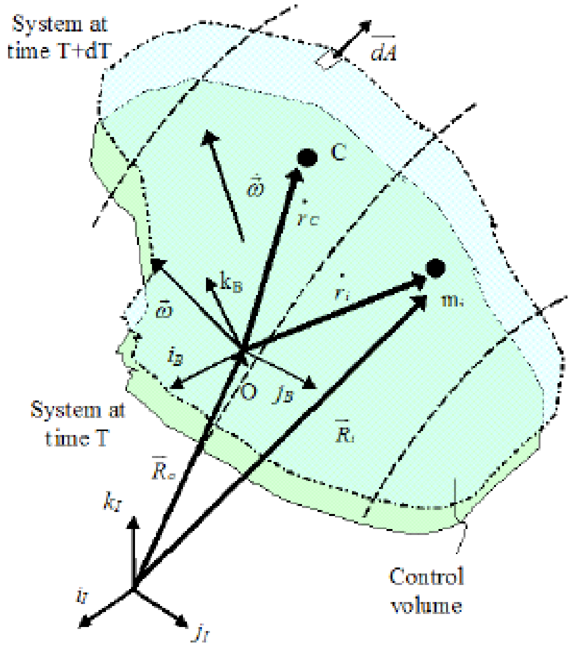

FIG. 7 – Reference frame and control volume (see [1])

So the derivation of a vector in a non inertial reference frame can be expressed as:

$$\dot{\vec{r_\iota}} = +\overrightarrow{\dot{r_\iota}} + \vec{\omega}\, x\, \vec{r_\iota}$$

while in an inertial reference frame the following expression holds:

$$\dot{\vec{R_\iota}} = \overrightarrow{\dot{R_\iota}}$$

By similar arguments, the acceleration of the mass element in the inertial frame can be derived:

$$\ddot{\vec{R_\iota}} = \ddot{\vec{R_0}} + \dot{\vec{\omega}}\, x\, \vec{r_\iota} + \vec{\omega}\, x\, (\vec{\omega}\, x\, \vec{r_\iota}) + \ddot{\vec{r_\iota}} + 2\vec{\omega}\, x\, \dot{\vec{r_\iota}}$$

In the case of solid-fuel launchers, the one studied in this presentation, the mass elements can be split between the ones related to the launcher structure and unburned fuel and the burned fuel ones. When deriving the velocity expression, it appears the contribution of the relative motion of the burned fuel mass elements with respect to the body reference frame.

$$\int_m \dot{\vec{r}}\, dm = \int_{m_{str}} \overrightarrow{\dot{r}_{str}}\, dm + \int_{m_f} \overrightarrow{\dot{r}_f}\, dm = \int_{m_{str}} \overrightarrow{\dot{r}_{str}}\, dm + \int_{m_f} \overrightarrow{v_f}\, dm$$

$$\dot{\vec{r}}_f = \overrightarrow{\dot{r}_{str}} + \overrightarrow{v_f}$$

From the previous equations for the control volume it can be derived the expression for the force equation

$$\vec{F} = \int_m \overrightarrow{\ddot{R}_\iota}\, dm$$

$$= \int_m [\overrightarrow{\ddot{R}_0} + \overrightarrow{\ddot{r}_{str}} + \dot{\vec{\omega}}\, x\, \overrightarrow{r_{str}} + \vec{\omega}\, x\, (\vec{\omega}\, x\, \overrightarrow{r_{str}}) + 2\vec{\omega}\, x\, \overrightarrow{\dot{r}_{str}}]dm$$

$$+ \int_m [\overrightarrow{\dot{v}_f} + 2\vec{\omega}\, x\, \vec{v}_f]\, dm$$

The previous equation can be rearranged as:

$$\vec{F} + \overrightarrow{F_u} + \overrightarrow{F_c} + \vec{F}_R = \int_m [\overrightarrow{\ddot{R}_0} + \overrightarrow{\ddot{r}_{str}} + \dot{\vec{\omega}}\, x\, \overrightarrow{r_{str}} + \vec{\omega}\, x\, (\vec{\omega}\, x\, \overrightarrow{r_{str}}) + 2\vec{\omega}\, x\, \overrightarrow{\dot{r}_{str}}]dm$$

with:

$$\overrightarrow{F_c} = -2\vec{\omega}\, x \int_{m_f} \overrightarrow{v_f}\, dm_f\,, \qquad Coriolis\ term$$

$$\overrightarrow{F_u} = -\frac{\partial}{\partial t} \int_{m_f} \overrightarrow{v_f}\, dm_f\,, \qquad Unsteadiness\ of\ the\ glass\ flow\ w.r.t\ the\ vehicle$$

$$\overrightarrow{F_R} = -\int_{A_f} \overrightarrow{v_f}\, (\rho_f \overrightarrow{v_f} \cdot \overrightarrow{dA}) = -\dot{m}\vec{u},\ \ m_f = \rho_f v_f, \qquad Reactive\ force\ term$$

The Reactive force needs a particular attention as it expresses the thrust force due to the exhausted gas.

The term $\dot{m}$ is the mass flow rate and $\vec{u}$ is the gas velocity vector.

The other terms on the right side of the forces equation are the typical terms of a rigid body with constant mass characteristics, with:

- D' Alambert term → $m\overrightarrow{\ddot{R}_0}$
- Euler force term → $\int_m \dot{\vec{\omega}}\ x\ \overrightarrow{r_{str}}\ dm$
- Coriolis force term → $\int_m \overrightarrow{\dot{r}_{str}}\ dm$
- Centrifugal force term → $\vec{\omega}\ x\ (\vec{\omega}\ x\ \int_m \overrightarrow{r_{str}}\ dm)$
- Force applied by structural dynamics → $\int_m \overrightarrow{\ddot{r}_{str}}\ dm$

Concerning the moments:

$$\overrightarrow{M_0} = \int_m \vec{r}\ x\ \ddot{\vec{R}}\ dm$$

can be rearranged by similar arguments as for the force equations:

$$\overrightarrow{M_0} + \overrightarrow{M_u} + \overrightarrow{M_c} + \overrightarrow{M_R} = \int_m \vec{r}_{str}\ \mathrm{x}\ [\,\overrightarrow{\ddot{R}_0} + \overrightarrow{\ddot{r}_{str}} + \dot{\vec{\omega}}\ x\ \overrightarrow{r_{str}} + \vec{\omega}\ x\ (\vec{\omega}\ x\ \overrightarrow{r_{str}}) + 2\vec{\omega}\ x\ \overrightarrow{\dot{r}_{str}}]dm$$

with:

$$\overrightarrow{M_c} = -2\int_{m_f} \overrightarrow{r_{str}}\ x\ (\vec{\omega}\ x\ \vec{v}_f)\ dm_f$$

$$\overrightarrow{M_u} = -\frac{\partial}{\partial t}\int_{m_f} \overrightarrow{r_{str}}\ x\ \ \vec{v}_f\ dm_f$$

$$\overrightarrow{M_R} = -\int_{A_f} (\overrightarrow{r_{str}}\ x\ \ \vec{v}_f)(\rho_f \vec{v}_f\ \cdot \overrightarrow{dA}$$

For more details refer to [1].

Together with the equations for rotational and translational dynamics, the flexible contributions must be taken into account (see [23]).

Considering the linear elastic framework, the modal representation can be formulated by means of the following expressions:

$$\vec{x}(t) = \sum_{i=0}^{\infty} \vec{\varphi}_i \cdot \xi_i(t)$$

with:

- $\xi_i(t)$ → i-th modal coordinate
- $\vec{\varphi}_i$ → i-th modal shape

The rigid body equation for the translation dynamics can be stated as:

$$\sum \overrightarrow{F_{ext}} = m(t)\overrightarrow{a_{CoG}} + \sum_i \dot{m}_i \vec{v}_i$$

This equation brings back to the classical Newton formulation for the rigid body motion where the presence of the variable mass is just approximated as an external force term (Thrust force).

With the approximation of the so called "Solidification Principle" (see [1]): "*The equations of the body of a launch vehicle at any instant t can be written in the*
form of equation of motion of a fictitious flexible body which would be obtained if it «solidified» at the instant t. To this «solidified» flexible body there should be applied, in addition to the external forces (F) on the vehicle, the only the reaction active."

Therefore, by neglecting the following terms:

$$\int_m \vec{\dot{r}}_{str}\, dm \approx 0, \qquad \int_m \vec{\ddot{r}}_{str}\, dm \approx 0$$

and defining the center of mass as:

$$\int_m \vec{r}_{str}\, dm = m\vec{r}_c$$

The translation equations can be stated as follows:

$$\vec{F} + \overrightarrow{F_R} = m \cdot [\overrightarrow{\ddot{R}_0} + \dot{\vec{\omega}}\, x\, \vec{r_c} + \vec{\omega}\, x\, (\vec{\omega}\, x\, \vec{r_c})]$$

By introducing the following expressions:

$$\vec{\dot{r}_0} = \begin{bmatrix} U \\ V \\ W \end{bmatrix}, \quad \vec{\ddot{r}_0} = \begin{bmatrix} \dot{U} \\ \dot{V} \\ \dot{W} \end{bmatrix}, \quad \vec{r_c} = \begin{bmatrix} x_{cg} \\ y_{cg} \\ z_{cg} \end{bmatrix}, \quad \vec{\omega} = \begin{bmatrix} P \\ Q \\ R \end{bmatrix}$$

The equations are rearranged as:

$$\vec{F_s} = m \begin{bmatrix} (\dot{U} + QW - RV) - x_{cg}(Q^2 + R^2) - y_{cg}(\dot{R} - PQ) + z_{cg}(\dot{Q} + PR) \\ (\dot{V} + RU - PW) + x_{cg}(\dot{R} + PQ) - y_{cg}(P^2 + R^2) + z_{cg}(\dot{P} - QR) \\ (\dot{W} + PV - QU) - x_{cg}(\dot{Q} - PR) + y_{cg}(\dot{P} + QR) + z_{cg}(P^2 + Q^2) \end{bmatrix}$$

By assuming that the center of mass is coincident with the origin of the body reference frame, the force equations are further simplified as:

$$\vec{F} = m \begin{bmatrix} \dot{U} + QW - RV \\ \dot{V} + RU - PW \\ \dot{W} + PV - QU \end{bmatrix}$$

Concerning the rotational dynamics, from Euler equations the moment expressions are:

$$\sum \vec{M}_{ExtCog} = \frac{d(I(t)\vec{\omega})}{dt} - \sum_i \vec{r_i}\, x\, (\dot{m}_i \vec{v_i})$$

Which can be written as (under the same assumptions considered for the translational equations plus that the cross inertia terms are negligible):

$$\vec{M} + \overrightarrow{M_R} = \begin{bmatrix} I_{xx}\dot{P} + (I_{zz} - I_{yy})QR \\ I_{yy}\dot{Q} + (I_{xx} - I_{zz})PR \\ I_{yy}\dot{R} + (I_{yy} - I_{xx})PQ \end{bmatrix}$$

About the elastic dynamics, in general a launch vehicle is characterized by having quite important bending modes with respect to the axial/torsion ones. For this reason, the elastic behavior of the vehicle can be modelled, in a first approximation, only considering the lateral deflections.

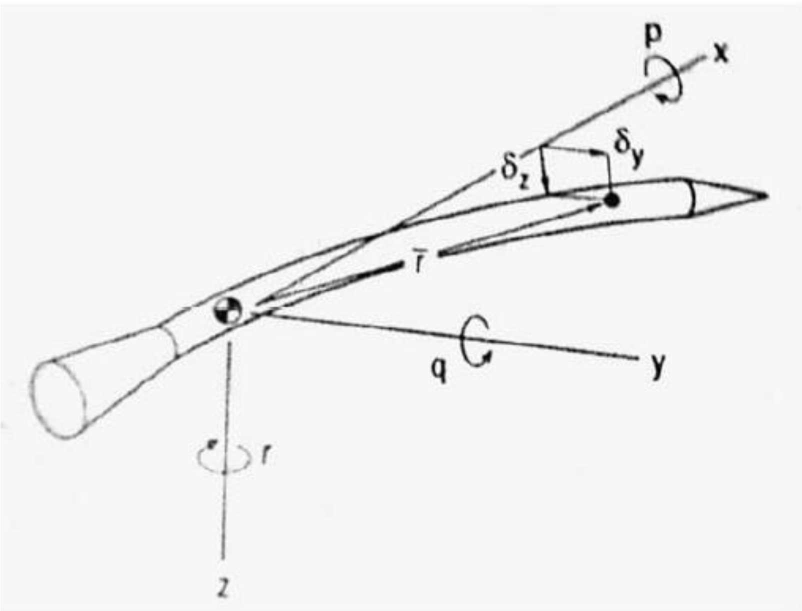

FIG. 8 – See ref [15]

The elastic deflections can be expressed as:

$$\delta_y(\eta, t) = \sum_{i}^{\infty} K_i(t) \cdot \phi_{y_i}(\eta) \approx \sum_{i}^{N_E} K_i(t) \cdot \phi_{y_i}(\eta)$$

$$\delta_z(\eta, t) = \sum_i^{\infty} \xi_i(t) \cdot \phi_{z_i}(\eta) \approx \sum_i^{N_E} \xi_i(t) \cdot \phi_{z_i}(\eta)$$

With ϕ_{y_i} representing the i-th mode shape on j_B axis and ϕ_{z_i} on k_B axis.

Due to the axial-symmetry property of most of the launch vehicles, it can be assumed that $\phi_{y_i} = \phi_{z_i} = \phi_i, for\ i = 1..N_E$, where N_E represents the number of degrees of freedom required to model the structural dynamics.

Considering the case of a "free-free" beam, the normal modes satisfy these conditions:

$$\int_L \phi_i(x)\phi_j(\eta)m(\eta)d\eta = \begin{cases} 0, i \neq j \\ M_i, i = j \end{cases}$$

$$\int_L \eta\phi_i(x)m(\eta)d\eta = 0$$

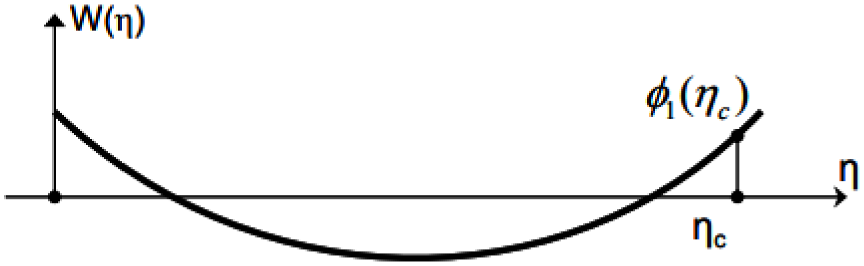

FIG. 9 – Mode shape for the case of a "free-free" beam See [4]

With M_i as the generalized mass due to elasticity for the i-th mode and $m(\eta)$ the mass per unit length.

The elastic angular deflections on j_B and k_B axes are determined by:

$$\gamma_y(\eta, t) \approx \sum_i^{N_E} \xi_i(t) \cdot \dot{\phi}_i(\eta)$$

$$\gamma_z(\eta, t) \approx \sum_{i}^{N_E} K_i(t) \cdot \dot{\phi}_\iota(\eta)$$

Using the Lagrange approach (detailed in Ref [15]), by means of the equations:

$$\frac{d}{dt}(\frac{\partial E_c}{\partial \dot{q}_i}) - \frac{\partial E_c}{\partial q_i} + \frac{\partial E_p}{\partial q_i} + \frac{\partial \mathrm{D}}{\partial q_i} = Q_i$$

With:

- qi → generalized coordinate i
- Ec → Kinetic energy
- Ep → Potential energy
- D → Damping energy
- Qi → generalized force associated to qi

The lateral elastic dynamics are found to be:

$$\ddot{K}_\iota + 2\varsigma_i\omega_i\dot{K}_\iota - 2P\dot{\xi}_\iota + (\omega_i^2 - P^2 - R^2)K_i + QR\xi_i = \frac{\int_L -F_y(\eta, t)\phi_i(\eta)d\eta}{\int_L m(\eta)\phi_i^2(\eta)d\eta}, i = 1..N_E$$

$$\ddot{\xi}_\iota + 2\varsigma_i\omega_i\dot{\xi}_\iota - 2P\dot{K}_\iota + (\omega_i^2 - P^2 - Q^2)\xi_i + QRK_i = \frac{\int_L -F_z(\eta, t)\phi_i(\eta)d\eta}{\int_L m(\eta)\phi_i^2(\eta)d\eta}, i = 1..N_E$$

From the previous two equations it can be highlighted the coupling between the rigid and elastic dynamics.

The elastic characteristics (mainly frequencies and mode shapes) vary during the flight and they are found by interpolation of different models computed at specific flight instants. For more details refer to [1].

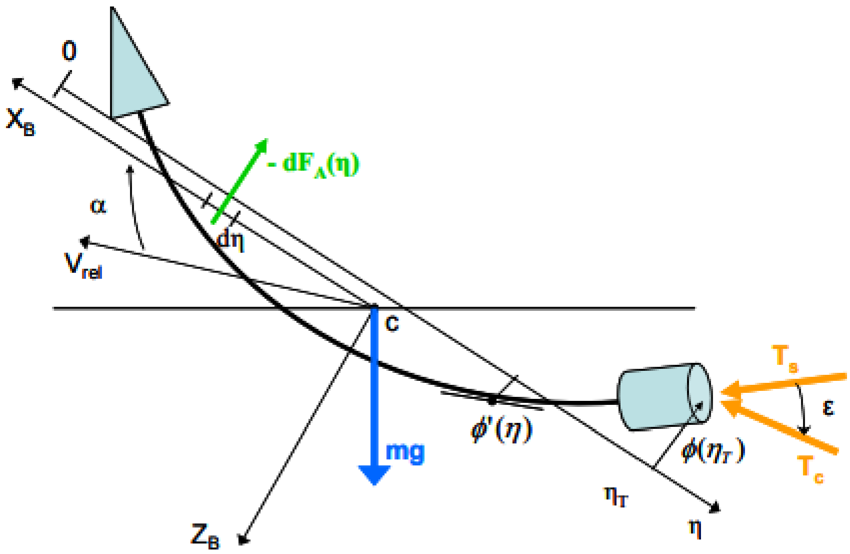

FIG. 10 – Schematic view of a launch vehicle with elastic deformation (pitch plane) See [1]

2.2.1. Forces and Moments

The main external terms that cause forces and moments to act on the launch vehicle can be summarized in the following categories:

- Aerodynamics
- Thrust
- Gravity

Other factors to be considered are the forces and moments caused by the nozzle and the sloshing of the propellant inside the tank.

2.2.1.1. Aerodynamics

The aerodynamics action is computed by means of parameters that express the forces and moments in the body axes.

$$\vec{F_A} = \frac{1}{2}\rho V_{rel}^2 S_R \begin{bmatrix} C_{F_{Ax}} \\ C_{F_{Ay}} \\ C_{F_{AN}} \end{bmatrix}$$

With:

- ρ → air density
- S_R → reference aerodynamic surface
- ρV_{rel}^2 → dynamic pressure
- $C_{F_{Ax}}, C_{F_{Ay}}, C_{F_{AN}}$ → nonlinear aerodynamic coefficients

The aerodynamic coefficients depend on many flight variables, so cannot be known apriori, but typically are obtained in terms of Mach number and the angle of attack. The uncertainty in the values of the aerodynamic parameters represents one of the main concerns for control design during the atmospheric flight.

Since the aerodynamic actions depend on the flight variables, the aerodynamic loads might be cause of coupling between the different axes. In general, the coupling forces are represented by using stability derivatives (See [16], [17]).

For axial-symmetric launch vehicles whit little lifting surfaces the main stability derivatives are related to the angle of attack (α) and the side-slip angle (β)

The expression for the aerodynamic forces can be summarized as:

$$F_{AX} = -\frac{1}{2}\rho V_{rel}^2 S_R [C_{A0} + \int_0^L \frac{\partial C_A(\eta)}{\partial \alpha} \alpha'(\eta) d\eta]$$

$$F_{AY} = -\frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha} \beta(\eta) d\eta]$$

$$F_{AZ} = -\frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha} \alpha(\eta) d\eta]$$

With α' total incidence angle

Concerning the moments contribution:

$$M_{AX} = 0$$

$$M_{AY} = \frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha} (\eta_{cg} - \eta)\beta(\eta)d\eta]$$

$$M_{AZ} = -\frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha} (\eta_{cg} - \eta)\alpha(\eta)d\eta]$$

In many aeronautical applications the forces are expressed in the wind reference frame as:

$$F_{AX_w} = -D$$

$$F_{AY_w} = Y$$

$$F_{AZ_w} = -L$$

With:

- D → drag force
- Y → side force
- L → lift force

In the wind frame a straightforward expression for the angle of attack and side slip is derived by means of:

$$\alpha = atan(\frac{W_r}{U_r})$$

$$\beta = atan(\frac{V_r}{U_r})$$

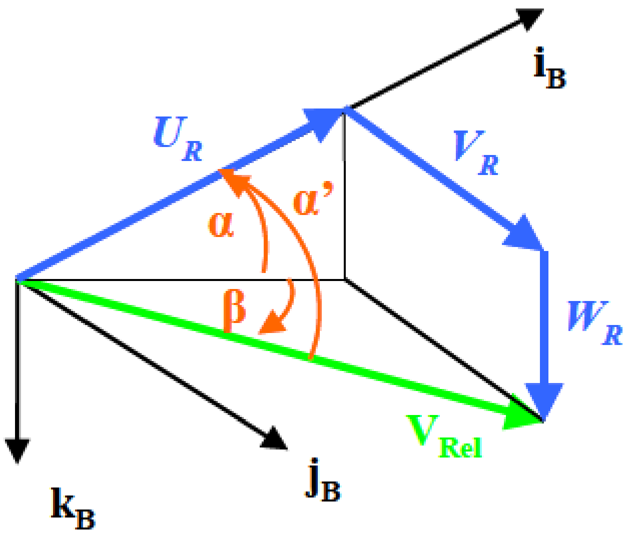

FIG. 11 – Aerodynamic angles (See [1])

For the contribution to the elastic dynamics (see [1]):

$$\int_L F_y(x,t)\phi_i(x)dx = \int_L (\frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha} \beta(\eta)d\eta)\, \phi_i(\eta)d\eta$$

$$\int_L F_z(\eta,t)\phi_i(\eta)d\eta = \int_L (\frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha} \alpha(\eta)d\eta)\, \phi_i(\eta)d\eta$$

2.2.1.2. Thrust

The propulsive force can be summarized as a force that is applied at the nozzle pivot point. The launcher is controlled by proper nozzle deflections via the so called "Thrust Vector Control" system generating the desired moments in order to keep the attitude and preserve stability all over the mission.

Commonly, the thrust is modeled with a simple profile that depends on the flight time in the proximity of the mission trajectory, since the thrust is influenced by the external pressure and thus depends on current altitude.

A common approximation consists in dividing the thrust component in two terms: swivelled and not swivelled thrust (See [1]).

$$T(t) = T_s(t) \; + \; T_c(t)$$

The forces can be computed as:

$$\overrightarrow{F_T} \; = \begin{bmatrix} T_s \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} c\varepsilon c\delta & -s\varepsilon & c\varepsilon s\delta \\ s\varepsilon c\delta & c\varepsilon & s\varepsilon s\delta \\ -s\varepsilon & 0 & c\varepsilon \end{bmatrix} \begin{bmatrix} T_c \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \gamma_z \\ \gamma_y \end{bmatrix} (T_c + \; T_s)$$

With $\delta \; and \; \varepsilon$ being the nozzle deflections in the pitch and yaw plane, respectively.
In case of small angles deflections the total forces, including elastic contributions, can be summarized as:

$$\overrightarrow{F_T} \; = \begin{bmatrix} T_s + T_c \\ T_c\varepsilon \; + \; (T_c + \; T_s) \sum_i^{N_E} K_i(t) \cdot \phi_i'(\eta_T) \\ -T_c\delta \; + \; (T_c + \; T_s) \sum_i^{N_E} \xi_i(t) \cdot \phi_i'(\eta_T) \end{bmatrix}$$

For the moments:

$$\overrightarrow{M_T} = \begin{bmatrix} M_x \\ l_c(-T_c\delta(T_c + T_s)\sum_i^{N_E} \xi_i(t)\cdot\phi_i'(\eta_T)) - (T_c + T_s)\sum_i^{N_E} \xi_i(t)\cdot\phi_i'(\eta_T) \\ -l_c(T_c\varepsilon(T_c + T_s)\sum_i^{N_E} K_i(t)\cdot\phi_i'(\eta_T)) - (T_c + T_s)\sum_i^{N_E} \xi_i(t)\cdot\phi_i'(\eta_T) \end{bmatrix}$$

For the elastic contributions, considering only the bending modes the generalized force lateral components are:

$$\int_L F_y(\eta,t)\phi_i(\eta)d\eta = -(T_c\delta + (T_c + T_s)\sum_i^{N_E} K_i(t)\cdot \phi_i'(\eta_T))\,\phi_i(\eta_T)$$

$$\int_L F_z(\eta,t)\phi_i(\eta)d\eta = -(T_c\varepsilon + (T_c + T_s)\sum_i^{N_E} \xi_i(t)\cdot \phi_i'(\eta_T))\,\phi_i(\eta_T)$$

2.2.1.3. Gravity

The gravity force is usually expressed in the inertial reference frame as:

$$\overrightarrow{F_G} = mg\begin{bmatrix} -s\theta_g c\varphi_g \\ c\theta_g c\varphi_g \\ -s\varphi_g \end{bmatrix}$$

By considering the typical mission length of an expendable launcher, the assumption to neglect the effects of the earth rotation can be done. Therefore, the angles $\varphi_g$and θ_g can be seen as the longitude and latitude at the current time of flight.

In the body frame the force component can be rearranged in order to get:

$$\overrightarrow{F_G} = m\begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix}$$

With g defined as:

$$g = \frac{GM_E}{R_E^2}$$

The moments can be neglected because the gravity field can be assumed to be uniform in the vicinity of the Earth (see [1]).

2.2.2. Final Equations

The system can be described by means of 6 D.o.f dynamics equations, plus N_E elastic equations and 3 kinematics equations (see [1]).

$$\begin{cases} m(\dot{U} + QW - RV) = -mg_x + (T_c + T_s) - \frac{1}{2}\rho V_{rel}^2 S_R [C_{A0} + \int_0^L \frac{\partial C_A(\eta)}{\partial \alpha}\alpha' d\eta] \\ m(\dot{V} + RU - PW) = T_c\varepsilon + (T_c + T_s)\sum_i^{N_E} K_i(t)\cdot \phi_i'(\eta_T) - \frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_A(\eta)}{\partial \alpha}\beta(\eta)d\eta - mg_y \\ m(\dot{W} + PV - QU) = -T_c\delta + (T_c + T_s)\sum_i^{N_E} \xi_i(t)\cdot \phi_i'(\eta_T) - \frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_A(\eta)}{\partial \alpha}\alpha(\eta)d\eta - mg_z \end{cases}$$

$$\begin{cases} I_{xx}\dot{P} + (I_{zz} - I_{yy})QR = M_x \\ I_{yy}\dot{Q} + (I_{xx} - I_{zz})PR = l_c(-T_c\delta + (T_c + T_s)\sum_i^{N_E} \xi_i(t)\cdot \phi_i'(\eta_T)) - (T_c + T_s)\sum_i^{N_E} \xi_i(t)\cdot \phi_i(\eta_T) + \frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha}(\eta_{cg} - \eta)\alpha(\eta)d\eta \\ I_{zz}\dot{R} + (I_{yy} - I_{xx})PQ = -l_c(T_c\varepsilon + (T_c + T_s)\sum_i^{N_E} K_i(t)\cdot \phi_i'(\eta_T)) - (T_c + T_s)\sum_i^{N_E} K_i(t)\cdot \phi_i(\eta_T) + \frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha}(\eta_{cg} - \eta)\beta(\eta)d\eta \end{cases}$$

$$(\ddot{K}_\iota + 2\varsigma_i\omega_i\dot{K}_\iota - 2P\dot{\xi}_\iota + (\omega_i^2 - P^2 - R^2)K_i + QR\xi_i)\int_L m(x)\phi_i^2(x)dx = \int_L (\frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha}\alpha(\eta)d\eta) - (-T_c\Delta\varepsilon(T_c + T_s)\sum_i^{N_E} K_i(t)\cdot \Phi_i'(\eta_T))\Phi_i(\eta_T), i = 1..N_E$$

$$(\ddot{\xi}_\iota + 2\varsigma_i\omega_i\dot{\xi}_\iota + 2P\dot{K}_\iota + (\omega_i^2 - P^2 - Q^2)K_i + QRK_i)\int_L m(x)\phi_i^2(x)dx = \int_L (\frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha}\alpha(\eta)d\eta) + (T_c\Delta\delta(T_c + T_s)\sum_i^{N_E} \xi_i(t)\cdot \Phi_i'(\eta_T))\Phi_i(\eta_T), i = 1..N_E$$

$$\begin{cases} \dot{\varphi} = p + s\varphi tn\theta \cdot q + c\varphi tn\theta \cdot r \\ \dot{\theta} = c\varphi \cdot q - s\varphi \cdot r \\ \dot{\psi} = \frac{s\varphi}{c\theta}\cdot q + \frac{c\varphi}{c\theta}\cdot r \end{cases}$$

2.2.3. Addionatal Elements

In order to better characterize the whole system dynamics, several other effects might be taken into account (see [1])., namely:

- Sloshing effect → due to the movement of the propellent inside the tank
- Actuator dynamics → The control action required by the controller cannot be instantaneously implemented and the actuators might not be able to dispense all the required actions (saturation and discrete time evolution)
- Measurements → the sensors devoted to measure the states of the system are subjects to errors like drift, bias and measures are corrupted by noise
- Engine inertia → the nozzle is a rigid body attached to the launcher. They exchanges forces each other

2.2.3.2. Sensor Dynamics

The sensor dynamics can have a huge impact on control design. In fact the measurements are discrete and available for processing over time intervals (depending on sensor capabilities).
The measurement done for example by an IMU, through gyros and accelerometers, are affected also by the elastic dynamics, the sensors imprecision and added noise.

The main measured variables can be approximatively modeled in this way (see [1]).:

$$\theta_T = \theta_R + \theta_E + \theta_{noise} = \theta_R - \sum_i^{\infty} \phi_i'(\eta_G)\xi_i + \theta_{noise}$$

$$q_T = q_R + q_E + q_{noise} = q_R - \sum_i^{\infty} \phi_i'(\eta_G)\xi_i + q_{noise}$$

And the lateral velocity measurement

$$w_T = w_R + w_E + w_{noise} = w_R - l_{IMU} q_R - \sum_i^{\infty} \phi_i'(\eta_G)\dot{\xi}_i + w_{noise}$$

2.2.3.3. Actuator dynamics

The nozzle and TVC dynamics can be added by considering the elastic characteristics and the actuated deflection command with respect to the commanded one.

$$\Delta\delta_{\mathrm{T}} = \Delta\delta_{\mathrm{TR}} + \Delta\delta_{\mathrm{TE}}$$

Typically, the actuator dynamics can be modeled with transfer function (see [1]), in order to characterize the delay in the actuation execution. Indeed when the actuator is not modelled it is implicitly assumed a transfer function equal to 1 (instantaneous execution). Modeling the actuators can allow to realize other limitations in controller design for example the saturation imposed by the actuation system and so on.

$$\Delta\delta_{\mathrm{TR}}(\mathrm{s}) = \frac{\omega_a^2}{s^2 + 2\varsigma\omega_a s + \omega_a^2}\,\Delta\delta_{\mathrm{Tc}}$$

2.3. Simplified Model

The behavior of a launch vehicle during the mission is commonly divided in two parts, named "long period dynamics" and "short period dynamics".
The “long period dynamics” derives its name from the long period response to the oscillations about the nominal trajectory. During this analysis, the vehicle is modelled as a point mass.
However, for example it is not realistic to assume that the propulsion force can be instantaneously oriented since it is strictly linked with the vehicle attitude and actuator dynamics. The induced Oscillations about the vehicle mass center must be damped out to satisfy the mission requirements. These oscillations have a relative short period response time and their study take the name of “short period dynamics” (see [4]).

The control design has a huge impact on the trajectory errors especially in the atmospheric phase, by managing conflicting requirements like minimizing these errors and keeping the angle of attack small for the structure integrity (see [1]).
Linearization is a common approach for designing ad hoc controller with simplified models on specific flight conditions.

2.3.1. Model Linearization

The control design for complex dynamical systems is often prohibitive without proper assumptions. There could be models with too many state variables or including disturbances and uncertainties which may render unfeasible the design of a control law that is capable of satisfying the performance and stability requirements.
In many cases, as it is common for aerospace applications, it is used to approximate the model to be able to use standard control methods. To do this it is important for the control engineer to have a good understanding of the physics of the system considered and to know that approximations are only valid locally. As an example in the aeronautic field, it is common to consider "trimming" conditions.

The system linearization around specific dynamic conditions is a common method used since it allows to approximate nonlinear models with linear time invariant system and so to design controllers and verify standard stability margins (for example gain and phase margins) with standard methods.

The linearization, or 1st order Taylor series expansion, consists in approximating the system by truncating the Taylor series expansion at the 1st order term.

Consider the nonlinear time invariant system described by:

$$\dot{x} = f(x, u)$$

The equilibrium point for the system is given by the values of x, u (x_0, u_0) for which:

$$\dot{x}_{|(x_0 u_0)} = f(x,u)_{|(x_0 u_0)} = \vec{0}$$

The linearized system around the equilibrium point is given by:

$$\dot{\tilde{x}} = \frac{\partial f}{\partial x}_{|(x_0 u_0)} \tilde{x} + \frac{\partial f}{\partial u}_{|(x_0 u_0)} \tilde{u}$$

$$\tilde{x} = x - x_0, \ \tilde{u} = u - u_0$$

It is possible to linearize the system also around a "non-equilibrium" point. In this case further terms (the evaluation of f around that point) must be added to the linearized equation.

This is common in the gain scheduling approach, where the system is linearized at chosen trajectory instants to update the control law gains values.

3. Control

As mentioned in the previous chapters the control design for a launch vehicle is a challenging task.
The equations of motion are highly nonlinear and the number of states can easily increase. Therefore, the direct approach on the full model is practically unfeasible.
To make this task possible different assumptions and conditions are considered while designing the controller. In particular different approximations are assumed at different flight instants.
This is the reason why the main technique used to control this type of system is what is known in literature as "Gain Scheduling".
The "Gain Scheduling" (See [10], [18]), as the name suggests, consists in freezing the model in different time instants over the trajectory and designing a standard control on a simplified linearized model over that interval. Stability margins criteria are guaranteed by performing standard linear stability analysis techniques. The "Gain Scheduling" is in practice very simple also at implementation level and has been proven to be efficient for launcher control.
Of course, one of the main problem with this approach is due to the linearization applied to the model, which simplifies many of the heavy nonlinear couplings that can be appreciated on the equations previously reported.

Another major challenge in the control design for launch vehicles is the time varying nature of the system. This is due to the high amount of burned fuel, especially in the first mission phases, that changes the dynamic properties of the vehicle (e.g. mass, inertia, thrust value, etc) together with the aerodynamic coefficients and elastic properties.

In the literature, Zhu et al. (See [6], [8]) have done several studies in order to define new methods for the analysis of time varying systems. In particular, they have introduced the concept of "PD eigenvalues" to recall the eigenvalues that characterize the modes in the linear time invariant case. Zhu has also published an article with NASA (See [19]) where an adaptive controller for launcher has been developed based on this new theory.

Another approach used in the industry is based on "Linear Fractional Transformation" (LFT). This method, which allows to perform robust control design, even considering linear time invariant models, defines a structure for the system uncertainties, thus providing bounds on parameters variation. A similar approach to LFT takes the name of "Linear Parameter Varying", where the parameters are allowed to vary inside some functions defining a polytope. These methods involve the solution of complex Linear Matrix inequalities problems and are based on numerical optimization algorithms. Another problem to take into account is the order of the resulting controller.

Nonlinear design techniques showed to be very powerful in different control problem design. The possibility to avoid approximations introduced by classical linearization and the analysis tools that can be applied directly to the original system can allow to give a better insight into the controller design problem. The feedback linearization and zero dynamics (See [2]) relies on the properties of the original system in order to set-up a linearizing controller. Again the main drawback is related to the accuracy of the plant model, the availability of full state information and the class of nonlinear systems where these methods can be applied.

3.1. Gain Scheduling

The design of the "gain-scheduled" controller can be summarized in 4 iterations:

- **Linearizing the model**

  The original model plant (nonlinear most of the times) is simplified and linearized around operating conditions. Some parts of the dynamics are neglected in order to reduce the order and the complexity of the model.

- **Design the controller**

  For the LTI (Linear Time Invariant) characterization of the plant in the different time instants.

- **Scheduling**

  All the obtained controller are scheduled so that the gains are coherently adapted based on the evolution of the system along the studied path.

- **Perform stability analysis**

  Local stability and margins are checked on the closed loop systems and multiple simulations are run.

Typical "gain-scheduled" controller methods are based on:

- Eigenvalues assignment
- LQR
- PID control

The reason why "Gain-scheduling" is so used is the ability to perform standard and well-known control techniques on several generated LTI models. Local stability can be also checked with standard methods and all the generated controller gains can be easily implemented based on mission timeline or flight dynamics conditions.

The main drawback is that these controllers are designed on simplified model descriptions that neglects many nonlinearities and the time varying nature of the launcher in this case.

Therefore, many simulations are required in order to guarantee that the controller is capable of performing its task on several environmental and off-nominal conditions. So, the performance is not guaranteed.

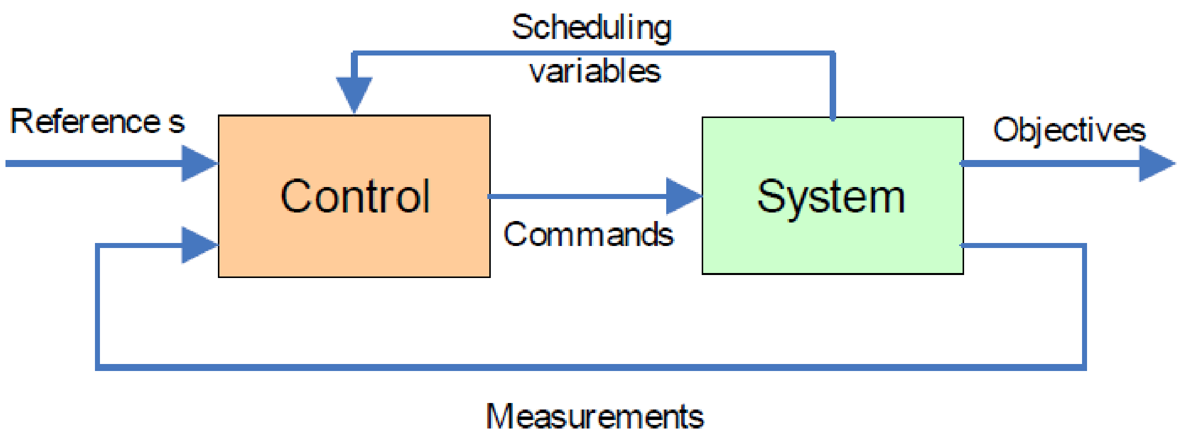

FIG. 12 – Typical Gain Scheduling controller architecture

3.1.1. LQR

The linear quadratic regulation problem can be so formulated: given a linear controllable and exponentially stable system described by

$$\begin{cases} \dot{x} = A(t)x + B(t)u \\ y = Cx \end{cases} \tag{4.26}$$

with A and B bounded and with elements of C^1 class, and given an initial time instant t_i, it is required to find the optimal regulator that optimizes the control action $\boldsymbol{u} \in C^0[t_i, \infty)$ and the system state $\boldsymbol{x} \in C^1[t_i, \infty)$ by minimizing the cost function

$$J = \frac{1}{2}\int_{t_i}^{\infty} [x^T Q x + u^T R u] dt \tag{4.27}$$

Q is a symmetric positive semidefinite matrix, and R a symmetric positive definite matrix, both with elements bounded and of C^1 class.

These matrices define respectively the weight on the state vector and on the control input vector. This is also referred to as the infinite horizon LQR problem.

The Riccati equation is introduced to solve the problem:

$$\dot{K}(t) = K(t)B(t)R^{-1}(t)B^T(t)K(t) - K(t)A(t) - A^T(t)K(t) - K(t)A(t) - Q(t) \quad (4.28)$$
$$K(T) = F$$

The Riccati equation has a unique solution.

The control input that minimizes the cost index is in fact given by:

$$u(t) = -R^{-1}(t)B^T(t)K(t)\boldsymbol{x}(t)$$

K is the solution of the Riccati equation associated to the original system:

$$\dot{K}(t) = K(t)B(t)R^{-1}(t)B^T(t)K(t) - K(t)A(t) - A^T(t)K(t) - K(t)A(t) - Q(t)$$
$$\lim_{T\to\infty} K(T) = 0$$

The cost index assumes the following value:

$$J(x,u) = \frac{1}{2}x_i^T K(t_i)x_i$$

Where x_i is the initial condition on the state.

In case A, B, Q and R are constants, and Q is also positive definite (so considering the infinite horizon linear time invariant problem), the algebraic Riccati equation solution (K) is given by

$$0 = KBR^{-1}B^TK - KA - A^TK - KA - Q$$

For more details refer to [34], [35].

3.2. Feedback Linearization

Given a nonlinear system of the form

$$\begin{cases} \dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} \\ \boldsymbol{y} = h(\boldsymbol{x}) \end{cases}$$

with **x** being the n-dimensional state vector, **u** the m-dimensional vector of the control variables and **y** being the output vector of the same dimension of the input variable (square MIMO system), the objective is to find a nonlinear controller that can completely linearize the original systems, in order to successively design a linear controller based on the well known linear control theory for LTI systems.
The major reference for the feedback linearization theory is provided in [2].

3.2.1. SISO Feedback Linearization

First step for the analysis is to introduce the concept of relative degree. The type of system considered is "single input single output" (SISO).
Refer to the previously introduced affine in the control SISO nonlinear system , the relative degree of the system in $x_0$is equal to "r" if:

$$L_g L_f^k h(x) = 0 \;\; \forall x \; \in neighborhood\ of\ x_0,\ k < r - 1$$
$$L_g L_f^{r-1} h(x_0) \neq 0$$

Physically the concept of relative degree can be seen as the exact number the output function $y = h(x)$ time derivative needed to make the input u appear explicitly in the equation. The output time derivative can be expressed as:

$$y = h(x)$$
$$\dot{y} = \frac{dh}{dt} = \frac{\partial h}{\partial x}\frac{\partial x}{\partial t} = \frac{\partial h}{\partial x}(f(x) + g(x)u) = L_f h(x) + L_g h(x) u$$
$$\ddot{y} = {L_f}^2 h(x) + L_g L_f h(x) u$$
$$\vdots$$
$$y^r = {L_f}^r h(x) + L_g {L_f}^{r-1} h(x) u$$

In case of relative degree r the result of this process yields:

$$
\begin{aligned}
y &= h(x) \\
\dot{y} &= L_f h(x) \\
\ddot{y} &= L_f{}^2 h(x) \\
&\vdots \\
y^{r-1} &= L_f{}^{r-1} h(x) \\
y^r &= L_f{}^r h(x) + L_g L_f{}^{r-1} h(x) u
\end{aligned}
$$

If no relative degree can be defined around a point, the system will depend only on its initial condition and not on the input.

If the relative degree equals the state dimension "n", a coordinate transformation can be defined locally around x_0:

$$
\begin{aligned}
\phi_1(x) &= h(x) \\
\phi_2(x) &= L_f h(x) \\
\phi_3(x) &= L_f{}^2 h(x) \\
&\vdots \\
\phi_r(x) &= L_f{}^{r-1} h(x)
\end{aligned}
$$

In case $r < n$, the coordinate transformation would not be complete, but it is always possible to find n+r more functions such that

$$
\mathbf{\Phi} = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_n(x) \end{bmatrix}
$$

has a nonsingular Jacobian in the evaluated point x_0, so that Φ is an admissible local coordinate transformation (Diffeomorphism). It can be shown that this choice can be done in order to get

$$
L_g \phi_i(x) = 0 \; \forall \, r + 1 \leq i \leq n, \forall x \in neighborhood\ of\ x_0
$$

The new system is described by:

$$\dot{\phi}_1 = \phi_2(x)$$
$$\dot{\phi}_2(x) = \phi_3(x)$$
$$\dot{\phi}_3(x) = \phi_4(x)$$
$$\vdots$$
$$\dot{\phi}_r(x) = L_f{}^r h(x) + L_g L_f{}^{r-1} h(x) u$$
$$\dot{\phi}_{r+1}(x) = L_g \phi_{r+1}(x)$$
$$\vdots$$
$$\dot{\phi}_n(x) = L_g \phi_n(x)$$

and the nonlinear controller is set to:

$$u = \frac{v - L_f{}^r h(x)}{L_g L_f{}^{r-1} h(x)}$$

So, if r = n, the system becomes

$$\dot{\phi}_1 = \phi_2(x)$$
$$\dot{\phi}_2(x) = \phi_3(x)$$
$$\dot{\phi}_3(x) = \phi_4(x)$$
$$\vdots$$
$$\dot{\phi}_r(x) = v$$

where v is the new auxiliary input. The resulting system is controllable and linear, and v can be used to assign the desired system behavior.

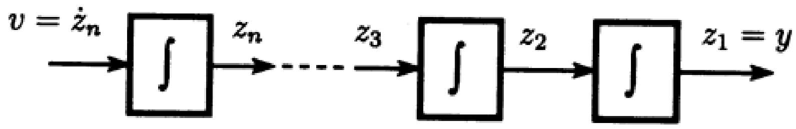

FIG. 13 – Chain of integrators (see [2])

In case $r < n$, the system is described by the following set of equations

$$\dot{\phi}_1 = \phi_2(x)$$
$$\dot{\phi}_2(x) = \phi_3(x)$$
$$\dot{\phi}_3(x) = \phi_4(x)$$
$$\vdots$$
$$\dot{\phi}_r(x) = v$$
$$\dot{\phi}_{r+1}(x) = L_g \phi_{r+1}(x)$$
$$\vdots$$
$$\dot{\phi}_n(x) = L_g \phi_n(x)$$

This system can be split in a linear subsystem with the same characteristic of the previous one and a subsystem totally independent by the input and not affecting the output value.

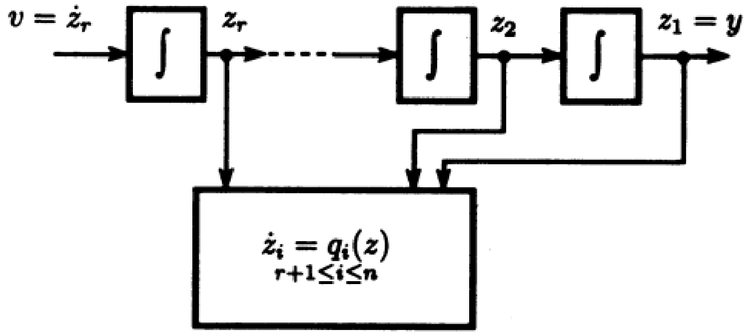

FIG. 14 – System representation in case r < n (see [2])

In this case, after the coordinate transformation, it is highlighted the system zero dynamics. The importance of studying this dynamic is related to the fact that is it cancelled, so made unobservable, by the nonlinear controller action. So, its behavior has to be stable to avoid problems.

The zero dynamics describe the internal behavior of the system when both input and initial conditions are chosen such that the output is zero.

Considering the previous system in normal form, with

$$\boldsymbol{z} = \begin{bmatrix} \phi_1(x) \\ \vdots \\ \phi_r(x) \end{bmatrix}$$

$$\begin{aligned} \dot{z}_1 &= z_2 \\ &\vdots \\ \dot{z}_{r-1} &= z_r \\ \dot{z}_r &= b(\boldsymbol{z},\eta) + a(\boldsymbol{z},\eta)u \\ \dot{\eta} &= q(\boldsymbol{z},\eta) \end{aligned}$$

The zero dynamics of the system are described by

$$\dot{\eta} = q(\mathbf{0},\eta)$$

3.2.2. Square MIMO feedback linearization

In the case of MIMO square systems (so multi input multi output systems with the same number of inputs and outputs) the same theory of the SISO case can be adapted in order to define a nonlinear linearizing controller. Firstly, when dealing with MIMO systems, must be introduced the concept of vector relative degree.

A system has a vector relative degree $[r_1 \quad \cdots \quad r_m]$ in x_0 if:

$$L_{g_j} L_f^k h_i(x) = 0$$
$$\forall \in neighborhood\ of\ x_0, \forall\ j\ such\ that\ 1 \leq j \leq m, k < r - 1, \forall\ i\ such\ that\ 1 \leq j \leq m$$

and the following matrix:

$$\Delta(x) = \begin{bmatrix} L_{g_1} L_f^{r_1-1} h_1(x) & \dots & L_{g_m} L_f^{r_1-1} h_1(x) \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m-1} h_1(x) & \dots & L_{g_m} L_f^{r_m-1} h_m(x) \end{bmatrix}$$

has its determinant computed in x_0 different from zero. The matrix $\Delta(x)$ is referred to as decoupling matrix.

Under the previous conditions, by coordinate transformation, and applying the control input

$$u = \Delta^{-1}(x)(v - a(x))$$

with $a(x)$ being

$$a(x) = \begin{bmatrix} L_f^{r_1} h_1(x) \\ \vdots \\ L_f^{r_m} h_m(x) \end{bmatrix}$$

the system is linearized by nonlinear static state feedback.

The condition of having the decoupling matrix nonsingular around x_0 is due to the fact that its inverse appears in the control law in order to obtain a linearized MIMO system.

If $r_1 + r_2 + \ \dots + \ r_m = n$ no zero dynamics are present and the system is written as

$$\dot{z}_1^1 = z_2^1$$
$$\vdots$$
$$\dot{z}_{r-1}^1 = z_r^1$$
$$\dot{z}_r^1 = v_1$$
$$\dot{z}_1^2 = z_2^2$$
$$\vdots$$
$$\dot{z}_r^2 = v_2$$
$$\vdots$$
$$\dot{z}_r^m = v_m$$

This system is described by m decoupled, linear and controllable subsystems.

If zero dynamics are present, their behavior have to be analyzed for assuring the asymptotic stability of the system.

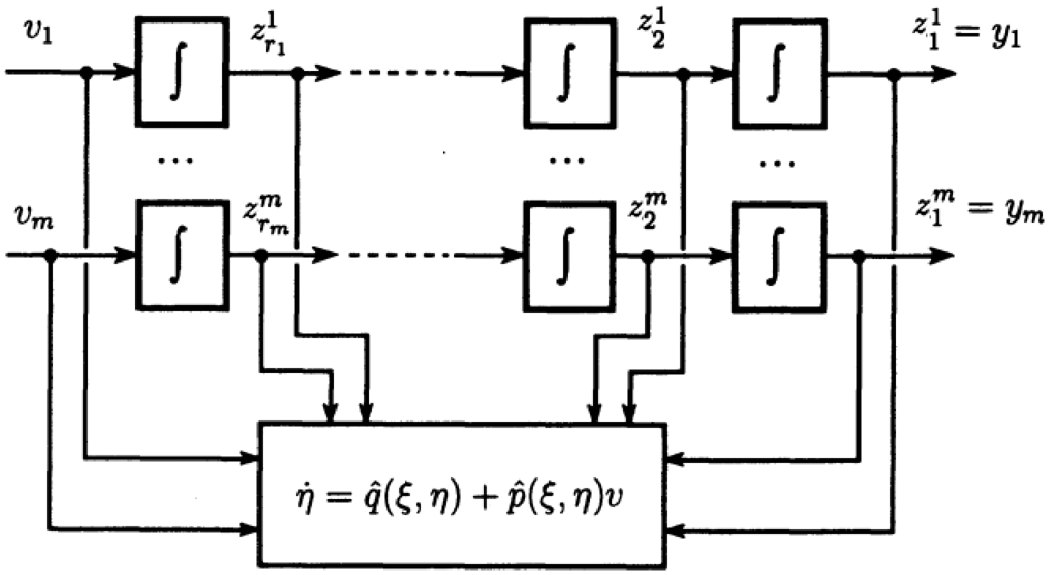

FIG. 15 – Subsystem resulting from MIMO feedback linearization with zero dynamics(see [2])

3.3. Observation problem

One of the drawbacks of the feedback linearization is the request of access to the full state information in order to design the nonlinear feedback that linearizes the original systems exactly.

Rarely in physical systems are available sensors that are capable of measuring all the state variables thus allowing to implement a full state feedback control.

The problem of reconstructing the state of a system from the plant model knowledge and the available measurements is known as "Observability problem", that can be seen as the dual of the controllability problem.

Further in applications the sensors that are devoted to provide the state information to the control algorithms are subject to noises and errors (bias etc.).

This is the reason why the Kalman filter, since its definition, is still nowadays the most used algorithm to reconstruct the state dynamics and clean measurements from added noise.

3.3.1. State Estimators

3.3.1.1. Bayesian Estimators

The Kalman filter is a recursive implementation of the Bayesian MMSE (minimum mean square error) estimator in case of random variables described by Gaussian distributions. In the following, in order to start and understand the properties of Bayesian filters,
some details are presented (see [5]).

Firstly, Bayesian filters deal with random variables (θ).

The problem is of the following type: consider a generic non negative function $\Phi: \mathbb{R}^p \to R$, and define

$$J(\alpha) \coloneqq E[\Phi(\theta - \alpha)|y^k]$$

which is the conditional expectation given the measurements.

This expression can be explicited as:

$$J(\alpha) = E[\Phi(\theta - \alpha)|y^k] = \int_{\Omega_{\theta|y^k}} \Phi(\theta - \alpha)\, p_{\theta|y^k}(y^k, \theta) d\theta$$

Since $p_{\theta|y^k}(y^k, \theta)$ (conditioned probability density) is not an a-priori information it is not always possible to find $J(\alpha)$.

The Bayesian estimator is given by:

$$\hat{\theta}_{|k} = argmin_{\alpha \in \Omega_{\theta|y^k}} J(\alpha)$$

From bayes rule $p_{\theta|y^k}(y^k, \theta)$ can be written as:

$$p_{\theta|y^k}(y^k, \theta) = \frac{p_{\binom{\theta}{y^k}}(\xi^k, \theta)}{p_{y^k}(\xi^k)}$$

Where $p_{\binom{\theta}{y^k}}(\xi^k, \theta)$ is the joint density, and $p_{y^k}(\xi^k)$ is the marginal density.

$$p_{y^k}(\xi^k) = \int_{\Omega_\theta} p_{\binom{\theta}{y^k}}(\xi^k, \theta)\, d\theta$$

Thus the joint density is required. It can be derived making use of a-priori information:

$$p_{\binom{\theta}{y^k}}(\xi^k, \theta) = p_{y^k|\theta}(\xi^k, \theta)\, p_\theta(\theta)$$

The a-priori information typically available are:

$$p_{v(i)}(\eta(i))$$
$$p_\theta(\theta)$$

$p_{y^k|\theta}(\xi^k, \theta)$ can be obtained from a-priori information on measurement noise sequence:

$$p_{\nu^k}(\eta^k) \rightarrow p_{y^k|\theta}(\xi^k, \theta)$$

Consider this type of measurement equation:

$$y^k = g^k(\theta) + \nu^k$$

Knowing the noise probability distribution it results:

$$p_{y^k|\theta}(\xi^k, \theta) = \ p_{\nu^k}(\xi^k - g^k(\theta))$$

In particular for a "Minimum Mean Square Estimator"(MMSE) the cost function is:

$$J(\alpha) = E\left[||\theta - \alpha||^2 | y^k\right]$$

Another important characteristic of this type of estimators is that it's possible to get a closed form solution for the optimum.

Suppose that $\Omega_{\theta|y^k}$ is a convex set (this assumption guarantees the fact that the solution of the optimization problem is admissible), then:

$$\hat{\theta}_{|k} = argmin_{\alpha \in \Omega_{\theta|y^k}} J(\alpha) = argmin_{\alpha \in \Omega_{\theta|y^k}} E\left[||\theta - \alpha||^2 | y^k\right] = E[\theta | y^k]$$

In fact:

$$J(\alpha) = E\left[||\theta - \alpha||^2 | y^k\right] = \int_{\Omega_{\theta|y^k}} ||\theta - \alpha||^2 p_{\theta|y^k}(y^k, \theta)$$

From the first order necessary condition for finding the extremals:

$$\left.\frac{dJ}{d\alpha}\right|_{\alpha=\alpha^*} = 0$$

$$\frac{dJ}{d\alpha} = \frac{d}{d\alpha}\int_{\Omega_{\theta|y^k}} ||\theta-\alpha||^2 \, p_{\theta|y^k}(y^k,\theta)d\theta = \frac{d}{d\alpha}\int_{\Omega_{\theta|y^k}} (\theta-\alpha)^T(\theta-\alpha) \; p_{\theta|y^k}(y^k,\theta)d\theta$$

$$= -\int_{\Omega_{\theta|y^k}} 2(\theta-\alpha^*)\, p_{\theta|y^k}(y^k,\theta)d\theta = 0$$

$$0 = \int_{\Omega_{\theta|y^k}} 2\theta\, p_{\theta|y^k}(y^k,\theta)d\theta - \int_{\Omega_{\theta|y^k}} 2\alpha^*\, p_{\theta|y^k}(y^k,\theta)d\theta =$$

$$= \int_{\Omega_{\theta|y^k}} \theta\, p_{\theta|y^k}(y^k,\theta)d\theta - \alpha^* \int_{\Omega_{\theta|y^k}} p_{\theta|y^k}(y^k,\theta)d\theta.$$

Since $\int_{\Omega_{\theta|y^k}} p_{\theta|y^k}(y^k,\theta)d\theta = 1$ it follows that there is just one extremal that is:

$$\alpha^* = \int_{\Omega_{\theta|y^k}} \theta\, p_{\theta|y^k}(y^k,\theta)d\theta = E[\theta|y^k]$$

From second order necessary condition:

$$\frac{\partial^2 J}{\partial\alpha^2} = \frac{\partial}{\partial\alpha}\left(\frac{\partial\alpha}{\partial\alpha}\right)^T = 2I\int_{\Omega_{\theta|y^k}} p_{\theta|y^k}(y^k,\theta)d\theta = 2I_{pxp}$$

Since the Hessian matrix is positive definite then α^* is a global minimum.

The MMSE of θ, $\hat{\theta}_{|k} = E[\theta|y^k]$ is always centered:

$$E\left[\hat{\theta}_{|k}\right] = E[\theta|y^k] = E[\theta]$$

The previous problem can be generalized for random time varying parameters, in particular:

$$\hat{\theta}_{|k} \rightarrow \hat{\theta}(i|k)$$

Where depending on "i" different problems are defined:

$i < k$ leads to the problem of interpolation

$i = k$ leads to the problem of filtering

$i > k$ leads to the problem of prediction

Another property of MMSE is the affinity in the measurements.

Suppose again that the support $\Omega_{\theta|y^k}$ is a convex set and that the joint probability $p_{\binom{y^k}{\theta}}(\xi^k, \theta)$ is jointly Gaussian then:

$$E[\theta|y^k]\ is\ affine\ in\ the\ measurement\ (My^k + H)$$

If this condition is holding, as a result:

$\hat{\theta}(i|k)$ is linear in the measurements (this problem is solved by Kalman filter)

$\hat{\theta}_{|k}$ is linear in the measurements

In particular it can be shown that under these conditions the conditioned probability $p_{\theta|y^k}(\xi^k, \theta)$, for the static case (but this can be easily generalized to time varying case), is Gaussian with this mean and covariance matrix:

$$E[\theta|y^k] = \bar{\theta} + \Psi_{\theta y^k}\Psi_{y^k}^{-1}(y^k - \bar{y}^k)$$

$$\Psi_{\theta|y^k} = \Psi_{\theta} - \Psi_{\theta y^k}\Psi_{y^k}^{-1}\Psi_{\theta y^k}^{T}$$

Where:

$$\Psi_{\theta y^k} = E[(\theta - \bar{\theta})(y^k - \bar{y}^k)^T] = \Psi_{y^k\theta}^{T}$$

$$\bar{\theta} = E[\theta]$$

$$\bar{y}^k = E[y^k]$$

For the proof and more details on Bayesian filters see [5].

The MMSE is given by:

$$\hat{\theta}_{|k} = \bar{\theta} + \Psi_{\theta y^k}\Psi_{y^k}^{-1}(y^k - \bar{y}^k)$$
$$\Psi_{\hat{\theta}|k} = \Psi_{\theta} - \Psi_{\theta y^k}\Psi_{y^k}^{-1}\Psi_{\theta y^k}^{T}$$

It can be noticed from the previous expression that as the number of measurements increase (the vector y^k) the computational effort increases as well and can easily become expensive. This leads to the necessity of a recursive implementation.

Furthermore it is required to verify that the joint probability $p_{\binom{\theta}{y^k}}(\xi^k, \theta)$ is Gaussian in order to guarantee affinity. For this two properties are needed:

- Linearity in the measurement equation: $y^k = C^k\theta + v^k$
- θ, v^k must be Gaussian and uncorrelated

Not only each component of the noise sequence has to be uncorrelated with θ, but the sequence must be internally uncorrelated.

Considering the time varying case:

Linearity of measurement and model equations:

$$y(i) = C(i)\theta(i) + v(i)$$
$$\theta(i+1) = A(i)\theta(i) + m(i)$$

If $\theta(i+1)$ is linear then $\theta(i)$ is linear with respect to $\theta(0)$ and the noise sequence $\{m(i)\}$, then also $y(i)$ is linear with respect to $\theta(0)$ and $\{m(i)\}$

For the gaussianity of the joint probability then are needed the assumptions of gaussianity and uncorrelation on $\theta(0)$, $\{m(i)\}$ and $\{v(i)\}$

Another fundamental property for a Bayesian estimator, which guarantees a necessary and sufficient condition, to be "minimum mean square error" is the so called "Orthogonality Principle" which expresses some orthogonality condition between the estimation error and the measurement equation.

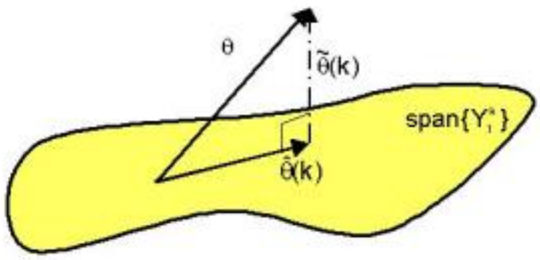

FIG. 16 – MMSE orthogonal to $\hat{\theta}_{|k}$ (see [25])

Assume that Ψ_{y^k} is invertible, and consider the family of centered and affine in the measurement estimators $\{\hat{\theta}_{|k}\}$ (for static case) and $\{\hat{\theta}(i|k)\}$ (for dynamic case).
Necessary and sufficient condition to be MMSE is (see [5]):

$$E\left[\hat{e}_{|k} {y^k}^T\right] = 0, \text{ for static case}$$
$$E\left[\hat{e}(i|k) {y^k}^T\right] = 0, \text{ for dynamic case}$$

3.3.1.2. Kalman Filter

The idea behind the Kalman filter is a recursive implementation of Bayesian MMSE estimator.
The Kalman Filter theory was developed by R.Kalman and R.Bucy around 1960. It's used in order to compute state estimation considering a physical system affected by measurement and process noises.

Consider a dynamical system where the random variable is the state to be estimated, with $\{u(i)\} and \ \{v(i)\}$ Gaussian sequences:

$$x(i+1) = A(i)x(i) + B(i)u(i), \qquad i = 0, \dots, k-1$$
$$y(i) = C(i)x(i) + v(i), \qquad i = 1, \dots, k$$

Thus: $x(i) = \theta(i)$.

Consider $x(i)$ to be a linear function of $x(0)$ and $\{u(i)\}$, then:

1. $y(i)$ is a linear function of $v(i), x(0), u(i)$
2. y^k is a linear function of $\{v(i)\}, x(0), \{u(i)\}$
3. If $v(i), u(i), x(0)$ are Gaussian and uncorrelated (mutually and internally) then the joint probability $p_{\binom{x(i)}{y(i)}}$ is Gaussian.

The affinity in measurements property (given by the above statements) is fundamental for getting a recursive expression for the filter

Consider the following general equation for the filter:

$$\hat{x}(i+1|i+1) = G(i+1) \ \hat{x}(i|i) + F(i+1)y(i+1)$$

$G(i+1)$ is the recursive term (see dependence from previous filtering action) and $F(i+1)$ is the correction term for the current measurement.

The matrices G, F must be chosen in order for the filter to be centered and optimal in Bayesian sense. Furthermore, even if the system considered is stationary the Kalman filter is not.

Regarding the centering property of the estimator, suppose that $\hat{x}(i|i)$ is centered and that the process and measurement noise have zero mean, it is required to ensure that (*see* [5]):

$$E[\hat{x}(i+1|i+1)] = E[[G(i+1)\ \hat{x}(i|i) + F(i+1)y(i+1)]$$
$$= G(i+1)E[\hat{x}(i|i)] + F(i+1)E[y(i+1)]$$
$$= G(i+1)E[x(i)] + F(i+1)E[C(i+1)x(i+1) + v(i+1)]$$
$$= G(i+1)E[x(i)] + F(i+1)E\big[C(i+1)\big(A(i)x(i) + B(i)u(i)\big) + v(i)\big]$$
$$= G(i+1)E[(x(i)] + F(i+1)C(i+1)A(i)E[x(i)]$$

Centering implies $E[\hat{x}(i+1|i+1)] = E[x(i+1)]$, then:

$$E[x(i+1)] = G(i+1)E[(x(i)] + F(i+1)C(i+1)A(i)E[x(i)]$$
$$E[x(i+1)] = [G(i+1) + F(i+1)C(i+1)A(i)]E[x(i)]$$

Since :

$$E[x(i+1)] = E[A(i)x(i) + B(i)u(i)]$$

and $\{u(i)\}$ has zero mean value:

$$E[x(i+1)] = E[\big(G(i+1) + F(i+1)C(i+1)A(i) + A(i) - A(i)\big)x(i) + B(i) + u(i)]$$

Then it's clear that for this to be satisfied it must be verified:

$$G(i+1) + F(i+1)C(i+1)A(i) - A(i) = 0$$

From which:

$$G(i+1) = A(i) - F(i+1)C(i+1)A(i)$$

It has been supposed that $E[\hat{x}(i|i)]$is centered, so for proper initialization:

$$E[\hat{x}(0|0)] = E[x(0)]$$

This highlights the importance of a good filter initialization values, but this requires the knowledge of the mean at the initial step which could not be provided.

In practice the main implemented version for linear systems is the stationary filter (see [11]), which under some hypothesis (Controllability, observability or stabilizability) converges as the number of iterations increase despite of the initializations (is asymptotically optimal).

Another key property to be verified is the optimality (see [5]). This will be guaranteed by appropriate choice of F matrix, starting from the already mentioned principle of orthogonality between the estimation error and the measurements.

It must be verified that:

$$E\left[\hat{e}(i+1|i+1)z^{(i+1)^T}\right] = 0$$

With proper manipulations:

$$E[\hat{e}(i+1|i+1)z(\tau)^T] = 0, \tau = 1, \dots, i$$
$$E[\hat{e}(i+1|i+1)z(i+1)^T] = 0$$

For (1):

$$\begin{aligned}\hat{e}(i+1|i+1) &= x(i+1) - \hat{x}(i+1|i+1)\\ &= A(i)x(i) + B(i)u(i) - G(i+1)\ \hat{x}(i|i) - F(i+1)y(i+1)\\ &= A(i)x(i) + B(i)u(i) - (A(i) - F(i+1)C(i+1)A(i))\hat{x}(i|i) - F(i+1)y(i+1)\end{aligned}$$

The proof is quite involved in terms of manipulations (see [5]), but it can be demonstrated that the final equation of F is given by:

$$F(i+1) = \Psi_{\hat{e}(i+1|i+1)}C^T(i+1)\Psi_{\nu(i+1)}^{-1}$$

So the correction term depends on covariance of estimation error and the noise covariance. F is like the gain on the input (the actual measurement).
Since there is the dependence on the covariance of estimation error it's straightforward that the filter can't be stationary.

For the covariance of the estimation error:

$$\Psi_{\hat{e}(i+1|i+1)} = \Gamma\left(\Psi_{\hat{e}(i|i)}\right)\left(I + C^T(i+1)\Psi_{v(i)}^{-1}C(i+1)\Gamma\left(\Psi_{\hat{e}(i|i)}\right)\right)^{-1}$$

With:

$$\Gamma\left(\Psi_{\hat{e}(i|i)}\right) = A(i)\Psi_{\hat{e}(i|i)}A^T(i) + B(i)\Psi_{u(i)}B(i)^T$$

Even in this case it is fundamental the initialization (for the filter optimality):

$$\Psi_{\hat{e}(0|0)} = E\left[\left(x(0) - \hat{x}(0|0)\right)\left(x(0) - \hat{x}(0|0)\right)^T\right] = E[(x(0) - E[(x(0)])(x(0) - E[(x(0)])^T] = \Psi_{x(0)}$$

A failure in the initialization leads to possible errors in the state estimation.

Now the final form of the Kalman filter can be obtained (see [5]):

$$\begin{aligned}\hat{x}(i+1|i+1) &= G(i+1)\ \hat{x}(i|i) + F(i+1)y(i+1)\\ &= \left(A(i) - F(i+1)C(i+1)A(i)\right)\ \hat{x}(i|i)\\ &+\ \Psi_{\hat{e}(i+1|i+1)}C^T(i+1)\Psi_{v(i+1)}^{-1}y(i+1)\\ &= A(i)\ \hat{x}(i|i) + \Psi_{\hat{e}(i+1|i+1)}C^T(i+1)\Psi_{v(i+1)}^{-1}[y(i+1) +\\ &\qquad -C(i+1)A(i)\ \hat{x}(i|i)]\end{aligned}$$

$$\Psi_{\hat{e}(i+1|i+1)} = \Gamma\left(\Psi_{\hat{e}(i|i)}\right)\left(I + C^T(i+1)\Psi_{v(i)}^{-1}C(i+1)\Gamma\left(\Psi_{\hat{e}(i|i)}\right)\right)^{-1}$$

$$\hat{x}(0|0) = \bar{x}(0)$$

$$\Psi_{\hat{e}(0|0)} = \Psi_{x(0)}$$

Looking at last term in the state estimation equation , $[y(i+1) - C(i+1)A(i)\ \hat{x}(i|i)]$, there is the "Innovation Process" that represents the difference between the last available measure and the result that it would have in absence of noise at step (i+1) considering the state one step prediction.

It evaluates the new information content with respect to the previous (if the innovation is zero then the prediction and measurement agree).

The filter works online because it requires the measurement at current step $(i+1)$ as can be noticed in the innovation process term.

Instead the covariance of the estimation error can in principle be computed offline, because uses just the a-priori information on $\{\Psi_{u(i)}\}, \{\Psi_{v(i)}\}, \Psi_{x(0)}$.

If the gaussianity is not assumed for $x(0), \{u(i)\}, \{v(i)\}$, the Kalman filter gives the optimal MMSE among all possible affine and centered estimates (see [5][).

In case of deterministic and stochastic inputs, the previous expression can be split into deterministic and stochastic components (see [5]):

$$x(i) = x_s(i) + x_d(i)$$
$$\hat{x}(i+1|i+1) = \hat{x}_s(i+1|i+1) + x_d(i+1)$$

The deterministic part evolves on model prediction, the stochastic is estimated via Kalman filter algorithm.

All the probability distribution functions assumed in the Kalman filter are Gaussian so it is possible to consider the corresponding locus of equal probability around the predicted and estimated values of the state, that constitute the mean of the conditional probability distribution function that is propagated by the filter (see [25]).
From the next figures a graphical interpretation is given.
The ellipses represent the contour of equal probability around the mean, the dashes lines correspond to the actual filter dynamics that involves the mean values, and the solid lines are the exact values of the random variables estimated (see [25]).

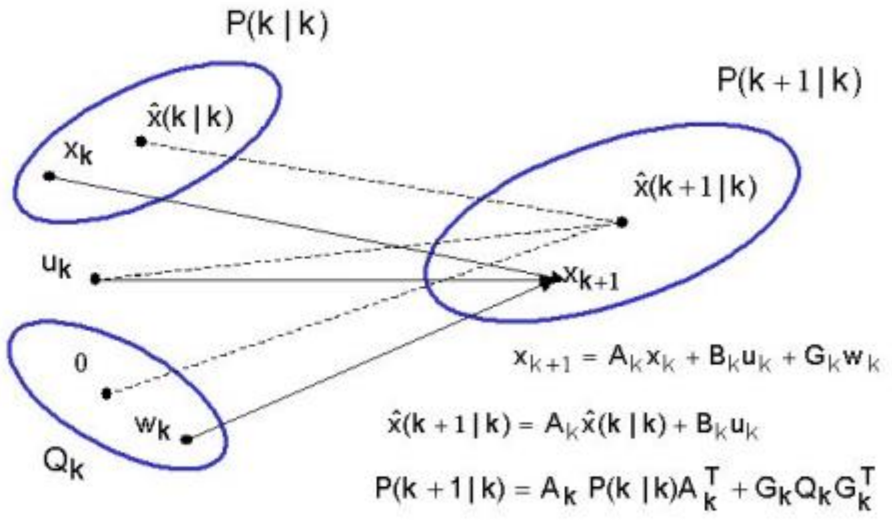

FIG. 17 – Error ellipsoid propagation in the Kalman filter prediction cycle (see [25])

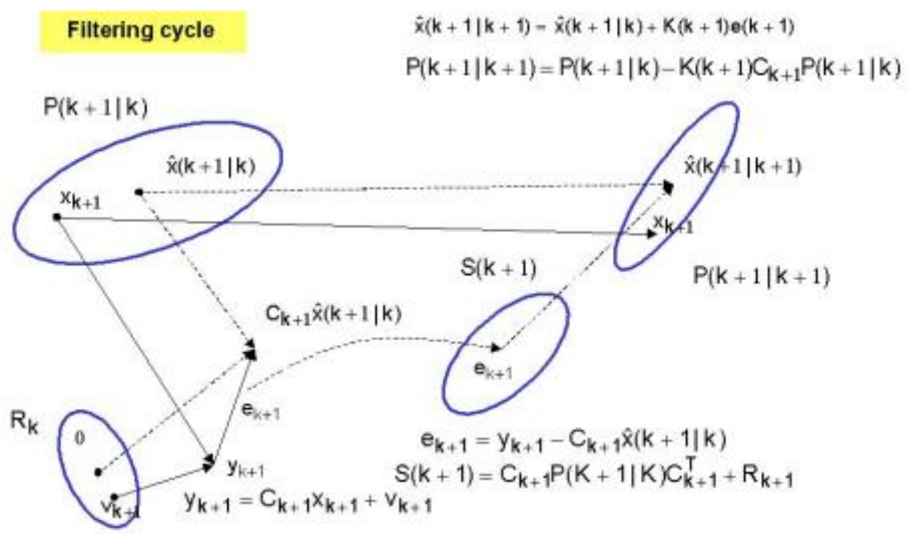

FIG. 18 – Error ellipsoid propagation in the Kalman filter filtering cycle **(see [25])**

There is also a steady state implementation of the Kalman filter that can ensure the asymptotic optimality despite of the initialization.

The KF can also handle the case where process and measurement noises are correlated (see [5]).

In the next paragraph will be instead presented the most used approximation of the original Kalman Filter, the Extended Kalman Filter and after a more recent implementation named “Unscented Kalman filter”.

3.3.1.3. Extended Kalman Filter

The extended Kalman filter needs arise from the necessity of using a Kalman filter for nonlinear state estimation, since most of physical systems are governed by nonlinear dynamics. Being an approximation of its original version implies that it is no more optimal because it does not minimize the minimum mean square error.

However because the filter performs very well in many practical cases it has become the standard for KF algorithm.

Given a discrete nonlinear dynamical system with additive process and measurement noise:

$$x(i+1) = \phi\big(x(i), u(i)\big) + w(i)$$
$$y(i) = h\big(x(i)\big) + v(i)$$

The algorithm steps are:

1. State Prediction:

   $$\hat{x}(i+1|i) = \phi(\hat{x}(i|i), u(i))$$

2. Covariance Prediction:

   $$\Psi_{\hat{e}(i+1|i)} = \tilde{A}(i)\Psi_{\hat{e}(i|i)}\ \tilde{A}(i)^T + \tilde{B}(i)\Psi_{w(i)}\ \tilde{B}(i)^T$$

3. Covariance update:

   $$\Psi_{\hat{e}(i+1|i+1)} = \Psi_{\hat{e}(i+1|i)}\big(I + C^T(i+1)\Psi_{v(i)}^{-1}C(i+1)\Psi_{\hat{e}(i+1|i)}\big)^{-1}$$

4. State update:

   $$\hat{x}(i+1|i+1) = \hat{x}(i+1|i) + \Psi_{\hat{e}(i+1|i+1)}C^T(i+1)\Psi_{v(i+1)}^{-1}[y(i+1) + -h(\hat{x}(i+1|i), i+1)]$$

Even in this case is fundamental a good filter initialization for the state and covariance of estimation error.

As can be noticed in step (2) the covariance one step prediction is based on state transition matrix and input matrix.

Those matrices are obtained at each instant linearizing around the previous prediction step. In particular the linearization step is performed in this way:
At first step:

$$\tilde{A}(0) = \left.\frac{\partial\phi\big(x(0), u(0)\big)}{\partial x(0)}\right|_{\hat{x}(0|0)}$$

At generic "i" step:

$$\tilde{A}(i) = \left.\frac{\partial\phi\big(x(i), u(i)\big)}{\partial x(i)}\right|_{\hat{x}(i|i-1)}$$

It's evident that at each filtering step, after the one step prediction there is the linearization process which will be used for next prediction step.

Considering the case of a discrete nonlinear dynamical system in presence of input and measurement noise:

$$x(i+1) = \phi(x(i), \tilde{u}(i)), \qquad \tilde{u}(i) = u(i) + w(i)$$
$$y(i) = h\big(x(i)\big) + v(i)$$

In this case will change the prediction step of the covariance of estimation error and the system dynamics will be linearized with respect to the stochastic component in order to derive the input matrix.

At generic "i-th" step:

$$\tilde{A}(i) = \left.\frac{\partial\phi\big(x(i),u(i)\big)}{\partial x(i)}\right|_{\hat{x}(i|i-1)}$$

$$\tilde{B}(i) = \left.\frac{\partial\phi\big(x(i),u(i)\big)}{\partial u(i)}\right|_{\hat{x}(i|i-1),u(i-1)}$$

The prediction step relies on system linearization and this could lead to estimation error in case of highly nonlinear dynamics because of the first order approximation introduced. For more details see [5].

An intuitive example that can give a visualization of the error that possibly is introduced by these assumptions is shown in the next two figure (see [26])

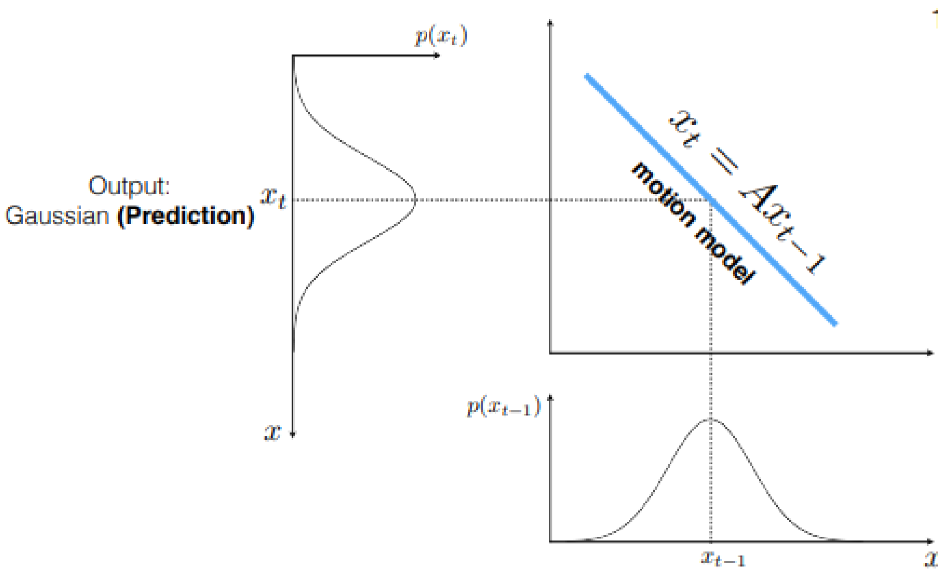

FIG. 19 – One step gaussian prediction on the linear system case (Image from https://www.cs.cmu.edu/~16385/s17/Slides/16.4_Extended_Kalman_Filter.pdf)

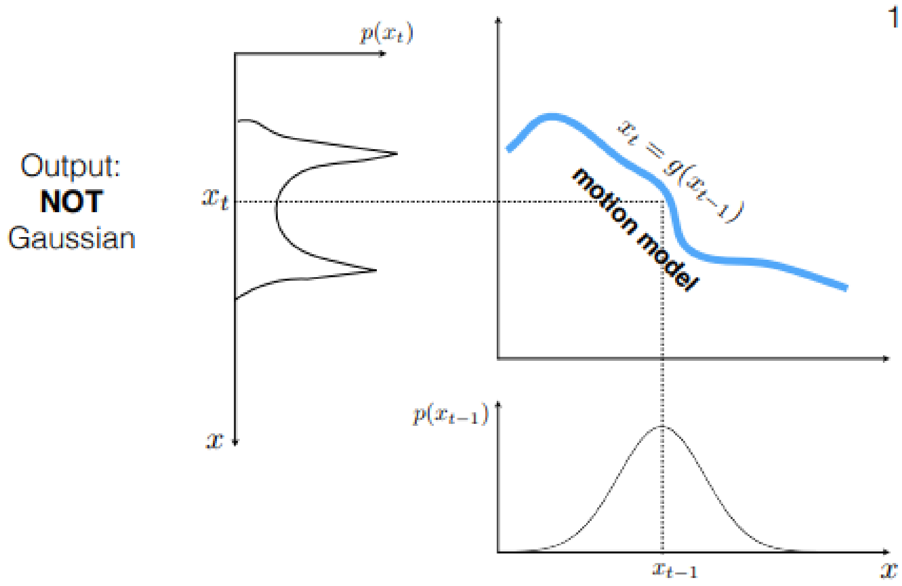

FIG. 20 – One step gaussian prediction on the nonlinear system case (Image from https://www.cs.cmu.edu/~16385/s17/Slides/16.4_Extended_Kalman_Filter.pdf)

3.3.1.4. Unscented Kalman Filter

The main drawback of the Extended Kalman Filter is that the assumption of gaussianity might be lost or not applicable due to the nonlinear nature of physical systems.

While the sum of gaussian variables returns a gaussian variable, this doesn't happen when nonlinearities characterize the system evolution.

Further, the optimal Bayesian solution requires the propagation of the description of the full probability density function (pdf) (see [28]).

Despite all the estimators use some kind of approximation for the solution, in practice due to the implementation unfeasibility, the Kalman Filter remains the most used algorithm in the industry, since it uses only the first two statistical moments for the state evolution and update (mean and covariance).

Julier and Uhlmann described the "Unscent Transformation" to address the deficiencies of the 1st order approximation introduced by the EKF and to have a more direct way to transform the mean and covariance information.

Basically, this consists in choosing a set of points, called "sigma points", in order to represent the state mean and covariance. The nonlinear function is applied to each point to obtain a set of transformed points.

Then the statistics of the points is computed to get an estimate of the nonlinear transformation of the mean and covariance (See [27]).

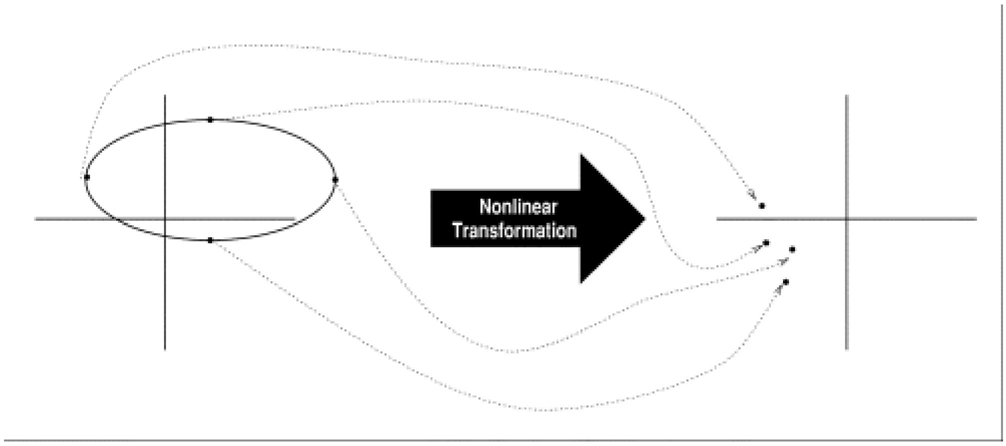

FIG. 21 – The sigma points under the Unscent Transformation (Image from **[27]**)

The idea behind the UKF is not to approximate the nonlinear system to a linear system (process and observation model) but to approximate the covariance and the mean.

The sigma points are computed as follows:

Consider a nonlinear process of the following form:

$$\dot{x} = f(x)$$

$$y = g(x)$$

- Choose a set of sigma points in order to characterize the true mean and covariance of the random variable "x" (the number is 2n+1)

$$X_0 = \hat{x}, \qquad W_0 = \frac{\lambda}{(n + \lambda)}, \qquad i = 0$$

$$X_i = \hat{x} + (\sqrt{(n+\lambda)\Psi(x)}_i, \qquad W_i = \frac{1}{2(n + \lambda)}, \qquad i = 1, \ldots, n$$

$$X_i = \hat{x} - (\sqrt{(n+\lambda)\Psi(x)}_i, \qquad W_i = \frac{1}{2(n + \lambda)}, \qquad i = n+1, \ldots, 2n$$

With "λ" as a scaling parameter and $(\sqrt{(n+\lambda)\Psi(x)}_i$ i-th row of the matrix $\sqrt{(n+\lambda)\Psi(x)}$.

The weights W_i associated to each point are such that their total sum is 1

$$\sum_{i=0}^{2n} W_i = 1,$$

- The sigma points are propagated through the nonlinear output function

$$Y_i = g(X_i), \qquad i = 0, \ldots, 2n$$

- The estimated mean and covariance is computed accordingly:

$$\hat{y} = \sum_{i=0}^{2n} W_i \cdot Y_i$$

$$\Psi(x) = \sum_{i=0}^{2n} W_i \cdot (Y_i - \hat{y}) \cdot (Y_i - \hat{y})^T$$

Accurate up to the 2^{nd} order of Taylor series expansion

The algorithm steps are:

- Prediction

The state estimate is augmented with the mean and covariance of the process noise

$$X^{aug}_{(k-1)|(k-1)} = \begin{bmatrix} \hat{x}^T_{(k-1)|(k-1)} \\ E[w^T_k] \end{bmatrix}$$

$$\Psi(x) = \begin{bmatrix} \Psi_{(k-1)|(k-1)} & 0 \\ 0 & Q_k \end{bmatrix}$$

The set of $2n_{aug} + 1$ sigma points (n_{aug} being the dimension of the state plus the process noise) is computed as described previously (the square root of $(n + \lambda)\Psi(x)$ could be computed by means of Cholesky decomposition for computational effort).

The sigma points are propagated through the system nonlinear dynamics

$$X^i_{k|k-1} = f(X^i_{k-1|k-1}), \qquad i = 0,\ldots,2n^{aug}$$

The weighted sigma points are summed in order to give the state and covariance prediction for the next iteration step

$$\hat{x}_{k|k-1} = \sum_i W^s_i X^i_{k|k-1}, \qquad i = 0,\ldots,2n^{aug}$$

$$\Psi_{k|k-1} = \sum_i W^c_i [X^i_{k|k-1} - \hat{x}_{k|k-1}][X^i_{k|k-1} - \hat{x}_{k|k-1}]^T, \; i = 0,\ldots,2n^{aug}$$

Where the weights for the covariance matrix, W^c_i, are computed as the ones for the state with the attention on the 1st weight that is derived from:

$$W^c_0 = \frac{\lambda}{(n + \lambda)} + (1 - \alpha^2 + \beta), \qquad i = 0$$

And the scaling parameter "λ" is now computed as

$$\lambda = \alpha^2(n^{aug} + k) - n^{aug}, \qquad i = 0$$

- Update

The state and covariance predictions are augmented with the statistics of the measurement noise

$$X_{k|(k-1)}^{aug} = \begin{bmatrix} \hat{x}_{k|(k-1)}^{T} \\ E[v_k^T] \end{bmatrix}$$

$$\Psi(x) = \begin{bmatrix} \Psi_{(k-1)|(k-1)} & 0 \\ 0 & R_k \end{bmatrix}$$

The sigma points are computed from the augmented state and covariance

The sigma points are propagated through the measurement function "h(x)"

$$\gamma_{k|k-1}^{i} = h(X_{k-1|k-1}^{i}), \qquad i = 0,\ldots,2n^{aug}$$

The predicted state and covariance measurement are computed via the sigma points

$$\hat{z}_{k|k-1} = \sum_i W_i^s \, \gamma_{k|k-1}^{i}, \qquad i = 0,\ldots,2n^{aug}$$

$$\Psi_{z_k z_k} = \textstyle\sum_i W_i^c \, [\gamma_k^i - \hat{z}_k][\gamma_k^i - \hat{z}_k]^T, \; i = 0,\ldots,2n^{aug}$$

The cross covariance matrix (between the state and the measurement) is used to get the UKF gain

$$\Psi_{x_k z_k} = \sum_i W_i^c \, [X_{k|k-1}^{i} - \hat{x}_{k|k-1}][\gamma_k^i - \hat{z}_k]^T, \qquad i = 0,\ldots,2n^{aug}$$

$$K_k = \Psi_{x_k z_k} \Psi_{z_k z_k}^{-1}$$

Finally, the updated state estimate and covariance matrix are obtained weighting with the Kalman gain

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - \hat{z}_k)$$
$$\Psi_{k|k} = \Psi_{k|k-1} - K_k \Psi_{z_k z_k} K_k^T$$

During the Unscent transformation the sigma points are chosen so that as the dimension of the system increases the same happens to the sphere that bounds all the sigma points. Despite the fact that it correctly captures the mean and covariance of the prior distribution, it samples nonlocal effects. To counter this, the sigma points can be scaled either close or far from the mean of the prior distribution through the scaling parameter "λ".

As further remark the main drawback of this algorithm concerns the choice of the sigma point via the weights selection, which can be different depending on the system considered and so it makes not easy the implementation and the success of the estimation.

4. Case of Study

In this paragraph, a case of study for attitude control applied on launcher vehicle based on nonlinear output regulation feedback design will be presented.

The objective of the controller is to keep the desired attitude during the flight, so that the attention can be focused on the rotational dynamics:

$$\begin{cases} I_{xx}\dot{P} + (I_{zz} - I_{yy})QR = M_x \\ I_{yy}\dot{Q} + (I_{xx} - I_{zz})PR = -l_c T_c \delta + \frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha}(\eta_{cg} - \eta)\alpha(\eta)d\eta \\ I_{zz}\dot{R} + (I_{yy} - I_{xx})PQ = -l_c T_c \varepsilon - \frac{1}{2}\rho V_{rel}^2 S_R \int_0^L \frac{\partial C_N(\eta)}{\partial \alpha}(\eta_{cg} - \eta)\beta(\eta)d\eta \end{cases}$$

$$\begin{cases} \dot{\varphi} = p + s\varphi tn\theta \cdot q + c\varphi tn\theta \cdot r \\ \dot{\theta} = c\varphi \cdot q - s\varphi \cdot r \\ \dot{\psi} = \frac{s\varphi}{c\theta} \cdot q + \frac{c\varphi}{c\theta} \cdot r \end{cases}$$

The aerodynamic moments can be considered as an external disturbance acting on the system. When the angle of attack is kept very close to zero also their presence tends to diminish.

As first approximation we can rearrange the rotational equations plus kinematics in order to get a nonlinear systems affine in the control, namely:

$$\begin{cases} \dot{x} = f(x,t) + g(x,t)u(t) \\ y = h(x,t) \end{cases}$$

With:

$$f(x) = \begin{bmatrix} -\,(I_{zz} - I_{yy})QR \big/ I_{xx} \\ -\,(I_{xx} - I_{zz})PR \big/ I_{yy} \\ -\,(I_{yy} - I_{xx})PQ \big/ I_{zz} \\ p + s\varphi tn\theta \cdot q + c\varphi tn\theta \cdot r \\ c\varphi \cdot q - s\varphi \cdot r \\ \frac{s\varphi}{c\theta} \cdot q + \frac{c\varphi}{c\theta} \cdot r \end{bmatrix}$$

$$g(x) = \begin{bmatrix} -\,(T_c - l_c) \big/ I_{xx} & 0 & 0 \\ 0 & -\,(T_c - l_c) \big/ I_{yy} & 0 \\ 0 & 0 & -\,(T_c - l_c) \big/ I_{zz} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$y(x) = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix}$$

The input acting on the roll dynamics was added in order to get a square system, so with the same number of inputs and outputs. Anyway, it could be considered to be the presence of external thrusters that act just to stabilize the roll motion during the flight. Another important statement is that due to the axial symmetry of the launcher, it is very common to decouple the yaw-pitch motion from the roll dynamics, which is very close to zero and at least stabilized by a devoted actuation system in general.

So for the simulations the roll motions will be considered almost zero and it will be seen as it is almost a stable dynamics over time.

$$L_{g_1}h_1(x) = [\,0\;0\;0\;1\;0\;0\,]g_1(x) = 0$$

$$L_{g_2}h_1(x) = [\,0\;0\;0\;1\;0\;0\,]g_2(x) = 0$$

$$L_{g_3}h_1(x) = [\,0\;0\;0\;1\;0\;0]g_3(x) = 0$$

$$L_f h_1(x) = [\,0\;0\;0\;1\;0\;0]f(x) = p + s\varphi tn\theta q + c\varphi tn\theta r$$

$$L_{g_1}L_f h_1(x) = -\frac{T_c l_c}{I_{xx}}$$

$$L_{g_2}L_f h_1(x) = -\frac{T_c l_c}{I_{yy}} s\varphi tn\theta$$

$$L_{g_3}L_f h_1(x) = -\frac{T_c l_c}{I_{zz}} c\varphi tn\theta$$

$$L_{g_1}h_2(x) = [\,0\;0\;0\;0\;1\;0]g_1(x) = 0$$

$$L_{g_2}h_2(x) = [\,0\;0\;0\;0\;1\;0]g_2(x) = 0$$

$$L_{g_3}h_2(x) = [\,0\;0\;0\;0\;1\;0]g_3(x) = 0$$

$$L_f h_2(x) = [\,0\;0\;0\;0\;1\;0]f(x) = c\varphi Q - s\varphi R$$

$$L_{g_1}L_f h_2(x) = 0$$

$$L_{g_2}L_f h_2(x) = -\frac{T_c l_c}{I_{yy}} c\varphi$$

$$L_{g_3}L_f h_2(x) = \frac{T_c l_c}{I_{zz}} s\varphi$$

$$L_{g_1}h_3(x) = [0\ 0\ 0\ 0\ 0\ 1]g_1(x) = 0$$

$$L_{g_2}h_3(x) = [0\ 0\ 0\ 0\ 0\ 1]g_2(x) = 0$$

$$L_{g_3}h_3(x) = [0\ 0\ 0\ 0\ 0\ 1]g_3(x) = 0$$

$$L_f h_3(x) = [0\ 0\ 0\ 0\ 0\ 1]f(x) = \frac{s\varphi}{c\theta}Q + \frac{c\varphi}{c\theta}R$$

$$L_{g_1}L_f h_3(x) = 0$$

$$L_{g_2}L_f h_2(x) = -\frac{T_c l_c}{I_{yy}}\frac{s\varphi}{c\theta}$$

$$L_{g_3}L_f h_2(x) = \frac{T_c l_c}{I_{zz}}\frac{c\varphi}{c\theta}$$

The decoupling matrix obtained is:

$$A(x) = \begin{bmatrix} \frac{-T_c l_c}{I_{xx}} & -\frac{T_c l_c s\varphi tn\theta}{I_{yy}} & -\frac{T_c l_c c\varphi tn\theta}{I_{zz}} \\ 0 & \frac{-T_c l_c c\varphi}{I_{yy}} & \frac{T_c l_c s\varphi}{I_{zz}} \\ 0 & \frac{-T_c l_c s\varphi}{c\theta I_{yy}} & \frac{-T_c l_c c\varphi}{c\theta I_{zz}} \end{bmatrix}$$

which gets singular for $\theta = \frac{\pi}{2} + k\pi$

However, for that value of the pitch angle also the Euler angles representation gets singular, so it's a standard condition to be avoided.

The inverse is:

$$A(x)^{-1} = \begin{bmatrix} \frac{-I_{xx}}{T_c l_c} & 0 & -\frac{I_{xx} s\theta}{T_c l_c} \\ 0 & -\frac{I_{yy} c\varphi}{T_c l_c} & -\frac{I_{yy} c\theta s\varphi}{T_c l_c} \\ 0 & \frac{I_{zz} s\varphi}{T_c l_c} & -\frac{I_{zz} c\theta c\varphi}{T_c l_c} \end{bmatrix}$$

The system has uniform vector relative degree with value 2 with respect to every output, so no zero dynamics exists and can be defined a coordinate transformation that is a diffeomorphism.

$$b(x) = \begin{bmatrix} L_f^2 h_1(x) \\ L_f^2 h_2(x) \\ L_f^2 h_3(x) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & s\varphi tn\theta & c\varphi tn\theta & c\varphi tn\theta \cdot Q - s\varphi tn\theta \cdot R & (s\varphi \cdot Q + c\varphi \cdot R)/c\theta^2 & 0 \\ 0 & c\varphi & -s\varphi & -s\varphi \cdot Q - c\varphi \cdot R & 0 & 0 \\ 0 & {}^{s\varphi}/_{c\theta} & {}^{c\varphi}/_{c\theta} & (c\varphi \cdot Q - s\varphi \cdot R)/c\theta & (s\varphi \cdot Q + c\varphi \cdot R)s\theta/c\theta^2 & 0 \end{bmatrix} f(x)$$

The linearizing controller is defined as:

$$u(x) = A(x)^{-1}(\begin{pmatrix} v_1(x) \\ v_2(x) \\ v_3(x) \end{pmatrix} - b(x))$$

Assuming that there is no knowledge of the unmeasured state, the EKF is introduced in order to provide the state estimate to the nonlinear controller.

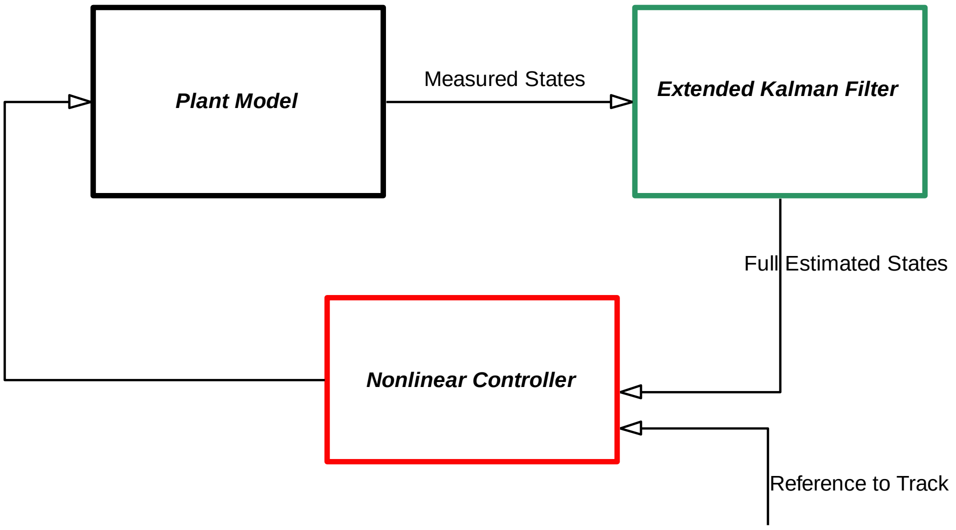

FIG. 11 – Nonlinear Output Feedback control Scheme

First is shown the behavior of the nonlinear controller in the case of full information feedback.

Provided an initial attitude for the vehicle, in the next figure it is shown the free evolution behavior:

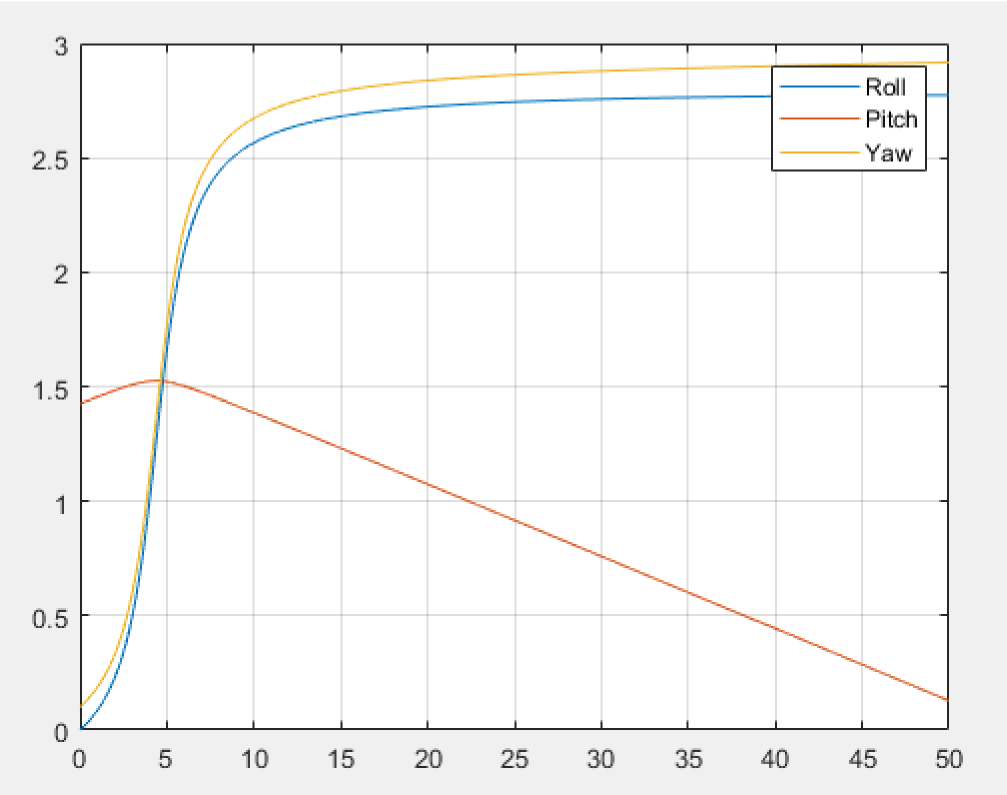

FIG. 12 – Free evolution of the uncontrolled system, Attitude behavior

Clearly the system is not stable in free evolution. Even if the roll motion, as previously stated, is characterized by a stable dynamics the divergent behavior of the pitch and the couplings between the three rotational dynamics make it unstable too.

Applying the nonlinear feedback and required a desired attitude of:

$$\theta = 1\ [rad], \qquad \psi = 0.1\ [rad]$$

The action of the controller makes the system stable and the desired reference is reached.

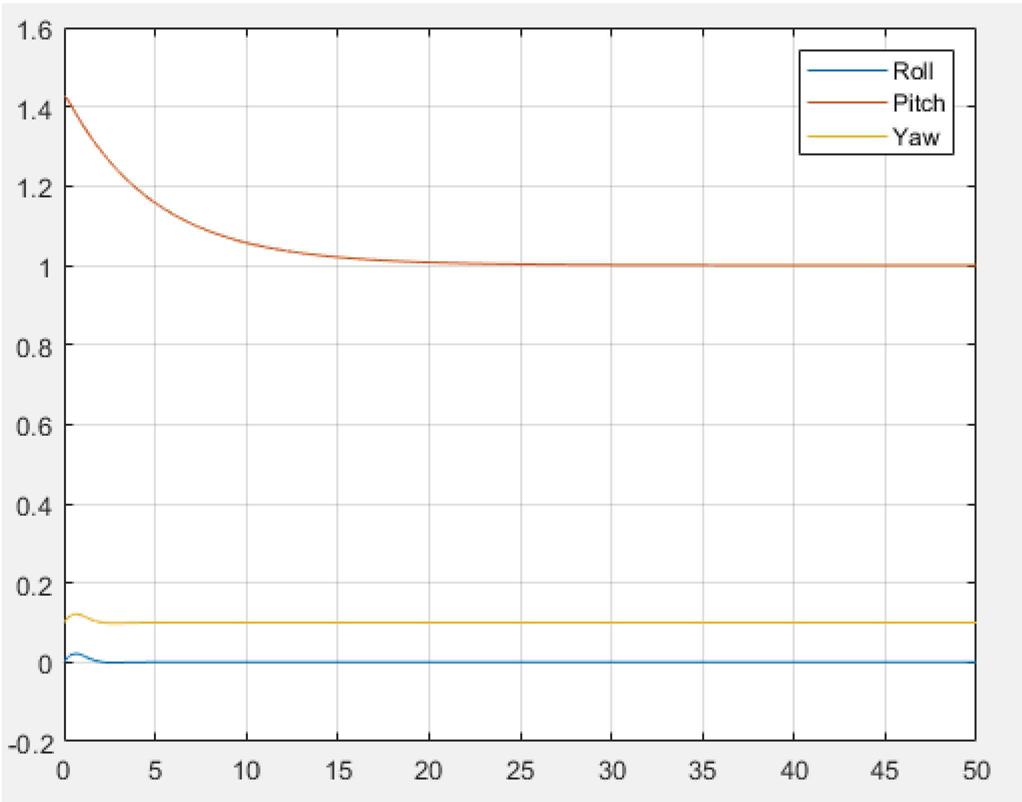

FIG. 13 – System under the action of the full state nonlinear feedback, Attitude behavior

It is again important to underline the fact that the first controller was added to get a square MIMO system but the actual controller computation is not applied to the original system.

Anyway from the previous figure it can be seen that the roll motion is characterized by a stable dynamics and in order to keep it to 0 a separate controller can be used in specific flight instants.

Concerning the new virtual input that acts on the linearized system, "v", different methods can be used to assign the desired dynamics, for example standard "pole-placement" or LQR and so on. The idea is that limitation in the "TVC" actuation system performance can be taken into account when designing the linear controller in order not to ask instantaneous or not-realistic performances that could not be actuated.

Considering now the case where no full state information is available to the controller, the Extended Kalman filter is added to the controller architecture in order to reconstruct the unmeasured state dynamics.

Here is the state estimation error as output of the EKF algorithm during the system evolution.

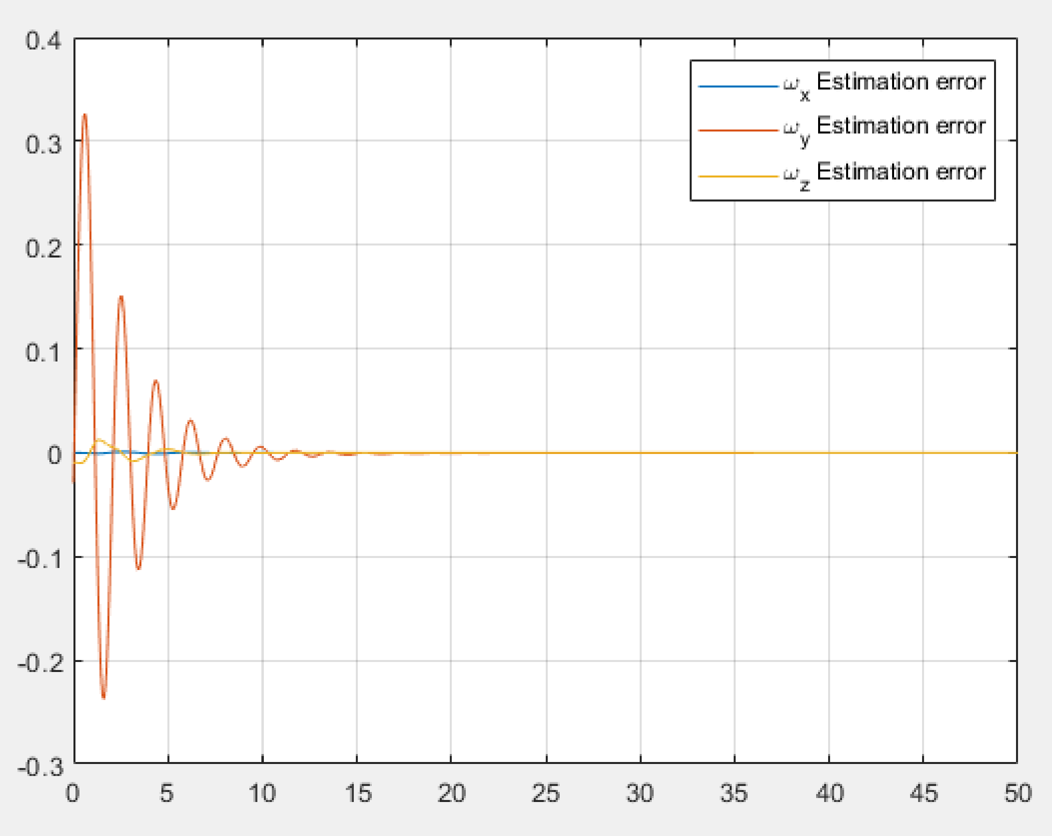

FIG. 14 – Estimation error [rad/s] of the angular velocities as output of the EKF

The EKF is able to reconstruct the state dynamics accurately. Of course a big impact on the time of convergence and good behavior of the filter is due to the initial guess of the unmeasured states, that anyway can be guessed due to the fact that the launcher has to be kept stable all over the flight (high values of these states will lead to the instability of the vehicle).

The result of the nonlinear output feedback controller is the following:

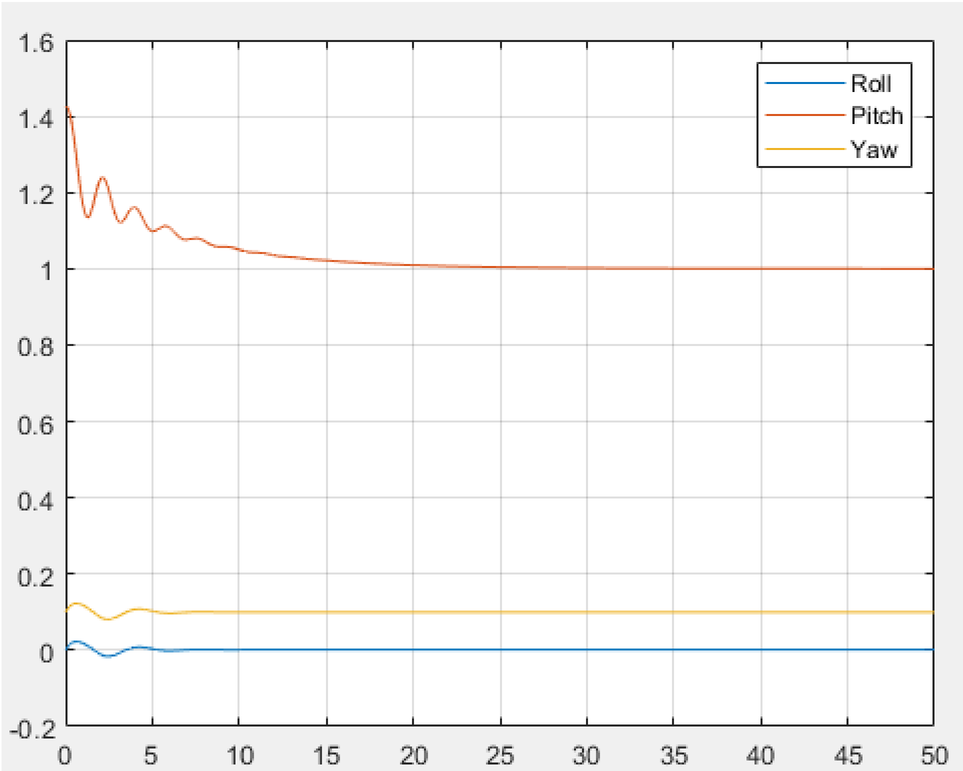

FIG. 15 – System under the action of the nonlinear output feedback controller. Attitude behavior [rad]

Clearly until the EKF has almost estimated the angular velocities the controller does not perform exactly the linearization and this implies an oscillatory behavior on the system response with respect to the case of the full state feedback. However these oscillations are acceptable.

5. Conclusions

The present work had the objective to describe the architecture of a nonlinear output feedback for the control of the attitude of a launch vehicle. Briefly were recalled the equations of motion that can be used to model this type of systems with all the references for a more comprehensive reading.

The classical architecture used for launcher control was shortly described together with some references to innovative methodologies to deal with such complex dynamical system (like LPV).

The challenges for designing an efficient controller are many and from theoretical analysis to possible implementation a lot of problems arise due also to the limits in the computations, on the system knowledge and the sensors and actuators present on board.

The nonlinear control techniques can however be seen as a more innovative solution to handle this system, even considering all the limitations illustrated in the previous chapters.

The major challenge is related to the time varying nature of the system and the future goal would be to try to take into account this aspect in the definition of an adaptive controller and of stability criteria which do not rely on the first order system approximation and LTI methodologies.

Bibliography

[1] G. Baldesi, "Modeling, control and simulation for a launch vehicle: from linear to nonlinear methods", 2007.

[2] A. Isidori "Nonlinear control system", Springer, third edition, 1995.

[3] Timothy Barrows and Jeb Orr ,"Dynamics and Simulations of flexible rockets", Academic Press, first edition, 2021.

[4] A. L. Greensite ,"Analysis and Design of Space Vehicle Flight Control Systems. Volume I – Short Period Dynamics", 1967.

[5] C. Bruni, C. Ferrone,"Metodi di stima per il filtraggio e l'identificazione dei sistemi", Aracne, 2009.

[6] A. S. Hodel, Whorton, J. Jim Zhu ,"Stability Metrics for Simulation and Flight-Software Assessment and Monitoring of Adaptive Control Assist Compensators", 2008.

[7] Hassan K. Khalil ,"Nonlinear Systems", Pearson, third edition, Pearson, 2001.

[8] J. Jim Zhu, Xiaojing Yang and A. Scottedward Hodel ,"A Singular Perturbation Approach for Time-Domain Assessment of Phase Margin", American Control Conference, 2010.

[9] S. Skogestad and I. Postlethwaite ,"Multivariable Feedback Control Analysis and design", Wiley, second edition, 2005.

[10] D.J. Leith and W.E. Leithead, "Survey of gain-scheduling analysis and design", International Journal Of Control, 2000.

[11] A. Isidori ,"Sistemi di Controllo", Vol.1, Siderea, Roma, Italy, 1996.

[12] H. Kimura, "Pole assignment by gain output feedback", IEEE Transactions on Automatic Control, 1975.

[13] A. J. Krener ,"On the Equivalence of Control Systems and Linearizations of Nonlinear Systems," SIAM Journal of Control and Optimization, 1973.

[14] A. Isidori, A. J. Krener, C. Gori-Giorgi, S. Monaco ,"Nonlinear decoupling via feedback: a differential geometric approach", IEEE Transactions on Automatic Control, Vol. 26, 1981.

[15] D. H. Platus ,"Aeroelastic Stability of Slender Spinning Missile", Journal Of Guidance,

Dynamics And Control.
[16] B. Etkin ,"Dynamics Of Flight", Wiley, New York, 2nd edition, 1982.
[17] Duke, Antoniewicz and Krambeer ,"Derivation and definition of a linear aircraft model",Nasa RP–1207, 1988.
[18] W.J. Rugh and J.S. Shamma ,"Research on gain scheduling", Automatica, 2000.
[19] Tannen VanZwieten, J. Jim Zhu Tony Adami Kyle Berry and Alex Grammar, Jeb S. Orr, Eric A. Best ,"Stability Assessment and Tuning of an Adaptively Augmented Classical Controller for Launch Vehicle Flight Control".
[20] L. Sonneveldt Q.P. Chu, J.A. Mulder "Adaptive Backstepping Flight Control for Modern Fighter Aircraft", Advances in Flight Control Systems, 2011.
[21] James Diebel, Stanford University ,"Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors".
[22] W. T. Thomson, J. Wiley & Sons ,"Introduction to Space Dynamics", Inc., New York, 1963.
[23] A. Banerjee ,"Dynamics Of A Variable Mass, Flexible Body System", Journal Of Guidance, Dynamics And Control, Vol. 23, No. 3, pp 501-508, 2003.
[24] L. Meirovitch ,"General Motion of a Variable Mass Flexible Rocket with Internal Flow", Journal of Spacecraft and Rocket, Vol. 7, No. 2, pp.186-195, 1970.
[25] Maria Isabel Ribeiro ,"Kalman and Extended Kalman Filters: Concept, Derivation and Properties", Institute for Systems and Robotics Instituto Superior Tecnico.
[26] Axel Barrau, Silvère Bonnabel ,"Extended Kalman Filtering with Nonlinear Equality Constraints: a Geometric Approach.", IEEE Transactions on Automatic Control, 2020.
[27] Simon J. Julier, And, Jeffrey K. Uhlmann ,"Unscented Filtering and Nonlinear Estimation", Proceedings of the IEEE, vol. 92, 2004.
[28] "Adaptive model architecture and extended Kalman – Bucy filters", IEEE Trans. Aerosp. Electron. Syst., 1994.
[29] Särkkä, "On Unscented Kalman Filtering for State Estimation of Continuous-Time Nonlinear Systems", IEEE Transaction on Automatic Control, vol. 52, no. 9, 2007.
[30] Gabriel A. Terejanu, " Unscented Kalman Filter Tutorial", Department of Computer Science and Engineering University at Buffalo, Buffalo, NY 14260.

[31] Ryan Turner and Carl Edward Rasmussen, "Model Based Learning of sigma points in unscented Kalman filtering", Neurocomputing, 2012.
[32] A. Astolfi, L. Marconi, "Analysis and Design of Nonlinear Control Systems", Springer, 2007.
[33] Wei Du, Bong Wie, Mark Whorton, "Dynamic Modeling and Flight Control Simulation of a Large Flexible Launch Vehicle", AIAA Guidance, Navigation and Control conference, 2008.
[34] C. Bruni, G. Di Pillo, "Metodi variazionali per il controllo ottimo", Aracne, 2007.
[35] D. Liberzon, "Calculus of Variations and Optimal Control Theory: A Concise Introduction", Princeton University Press.