Modelling and Controlling the landing procedure of an F/A Hornet

Alessandro Gallo

February 2024

A Proof of Concept

1 Finding the right mathematical model

This section will try to model the landing maneuver of the jet. Keep in mind that this is a simplified model, as a matter of fact the jet will only be able to perform maneuvers on the x and y axis and will only be able to perform a *pitch* maneuver.
The jet is constrained by some conditions in various instants of the process:

1) Initial horizontal velocity $v_{x_0} = 1000\frac{km}{h}$
2) When altitude is greater than $15m$: $-10^o < \alpha < 10^o$
3) When altitude is lower than $15m$: $\alpha = 3$
4) Maximum touchdown horizontal velocity $v_{touchdown} \leq 180\frac{km}{h}$
5) Wind is a Poisson random variable of parameter λ

1.1 Approach

The type of approach that will be used is an automatic approach as it best fits the problem. The maneuver, is considered a discrete time system that evolves in the following way:

$$X_i = \hat{F}(X_{i-1}) \tag{1}$$

$$X_i = \begin{pmatrix} x_i & y_i & vx_i & vy_i \end{pmatrix}^T \quad , \quad \text{state vector in the i-th instance of time.}$$

The function $\hat{F}$ with which the system evolves changes once the jet touches the runway, this is why it's useful to split the model in "2 phases"

1) *Landing Phase* if altitude $y_i > 0$

2) *Breaking Phase* if altitude $y_i = 0$

To find $\hat{F}$ the following definitions of velocity and acceleration will be used:

$$v = \frac{\Delta x}{\Delta t} = \frac{x_i - x_{i-1}}{\Delta t} \quad , \quad a = \frac{\Delta v}{\Delta t} = \frac{v_i - v_{i-1}}{\Delta t} \tag{2}$$

inverting (2) the components of the state vector are obtained:

$$\begin{cases} x_i = x_{i-1} + v_{i-1_x} \cdot \Delta t \\ y_i = y_{i-1} + v_{i-1_y} \cdot \Delta t \\ vx_i = v_{i-1_x} + a_{i-1_x} \cdot \Delta t \\ vy_i = v_{i-1_y} + a_{i-1_y} \cdot \Delta t \end{cases} \tag{3}$$

1.2 Wind

Approximately 10 gusts of wind of different intensity (velocity) are expected during the Landing phase. The system (3) doesn't include this effect. For simplicity purposes I will be considering the expected value of the Poisson random variable W (wind) as a horizontal velocity opposing the velocity of the jet.

$$E[W] = \lambda$$
$$\lambda = 10\frac{km}{h} = 2,77\frac{m}{s}$$

1.3 Forces

The acceleration along both axis depends on the altitude. While the jet is in the Landing Phase it is subject to a horizontal aerodynamic force defined as follows:

$$F_i = \frac{1}{2}CA\rho v_i^2 \quad , \quad A = A_h \sin\alpha + 4 \tag{4}$$

A being the area 'under the body of the jet' exposed to this force
C being the aerodynamic coefficient
ρ being the air density

α being the current *pitch* angle (also known as the *attack angle*)

On the other hand, while the jet is in the Breaking Phase (meaning it's on the runway) a tailhook on the jet snags onto a cable attached to a spring (this procedure is called carrier landing), to further break the jet, as an elastic force defined as follows:

$$F_k = -K \cdot (x_i - d) \tag{5}$$

d being the distance from the runway at which the landing procedure starts
K being the stiffness of the spring

1.4 Acceleration

The acceleration is defined following Newton's second law coherently with the forces cited in (5) and (4)

$$\begin{cases} a_{x_i} = \begin{cases} -\frac{F}{m} \quad \text{if} \quad y_i \geq 0 \\ -\frac{F+F_k}{m} \quad \text{if} \quad y_i = 0 \end{cases} \\ a_{y_i} = \begin{cases} -g \quad \text{if} \quad y_i \geq 0 \\ 0 \quad \text{if} \quad y_i = 0 \end{cases} \end{cases} \tag{6}$$

1.5 The Model

By substituting the Forces in play in (6) and then plugging in (2), a model can be obtained for the two phases of the system:

Landing Phase

$$\begin{cases} x_i = x_{i-1} + v_{i-1_x} \cdot \Delta t \\ y_i = y_{i-1} + v_{i-1_y} \cdot \Delta t \\ vx_i = v_{i-1_x} - \dfrac{1}{m}(0.5 \cdot CA\rho v_{i-1_x}^2)\Delta t - \dfrac{10 \cdot E[W]}{N} \\ vy_i = v_{i-1_y} - g \cdot \Delta t \end{cases} \tag{7}$$

Breaking Phase

$$\begin{cases} x_i = x_{i-1} + v_{i-1_x} \cdot \Delta t \\ y_i = 0 \\ vx_i = v_{i-1_x} - \dfrac{1}{m}(0.5 \cdot CA\rho v_{i-1_x}^2 + K(x_{i-1} - d))\Delta t \\ vy_i = 0 \end{cases} \tag{8}$$

The model is coherent with (1) where the i-th value of the state vector solely depends on the (i-1)-th value

2 Controlling the System

The natural evolution of the model is described in 1.5.
In reality the model includes an input that is meant to be interpreted as the pilot's decisions. Ideally the pilot's controls make the jet move in a smooth manner and stop it within the $200m$ of the runway
The model including the pilot's input is:

Landing Phase

$$\begin{cases} x_i = x_{i-1} + v_{i-1_x} \cdot \Delta t + u_{i-1x} \cdot \Delta t \\ y_i = y_{i-1} + v_{i-1_y} \cdot \Delta t + u_{i-1y} \cdot \Delta t \\ vx_i = v_{i-1_x} - \frac{1}{m}(0.5 \cdot CA\rho v_{i-1_x}^2)\Delta t + u_{i-1x} - \frac{10 \cdot E[W]}{N} \\ vy_i = v_{i-1_y} + u_{i-1y} - g \cdot \Delta t \end{cases} \tag{9}$$

Breaking Phase

$$\begin{cases} x_i = x_{i-1} + v_{i-1_x} \cdot \Delta t + u_{i-1x} \cdot \Delta t \\ y_i = 0 \\ vx_i = v_{i-1_x} + u_{i-1x} - \frac{1}{m}(0.5 \cdot CA\rho v_{i-1_x}^2 + K(x_{i-1} - d))\Delta t \\ vy_i = 0 \end{cases} \tag{10}$$

There are two main ways to approach this problem: *optimal control theory* and *dynamic programming.*
The *optimal control* approach aims essentially at minimizing a pre-determined cost function. The goal of the following sections is to build a QP (quadratic program) to solve the optimal control optimization problem:

$$\min_u (V(x,u)) = \min_u \left(x_L^T Q_L x_L + \sum_{k=1}^{L} x_k^T Q_k x_k + u_k^T R u_k \right) \tag{11}$$

$V(x,u)$ being a cost function related to the problem. The down side of this "simple" formulation is that the discrete time system has to be linear, in the way that $X_{i+1} = Ax_i + Bu_i$.

Linearizing a non-linear system is a local concept, therefore there is no way to globally make a nonlinear system linear. A further simplification to easily make the system linear is considering the aerodynamic force defined in equation (4) like $F \propto \beta v$ instead of $F \propto \beta v^2$. In simple terms this means substituting an aerodynamic force with a simple drag force. The difference between the two is that aerodynamic forces include the effect of pressure and viscous forces while a simple drag force describes on object moving through a fluid (air in our case). Now it's possible to write the system in the form $X_{i+1} = Ax_i + Bu_i$

Landing Phase

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \\ v_{x_{i+1}} \\ v_{y_{i+1}} \\ -10 \cdot E[W] \\ -g \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta & 0 & 0 \\ 0 & 0 & (1-\beta\Delta) & 0 & \frac{1}{N} & 0 \\ 0 & 0 & 0 & 1 & \Delta & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \\ v_{x_i} \\ v_{y_i} \\ -10 \cdot E[W] \\ -g \end{bmatrix} + \begin{bmatrix} \Delta & 0 \\ 0 & \Delta \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} U_{x,i} \\ U_{y,i} \end{bmatrix}$$

$$\beta = \frac{1}{m} 0.5 C A \rho$$

Breaking Phase

$$\begin{bmatrix} x_{i+1} \\ 0 \\ v_{x_{i+1}} \\ 0 \\ \gamma \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -K\Delta & 0 & (1-\beta\Delta) & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ 0 \\ v_{x_i} \\ 0 \\ \gamma \\ 0 \end{bmatrix} + \begin{bmatrix} \Delta \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \cdot \begin{bmatrix} U_{x,i} \end{bmatrix}$$

$$\gamma = K \Delta d$$

The state vector has 6 components because it includes the constants of the model.

Now, the matrices in the objective function $V(x, u)$ are defined in the following way, where the i-th element on the diagonal corresponds to the i-th component of the state vector.

Q_L is the terminal cost matrix and it quantifies the cost of deviations from the desired final state at step L

$$Q_L = \begin{bmatrix} q_{L_x} & 0 & 0 & \dots & 0 \\ 0 & q_{L_y} & 0 & \dots & 0 \\ 0 & 0 & q_{L_{vx}} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & q_{L_{-g}} \end{bmatrix}$$

Q is the evolution cost matrix and it quantifies the cost of deviation from the desired states at each step of the system's evolution

$$Q = \begin{bmatrix} q_x & 0 & 0 & \dots & 0 \\ 0 & q_y & 0 & \dots & 0 \\ 0 & 0 & q_{vx} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & q_{-g} \end{bmatrix}$$

R is the control cost matrix and it quantifies the cost of the control inputs. When its values are low it means that the system can be controlled in a more smooth way since controlling it in the first place doesn't cost much. On the other hand when its values are high the system will be controlled in a more strict way therefore making the jet maneuvers less smooth and natural.

$$R = \begin{bmatrix} R_{u_x} & 0 \\ 0 & R_{u_y} \end{bmatrix}$$

3 Simulating the Controlled Breaking Phase

The MATLAB simulations are based on the script "build_qp.m" (*Daniel K. Mills (2023). Discrete Linear Optimal Control (https://it.mathworks.com /matlabcentral/fileexchange), MATLAB Central File Exchange. Retrieved February 4, 2023.*).
It provides an interface to build a quadratic program to solve the optimization problem in eq. (11)
It's worth noting that during the breaking phase $\alpha = 0$

Different results will be shown to demonstrate how the controlled system evolves when distinct values of the coefficients in the matrices Q, Q_L, R are plugged in.

3.1 First Simulation

In this case the matrices are defined as follows:

```
QL = [1e9 0 0 0 0 0;   % Terminal cost matrix
      0 1e-12 0 0 0 0;
      0 0 1e6 0 0 0;
      0 0 0 1e-12 0 0;
      0 0 0 0 1e-12 0;
      0 0 0 0 0 1e-12;
     ];

Q = [1e3 0 0 0 0 0;     % Evolution cost matrix
     0 1e-12 0 0 0 0;
     0 0 1e-3 0 0 0;
     0 0 0 1e-12 0 0;
     0 0 0 0 1e-12 0;
     0 0 0 0 0 1e-12;
    ];

R = 1e10;           % Control cost matrix

```

Q_L's coefficients impose heavy cost contribution on not reaching the desired final position on the runway at the desired speed.
Q's coefficients imply that during the breaking phase horizontal velocity can vary without influencing much of the cost at the same time, meanwhile the

position has to pretty much follow nominal trajectory.
R has a single coefficient since the control input during the breaking phase is only one. It implies a very high cost when there are high control values.
The results can be seen in the figure below:

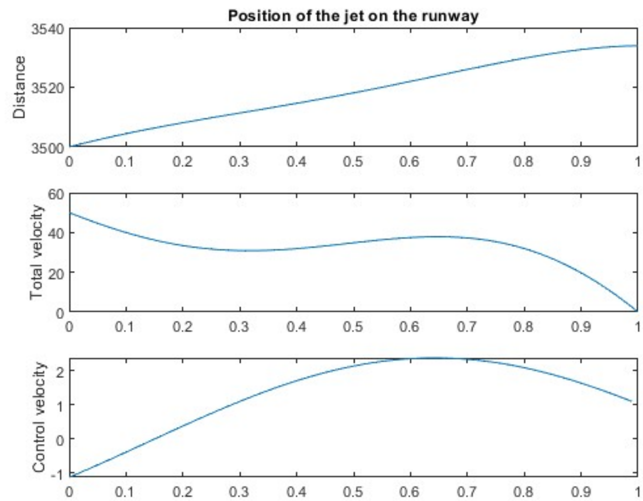


Figure 1: Sim. 1 Results

3.2 Second Simulation

In this case the matrices are defined as follows:

```
QL = [1e-3 0 0 0 0 0;    % Terminal cost matrix
       0 1e-12 0 0 0 0;
       0 0 1e-3 0 0 0;
       0 0 0 1e-12 0 0;
       0 0 0 0 1e-12 0;
       0 0 0 0 0 1e-12;
      ];

Q = [1e9 0 0 0 0 0;      % Evolution cost matrix
      0 1e-12 0 0 0 0;
      0 0 1e9 0 0 0;
      0 0 0 1e-12 0 0;
      0 0 0 0 1e-12 0;
      0 0 0 0 0 1e-12;
     ];
```

```
R = 1e-6;              % Control cost matrix
```

Q_L's low coefficients mean that even if the jet was not to reach the desired final state the cost would not be so high
Q's coefficients, instead, are meant to balance Q_L's coefficients implying that during the breaking phase the jet must 'stay close' to the nominal trajectory or else the cost would be very high
R's value, in this case, is indicating flexibility on the control values as they would not raise the cost
The results can be seen in the figure below:

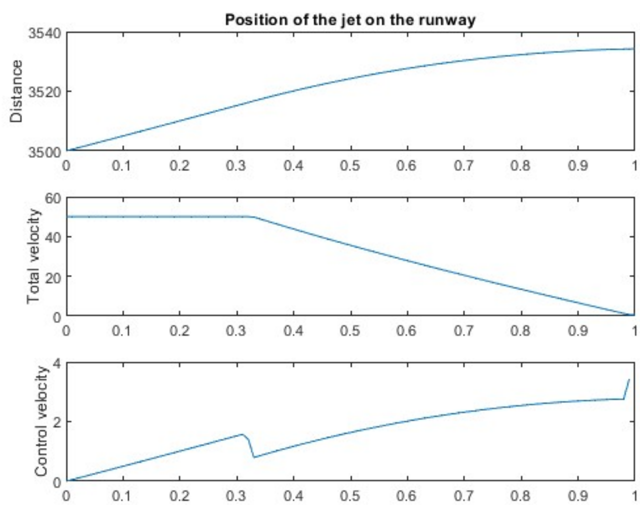


Figure 2: Sim. 2 Results

4 Simulating the Controlled Landing Phase

The Landing Phase simulation is done using the same scripts and functions as before. In this case to keep the system time invariant α is kept at a specific value $\alpha = 3$. This case seems to be more "complex" than the breaking phase since the control is a vector therefore B and R becomes a matrices.

4.1 First Simulation

The matrices are defined as follows:

```
        QL = [1e9 0 0 0 0 0;    % Terminal cost matrix
               0 1e9 0 0 0 0;
               0 0 1e12 0 0 0;
               0 0 0 1e9 0 0;
               0 0 0 0 1e-12 0;
               0 0 0 0 0 1e-12;
              ];


        Q = [1e9 0 0 0 0 0;      % Evolution cost matrix
              0 1e-12 0 0 0 0;
              0 0 1e9 0 0 0;
              0 0 0 1e-12 0 0;
              0 0 0 0 1e-12 0;
              0 0 0 0 0 1e-12;
             ];

        R = [1 0;            % Control cost matrix
              0 1
             ];

```

This way a higher cost is placed upon not reaching the desired final state and nominal trajectory for the distance and velocity along the x-axis. It also makes sense that during landing procedure there is a lot of flexibility for the control inputs to have a smooth touchdown.
During the landing phase it's worth mentioning the bounds that the state vector is subject to:

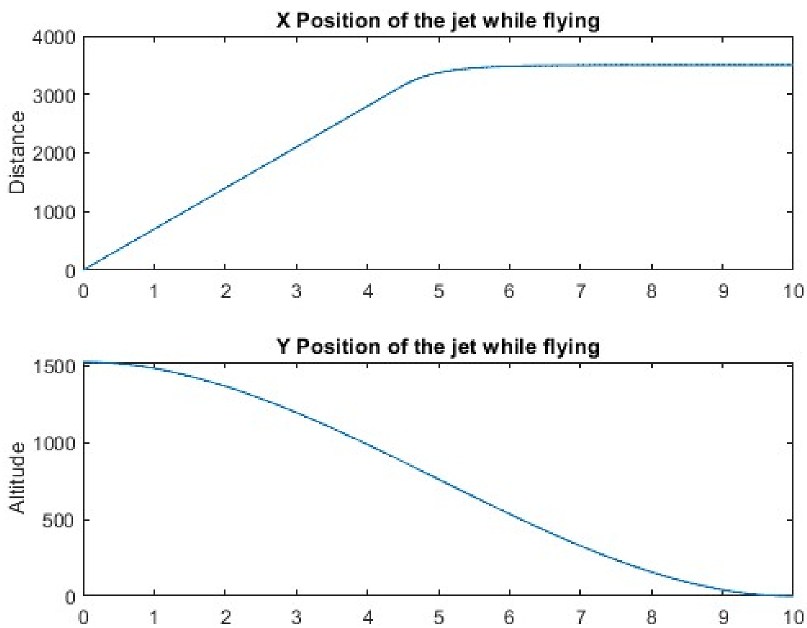
```
        % Box Constraints
        xmax = [d; h; 350; 150; -10*lambda; -g];
        xmin = [0; -h; -350; -150; -10*lambda; -g];
        umax = [350; 150];
        umin = [-350; -150];

```

The results are shown in three figures representing:

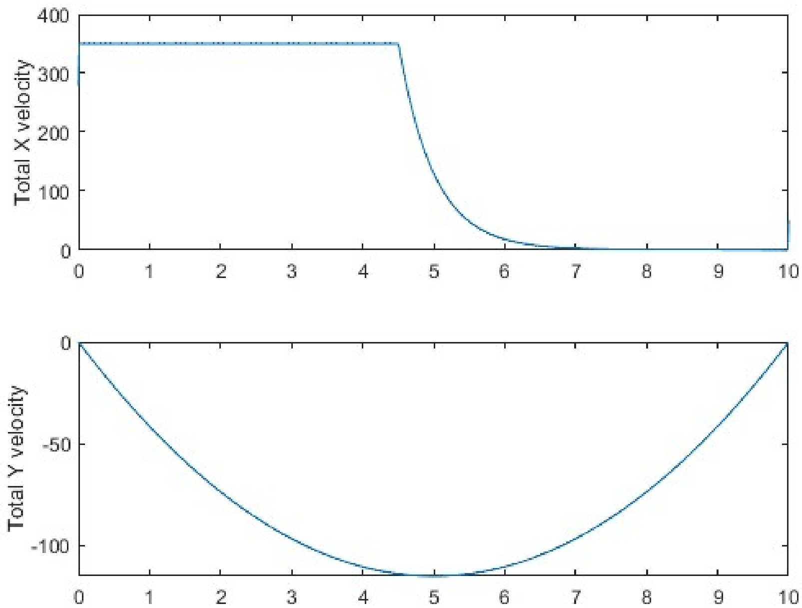
I) Distance and Altitude

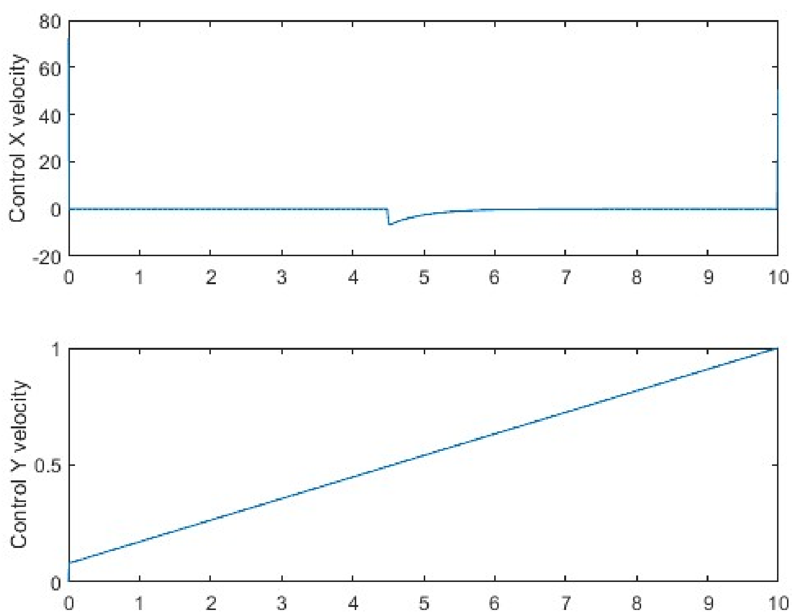
II) Total X and Y Velocity

III) Control X and Y velocity



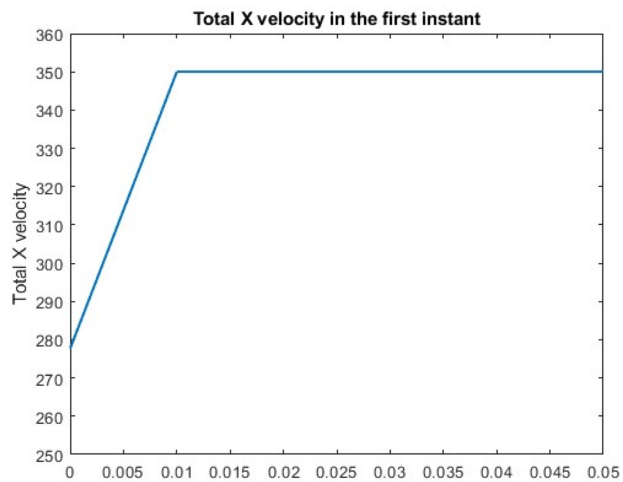
I

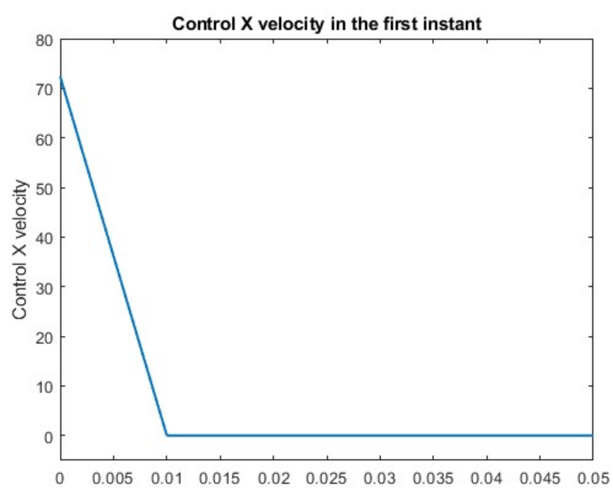

Total X velocity
Total Y velocity
400
300
200
100
0
-50
-100
0
1
2
3
4
5
6
7
8
9
10


Control X velocity
Control Y velocity
80
60
40
20
0
-20
1
0.5
0
1
2
3
4
5
6
7
8
9
10

II, III

It's clear that the jet doesn't need any excessive control to perform a smooth touchdown. A clear control maneuver is at the beginning. The maximum x-velocity of the jet is $350\frac{m}{s}$ and its initial velocity is only $277.7\frac{m}{s}$. In the first instant of the simulation the jet gains x-velocity reaching its max thanks to the control input:





The same controlling effect is evident in the last instant to reach a non null touchdown x-velocity.

5 Conclusion

The optimal control sequence of the vector U seems to be more natural for a jet like an F/A Hornet in the breaking phase.

It's evident that during the landing phase after about 5 steps in, the jet reaches the desired position along the x-axis, as a matter of fact velocity on the x-axis is also close to zero at that point. On y-axis the jet slowly and smoothly descends reaching the target altitude.

During the landing phase the control inputs and the trajectories of the stats vector's components are not very natural for a jet like this one but rather seem maneuvers that a VTOL (Vertical Take Off and Landing) jet or a drone would use.

Anyways, in all cases the control sequence is optimal and it is the best possible solution to the problem defined in (11) coherently with the matrices Q_L, Q, R.