Jeff’s Laboratory

NMSW05 – Flight Manager Guidance, Control, and Allocation

Comments	Revision	Date	Author
Initial Release	A	March 2, 2025	J. Mays

1. References

1. Aircraft Control Allocation, Durham
2. NMSW02 - Flight Manager Events

2. Purpose

This document describes the guidance, control, and allocation software capabilities (SWC) as illustrated in Matlab/Simulink. The original implementation was written in Matlab/Simulink and then later derived into C/C++ software for compilation onto hardware. Due to the nature of the software, the actual code is not posted in this document. Custom Simulink and C/C++ libraries were created manually to align the two languages.

3. Design Description

Below is a high-level overview of the controller system. The following sections will document the innerworkings of each main module.

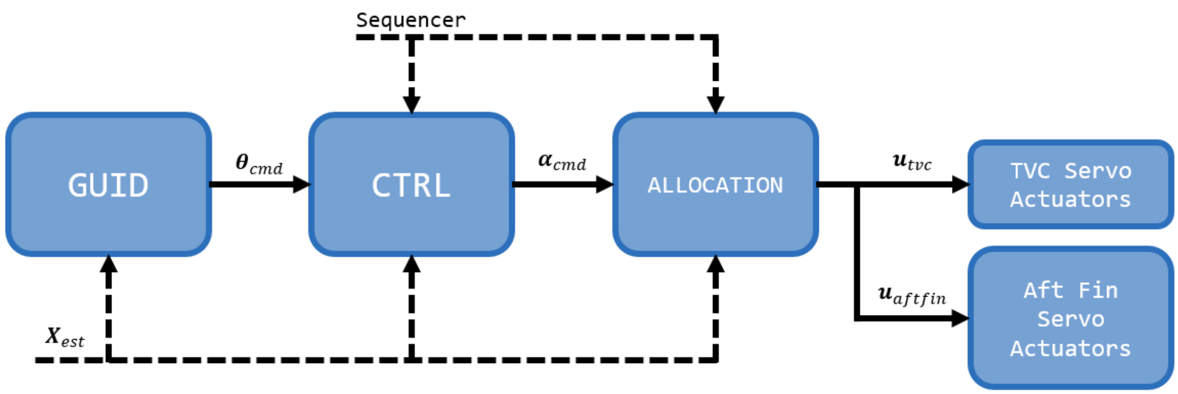

Figure 1: Guidance, Control, & Allocation Overview

3.1. Guidance

Guidance for New Mays is strategically simple to avoid raising suspicion about someone making their own "guided" rocket. While I do include "guidance" in this document, it is not the traditional guidance that might come to mind when thinking of a rocket-like vehicle. Knowing I wanted to write and make some of this code/logic public also deterred me from developing a more advanced guidance algorithm that could possibly violate some laws/regulations. Instead, I shifted my focus to the control and allocation aspect of the project.

The guidance algorithm is simply a series of lookup tables of Euler attitudes at predefined altitudes, also known as "Euler" guidance. This profile is meant to pitch the vehicle downrange to increase safety of the operators/observers of the launch. Requirement `FLT-900` describes the necessary downrange distance which is verified in Monte-Carlo simulations.

3.2. Control

The inner loop attitude controller follows a proportional-rate feedback control architecture. This can also be known as proportional-derivative (PD) control, but we are specifically using the angular velocity estimate from the gyroscope rather than taking the derivative of the estimated Euler attitude. For real applications, this works far better and creates simpler implementation. By using gain lookups based on vertical velocity, $\dot{h}$, the linear controller can apply to the entire flight phase of the system by providing the controller with the attitude, K_a, and rate, K_r, controller gains. Furthermore, by using the estimates of inertia, I, we can denote this controller specifically as an "angular acceleration" generator, where the controller's commanded effort is in radians per second squared.

$$\boldsymbol{e} = \boldsymbol{\theta}_{cmd}^{uf} - \boldsymbol{\theta}_{est}^{uf} \tag{1}$$

$$\boldsymbol{\alpha}_{cmd} = \frac{K_a(\dot{h})\boldsymbol{e} - K_r(\dot{h})\boldsymbol{\omega}_{ic}^{c}}{I} \tag{2}$$

This controller will only be useful in the ascent phases of flight, or when the flight software is in `Boost` and `Coast` sequencer segments.

3.3. Allocation

The algorithm used to mix the TVC + Aft Fins together in this project is known as Weighted Least Squares Pseudo-Inverse Control Allocation (WLS-PCA). This algorithm is attractive because it is (a) linear, (b) non-iterative, and (c) deterministic. Linearity of the WLS-PCA also allows us to use simple linear stability techniques for assessing controller stability requirements.

Let us first define a control matrix, B, such that when multiplied by a vector of control inputs, $\boldsymbol{u}$, it solves for the principal axis angular accelerations, $\boldsymbol{\alpha}$, generated on a body.

$$\boldsymbol{\alpha} = B\boldsymbol{u} \tag{3}$$

For control allocation, we need to invert the problem to solve for u given some desired angular acceleration, α_{des}, from an upstream controller. There may exist an infinite number of control vectors that could produce the desired angular acceleration depending on the problem's formulation. A simple and closed-form approach is the minimum norm pseudo-inverse, which both produces a solution and minimizes the sum of the squares of the control effector displacement, $\|u\|$. This therefore minimizes the consumed energy of all effectors. Using Lagrange multipliers and setting up the Hamiltonian yields,

$$\mathcal{H}(\boldsymbol{u}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{u}^T\boldsymbol{u} + \boldsymbol{\lambda}^T(\boldsymbol{\alpha}_{des} - B\boldsymbol{u}) \tag{4}$$

$\mathcal{H}$ will be a minimum (or maximum) when

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{u}} = 0, \qquad \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}} = 0 \tag{5}$$

Thus, providing us with

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{u}} = u^T - \boldsymbol{\lambda}^T B = 0, \qquad \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}} = \boldsymbol{\alpha}_{des} - B\boldsymbol{u} = 0 \tag{6}$$

Noting that $\boldsymbol{\alpha}_{des} = B\boldsymbol{u}$ and $u^T = B\boldsymbol{\lambda}^T$ from the above equations, we get

$$\alpha_{des} = Bu = BB^T\lambda \tag{7}$$

Since B is full rank, we can change BB^T to B^TB. With B^TB, it is square and invertible, this we can solve for the Lagrange multipliers.

$$\lambda = (B^TB)^{-1}\alpha_{des} \tag{8}$$

With $u = B^T\lambda$, we can solve for u, abd denote the variable P as the pseudo-inverse matrix.

$$u = B^T(B^TB)^{-1}\alpha_{des} = P\alpha_{des} \tag{9}$$

This completes the formulation of the minimum control vector norm pseudo-inverse.

A weight matrix, W, can be added to the above formulation to add effectiveness of some controls over the expense of others. Due to the lack of inequality constraints in the optimization problem, the weight matrix can be especially useful for real applications where effector saturation limits come into play. If the weight matrix is diagonal, $W = aI$, this control allocator is noted in literature as the weighted Moore-Penrose Pseudo-inverse [1]. Reformulating the Hamiltonian as the following

$$\mathcal{H}(\boldsymbol{u}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{u}^T W^T W \boldsymbol{u} + \boldsymbol{\lambda}^T(\boldsymbol{\alpha}_{des} - B\boldsymbol{u}) \tag{10}$$

And performing the same minimization steps, the optimization with a weight matrix yields the following optimal solution for a control vector

$$\boldsymbol{u} = W^{-1}B^T[BW^{-1}B^T]^{-1}\boldsymbol{\alpha}_{des} = P_W\boldsymbol{\alpha}_{des} \tag{11}$$

In this case, P_W denotes the now "weighted" pseudo-inverse matrix. This completes the formulation of the WLS-PCA.

3.3.1. TVC Only Allocation

In flight segments of low dynamic pressure, we want to only have TVC enabled allocation. The control matrix for the TVC can be solved by first writing the planer rigid body dynamics as,

$$M = J\alpha = Tl_{cg}\sin\delta \tag{12}$$

where J is the integrated vehicle's moment of inertia, T is the engine thrust, l_{cg} is the moment arm from the center of mass to the engine gimbal, and δ is the TVC deflection within the chosen plane. Formulating the control vector to apply to the pitch. δ_y, and yaw, δ_z, planes, we get $\boldsymbol{u}_{tvc} = [\delta_y \quad \delta_z]$. Using small angle approximation, our control matrix then becomes,

$$B_{tvc} = \begin{bmatrix} J_{yy}^{-1}Tl_{cg} & 0 \\ 0 & J_{zz}^{-1}Tl_{cg} \end{bmatrix} \tag{13}$$

resulting in

$$\begin{bmatrix} \alpha_{y_{des}} \\ \alpha_{z_{des}} \end{bmatrix} = B_{tvc}\boldsymbol{u}_{tvc} \tag{14}$$

$$\boldsymbol{u}_{tvc} = [B_{tvc}]^{-\mathbf{1}}\boldsymbol{\alpha}_{des} \tag{15}$$

Here, B_{tvc} is square and assumed invertible. Since B_{tvc} is trivially defined with only one non-zero value, we can simply use J_{ii}/Tl_{cg} to find the matrix inverse, simplifying the SW implementation.

3.3.2. TVC + Aft Fin Allocation

In times where aerodynamics are appreciable and overbearing on TVC only control, TVC and Aft Fin allocation can utilize WLS-PCA. The following table shows the relationship each fin has to the overall vehicle torque with the convention defined.

Table 1: Aft Fin Directions

Aft Fin #	Pointing Direction in FAB	FAB torque due to Positive Aft Fin Deflection
1	+Y	$-M_y$ (pitch down)
2	+Z	$-M_z$ (yaw left)
3	-Y	$+M_y$ (pitch up)
4	-Z	$+M_z$ (yaw right)

Based on the configuration of the fins, it is possible to use fins 1 & 3 for pitch axis control, and fins 2 & 4 for yaw axis control. This is known as effector ganging, where effectors are intentionally combined to serve the same purpose. Similarly, all for fins can be ganged for roll control. The control vector with both TVC and Aft Fin deflection commands would be formulated as

$$\boldsymbol{u} = [\delta_y \quad \delta_z \quad \delta_{f-1} \quad \delta_{f-2} \quad \delta_{f-3} \quad \delta_{f-4}] \tag{16}$$

where $\delta_{f-[\,]}$ corresponds to the deflection of one of the four Aft Fins. Formulating the control matrix for only the Aft Fins can be performed as follows, where $\bar{q}$ is the dynamic pressure, S and L are the aerodynamic characteristic area and length, and $C_{ll}/C_{lm}/C_{ln}$ are the fin specific xyz-axis aero moment coefficients for each fin.

$$B_{aftfin} = J^{-1}\bar{q}SL \begin{bmatrix} C_{ll_{f-1}} & C_{ll_{f-2}} & C_{ll_{f-3}} & C_{ll_{f-4}} \\ C_{lm_{f-1}} & C_{lm_{f-2}} & C_{lm_{f-3}} & C_{lm_{f-4}} \\ C_{ln_{f-1}} & C_{ln_{f-2}} & C_{ln_{f-3}} & C_{ln_{f-4}} \end{bmatrix} \tag{17}$$

Note that due to symmetry, for any given fin, we can confidently say that $C_{lm_f} \equiv C_{ln_f}$. The sign and non-zero value of each matrix entry is dependent on the pointing axis of the fin within the body frame. We can also safely ignore the inertia tensor's products of inertia. Combining the control matrices of the TVC and Aft Fins gives us,

$$B = [B_{tvc} \quad B_{aftfin}] = \begin{bmatrix} 0 & 0 & J_{xx}^{-1}\bar{q}SLC_{ll_{f-1}} & J_{xx}^{-1}\bar{q}SLC_{ll_{f-2}} & J_{xx}^{-1}\bar{q}SLC_{ll_{f-3}} & J_{xx}^{-1}\bar{q}SLC_{ll_{f-4}} \\ J_{yy}^{-1}Tl_{cg} & 0 & J_{yy}^{-1}\bar{q}SLC_{lm_{f-1}} & 0 & -J_{yy}^{-1}\bar{q}SLC_{lm_{f-3}} & 0 \\ 0 & J_{zz}^{-1}Tl_{cg} & 0 & J_{zz}^{-1}\bar{q}SLC_{ln_{f-2}} & 0 & -J_{zz}^{-1}\bar{q}SLC_{ln_{f-4}} \end{bmatrix} \tag{18}$$

This creates full control of the angular acceleration over all vehicle principal axes

$$\begin{bmatrix} \alpha_{x_{des}} \\ \alpha_{y_{des}} \\ \alpha_{z_{des}} \end{bmatrix} = B\boldsymbol{u} \tag{19}$$

The TVC and Aft Fin combined matrix would then be used to solve the weighted pseudo-inverse.

$$P_W = W^{-1}B^T[BW^{-1}B^T]^{-1} \tag{20}$$

The weight matrix

3.3.3. Weight Matrix

There exist many ways to select the weight matrix, which is usually problem dependent. Given the actuator degrees of freedom aligned with the principal axes of the vehicle, a suitable method for a diagonal weight matrix is the following documented in [3], which yields a good approximation to the maximum volume AMS.

$$W_i = \frac{a_i}{\|B_i\|} \tag{21}$$

where W_i is the i^{th} diagonal, a_i is the i^{th} actuator's total slew range, and B_i is the i^{th} column of B. This makes the weights applicable to the effector saturation limits while being normalized by relative control effectiveness. For rockets such as this one, the relative control authority on the rolling axis is typically many factors greater than the pitch/yaw axes. To account for this inequality, the computation of the weight matrix can prioritize the pitch/yaw planes by zeroing out the roll elements of B, weighting the effectors that drive roll action less-so than pitch/yaw action/ Note also that a weight matrix must be chosen such that $[BW^{-1}B^T]^{-1}$ is non-singular.

4. Software Logic

A high-level capture of the `FLIGHT_MANAGER` software logic is shown in Figure 2. The `ASCENT_GUID`, `ASCENT_CTRL`, and `ALLOCATION` software capabilities are highlighted. Note that the code below written in Simulink is NOT the actual code written for the model rocket. This was just the working model I used before re-writing all in C++ code for actual implementation (I do not own any Mathworks autocoders). This code did change slightly since transferring this code, but it is more or less a good representation of the SW that is deployed on the vehicle.

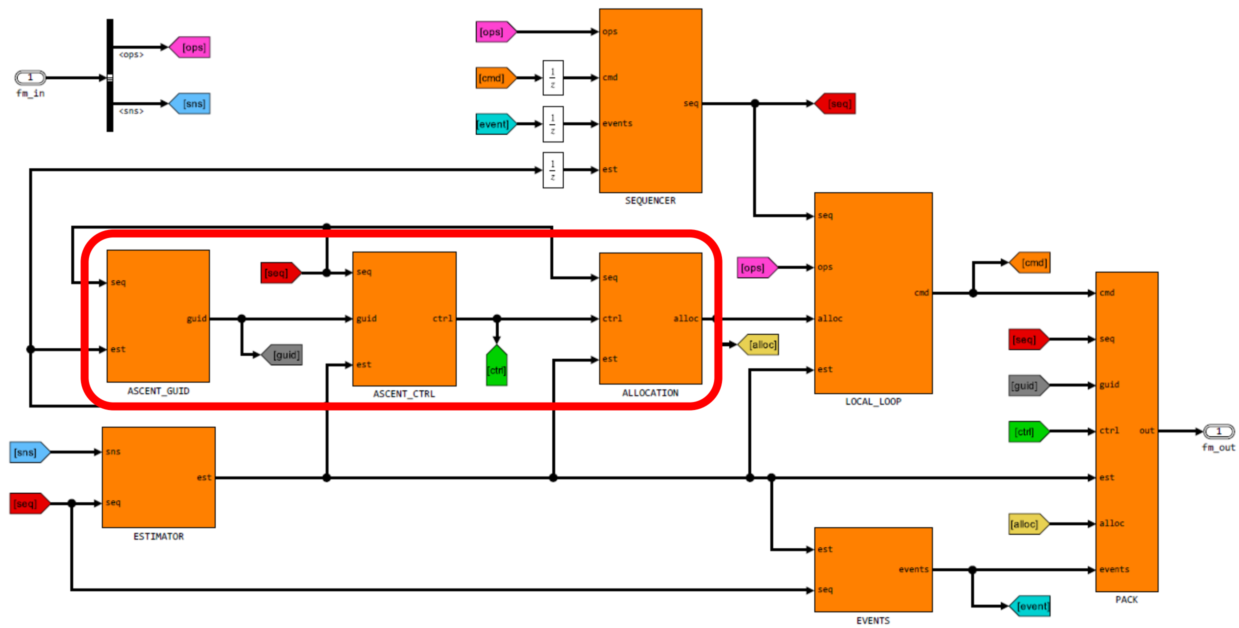

Figure 2: Flight Manager

Ascent guidance is a set of lookup tables based on altitude that set a reference attitude profile. This reference is used downstream in the control logic as the attitude controller setpoint.

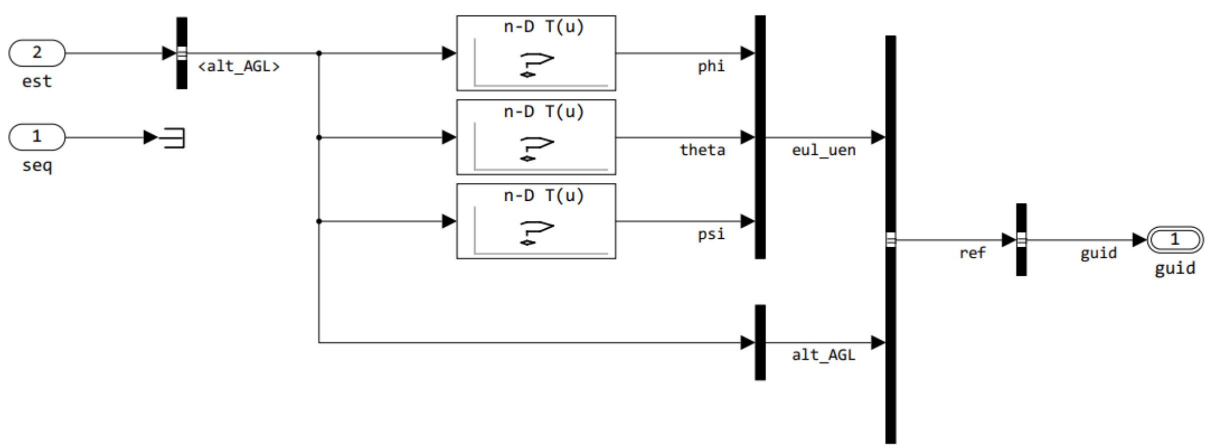

Figure 3: ASCENT_GUID

The control logic is shown below. Only during the Boost and Coast software modes can the controller generate non-zero attitude angular acceleration commands. We can also see the attitude and rate gains are a functional lookup from altitude rate, which is equivalent to vertical velocity.

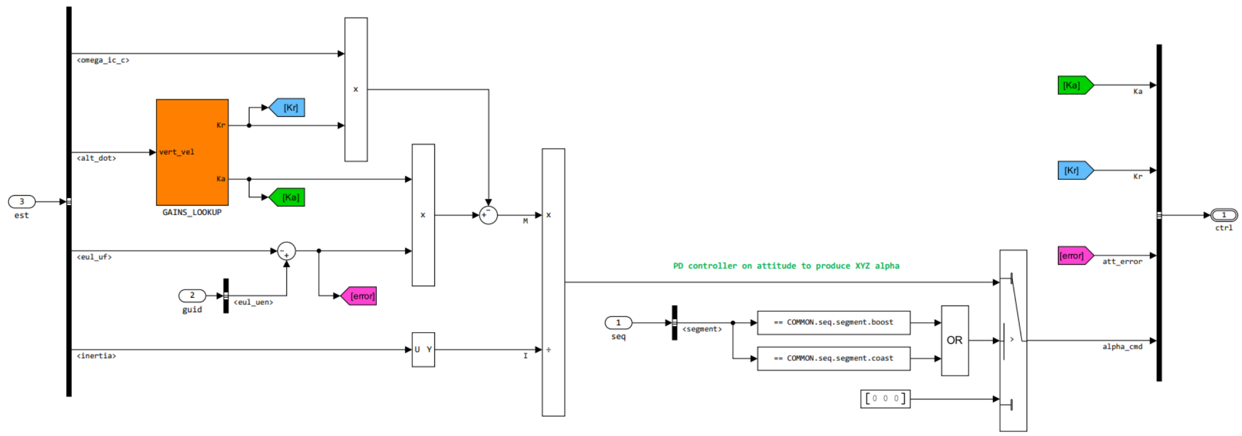

Figure 4: ASCENT_CTRL

Below is the gain-lookup logic for the attitude controller. These solve for the values used in equations 1 and 2.

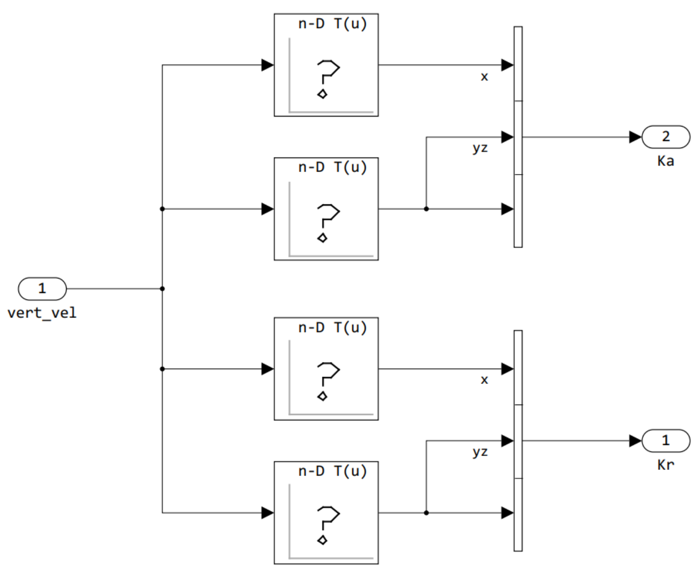

Figure 5: ASCENT_CTRL/GAINS_LOOKUP

Below is the top-level allocation logic. The `MODE` block is what enables/disables the different allocation modes and is largely a function of the sequencer mode and the dynamic pressure. As the vehicle ascends, the allocator mode logic will enable/disable the logic that pertains to the flight environment the software predicts it is in. Starting at liftoff, the allocator is in `TVC_ONLY` mode, where the aft fins are disabled but the TVC is able to vector the thrust from the SRM. This is because at this point in flight, dynamic pressure is too low to generate any appreciable aerodynamic forces from the Aft Fins. In this mode, the roll angle is open loop since the TVC can only generate moments in the pitch and yaw planes. Once aerodynamics are appreciable, the mode is switched to the `TVC_AFTFIN` mode. This mode enables both the TVC and Aft Fin effectors at the same time and utilizes the WLS-PCA logic as previously described. Depending on the control matrix, B, and weighted matrix, W, the desired angular acceleration will be divided up between the TVC and Aft Fins to generate the resultant desired torque, thus controlling the vehicle's attitude with two effectors in parallel. Once the SRM burns out, the TVC is no longer effective. Once the software senses SRM burnout through the `EVENTS` SWC[2], the mode is switched to `AFTFIN_ONLY` where the Aft Fins take full control of all vehicle axes. Once dynamic pressure is low, the mode is switched to `NULL_ALLOC` which commands all effectors to their based trim commands.

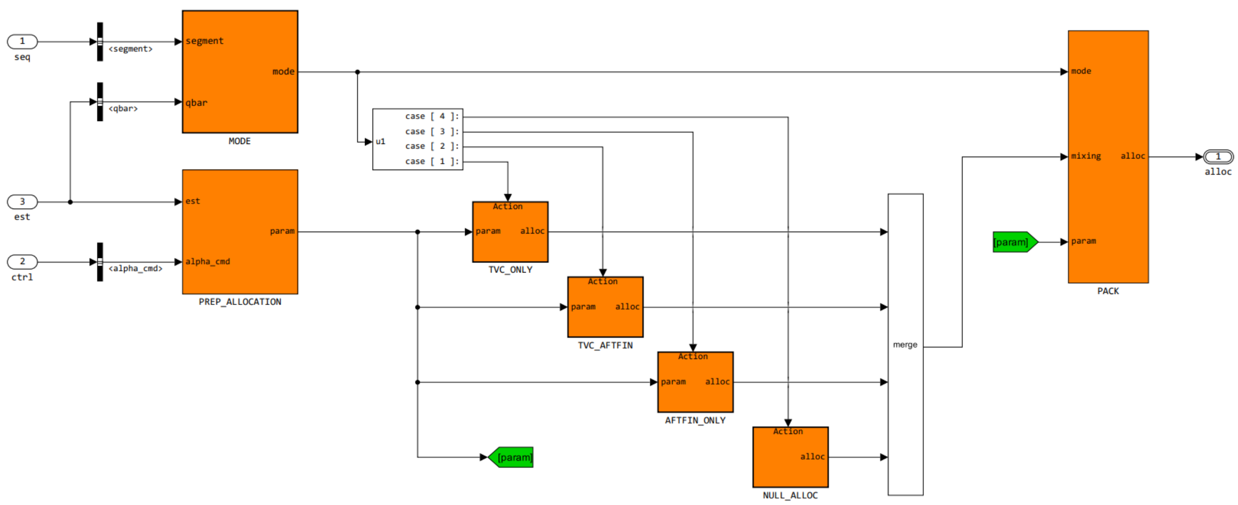

Figure 6: ALLOCATION

The `PREP_ALLOCATION` block is used to prepare the parameters required for the downstream allocation logic regardless of what allocation mode is active. This is where equations 13 and 18, for example, are solved.

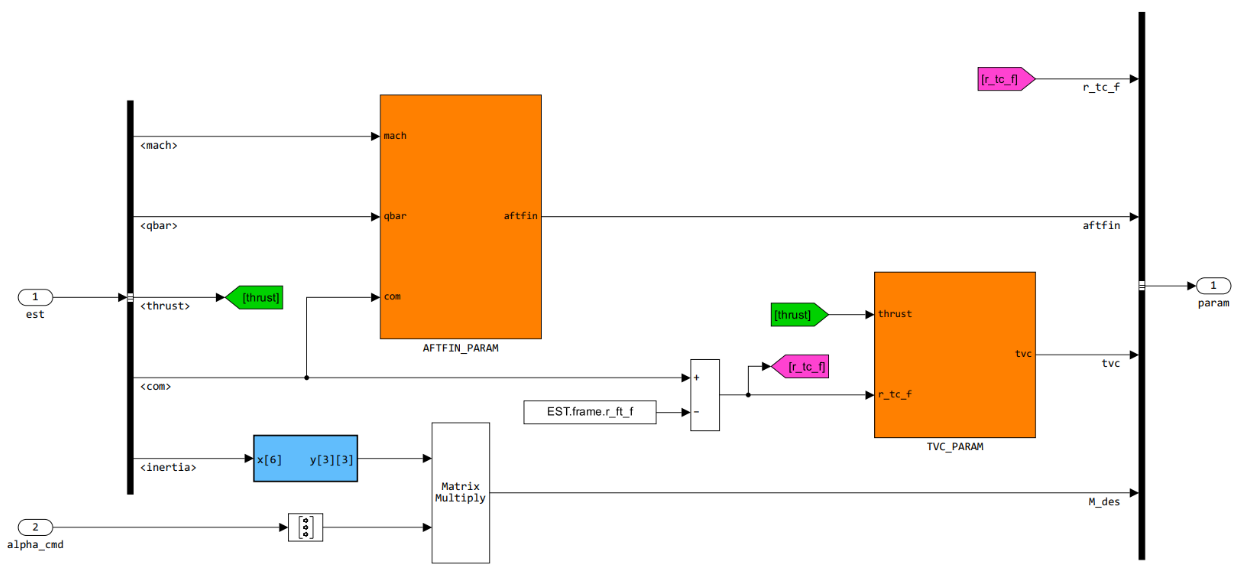

Figure 7: ALLOCATION/PREP_ALLOCATION

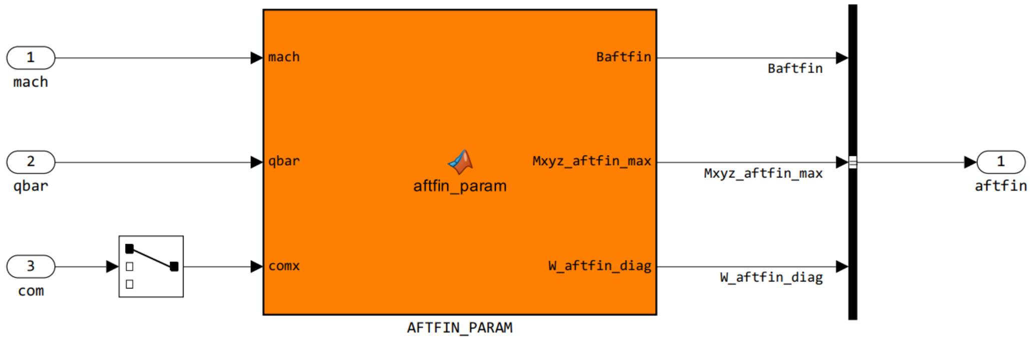

Figure 8: ALLOCATION/PREP_ALLOCATION/AFTFIN_PARAM

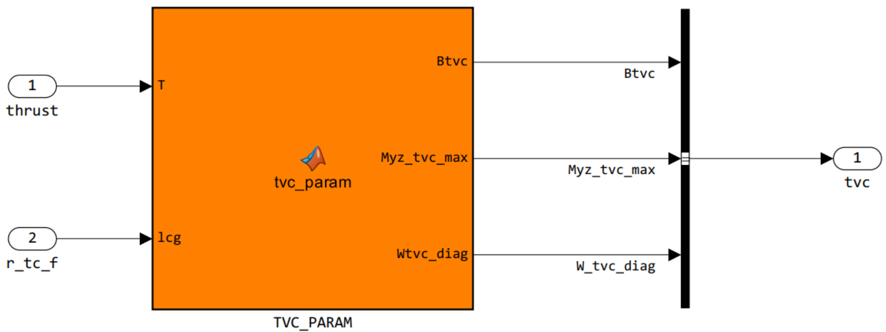

Figure 9: ALLOCATION/PREP_ALLOCATION/TVC_PARAM

The TVC_ONLY block performs elliptical limiting on the desired moment command based on the TVC constraints and generates the TVC specific control vector required to control the vehicle.

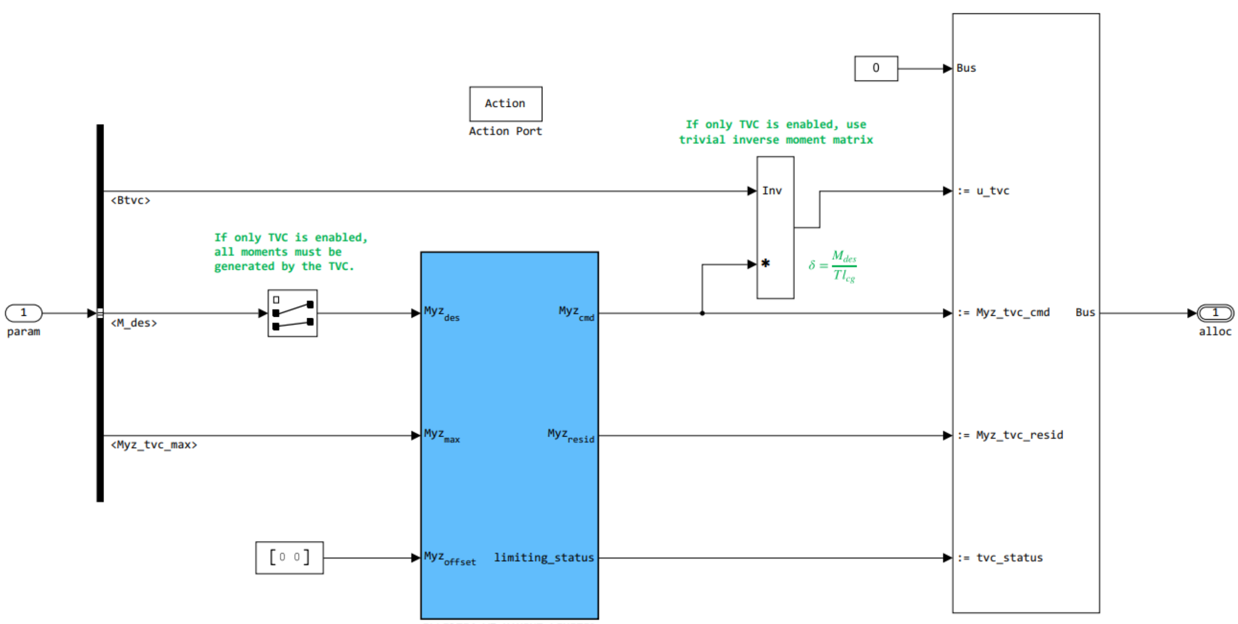

Figure 10: ALLOCATION/TVC_ONLY

The `TVC_AFTFIN` block performs the WLS-PCA algorithm and limits the TVC and Aft Fins before being added to the output bus for final sendoff to the actuator hardware.

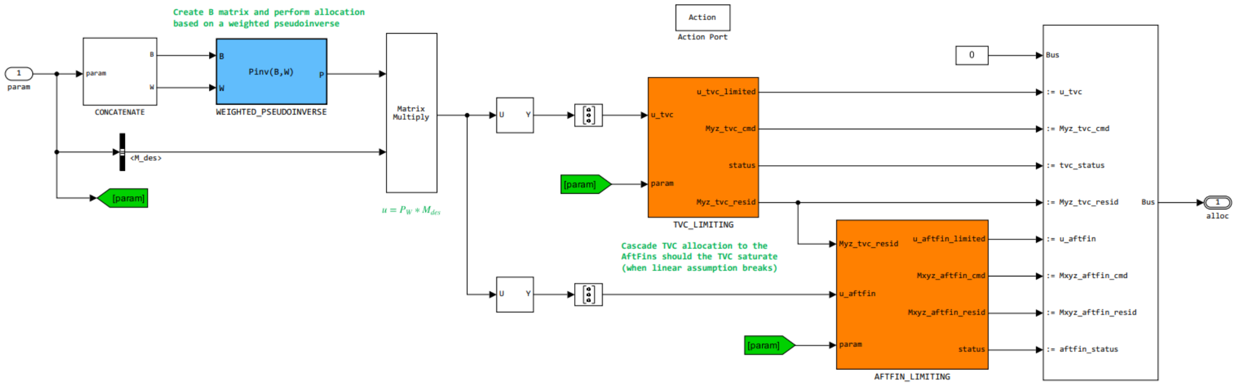

Figure 11: ALLOCATION/TVC_AFTFIN

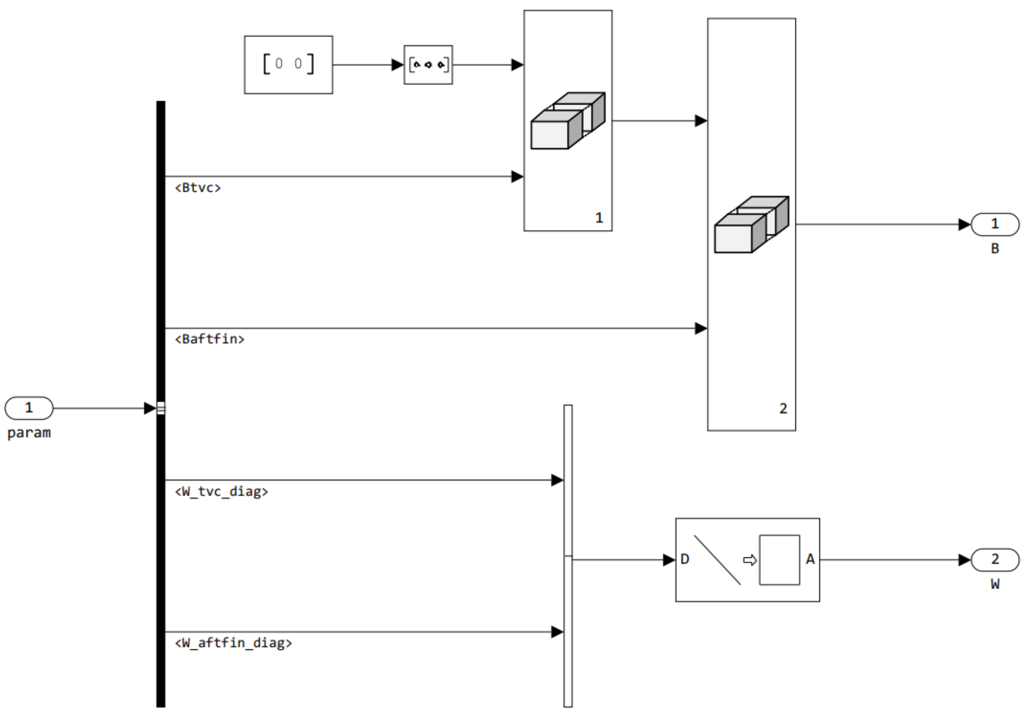

Figure 12: ALLOCATION/TVC_AFTFIN/CONCATENATE

The `AFTFIN_ONLY` limits the Aft Fin commands within their constraints and generates the necessary fin angles to generate the full moment as requested from the upstream controller.

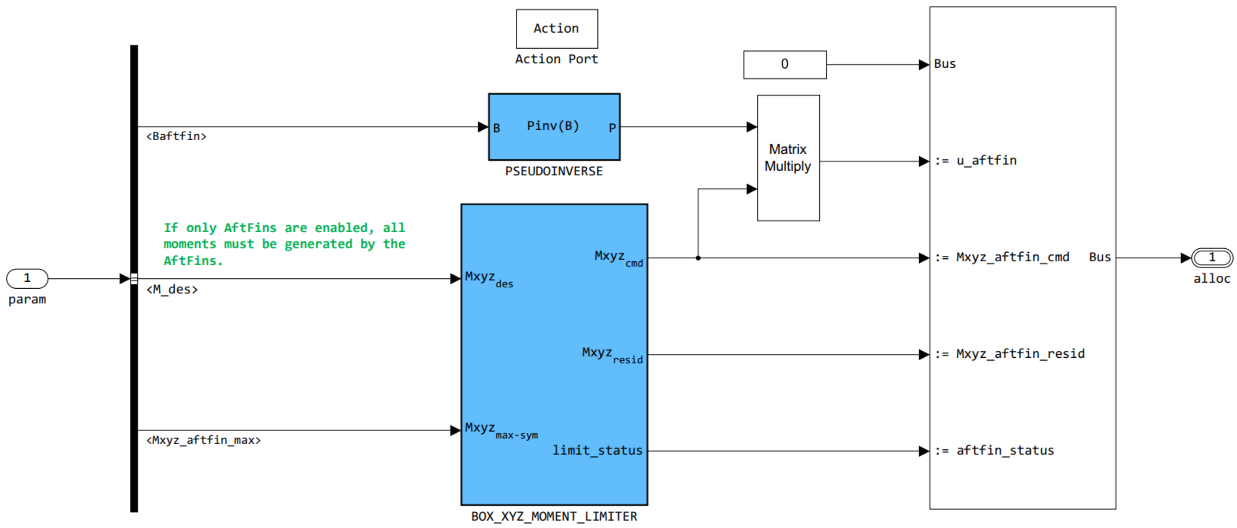

Figure 13: ALLOCATION/AFTFIN_ONLY

The `NULL_ALLOC` block simply sends the null bus to essentially set all actuators to their zero commands.

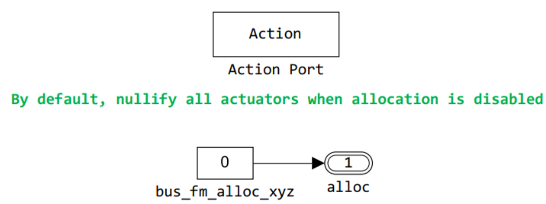

Figure 14: ALLOCATION/NULL_ALLOC