

E.T.S. de Ingeniería Industrial, Informática
y de Telecomunicación

MODELADO, SIMULACIÓN Y CONTROL DE UN QUADROTOR CON EL FIRMWARE ARDUROPTER



Máster en Ingeniería Mecánica Aplicada y
Computacional

Trabajo Fin de Máster

Alumno: Miguel Zapata Benito

Tutor: Aingeru Barrio López

Pamplona, Junio de 2021

upna

Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

RESUMEN

El objetivo final del proyecto es obtener un entorno de simulación de alta fidelidad en el cual se pueda estudiar la influencia de los parámetros del control del sistema sin riesgo de caída de la aeronave.

Para ello, en primer lugar, se realizará un modelado dinámico de un tipo concreto de quadrotor, que simulará el movimiento real que realiza el vehículo y devolverá los datos que aportarían los sensores.

A continuación, se modelará el piloto automático de código abierto ArduCopter. Para ello se estudiará el código del firmware escrito en C++ y se replicarán los lazos de control en Simulink, entorno que nos dará gran flexibilidad. El controlador consiste en una serie de PIDs en cascada que comandan posición y orientación.

Finalmente se realizará la validación del modelo desarrollado y se realizará una caracterización de la aeronave.

ABSTRACT

The final objective of the project is to obtain a high-fidelity simulation environment in which the influence of the control parameters of the system can be studied without the risk of the aircraft falling.

To do this, firstly, a dynamic modelling of a specific type of quadrotor will be carried out, which will simulate the real movement of the vehicle and return the data provided by the sensors.

Next, the open source ArduCopter autopilot will be modelled. To do this, the firmware code written in C++ will be studied and the control loops will be replicated in Simulink, an environment that will give us great flexibility. The controller consists of a series of cascaded PIDs that command position and orientation.

Finally, the developed model will be validated and a characterisation of the aircraft will be carried out.

PALABRAS CLAVE

UAV, PID, Model In the Loop, Simulink y ArduCopter.

KEYWORDS

UAV, PID, Model In the Loop, Simulink and ArduCopter.

1 CONTENIDO

2	ÍNDICE DE ILUSTRACIONES.....	3
3	INTRODUCCIÓN.....	7
4	OBJETIVOS.....	9
5	ESTADO DEL ARTE.....	9
6	EQUIPO FÍSICO.....	11
6.1	COMPONENTES PRIMARIOS.....	11
6.1.1	ESTRUCTURA.....	11
6.1.2	MOTORES.....	12
6.1.3	HÉLICES.....	13
6.1.4	VARIADORES ESC.....	14
6.1.5	PLACA DE DISTRIBUCIÓN DE POTENCIA.....	15
6.1.6	RADIO TELEMETRÍA.....	15
6.1.7	BATERÍA.....	16
6.2	CONTROLADOR DE VUELO: CUAV X7.....	16
6.3	SENSORES.....	17
6.3.1	GPS.....	17
6.3.2	ALTÍMETRO.....	18
6.4	FIRMWARE ARDUOPTER.....	19
6.4.1	CONTROL DE POSICIÓN (AC_PosControl).....	20
6.4.2	CONTROL DE ORIENTACIÓN (AC_AttitudeControl).....	21
6.4.3	LIBRERÍA DE MOTORES (AP_Motors).....	23
6.4.4	LIBRERÍA PID (AC_PID).....	23
7	MODELADO.....	23
7.1	MODELO DE LA PLANTA.....	24
7.1.1	ESTADOS DEL MODELO DE PLANTA.....	25
7.1.2	SISTEMAS DE COORDENADAS.....	26
7.1.3	PARÁMETROS DE LA PLANTA.....	29
7.1.4	EFFECTO AERODINÁMICO.....	33
7.1.5	ECUACIONES DE LA PLANTA.....	38
7.1.6	MODELO DE LA PLANTA EN SIMULINK.....	41
7.1.7	LÍMITES MUNDO REAL.....	48
7.2	MODELO DEL CONTROLADOR.....	50
7.2.1	PID CONTROLLER.....	51

8	AJUSTE CONTROLADOR.....	54
8.1	CONTROLADOR POSICIÓN Z.....	55
8.2	CONTROLADOR YAW.....	56
8.3	CONTROLADOR DE ROLL Y PITCH.....	58
8.4	AJUSTE POSICIÓN HORIZONTAL.....	61
8.5	TRAYECTORIA	65
9	VALIDACIÓN	69
10	CARACTERIZACIÓN UAV	75
10.1	ANGULOS MÁXIMOS	76
10.1.1	CASO 1.....	76
10.1.2	CASO 2.....	80
10.2	ACELERACIONES HORIZONTALES MAXIMAS.....	83
10.3	VELOCIDADES LINEALES MÁXIMAS.....	83
10.3.1	VELOCIDAD DE AVANCE	83
10.3.2	VELOCIDAD DE ASCENSO	89
10.3.3	VELOCIDAD DE DESCENSO	96
10.4	ACELERACIÓN VERTICAL MAXIMA	101
10.4.1	ACELERACION ASCENSO.....	101
10.4.2	ACELERACIÓN DESCENSO.....	104
10.5	VIENTO MÁXIMO.....	110
10.5.1	VIENTO HORIZONTAL UNIFORME	110
10.5.2	VIENTO HORIZONTAL TURBULENTO	114
10.6	MÁXIMA CARGA EXTRA.....	118
10.6.1	CARGA DE PAGO 2.3 kg.....	119
10.6.2	CARGA DE PAGO 2.4 KG	122
10.7	RESUMEN DE LOS PARÁMETROS	123
11	CONCLUSIONES	124
11.1	LINEAS FUTURAS	124
12	BIBLIOGRAFÍA.....	126

2 ÍNDICE DE ILUSTRACIONES

Ilustración 1- UAV para monitorización de paneles fotovoltaicos.....	8
Ilustración 2- UAV para agricultura de precisión.	8
Ilustración 3-Modelo en V propio del diseño basado en modelos.	11
Ilustración 4- Estructura del UAV.	12
Ilustración 5- Motores eléctricos.	12
Ilustración 6- Información de los motores del fabricante.....	13
Ilustración 7- Hélices 1045.	14
Ilustración 8- Parámetros de las hélices.....	14
Ilustración 9-Controladores ESC.....	15
Ilustración 10-Placa de distribución de potencia.	15
Ilustración 11-Radio Telemetría.	16
Ilustración 12- Batería 4S.	16
Ilustración 13- Piloto automático CUAV X7.....	17
Ilustración 14-GPS.	18
Ilustración 15-LiDAR como altímetro.	18
Ilustración 16-Flujo del proceso del control en ArduCopter.....	20
Ilustración 17- Estructura de un controlador ArduCopter.	21
Ilustración 18- Estructura de todo el sistema de control de orientación.	22
Ilustración 19-Eschema típico de un control con realimentación.	24
Ilustración 20-Sistema de coordenadas NED.	25
Ilustración 21- Sistema de coordenadas del cuerpo del sistema.....	27
Ilustración 22- Sistemas de coordenadas utilizados en el modelo.	27
Ilustración 23-Angulos de Euler.	28
Ilustración 24- Matriz de transformación de coordenadas al cuerpo.....	28
Ilustración 25-Matriz de transformación de coordenadas al sistema inercial.....	28
Ilustración 26- Esquema del sistema con los ejes de coordenadas.	29
Ilustración 27- Ensayo del empuje de los rotores.....	30
Ilustración 28- Estimación del empuje de los motores.....	31
Ilustración 29-Respuesta de un sistema de primer orden ante un escalón.....	32
Ilustración 30-Mallado en Ansys Fluent.....	33
Ilustración 31-Detalle mallado Ansys Fluent.....	34
Ilustración 32- Condiciones de contorno caso 1 Ansys Fluent.....	34
Ilustración 33-Convergencia iteraciones.....	35
Ilustración 34-Campo de presión del caso 1.	35
Ilustración 35-Campo de velocidades del caso 1.	36
Ilustración 36-Condiciones de contorno del caso 2.	36
Ilustración 37-Campo de presión del caso 2.	37
Ilustración 38-Campo de velocidad del caso 2.....	37
Ilustración 39- Distribución del empuje según el movimiento.	39
Ilustración 40-Sistema de coordenadas inercial frente a local.	41
Ilustración 41- Subsistema planta del dron.....	42
Ilustración 42- Subsistema de los rotores.....	42
Ilustración 43- Detalle del sistema Motor Subsystem.	42

Ilustración 44- Subsistema de dinámica rotacional.	43
Ilustración 45- Ecuación aceleración roll.	43
Ilustración 46- Ecuación de la aceleración de pitch.	44
Ilustración 47- Ecuación de la aceleración de yaw.	44
Ilustración 48- Subsistema de motores y dinámica rotacional.	45
Ilustración 49- Subsistema de dinámica lineal.	45
Ilustración 50- Ecuación de la aceleración lineal de X.	46
Ilustración 51- Ecuación de la aceleración lineal de Y.	46
Ilustración 52- Ecuación de la aceleración lineal en Z.	47
Ilustración 53- Subsistema de la dinámica lineal.	47
Ilustración 54- Subsistema del viento.	48
Ilustración 55- Planta del sistema.	48
Ilustración 56- Limite del suelo.	49
Ilustración 57- Limite sentido de giro de los motores.	49
Ilustración 58- Delimitar el rango de los ángulos.	50
Ilustración 59- Estructura del controlador.	50
Ilustración 60- Estructura tradicional de PID.	51
Ilustración 61- PID en Simulink.	51
Ilustración 62- Controlador de posición X.	52
Ilustración 63- Controlador de posición Y.	52
Ilustración 64- Controlador del roll.	53
Ilustración 65- Controlador de pitch.	53
Ilustración 66- Controlador de yaw.	53
Ilustración 67- Controlador de altura.	54
Ilustración 68- Subsistema Motor Mixer.	54
Ilustración 69- Respuesta ante escalón en Z.	55
Ilustración 70- RPM de los motores.	56
Ilustración 71- Respuesta ante escalón en Yaw.	57
Ilustración 72- Respuesta de Z ante escalón en Yaw.	57
Ilustración 73- RPM de los motores.	58
Ilustración 74- Respuesta ante escalón de roll.	59
Ilustración 75- Respuesta pitch ante escalón pitch.	59
Ilustración 76- Respuesta de Yaw ante escalón de roll.	60
Ilustración 77- Respuesta de Z ante escalón de roll.	60
Ilustración 78- RPM de motores.	61
Ilustración 79- Respuesta ante escalón en X.	62
Ilustración 80- Respuesta ante escalón en Y.	62
Ilustración 81- Respuesta del Roll.	63
Ilustración 82- Respuesta de Pitch.	63
Ilustración 83- Respuesta de Z.	64
Ilustración 84- RPM motores.	64
Ilustración 85- Trayectoria deseada.	65
Ilustración 86- Respuesta en X.	66
Ilustración 87- Respuesta en Y.	66
Ilustración 88- Respuesta en Z.	67
Ilustración 89- Respuesta de roll.	67
Ilustración 90- Respuesta de pitch.	68

Ilustración 91-Respuesta de Yaw.	68
Ilustración 92-Trayectoria de validación obtenida de un vuelo real.....	70
Ilustración 94-RPM motores.	71
Ilustración 95-Coordenada Z.	71
Ilustración 96-Coordenada X.....	72
Ilustración 97-Coordenada Y.....	72
Ilustración 98-Trayectoria vista en XY.....	73
Ilustración 99-Trayectoria en XYZ.	73
Ilustración 100- Roll validación.	74
Ilustración 101-Pitch validación.	74
Ilustración 102- Perturbación de ráfaga de viento.	76
Ilustración 103-RPM motores.	77
Ilustración 104-Coordenada Z.....	77
Ilustración 105- Angulo de Yaw.....	78
Ilustración 106-Angulo de Pitch.	79
Ilustración 107-Angulo de Roll.	79
Ilustración 108-Perturbación de ráfaga de viento.	80
Ilustración 109-RPM motores.	80
Ilustración 110-Coordenada Z.....	81
Ilustración 111- Angulo de Yaw.....	81
Ilustración 112-Angulo de Pitch.	82
Ilustración 113-Angulo de Roll.	82
Ilustración 114- Posición, velocidad y aceleración en X.....	84
Ilustración 115-Respuesta de coordenada X.....	84
Ilustración 116-RPM motores.	85
Ilustración 117- Coordenada Z.....	85
Ilustración 118- Angulo de Pitch.	86
Ilustración 119-Posición, velocidad y aceleración de X.....	86
Ilustración 120-respuesta de coordenada X.	87
Ilustración 121-RPM de los motores.....	87
Ilustración 122-Coordenada Z.....	88
Ilustración 123-Angulo de Pitch.	88
Ilustración 124-Posición, velocidad y aceleración en X.....	89
Ilustración 125-RPM de los motores.....	90
Ilustración 126- Respuesta de Z.....	90
Ilustración 127-Coordenada X.....	91
Ilustración 128-Coordenada Y.....	91
Ilustración 129-Angulo de Roll.	92
Ilustración 130-Angulo de Pitch.	92
Ilustración 131-Posición, velocidad y aceleración en Z.....	93
Ilustración 132-Respuesta en Z.....	93
Ilustración 133-RPM de los motores.....	94
Ilustración 134- Angulo de Roll.	94
Ilustración 135- Angulo de Pitch.	95
Ilustración 136-Posición, velocidad y aceleración en Z.....	96
Ilustración 137-RPM de los motores.....	97
Ilustración 138- Respuesta en Z.....	97
Ilustración 139- Angulo de roll.....	98

Ilustración 140- Angulo de Pitch.	98
Ilustración 141- Posición, velocidad y aceleración en Z.	99
Ilustración 142- RPM de los motores.	99
Ilustración 143- Respuesta en Z.	100
Ilustración 144- Angulo de Roll.	100
Ilustración 145- Angulo de Pitch.	101
Ilustración 146- Posición, velocidad y aceleración en Z.	102
Ilustración 147- RPM de los motores.	102
Ilustración 148- Respuesta en Z.	103
Ilustración 149- Angulo de Roll.	103
Ilustración 150- Angulo de Pitch.	104
Ilustración 151- Posición, velocidad y aceleración en Z.	105
Ilustración 152- RPM de los motores.	105
Ilustración 153- Respuesta en Z.	106
Ilustración 154- Angulo de Roll.	106
Ilustración 155- Angulo de Pitch.	107
Ilustración 156- Posición, velocidad y aceleración en Z.	107
Ilustración 157- RPM de los motores.	108
Ilustración 158- Respuesta en Z.	108
Ilustración 159- Angulo de Roll.	109
Ilustración 160 - Angulo de Pitch.	109
Ilustración 161- Coordenada X.	110
Ilustración 162- Coordenada Y.	111
Ilustración 163- RPM de los motores.	111
Ilustración 164- Coordenada X.	112
Ilustración 165- Coordenada Y.	112
Ilustración 166- RPM de los motores.	113
Ilustración 167- Perfil de viento horizontal.	114
Ilustración 168- Coordenada X.	115
Ilustración 169 - Coordenada Y.	115
Ilustración 170- RPM de los motores.	116
Ilustración 171- Perfil de viento horizontal.	116
Ilustración 172- Coordenada X.	117
Ilustración 173- Coordenada Y.	117
Ilustración 174- RPM de los motores.	118
Ilustración 175- Angulo de Roll.	119
Ilustración 176- Angulo de Pitch.	119
Ilustración 177- Coordenada Z.	120
Ilustración 178- RPM de los motores.	120
Ilustración 179- Coordenada Y.	121
Ilustración 180- Coordenada X.	121
Ilustración 181- Coordenada Z.	122
Ilustración 182- RPM de los motores.	122

3 INTRODUCCIÓN

Un *quadrotor* es un tipo de vehículo aéreo no tripulado, del inglés *Unmanned Aerial Vehicle* (UAV), que pertenecen a la familia de los UAV capaces de efectuar un despegue y aterrizaje vertical (VTOL) y que tiene cuatro rotores.

Dentro de los UAV se pueden encontrar:

- UAVs de ala fija, que se basan en un principio similar al de los aviones de pasajeros compuestos por un propulsor que proporciona el empuje delantero y unas alas, que permiten la sustentación, con una superficie controlable, para maniobrar.
- UAVs de alas rotativas, con una gran variedad de configuraciones entre las que se incluyen el helicóptero convencional con un rotor principal y otro en la cola, y los multirrotores, los cuales cuentan con multitud de configuraciones distintas entre las cuales se encuentra el *quadrotor*.

Un *quadrotor* es un tipo específico de multirrotor que cuenta con cuatro rotores equidistantes del centro de masas del vehículo y los cuales suelen estar dispuestos en forma cuadrada. La velocidad de los rotores se manipula para realizar diferentes maniobras, como el vuelo estacionario, el despegue y el aterrizaje.

Hasta hace poco tiempo, controlar un vehículo aéreo de manera autónoma era una tarea complicada debido a las limitaciones impuestas por el hardware hasta entonces existente. Lo que hizo posible la construcción de robots aéreos autónomos fue los recientes avances tecnológicos en actuadores y sensores en escala reducida, así como en el almacenamiento de energía y en el procesamiento de datos. Además, el diseño de controladores para este tipo de vehículos es complejo, debido principalmente a la dinámica presente en los sistemas aerodinámicos, los cuales son multivariantes, subactuados y además presentan diversas características no lineales. Esto significa que las leyes clásicas de control lineales y monovariantes pueden ver reducida su utilidad cuando se opera en condiciones no muy lejanas a las de equilibrio, ya que pueden aparecer inestabilidades.

Hoy en día, los drones se utilizan en varias áreas para diversos fines. A continuación, se indican algunas de las áreas más comunes de las aplicaciones de los drones:

- Fotografía aérea
- Búsqueda y rescate
- Agricultura
- Logística
- Inspección y mantenimiento
- Mapeo 3D
- Investigación y ciencia
- Vigilancia aérea
- Uso militar



Ilustración 1- UAV para monitorización de paneles fotovoltaicos.



Ilustración 2- UAV para agricultura de precisión.

4 OBJETIVOS

El objetivo principal del proyecto es el de desarrollar un modelo de multirrotor de alta fidelidad el cual pueda ser empleado para tareas de validación y ensayo en el programa Simulink. Este objetivo principal engloba varios subobjetivos los cuales se han ido cumpliendo a medida que avanzaba el proyecto y son:

1. Comprender las ecuaciones dinámicas y el comportamiento de los multirrotos mediante el modelado de la planta del dron.
2. Identificación de los parámetros físicos característicos del modelo.
3. Replicar la arquitectura del sistema de control del firmware de código abierto ArduPilot, el cual se utiliza para realizar misiones autónomas con el dron real.
4. Realizar el ajuste de los controladores PIDs tanto para la orientación como para la posición del dron.
5. Simular distintas situaciones para estudiar la respuesta del sistema.
6. Validar el modelo desarrollado comparándolo con datos de vuelo de una misión real.
7. Caracterización de UAV.

5 ESTADO DEL ARTE

En el marco de la ingeniería de sistemas, suele desarrollarse un plan inicial de verificación y validación del sistema al principio del proceso de ingeniería. Es importante diferenciar entre verificación y validación. Ambas actividades evalúan cómo se ha construido un sistema, pero las principales diferencias son las siguientes:

- **Verificación:** La verificación está relacionada con el conjunto de requisitos aprobados y puede realizarse en distintas fases del ciclo de vida del producto. Las pruebas de verificación son las pruebas oficiales "para que conste" realizadas en un sistema o elemento para demostrar que cumple los requisitos o especificaciones asignados, incluidas las interfaces físicas y funcionales. Las pruebas de verificación utilizan instrumentación y mediciones y, por lo general, son realizadas por ingenieros, técnicos o personal de pruebas del operador-mantenedor en un entorno controlado para facilitar el análisis de los fallos.
- **Validación:** pruebas de validación se llevan a cabo en condiciones realistas, pueden ser simuladas, en cualquier producto final para determinar la eficacia y la idoneidad del producto para su uso en operaciones de misión por parte de usuarios típicos y para evaluar los resultados de dichas pruebas.

Métodos de verificación/validación: Los procesos de verificación/validación necesarios pueden realizarse de forma recursiva a lo largo de todo el proyecto y sobre una amplia variedad de aspectos, mediante diferentes tipos de métodos que abarcan el análisis, la demostración, la inspección y el ensayo. Se podrían considerar algunos métodos prometedores:

- **Test Laboratorio:** pueden establecer y utilizar pruebas de laboratorio para albergar el entorno de prueba inicial, con la participación de varios clientes y aplicaciones para proporcionar capacidades de co-simulación de vuelo simulado y en vivo. Por ejemplo, podrían desarrollarse componentes de simulación multifacéticos para apoyar las pruebas de vuelo en vivo a corto plazo y además la exploración de conceptos a largo plazo.

- **Software-In-the-Loop:** representa la integración de código compilado en una simulación de modelo matemático, y proporciona un entorno de simulación práctico y virtual para el desarrollo y las pruebas. Consiste en reemplazar el modelo de la estrategia de control por un archivo compilado en C. El propósito de las pruebas SIL es verificar que el código generado automáticamente y el modelo utilizado para la generación de código sean coherentes en su comportamiento.
- **Processor-In-the-Loop:** La diferencia entre PIL y SIL es que SIL ejecuta el código generado en la plataforma Windows a través de una DLL, mientras que PIL ejecuta el código generado en el procesador de destino. Los compiladores utilizados en los dos modos también son diferentes. SIL usa compiladores en Windows, como Visual Studio C++ o compiladores LCC, mientras que PIL usa compiladores de destino.
Como se mencionó anteriormente, el propósito de las pruebas de equivalencia es evitar errores en el proceso de generación de código. De manera similar, pueden ocurrir errores en el proceso de compilación y el compilador puede tener errores. Los resultados de las pruebas de modelos integrales, las pruebas de SIL y las pruebas de PIL también pueden ayudarnos a descubrir los errores del compilador. La prueba PIL también puede obtener el tiempo de ejecución del algoritmo en el controlador real.
- **Hardware-In-the-Loop:** puede utilizarse en el desarrollo y las pruebas de sistemas integrados complejos en tiempo real, que proporciona una plataforma eficaz al añadir la complejidad de las entradas y salidas del mundo real bajo control a la plataforma de pruebas. HIL se usa para probar el sistema del controlador. El sistema del controlador incluye hardware, software subyacente y software de capa de aplicación. La prueba HIL generalmente conecta un controlador bajo prueba con una computadora industrial, y el modelo del objeto controlado se ejecuta en la computadora industrial, y se simulan algunas características del objeto controlado. En el escenario HIL, el controlador probado no sabe que está conectado a una computadora industrial y pensará que está conectado a un objeto real.
- **Model-In-the-Loop:** permite modelar todo el sistema, lo que hace posible simular un entorno completo para probar el concepto general, las capacidades o las estrategias antes de la validación funcional. En este tipo de prueba generalmente el ingeniero de sistemas usa el modelo de algoritmo de control para controlar el modelo de objeto controlado con el fin de verificar el algoritmo.
- **La demostración en vuelo:** puede utilizarse para examinar la aplicabilidad de diferentes conceptos, tecnologías y arquitecturas, para promover la cooperación de las partes interesadas y para identificar las necesidades, los problemas y los conceptos erróneos que deben abordarse antes de que el sistema UAV desarrollado se realice con éxito.

Las diferencias entre los objetivos que buscan son los siguientes: [2]

1. MIL se utiliza para medir algoritmos
2. SIL y PIL se utilizan para probar el código (coherencia con el modelo)
3. HIL se utiliza para probar el sistema del controlador

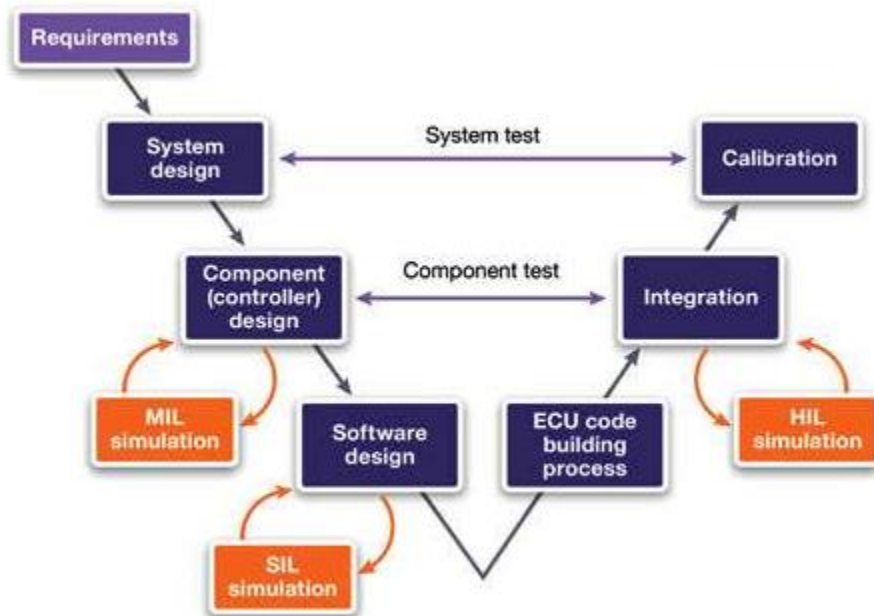


Ilustración 3-Modelo en V propio del diseño basado en modelos.

6 EQUIPO FÍSICO

En este apartado se hará una descripción del Firmware y el Hardware de control, de los sensores que se han empleado y de los componentes del `quadrotor`. Además, se explicará un poco el funcionamiento del Firmware de ArduCopter, de código abierto y el cual está escrito en C++.

6.1 COMPONENTES PRIMARIOS

6.1.1 ESTRUCTURA

Se trata de la estructura del vehículo. Está formado por cuatro brazos, dos patas de apoyo y dos placas centrales. En los extremos de los brazos se colocan los motores con sus respectivas hélices, mientras que las placas centrales se utilizan para fijar el controlador de vuelo, la batería y los distintos sensores.



Ilustración 4- Estructura del UAV.

6.1.2 MOTORES

Los motores empleados son de la marca Holybro, más en concreto el modelo Brushless Motor 2216-880KV.



Ilustración 5- Motores eléctricos.

A continuación, se muestra una serie de datos sobre el rendimiento del motor ofrecida por el fabricante. Estos datos que ofrece el fabricante se supondrán como valores teóricos de los cuales se pueden obtener parámetros característicos para modelar los motores. Aun así, para tener mayor precisión en el modelo se hará un ensayo con los motores integrados en el sistema para ver el empuje real que se obtiene.

Item No.	Propeller	Throttle	Voltage (V)	Torque (N*m)	Thrust (g)	Current (A)	RPM	Input power (W)	Efficiency (g/W)	Operating Temperature
AIR2216 KV880	T1045	50%	16	0.07	435	3.5	6015	56	7.77	53.5°C
		55%	16	0.08	527	4.6	6620	73.6	7.16	
		60%	16	0.09	608	5.6	7113	89.6	6.79	
		65%	16	0.11	702	6.8	7563	108.8	6.45	
		75%	16	0.13	888	9.5	8545	152	5.84	
		85%	16	0.15	1076	12.3	9442	196.8	5.47	
		100%	16	0.18	1293	16.2	10464	259.2	4.99	

Notes: Motor temperature is motor surface temperature @100% throttle running 10 mins.
 (Data above based on benchtest of 2018 are for reference only. Comparison with that of other motor types is not recommended.)

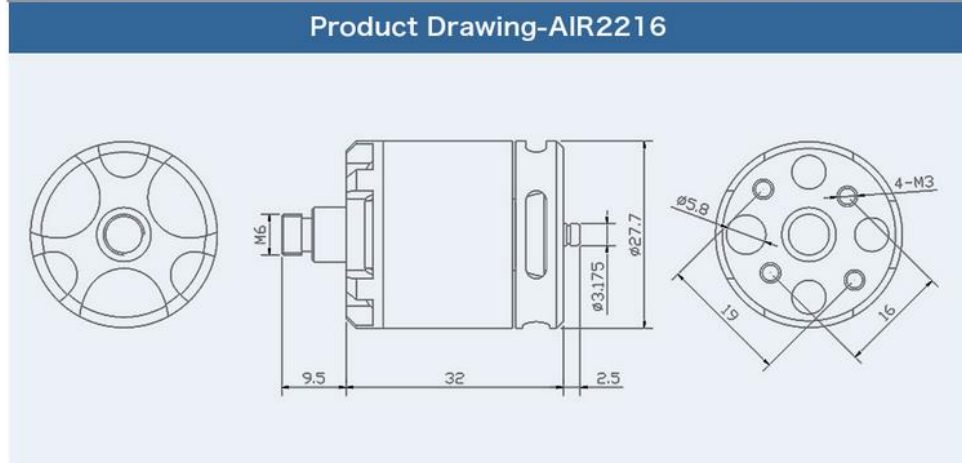


Ilustración 6- Información de los motores del fabricante.

6.1.3 HÉLICES

El dron utiliza cuatro hélices modelo 1045, dos giran en sentido horario y las otras giran en sentido antihorario. La numeración de las hélices es un dato muy importante a la hora de escoger unas hélices adecuadas para el sistema. El primer número denota el diámetro de la hélice de punta a punta, mientras que el segundo número hace referencia al pitch de la hélice, pero en este caso se denota como una longitud y no como un ángulo. Ambas medidas están expresadas en pulgadas.



Ilustración 7- Hélices 1045.

Por ejemplo, en este caso se tiene un modelo 1045 que significa:

- Un diámetro de hélice de 10 pulgadas (25,4 cm).
- Una distancia de pitch de 4,5 pulgadas (11,43 cm). Esto significa que, en condiciones ideales, una vuelta de la hélice provocara un desplazamiento de 4,5 pulgadas.

En la siguiente imagen se puede ver claramente que significan estos dos parámetros.

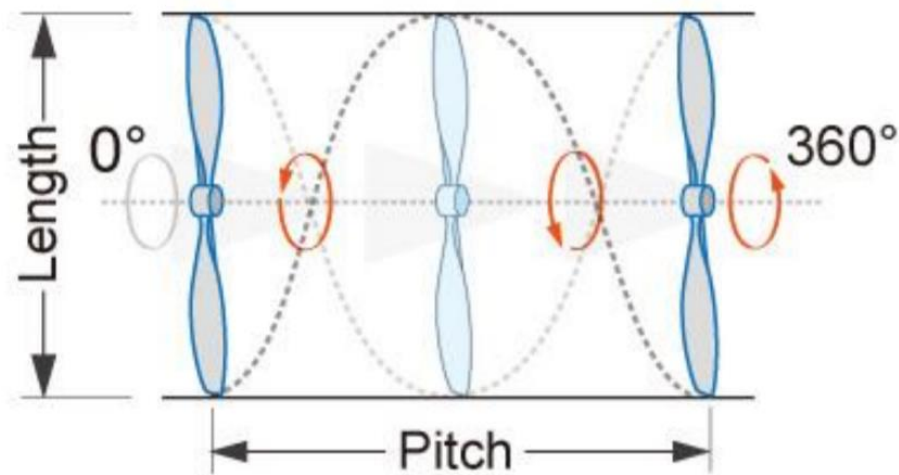


Ilustración 8- Parámetros de las hélices.

6.1.4 VARIADORES ESC

Los variadores (*Electronic Speed Controller*) se conectan a los motores y permiten controlar la velocidad de giro de estos. Cada variador va conectado a la batería para su alimentación, y además va conectado al receptor que obtiene la señal del piloto.



Ilustración 9-Controladores ESC.

6.1.5 PLACA DE DISTRIBUCIÓN DE POTENCIA

Su función es repartir la energía de una misma batería a los distintos variadores y motores, además del controlador.

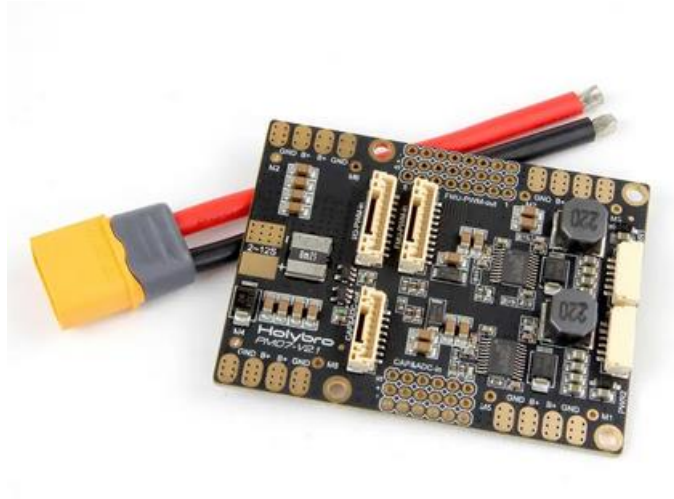


Ilustración 10-Placa de distribución de potencia.

6.1.6 RADIO TELEMETRÍA

Sirve para comunicar la estación de control con el dron. Esta conexión permite comandar al dron, recibir los datos de los sensores y además permite variar el valor de los parámetros del sistema en mitad del vuelo.



Ilustración 11-Radio Telemetría.

6.1.7 BATERÍA

Se ha adquirido una batería de tipo LiPo de la marca Zop Power y sus especificaciones son:

- Capacidad: 5000 mAh
- Voltaje: 14.8 V
- Nº de celdas: 4 celdas
- Tasa de descarga: 60 C



Ilustración 12- Batería 4S.

6.2 CONTROLADOR DE VUELO: CUAV X7

El CUAV X7 es un sistema de piloto automático, es decir, un ordenador de a bordo para el multirroto. De forma general está compuesto por los siguientes componentes:

- Procesador: se encarga de realizar los cálculos para procurar un vuelo estable.
- Giróscopo permite tomar medidas de velocidades angulares.

- Acelerómetro: permite tomar medidas de aceleración.
- Magnetómetro: se trata de una brújula.
- Barómetro: permite medir la altura de vuelo en la que se encuentra el multirroto.

En este componente es donde se ejecuta el firmware de control ArduPilot. Existen distintos tipos de software de control para distintos sistemas, en este caso se ha utilizado el firmware ArduCopter del cual se hablará más adelante.



Ilustración 13- Piloto automático CUAV X7.

6.3 SENSORES

Con los sensores que incorpora el controlador se podrían realizar operaciones dirigidas de forma manual por un piloto, pero el interés de este proyecto reside en demostrar la capacidad del sistema para realizar operaciones de forma autónoma. Es por eso por lo que se incorporan dos sensores adicionales para procurar que la navegación autónoma del sistema sea posible.

6.3.1 GPS

El sensor GPS es necesario si se quieren realizar trayectorias de forma autónoma. El GPS proporciona las coordenadas en las cuales se encuentra el dron, mientras que la brújula facilitará la dirección que se está encarando.



Ilustración 14-GPS.

6.3.2 ALTÍMETRO

Uno de los problemas que se observaron durante los primeros ensayos con el sistema es que la precisión de la posición en Z, es decir la altura, no era demasiado buena. Esta medida dependía del barómetro, el cual mide presión y a partir de ahí se calcula la altura a la cual se encuentra el sistema. Para mejorar la precisión en altura se ha decidido incorporar un LiDAR de un solo haz el cual se orientará mirando al suelo.

De esta forma se solucionan los problemas de pérdidas de altura en mitad de una misión debido a una ráfaga de aire, un cambio de temperatura en el ambiente, etc.



Ilustración 15-LiDAR como altímetro.

6.4 FIRMWARE ARDUROPTER

El código ArduCopter es el firmware desarrollado por la comunidad del proyecto ArduPilot que permite el vuelo y monitorización de varios tipos de vehículos que basan su funcionamiento, generalmente, en motores eléctricos con hélices que ejercen la fuerza actuadora.

La base de código de ArduPilot es muy extensa, alrededor de 700.000 líneas, y puede ser bastante intimidante al principio. Hay que remarcar que el código está escrito en su totalidad en C++. En este sentido, el alumno ha tenido que buscar personalmente conceptos pertenecientes a este lenguaje para entender el firmware lo suficiente para usarlo a su favor.

La última versión estable del código ArduCopter a estudiar se puede descargar íntegramente desde el repositorio oficial del Proyecto ArduPilot.[1]

De forma general el funcionamiento del firmware cuando opera de forma autónoma sería la siguiente:[3]

1. Se determina el estado actual del vehículo.
2. Se llama al módulo de navegación, el cual se encarga de calcular la trayectoria que el sistema debe seguir y a actualizar la posición deseada que entrara en los controladores de posición.
3. Se calculan los errores de posición, velocidad y aceleración y se calculan los ángulos de Pitch y Roll necesarios para satisfacer la posición deseada.
4. Se calculan los errores de orientación y de velocidad angular y se convierten en entradas para los motores del sistema.
5. Se convierten las entradas de los motores del sistema en salidas individuales de los motores.
6. Finalmente se escalan esas salidas y se envían las señales PWM a los variadores ESC.

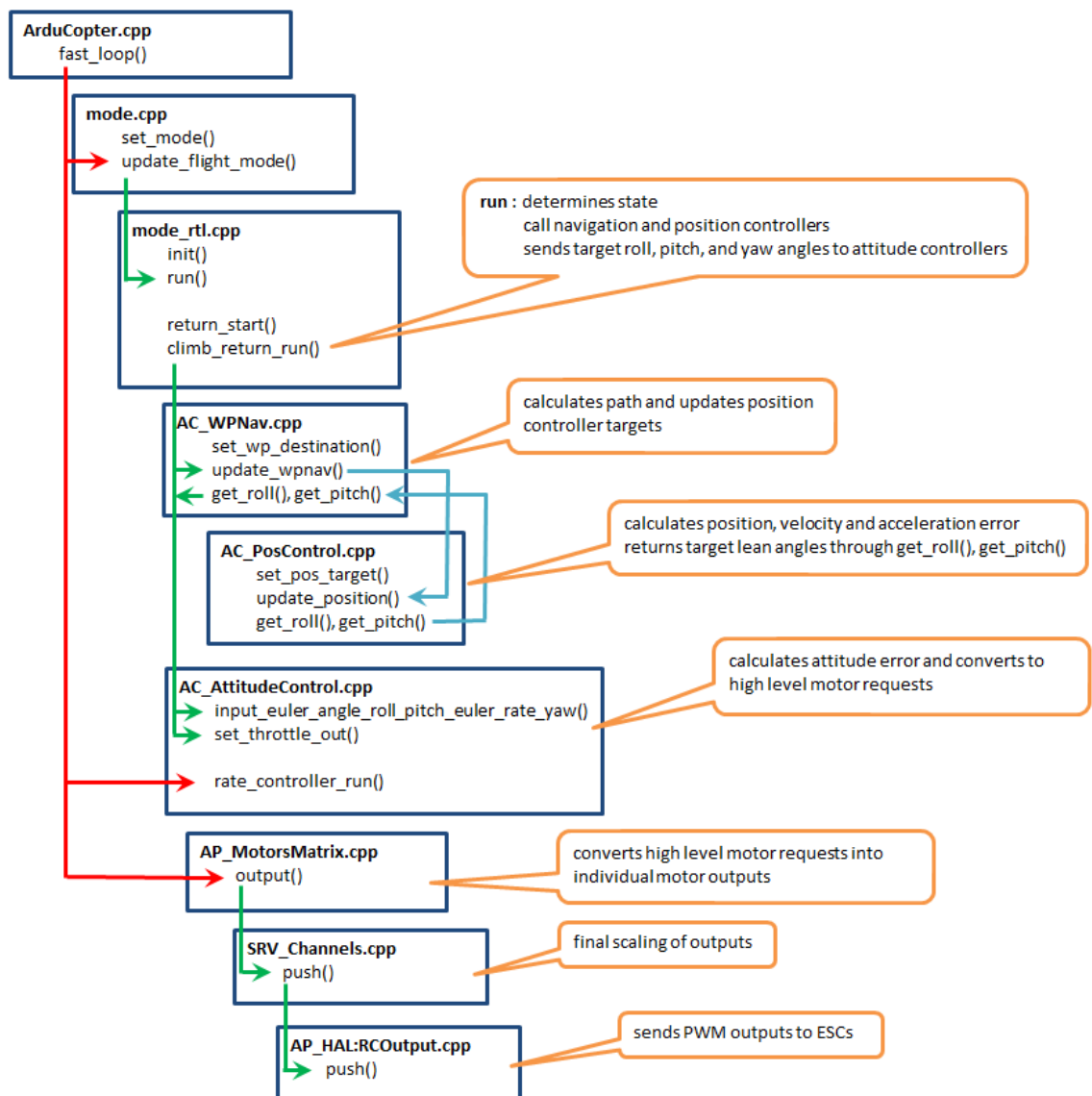


Ilustración 16-Flujo del proceso del control en ArduCopter.

6.4.1 CONTROL DE POSICIÓN (AC_PosControl)

Se utilizan interfaces separadas para el control horizontal (ejes X e Y) y vertical (eje Z). Se utilizan controladores PID por capas.

El eje XY utiliza un P de posición para convertir el error de posición en una velocidad objetivo. Un PID de velocidad convierte el error de velocidad en una aceleración deseada que luego se convierte en un ángulo de inclinación deseado que luego se envía a la biblioteca de control de orientación.

El eje Z utiliza un controlador de posición P para convertir el error de posición en una velocidad vertical objetivo (también conocida como tasa de ascenso). Un controlador de velocidad P convierte el error de velocidad en una aceleración deseada. Un PID de aceleración convierte el error de aceleración en la entrada para el comando de empuje que se enviara a los motores.

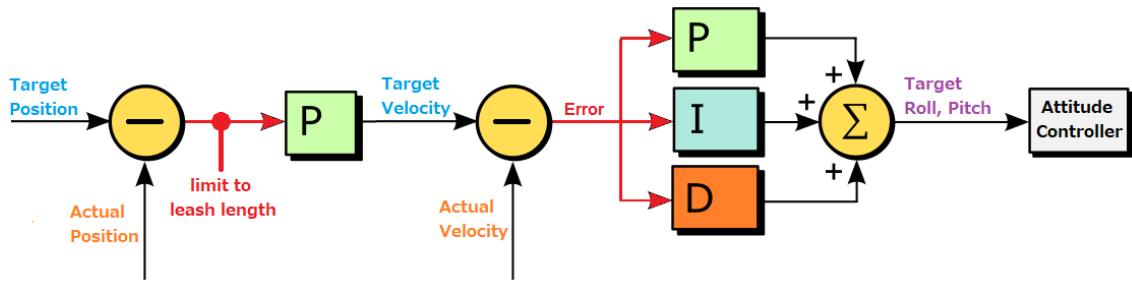


Ilustración 17- Estructura de un controlador ArduCopter.

6.4.2 CONTROL DE ORIENTACIÓN (AC_AttitudeControl)

La arquitectura del control de orientación es muy similar al control de posición que se acaba de explicar. Para cada giro, un controlador P convierte el error en el ángulo de orientación en una velocidad de rotación deseada la cual se compara con la velocidad de rotación actual y el error calculado se corrige mediante el uso de un control PID, cuya salida servirá para alimentar las señales de los motores.

ArduCopter V2.9 STABILIZE Roll, Pitch & Yaw PID's

100 Hz Update Rate

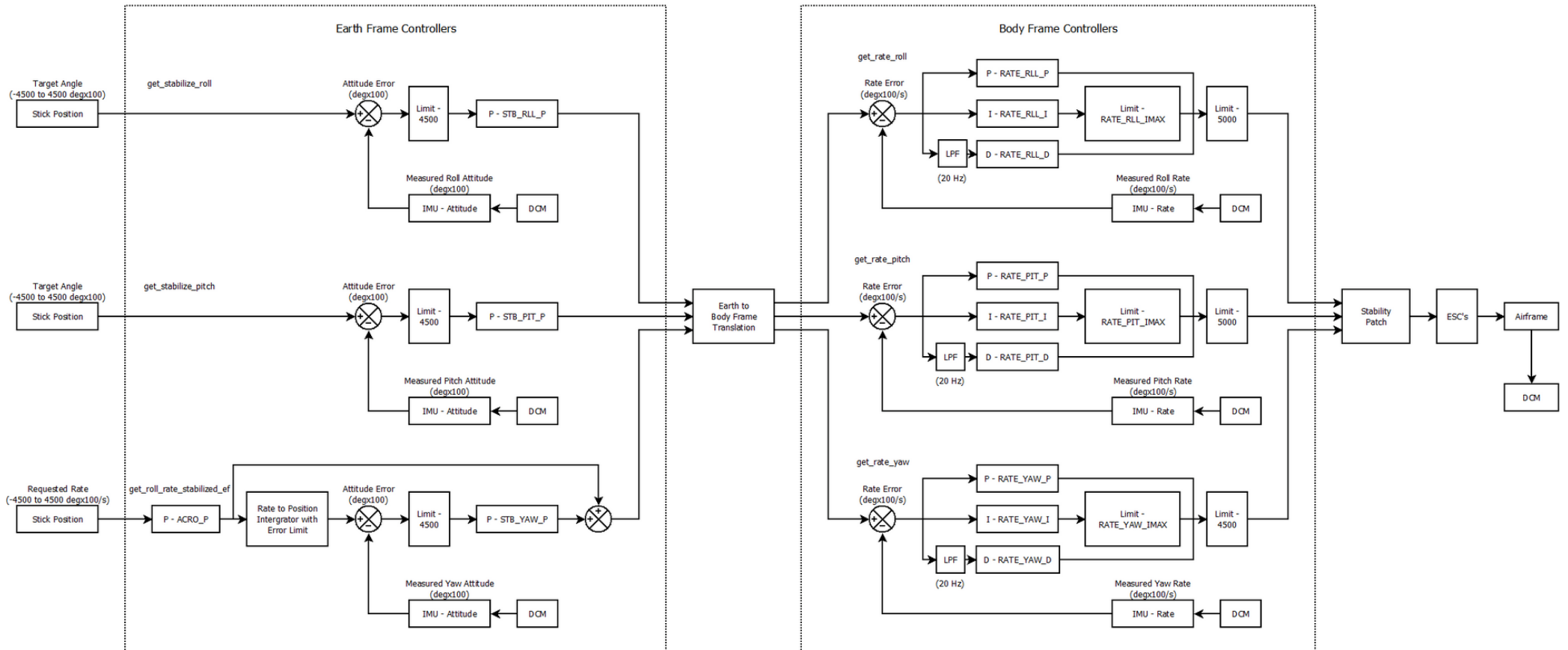


Ilustración 18- Estructura de todo el sistema de control de orientación.

6.4.3 LIBRERÍA DE MOTORES (AP_Motors)

ArduPilot soporta más de 22 estructuras diferentes de multirrotores. La librería Motors proporciona una interfaz común para que todos los vehículos puedan ser controlados de la misma manera sin necesidad de un manejo especial en el código de nivel superior.

Las entradas a la librería provienen del control de orientación, el cual comanda un valor entre -1 y 1 para Roll, Pitch y Yaw, y del control de posición vertical el cual comanda un valor entre 0 y 1 para el empuje.

La librería de motores en encarga de convertir estas entradas de alto nivel que llegan desde los controladores en salidas deseadas para cada motor, es decir se encarga de comandar la velocidad a la cual debe girar cada motor para cumplir que:

- Si la entrada de Roll es +1.0 se debería comandar una orientación igual al máximo ángulo permitido de Roll y el sistema se debería trasladar hacia la derecha.
- Si la entrada de Pitch es +1.0 se debería comandar una orientación igual al máximo ángulo permitido de Pitch y el sistema se debería trasladar hacia atrás.
- Si la entrada de Yaw es +1.0 se debería comandar una velocidad de giro máxima en Yaw en sentido horario.
- Si la entrada del empuje es +1.0 se debería comandar la aceleración, máxima de traslación vertical permitida.

6.4.4 LIBRERÍA PID (AC_PID)

Librerías que incluyen las funciones que definen los algoritmos de los controladores proporcionales P e integrales y derivativos PID, y las particularizaciones para el helicóptero. Si se desea cambiar la arquitectura de los controladores se debería cambiar esta parte del código.

7 MODELADO

El programa en el cual se ha realizado el modelo de la planta del UAV y del controlador es MATLAB/Simulink. Simulink es una herramienta de MATLAB que permite construir modelos usando diagramas de bloques. Cuando se realiza un modelo de UAV, se utiliza un script de MATLAB para almacenar todos los datos de la simulación que interesen y más tarde poder representarlos y analizarlos. Además, MATLAB/Simulink tiene la habilidad de convertir modelos de Simulink en C o C++. Esto es ideal debido a que la mayoría de los controladores de vuelo usan lenguaje C o C++ para el código del sistema de control.

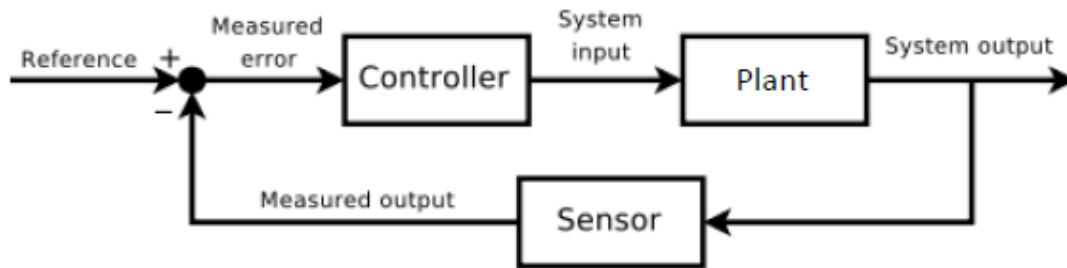


Ilustración 19-Esquema típico de un control con realimentación.

La planta del modelo es una función de transferencia de un sistema físico o un conjunto de ecuaciones que describen un sistema físico. Si se desea realizar una simulación y diseñar un sistema de control será necesario disponer de una planta del modelo. Con una planta del modelo, el sistema de control se puede ajustar y testear de forma rápida. La planta del modelo toma las velocidades angulares de los motores como entradas del sistema y devuelve el estado del UAV como salidas del sistema. En este trabajo se han utilizado una serie de ecuaciones que describen la dinámica del UAV, pero hay que decir que, aunque la física del sistema no cambie, existen distintas formas de modelar matemáticamente ese comportamiento.

Un controlador suele ser un sistema que coge el error medido de una señal, le aplica unas ganancias y su salida sirve de entrada al sistema del modelo de planta para que esta produzca una respuesta deseada. Existen muchos métodos distintos de control, pero en este trabajo se han utilizado controladores PID (Proporcional, Integral y Derivativo). Estos controladores son muy utilizados y son bastante simples de entender en comparación con otros métodos de control. El objetivo de estos controladores es básicamente el de corregir el error en la señal el máximo posible. En el caso de los controladores PID esto se realiza multiplicando el error en la señal por una ganancia (parte Proporcional), generando una ganancia a través del sumatorio del error (parte Integral), y creando una ganancia que sigue la pendiente del error de la señal (parte Derivativa).

El esquema general del control tiene una disposición tal que el controlador alimenta la planta y se realiza una realimentación de la salida de la planta para obtener las medidas de error de las señales a controlar. La realimentación en una simulación se suele realizar a partir de la salida del modelo de planta, sin embargo en la realidad esto no ocurre así, las mediciones de las salidas del sistema físico las realizan los sensores del UAV.

7.1 MODELO DE LA PLANTA

La planta del modelo es un sistema de ecuaciones que genera unas salidas en función de las entradas, parámetros y estados del sistema. Dentro de la planta del sistema se pueden diferenciar dos modelos de planta distintos, por un lado, la dinámica del vehículo y por otro los motores. El sistema comienza con las entradas, que en este caso serán las velocidades angulares de cada uno de los motores. Con estos comandos, los estados dentro de la planta cambian como son la altura, la velocidad angular del vehículo o la velocidad axial del vehículo. Los parámetros son constantes que afectan a los estados como son inercias del sistema, coeficientes de

rozamiento, etc. Las salidas de la planta dependen de las entradas, los parámetros y los estados previos.

7.1.1 ESTADOS DEL MODELO DE PLANTA

Un Quadrotor tiene 6 grados de libertad. Esto significa que el sistema se puede trasladar en direcciones X, Y y Z, y también puede rotar en direcciones Roll, Pitch y Yaw. Las rotaciones en Roll y Pitch son las que permiten al sistema desplazarse en las direcciones X e Y. Roll se puede definir como la rotación sobre el eje X, Pitch se puede definir como la rotación sobre el eje Y y Yaw se puede definir como la rotación sobre el eje Z. Las variables de estado son las variables dinámicas del sistema físico que incluyen posición y orientación. La mayoría de los estados dependen de las entradas y de otros estados. El sistema de coordenadas de referencia del UAV será NED, de sus siglas en inglés, *North-East-Down*.

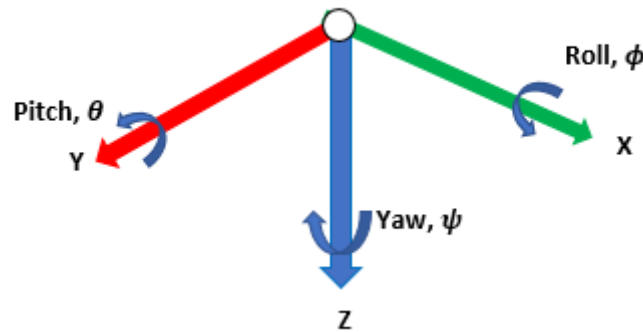


Ilustración 20-Sistema de coordenadas NED.

Un estado completamente observable es aquel que se puede medir directamente usando sensores. En este sistema no se da esta condición por lo que para realizar la estimación de todas las variables de estado habrá que utilizar estrategias de fusión sensorial, en este caso un filtro de Kalman. En la siguiente tabla se presentan las variables de estado, así como cuales son observables y cuales se tienen que obtener a partir de medidas indirectas.

Tabla 1- Estados del sistema.

Simbolo	Descripcion	Unidades(SI)	Unidades ArduCopter	Observabilidad
θ	Angulo de Pitch	rad	centideg	—
$\dot{\theta}$	Velocidad angular Pitch	rad/s	centideg/s	Giroscopio
$\ddot{\theta}$	Aceleracion angular Pitch	rad/s ²	centideg/s ²	—
ϕ	Angulo de Roll	rad	centideg	—
$\dot{\phi}$	Velocidad angular Roll	rad/s	centideg/s	Giroscopio
$\ddot{\phi}$	Aceleracion angular Roll	rad/s ²	centideg/s ²	—
ψ	Angulo de Yaw	rad	centideg	—
$\dot{\psi}$	Velocidad angular Yaw	rad/s	centideg/s	Giroscopio
$\ddot{\psi}$	Aceleracion angular Yaw	rad/s ²	centideg/s ²	—
X	Posicion X	m	cm	GPS
\dot{X}	Velocidad X	m/s	cm/s	—
\ddot{X}	Aceleracion X	m/s ²	cm/s ²	Acelerometro
Y	Posicion Y	m	cm	GPS
\dot{Y}	Velocidad Y	m/s	cm/s	—
\ddot{Y}	Aceleracion Y	m/s ²	cm/s ²	Acelerometro
Z	Posicion Z	m	cm	Barometro/ LiDAR
\dot{Z}	Velocidad Z	m/s	cm/s	—
\ddot{Z}	Aceleracion Z	m/s ²	cm/s ²	Acelerometro
$PWM1$	Motor 1 PWM	rad/s	pwm	Canales Salida Motores
$PWM2$	Motor 2 PWM	rad/s	pwm	Canales Salida Motores
$PWM3$	Motor 3 PWM	rad/s	pwm	Canales Salida Motores
$PWM4$	Motor 4 PWM	rad/s	pwm	Canales Salida Motores

7.1.2 SISTEMAS DE COORDENADAS

Hay que definir el sistema de coordenadas y los ángulos de Euler. Los ángulos de Euler son 3 ángulos que se utilizan para describir la orientación de un sólido rígido con respecto a un plano de referencia con origen en el centro de gravedad del sistema.

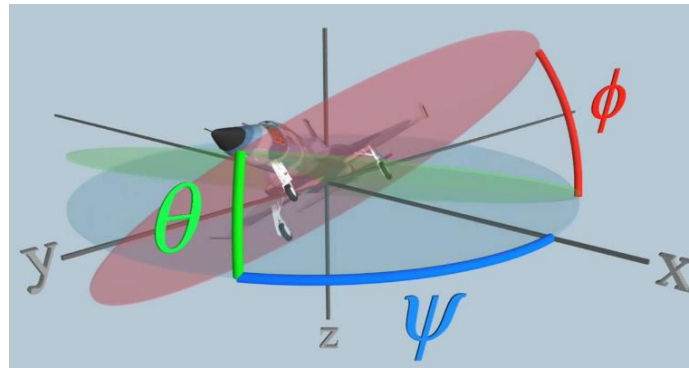


Ilustración 21- Sistema de coordenadas del cuerpo del sistema.

Se definirán dos sistemas de coordenadas, uno inercial, fijo al origen y otro en el cuerpo del vehículo similar al de la Ilustración 21.

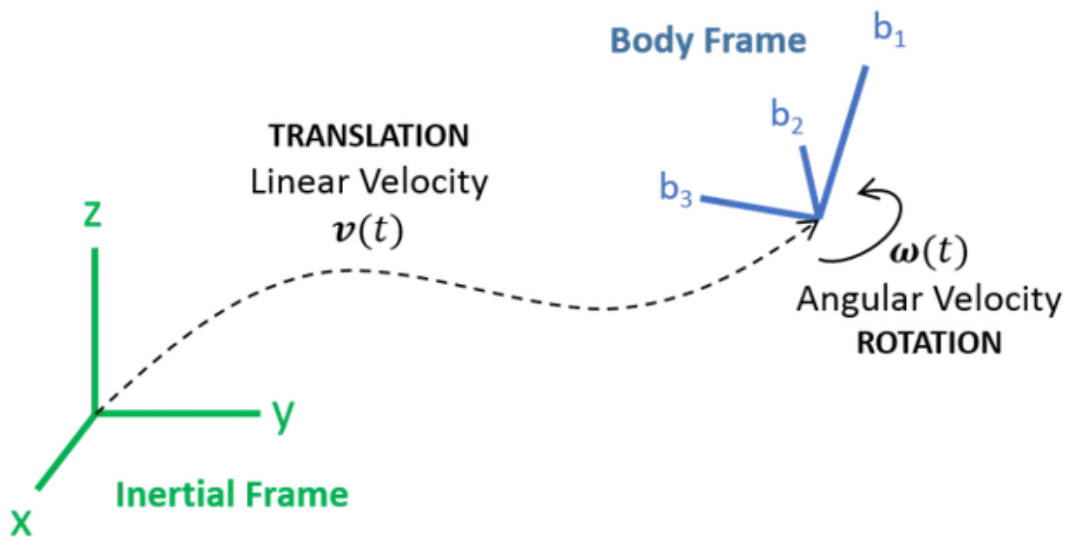


Ilustración 22- Sistemas de coordenadas utilizados en el modelo.

La idea de los ángulos de Euler es que se puede representar cualquier orientación final de un sistema de referencia como un conjunto de 3 rotaciones secuenciales.

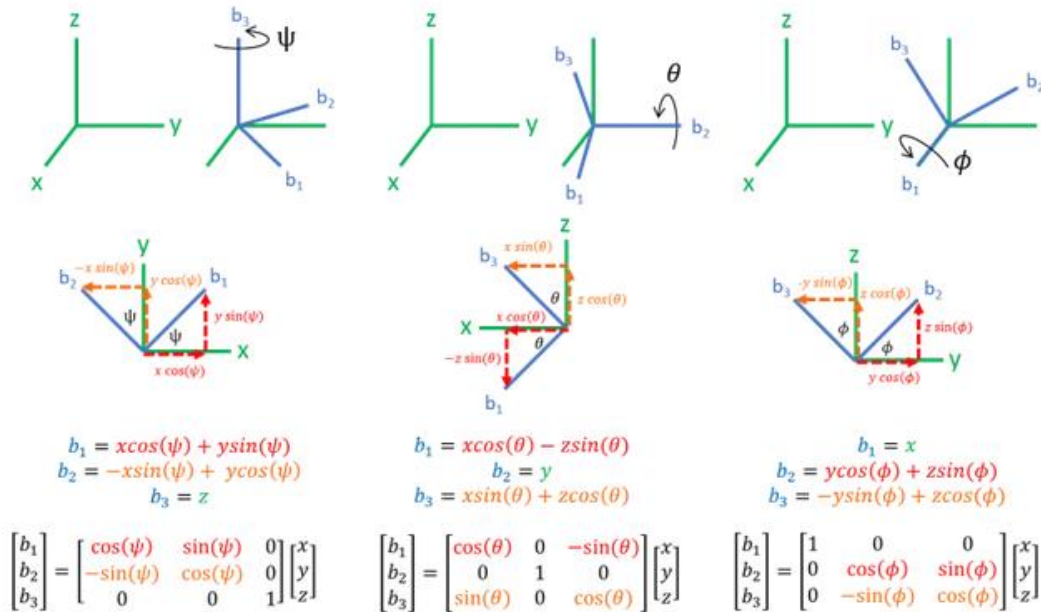


Ilustración 23-Angulos de Euler.

Se puede obtener una matriz de transformación entre el sistema de referencia inercial y el del cuerpo, concatenando las 3 matrices de rotación individuales para cada eje de rotación.

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

$$\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ -\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi) & \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi) & \sin(\phi)\cos(\theta) \\ \sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi) & -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Ilustración 24- Matriz de transformación de coordenadas al cuerpo.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos(\theta)\cos(\psi) & -\cos(\phi)\sin(\psi) + \sin(\phi)\sin(\theta)\cos(\psi) & \sin(\phi)\sin(\psi) + \cos(\phi)\sin(\theta)\cos(\psi) \\ \cos(\theta)\sin(\psi) & \cos(\phi)\cos(\psi) + \sin(\phi)\sin(\theta)\sin(\psi) & -\sin(\phi)\cos(\psi) + \cos(\phi)\sin(\theta)\sin(\psi) \\ -\sin(\theta) & \sin(\phi)\cos(\theta) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Ilustración 25-Matriz de transformación de coordenadas al sistema inercial.

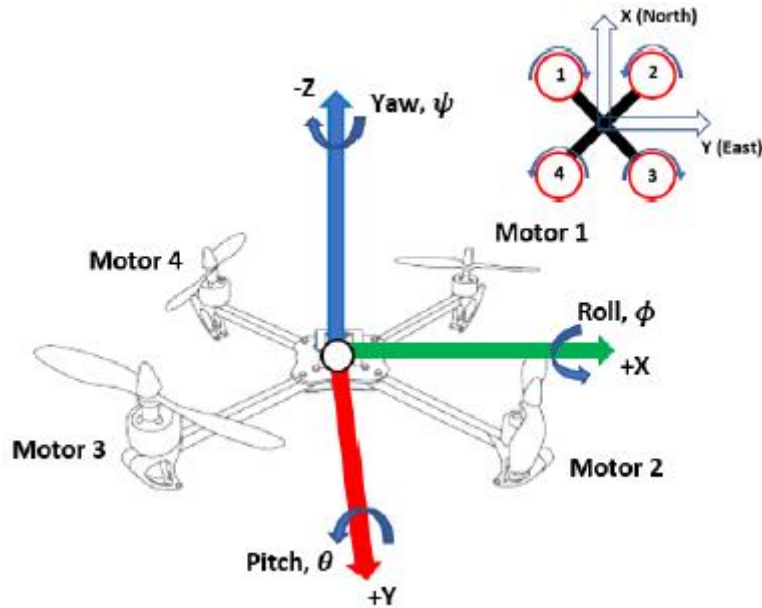


Ilustración 26- Esquema del sistema con los ejes de coordenadas.

7.1.3 PARÁMETROS DE LA PLANTA

Antes de definir las ecuaciones de movimiento, es necesario definir los parámetros del sistema. Los parámetros son constantes que tienen relación con las características físicas del sistema que se va a modelar. Los primeros parámetros que hay que definir son los momentos de inercia de cada eje. El momento de inercia es la cantidad de par que se necesita para obtener una determinada aceleración angular sobre un eje de rotación. Estos parámetros se pueden estimar de distintas formas, en este trabajo se han obtenido a partir de un modelo de CAD del sistema. Este modelo de CAD no es el objetivo de este proyecto por lo que no se le dará demasiada importancia, aunque hay que decir que estos parámetros son muy importantes para realizar una simulación representativa del sistema que se quiere modelar.

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} = \begin{bmatrix} 0,024 & 0 & 0 \\ 0 & 0,024 & 0 \\ 0 & 0 & 0,038 \end{bmatrix} \text{ kg m}^2$$

La inercia del rotor, que es el momento de inercia sobre el eje de rotación de las hélices, es otro de los parámetros necesarios para el modelado de la planta. [4]

$$J_r = 6,5 \times 10^{-7} \text{ kg m}^2$$

La distancia horizontal desde el centro de gravedad del UAV hasta el centro de los rotores también debe ser definida. En este caso se medirá del sistema real.

$$l = 0,24 \text{ m}$$

La distancia vertical desde el centro de gravedad del UAV al plano de los rotores se definirá de la misma forma que el parámetro anterior.

$$d_{CG} = 0,03 \text{ m}$$

La masa del vehículo es otro de los parámetros que hay que definir para realizar la simulación, este parámetro es uno de los más limitantes del modelo junto con el del coeficiente de empuje, ya que estos sistemas necesitan de un ratio empuje-masa adecuado para su correcto funcionamiento.

$$m = 1,663 \text{ kg}$$

Para obtener el coeficiente de Thrust o empuje se ha realizado un ensayo experimental en el cual se han tomado medidas de la velocidad angular de los motores y del empuje, y se ha obtenido un coeficiente de empuje estimado.

$$Thrust = b * \Omega^2$$

$$\Omega = 2\pi(RPM)/60$$



Ilustración 27- Ensayo del empuje de los rotores.

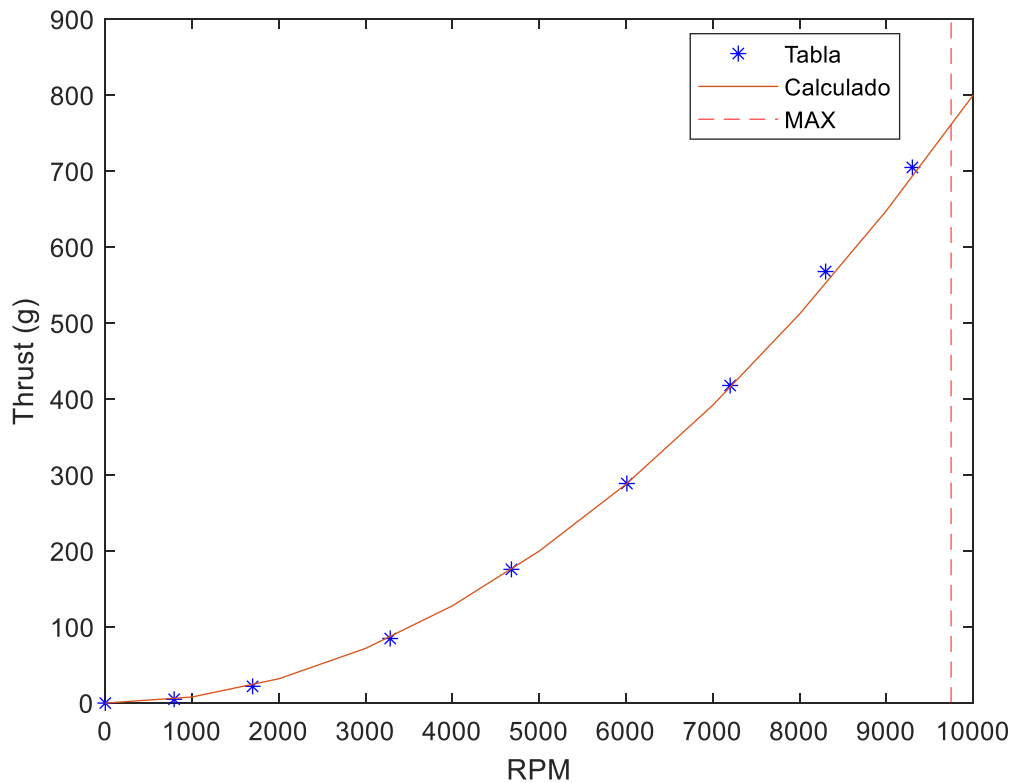


Ilustración 28- Estimación del empuje de los motores.

$$b = 7.296 \times 10^{-7} \text{ kg s}^2$$

Para realizar la estimación del coeficiente de Drag se podría plantear otro ensayo experimental, pero en este trabajo se ha obtenido de otra forma. Según distintas fuentes de bibliografía se puede ver que normalmente el coeficiente de drag es un orden menor al del coeficiente de empuje, por lo que se ha aproximado de esta forma [9]. Hay que tener en cuenta que las unidades del coeficiente de empuje no son las adecuadas para entrar en las ecuaciones dinámicas, habrá que multiplicar por la gravedad para obtener $N s^2$. De esta forma el coeficiente de drag será:

$$d = 7.2 \times 10^{-7} \text{ N m s}^2$$

También hay que establecer la velocidad máxima que pueden alcanzar los rotores, debido a que, si no se establece en el modelo un límite superior, nunca se alcanzará la saturación de los motores durante la simulación y este fenómeno de saturación de los motores es una de las cosas que más interesa estudiar.

$$\Omega_{MAX} = 9746 \text{ RPM}$$

Otro parámetro importante a la hora de modelar los motores eléctricos sería la constante de tiempo del motor. En este modelo se asume que el empuje no es instantáneo, es decir que, si se comanda a los motores que alcancen una determinada velocidad, esta velocidad tardara un poco de tiempo en alcanzarse y este tiempo depende de las características físicas del motor. En este caso se ha modelado este fenómeno mediante una función de transferencia de primer orden.

$$G(s) = \frac{1}{\tau_M s + 1}$$

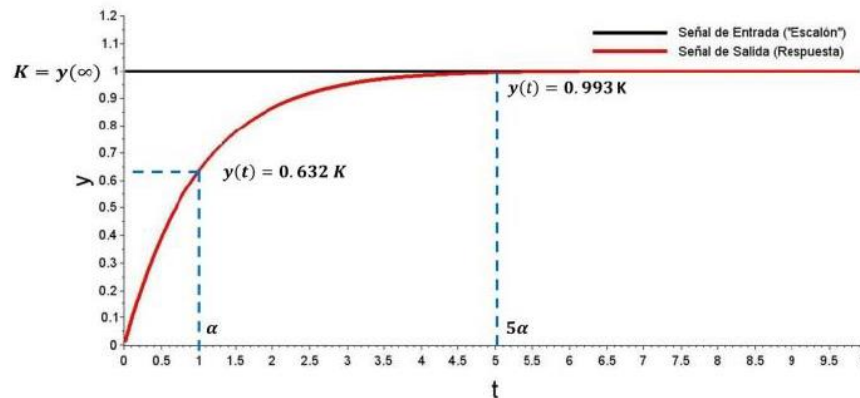


Ilustración 29-Respuesta de un sistema de primer orden ante un escalón.

La constante de tiempo se define como el tiempo que la respuesta del sistema tarda en alcanzar el 63,2% de la magnitud de la señal comandada.[12]

$$\tau_M = 0,01 \text{ s}$$

Los siguientes parámetros que se van a definir tienen que ver con la relación entre el UAV y el entorno. En el mundo real, los UAV experimentan efectos aerodinámicos, debido a la resistencia al aire y el viento. La resistencia al aire se puede definir como una fuerza que se opone al desplazamiento del vehículo. Además, existe también un par de resistencia al aire el cual afectará a la orientación del sistema. Medir la resistencia al aire es una tarea extremadamente compleja, y toma mayor dificultad para la resistencia rotacional ya que es un fenómeno que se suele despreciar y se suele simplificar los modelos estudiando solo la resistencia al desplazamiento. [5][18]

$$M_{pitch/roll} = 0,1522 \text{ kg} \frac{\text{m}}{\text{s}}$$

$$A_{xyz} = 0,11 \frac{\text{kg}}{\text{s}}$$

Tabla 2- Parámetros del modelo.

Símbolo	Descripción	Unidad
I_{XX}, I_{YY}, I_{ZZ}	Momentos de Inercia	kgm^2
J_r	Inercia del rotor	kgm^2
l	Distancia horizontal rotor al centro del sistema	m
b	Coficiente Thrust	kg/s^2
d	Coficiente de Drag	kg m /s^2
A_X, A_Y, A_Z	Coficiente de resistencia del aire	kg/s
$M_{ROLL}^{AIR}, M_{PITCH}^{AIR}$	Coficiente de resistencia rotacional del aire	kg m/s
m	Masa	kg
d_{CG}	Distancia vertical del centro de masa al plano de los rotores	m
τ_m	Constante de tiempo del motor	s

7.1.4 EFECTO AERODINÁMICO

La aportación del efecto aerodinámico sobre el sistema es uno de los términos más difíciles de modelar y según las fuentes que se consulten se puede ver como existen multitud de modelos distintos que intentan simular un mismo efecto físico. En este modelo además de incluir la resistencia al aire se pretende modelar la influencia del viento sobre el sistema y a continuación se explicará cómo se ha introducido este fenómeno en las ecuaciones de traslación y rotación.

Para realizar el cálculo de esta resistencia se va a utilizar la velocidad relativa del UAV y el viento, en vez de únicamente la velocidad lineal del vehículo, ya que, si el viento va en el mismo sentido de avance que el UAV, la fuerza resultante de este viento empujará en el sentido correcto mientras que si el viento incide en el sentido contrario de avance lo que hará es frenarlo.

$$R_X^{AIRE} = A_X(V_X^{VIENTO} - \dot{X})$$

De esta forma es como se modela la resistencia al desplazamiento y se le añade el efecto del viento.

En el caso del par de resistencia a la rotación el efecto que produce en el sistema no es tan obvio, es por eso por lo que se ha realizado una pequeña simulación en CFD con el programa Ansys Fluent, en el cual se quería estudiar el sentido de giro del par que genera el viento si incide en el mismo sentido de avance del UAV o en el sentido contrario.

Lo que se quiere estudiar es si dependiendo de estos sentidos de avance y del viento, el par de resistencia a la rotación tiende a estabilizar o inestabilizar el sistema. Es decir, que si provoca un aumento en la orientación del vehículo o si provoca una disminución de este. A continuación, se muestran unas imágenes donde se puede ver el proceso que se ha seguido en Ansys Fluent para estudiar este fenómeno. Debido a que el objetivo de este trabajo no es un estudio de la aerodinámica del vehículo no se profundizara en explicar que consideraciones se han tenido, solo se presentara la conclusión.

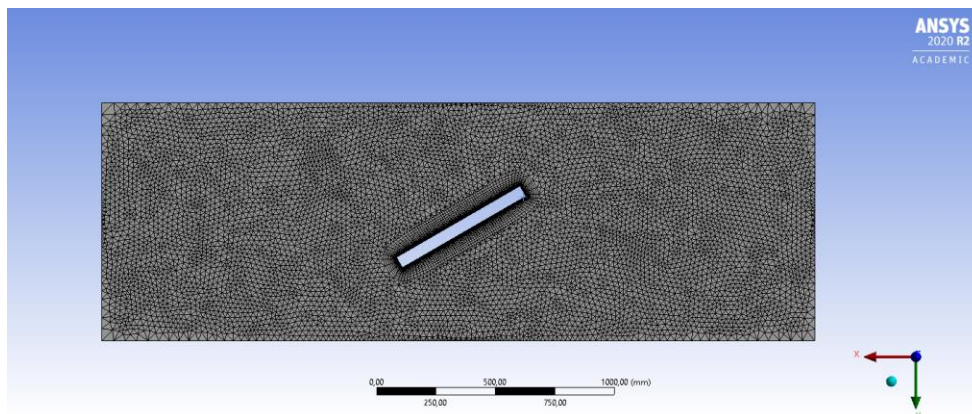


Ilustración 30-Mallado en Ansys Fluent.

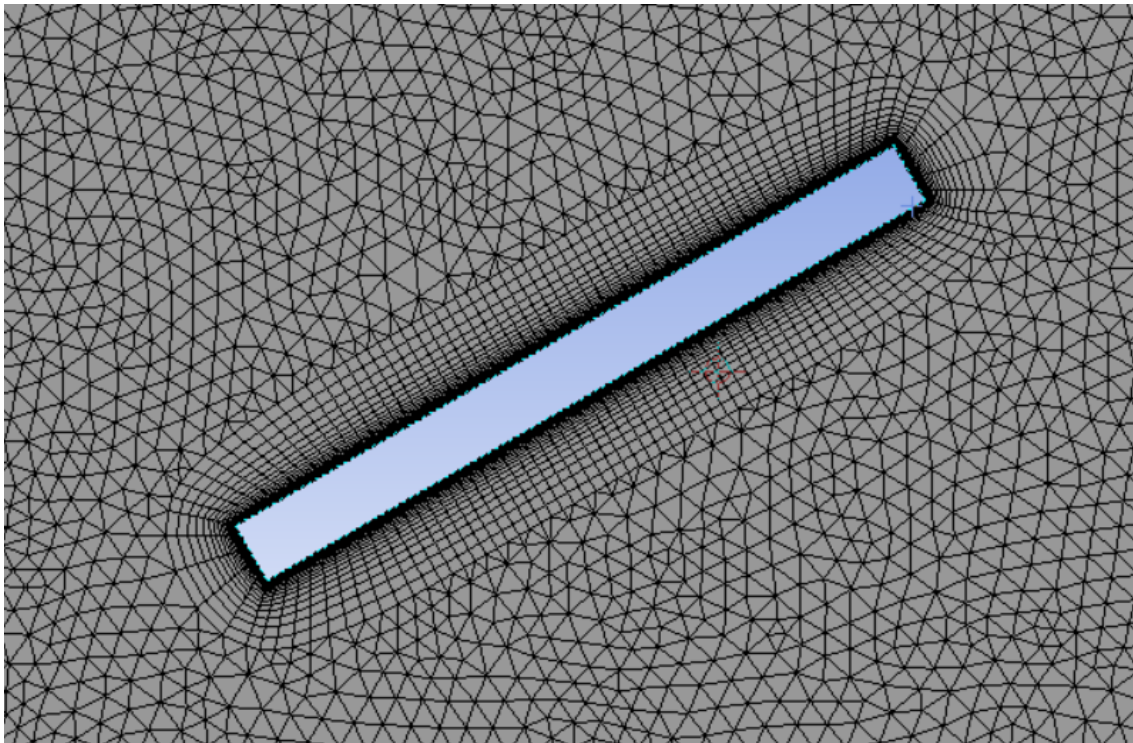


Ilustración 31-Detalle mallado Ansys Fluent.

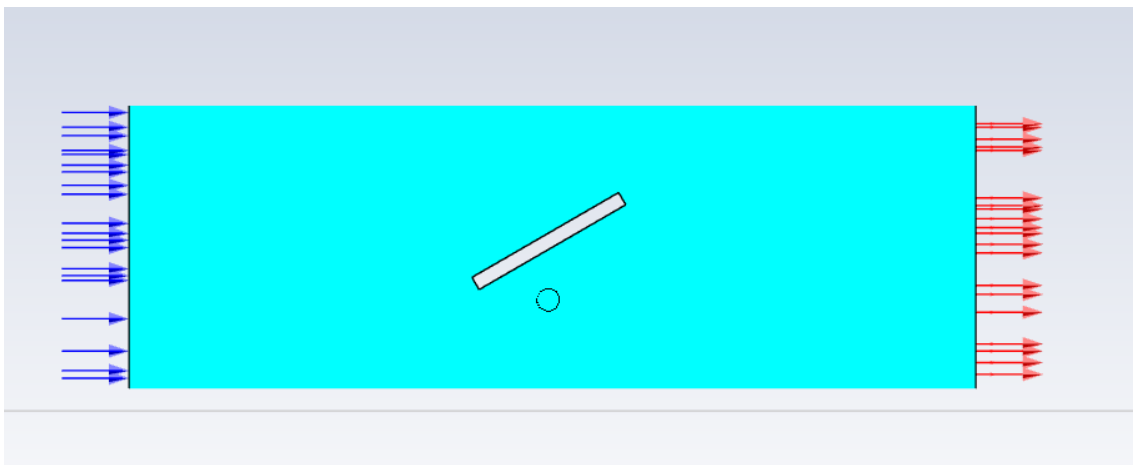


Ilustración 32- Condiciones de contorno caso 1 Ansys Fluent.

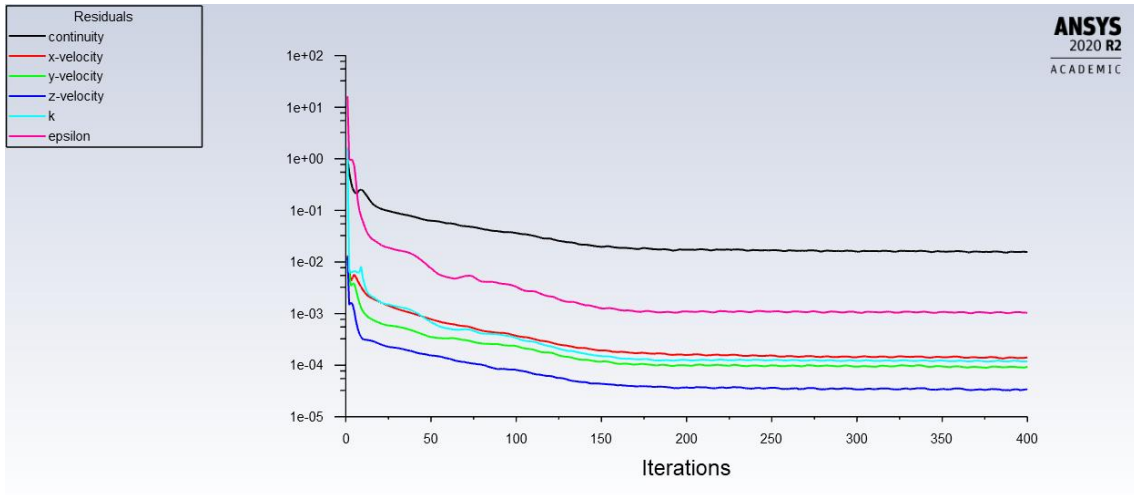


Ilustración 33-Convergencia iteraciones.

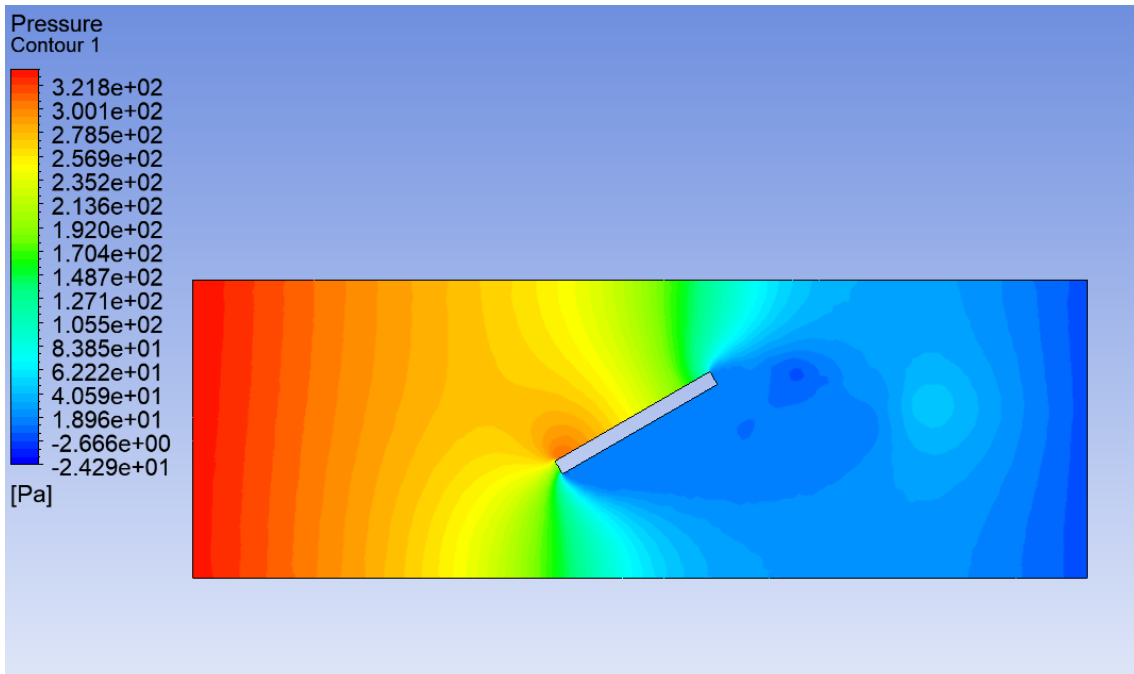


Ilustración 34-Campo de presión del caso 1.

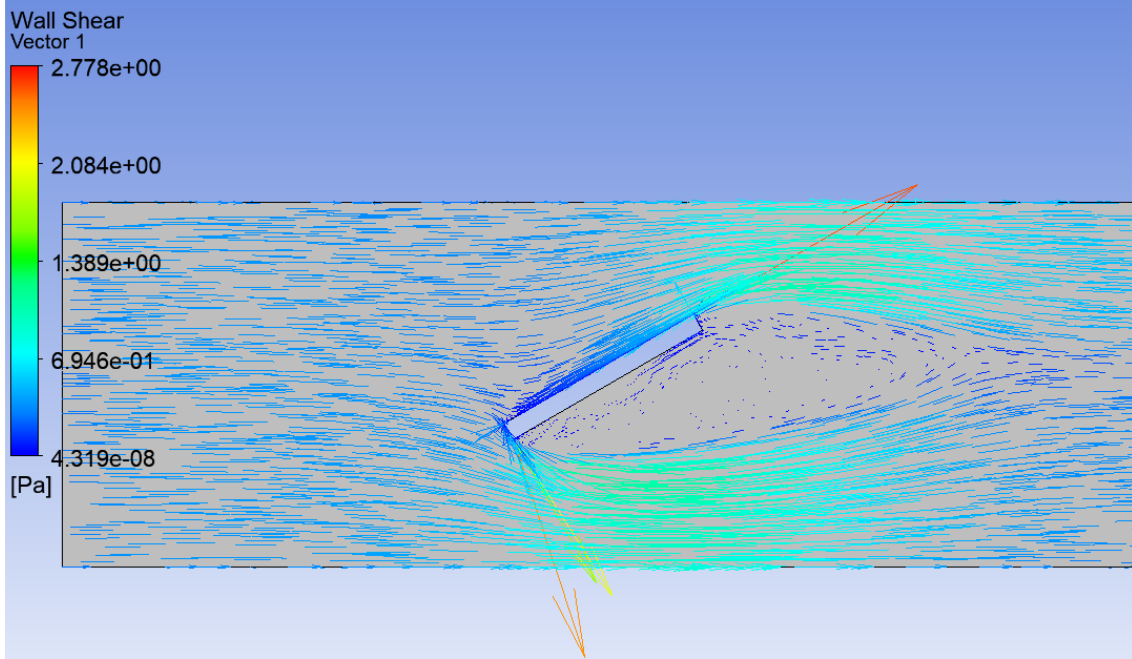


Ilustración 35-Campo de velocidades del caso 1.

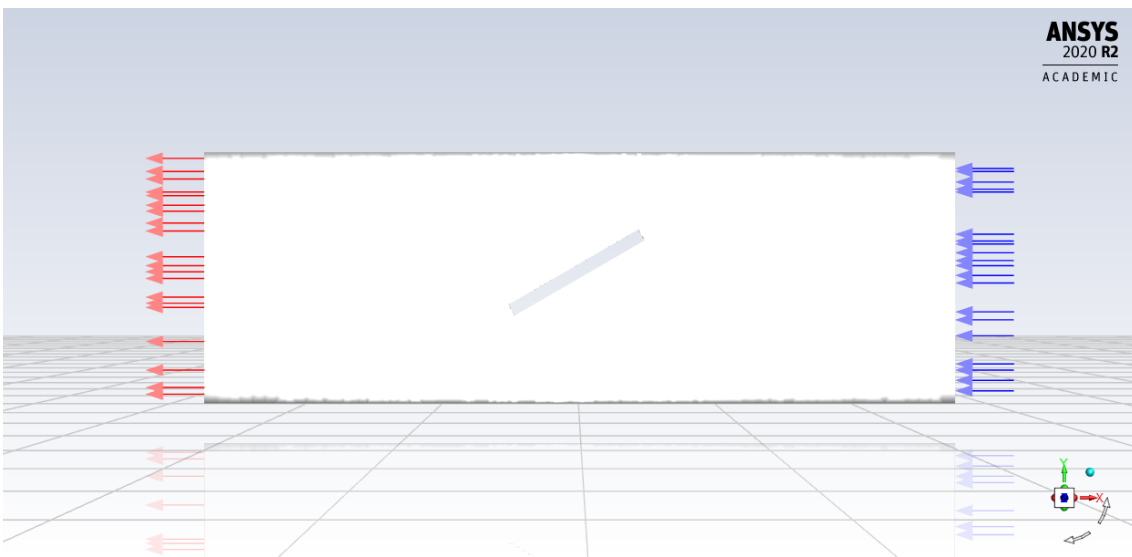


Ilustración 36-Condiciones de contorno del caso 2.

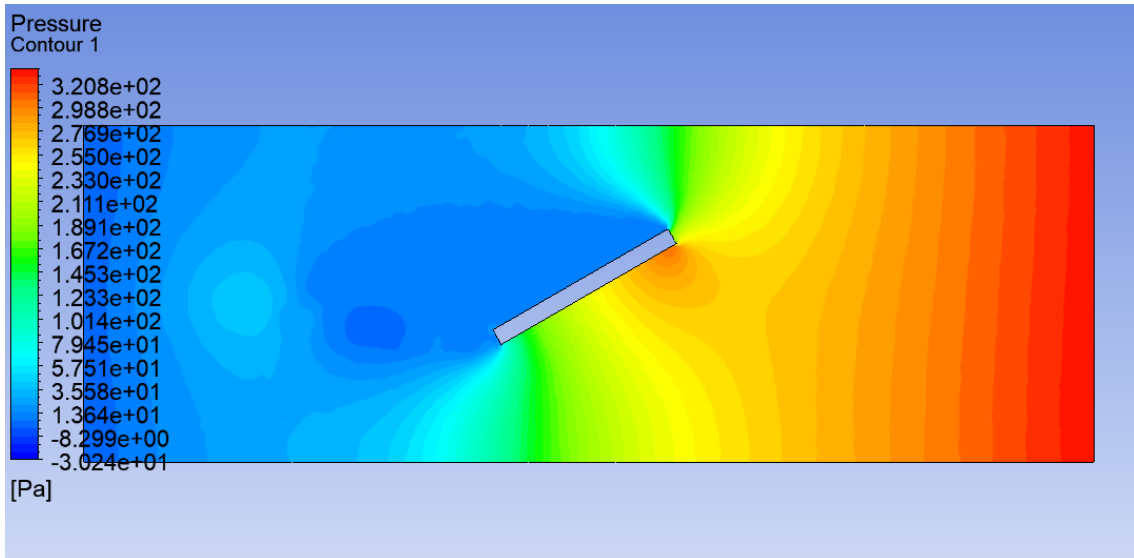


Ilustración 37-Campo de presión del caso 2.

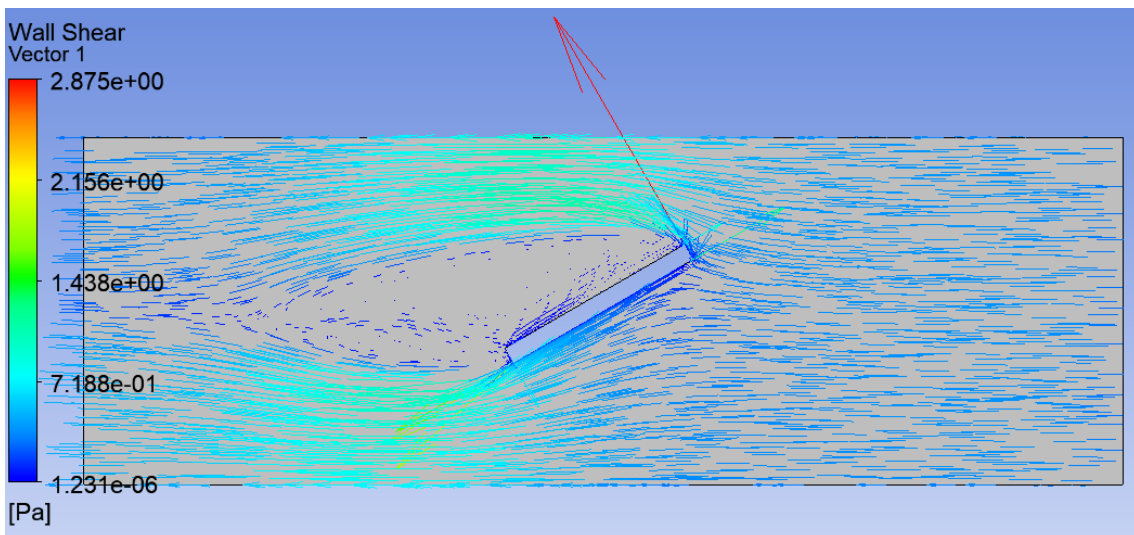


Ilustración 38-Campo de velocidad del caso 2.

Tras realizar las simulaciones de los dos casos que se quería estudiar (sentido del viento igual al de avance y sentido del viento contrario al de avance) se ha concluido que independientemente del sentido del vehículo o del viento, este último tiende a inestabilizar el sistema siempre. Por eso esta vez la fórmula matemática que represente este fenómeno no puede depender del signo de la velocidad relativa. A partir de la ecuación que modela este fenómeno en la referencia [10] se ha desarrollado estas dos fórmulas para este sistema en particular.

Cuando $\phi > 0$ $\dot{y} < 0$

$$M_{aero}^{\phi} = sign(\dot{y})(-1)M_{aero}(|w| + |\dot{y}|)$$

Cuando $\theta > 0$ $\dot{x} > 0$

$$M_{aero}^{\theta} = sign(\dot{x})(M_{aero}(|w| + |\dot{x}|)$$

7.1.5 ECUACIONES DE LA PLANTA

Antes de modelar el UAV en MATLAB/Simulink es necesario definir las ecuaciones matemáticas. Para ello se realizarán algunas suposiciones:

- El dron es una estructura rígida.
- La estructura y los componentes son simétricos.
- El Thrust y Drag son proporcionales con el cuadrado de la velocidad angular de las hélices.

Existen 4 ecuaciones principales del Thrust. Empuje vertical, empuje de Roll, empuje de Pitch y empuje de Yaw. Como el sistema tiene una configuración en 'x', las ecuaciones serian:

Tabla 3- Ecuaciones del empuje.

Input	Thrust Equation	Description
$U1_x$	$b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2)$	General Thrust
$U2_x$	$b \sin\left(\frac{p^i}{4}\right) (\Omega_1^2 - \Omega_2^2 - \Omega_3^2 + \Omega_4^2)$	Roll Thrust
$U3_x$	$b \sin\left(\frac{p^i}{4}\right) (\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2)$	Pitch Thrust
$U4_x$	$d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2)$	Yaw Thrust

$U1_x$ aplica un empuje general a todos los motores para aumentar la altura. $U2_x$ cambia el empuje de los motores para provocar un incremento del ángulo de Roll, de forma similar $U3_x$ y $U4_x$ cambian los empujes de los motores para provocar un aumento en el Pitch y el Yaw respectivamente.

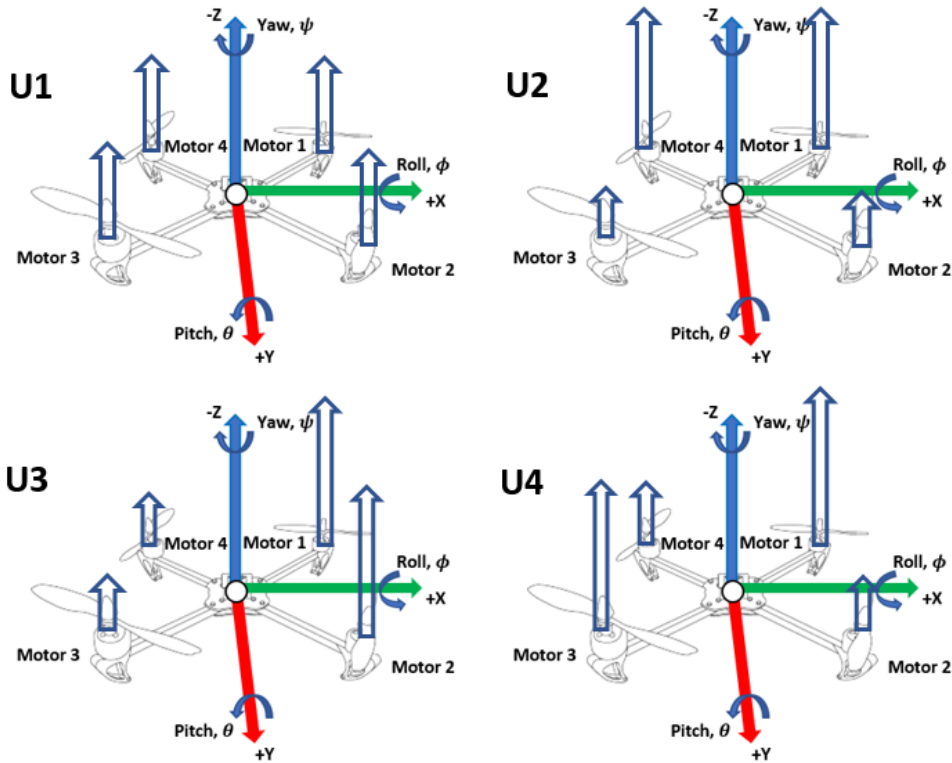


Ilustración 39- Distribución del empuje según el movimiento.

Se puede definir también la velocidad angular resultante. Esta variable se utilizará para calcular las aceleraciones angulares. En la ecuación se puede ver como hay dos rotores que giran en sentido contrario a los otros dos, ya que tienen signos distintos. Normalmente son los motores opuestos los que giran en el mismo sentido.

$$\Omega_r = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4$$

Ahora se definirán las ecuaciones del movimiento. El origen de estas ecuaciones es:

$$F = b \left(\sum_{i=1}^4 \Omega_i \right)^2$$

Donde b es el coeficiente de Thrust.

$$F = ma$$

$$M = I\alpha$$

Se empieza describiendo los efectos que influyen dentro de la dinámica rotacional del sistema y son:

- Efecto giroscópico de la estructura.
- Efecto giroscópico de las hélices.
- Fuerza de los motores y las hélices.
- Momento provocado por el centro de gravedad.
- Resistencia rotacional al aire y el viento.

Todos estos fenómenos provocan momentos los cuales se suman para formar las ecuaciones de las aceleraciones angulares. Para describir un poco las ecuaciones se va a poner un ejemplo de una de ellas:

$$\ddot{\phi} = \frac{\dot{\theta}\psi(I_{YY} - I_{ZZ}) + J_R\dot{\theta}\Omega_R + lU2 - mgd_{CG}(\sin \phi) + M_{aero}^{\phi}}{I_{XX}}$$

- Efecto giroscópico de la estructura: $\dot{\theta}\psi(I_{YY} - I_{ZZ})$
- Efecto giroscópico de las hélices: $J_R\dot{\theta}\Omega_R$
- Fuerza de los motores y las hélices: $lU2$
- Momento provocado por el centro de gravedad: $mgd_{CG}(\sin \phi)$
- Resistencia rotacional al aire y el viento: M_{aero}^{ϕ}

Aceleración angular de Roll:

$$\ddot{\phi} = \frac{\dot{\theta}\psi(I_{YY} - I_{ZZ}) + J_R\dot{\theta}\Omega_R + lU2 - mgd_{CG}(\sin \phi) + M_{aero}^{\phi}}{I_{XX}}$$

Aceleración angular de Pitch:

$$\ddot{\theta} = \frac{\dot{\phi}\psi(I_{ZZ} - I_{XX}) - J_R\dot{\phi}\Omega_R + lU3 - mgd_{CG}(\sin \theta) + M_{aero}^{\theta}}{I_{YY}}$$

Aceleración angular de Yaw:

$$\ddot{\psi} = \frac{\dot{\theta}\dot{\phi}(I_{XX} - I_{YY}) + lU4}{I_{ZZ}}$$

Lo siguiente será describir la dinámica de traslación. En este caso las ecuaciones son más sencillas ya que influyen menos términos en el cálculo de las aceleraciones lineales:

- Fuerza de los motores y hélices
- Resistencia al aire y viento

Estos dos fenómenos provocan fuerzas las cuales se suman para formar las ecuaciones de aceleraciones lineales, se va a poner de ejemplo la aceleración lineal en X:

$$\ddot{x} = \frac{(\sin \psi \sin \phi - \cos \psi \sin \theta \cos \phi)U1 + R_x^{AIRE}}{m}$$

- Fuerza de los motores y hélices: $(\sin \psi \sin \phi - \cos \psi \sin \theta \cos \phi)U1$
- Resistencia al aire y viento: R_x^{AIRE}

Aceleración lineal en X:

$$\ddot{x} = \frac{(\sin \psi \sin \phi - \cos \psi \sin \theta \cos \phi)U1 + R_x^{AIRE}}{m}$$

Aceleración lineal en Y:

$$\ddot{y} = \frac{(\cos \psi \sin \phi - \sin \psi \sin \theta \cos \phi)U1 + R_y^{AIRE}}{m}$$

Aceleración lineal en Z:

$$\ddot{z} = \frac{mg - (\cos \theta \cos \phi)U1 + R_Z^{AIRE}}{m}$$

Es importante mencionar que las ecuaciones dinámicas de rotación están expresadas en el sistema de coordenadas del cuerpo, mientras que las ecuaciones de dinámica de traslación están definidas en el sistema de coordenadas inercial.

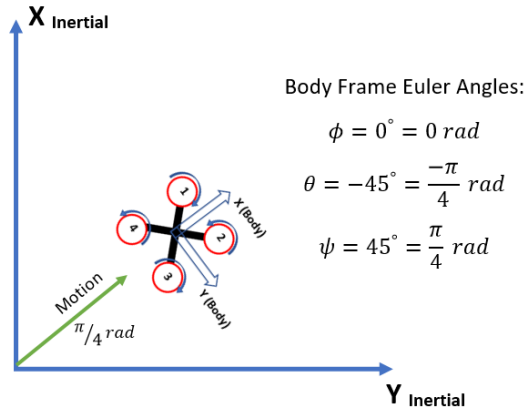


Ilustración 40-Sistema de coordenadas inercial frente a local.

En resumen, se han obtenido las fuerzas que actúan sobre cada uno de los 6 grados de libertad. A partir de estas, se pueden obtener las aceleraciones angulares y lineales. Hay que destacar de las ecuaciones obtenidas que las aceleraciones lineales dependen de la orientación del UAV. Debido a que las aceleraciones son las segundas derivadas de la posición, habrá que integrar estas aceleraciones para obtener velocidades y posiciones.

Tabla 4- Dependencias de los estados.

	Roll	Pitch	Yaw	X	Y	Z
Position	$\phi = \int \dot{\phi}$	$\theta = \int \dot{\theta}$	$\psi = \int \dot{\psi}$	$X = \int \dot{X}$	$Y = \int \dot{Y}$	$Z = \int \dot{Z}$
Velocity	$\dot{\phi} = \int \ddot{\phi}$	$\dot{\theta} = \int \ddot{\theta}$	$\dot{\psi} = \int \ddot{\psi}$	$\dot{X} = \int \ddot{X}$	$\dot{Y} = \int \ddot{Y}$	$\dot{Z} = \int \ddot{Z}$
Acceleration	$\ddot{\phi}$	$\ddot{\theta}$	$\ddot{\psi}$	\ddot{X}	\ddot{Y}	\ddot{Z}

7.1.6 MODELO DE LA PLANTA EN SIMULINK

Ahora que se han definido las ecuaciones básicas, se puede empezar a modelizar la planta en Simulink. Primero se crea un script donde se definirán los valores de los parámetros del modelo, los parámetros relacionados con la simulación y además, donde se lanzara la simulación y después se almacenaran los datos que nos interese para más tarde poder plotearlos y ver que conclusiones se pueden sacar de cada caso de la simulación.

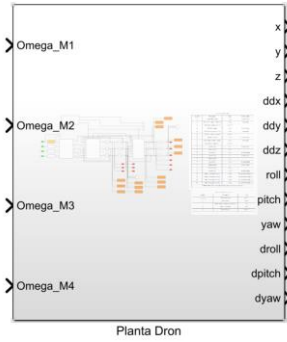


Ilustración 41- Subsistema planta del dron.

El primer paso para realizar el modelo de la planta es el de modelar la dinámica de los motores. Como se ha comentado anteriormente en esta misma memoria, el sistema de propulsión se modelará con una función de transferencia y las ecuaciones del Thrust definidas en la tabla 3.

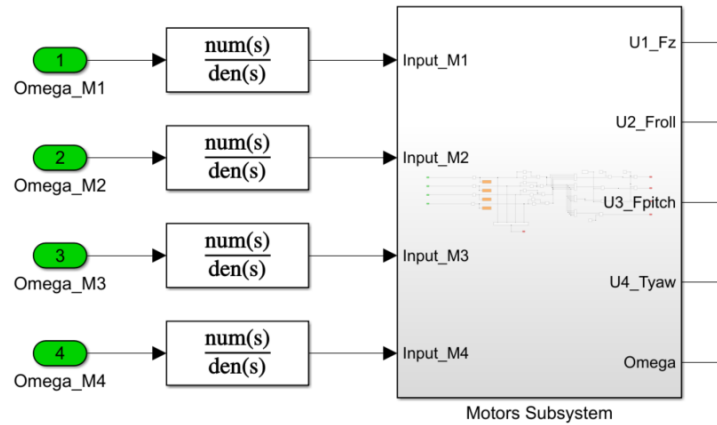


Ilustración 42- Subsistema de los rotores.

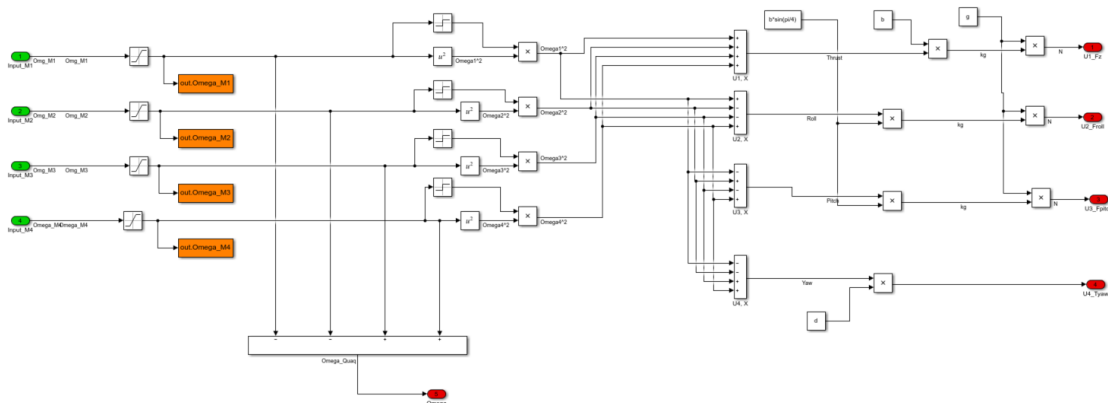


Ilustración 43- Detalle del sistema Motor Subsystem.

El siguiente paso será el de utilizar las ecuaciones de la dinámica rotacional para obtener las aceleraciones angulares. El subsistema cuenta con 12 entradas y 3 salidas. Dentro de este

subsistema hay otros 3 subsistemas dentro de los cuales se han modelado las ecuaciones, como se puede ver a continuación.

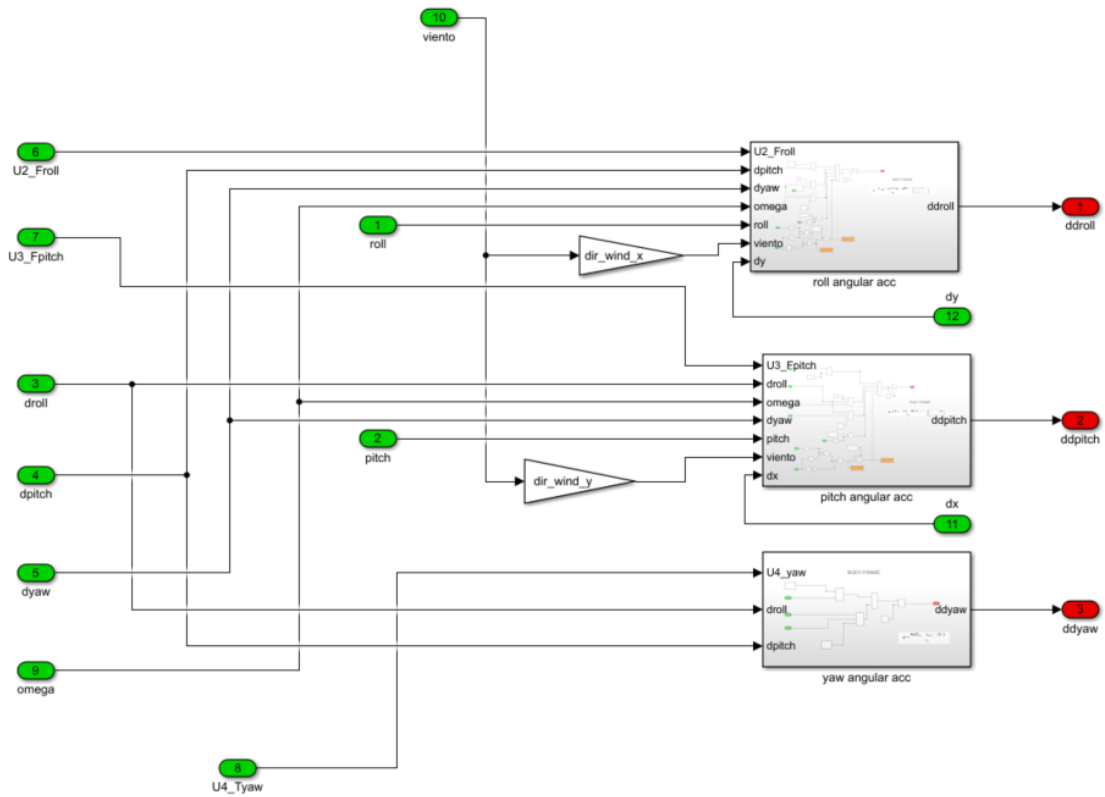


Ilustración 44- Subsistema de dinámica rotacional.

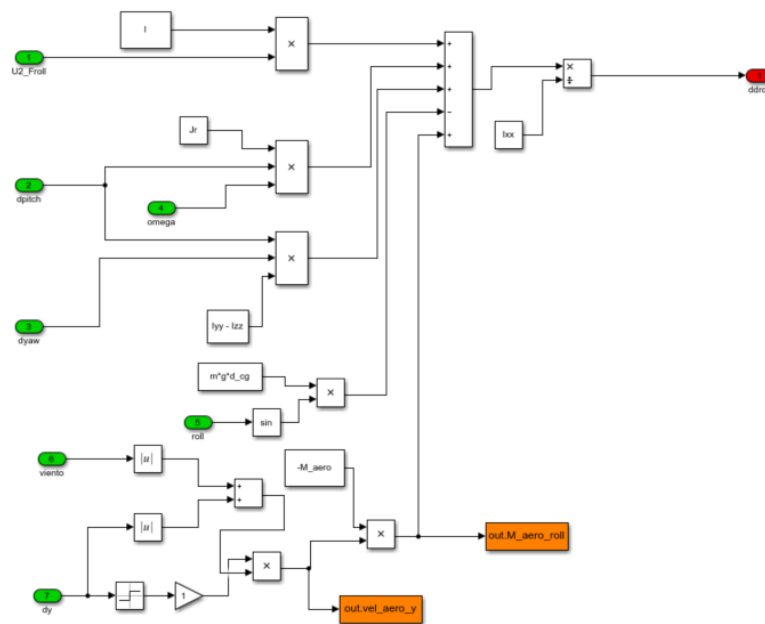


Ilustración 45- Ecuación aceleración roll.

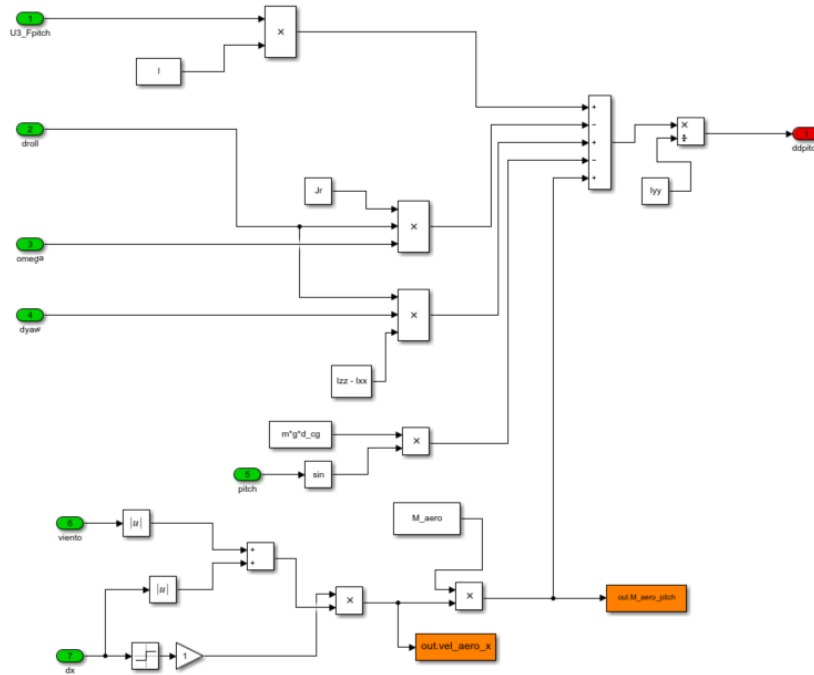


Ilustración 46- Ecuación de la aceleración de pitch.

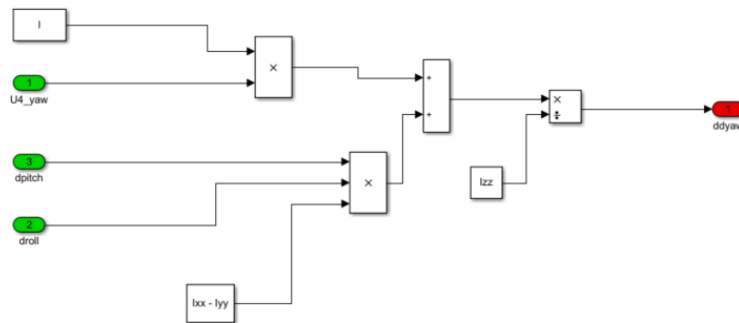


Ilustración 47- Ecuación de la aceleración de yaw.

Una vez obtenidas estas aceleraciones angulares se procede a obtener las velocidades angulares y los ángulos. Estos estados se pueden obtener integrando las aceleraciones angulares.

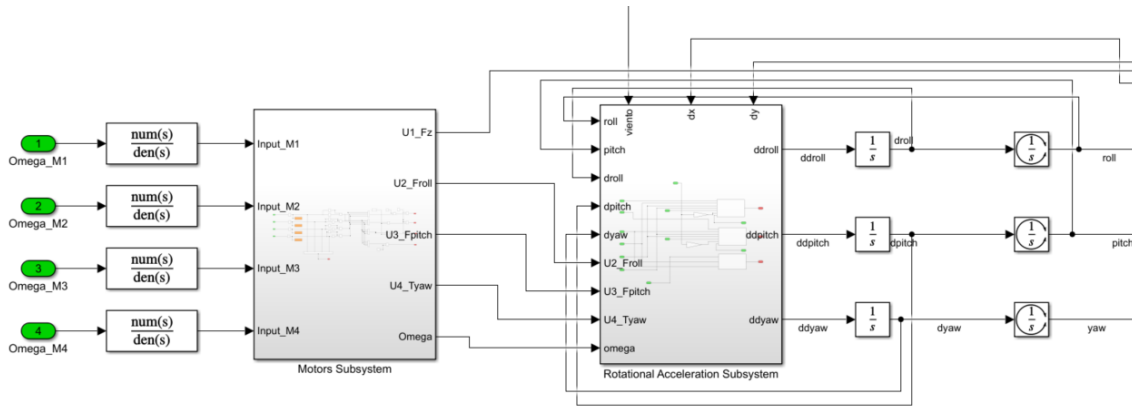


Ilustración 48- Subsistema de motores y dinámica rotacional.

El siguiente paso será el de obtener las aceleraciones lineales. Estas aceleraciones tienen que ser calculadas después de obtener los ángulos ya que depende de estos últimos. En este nuevo subsistema se modelarán las ecuaciones de la dinámica de traslación. El subsistema cuenta con 8 entradas y 3 salidas las cuales se pueden ver en la Ilustración de abajo. Dentro existen otros tres subsistemas donde se modelan las ecuaciones de las aceleraciones lineales.

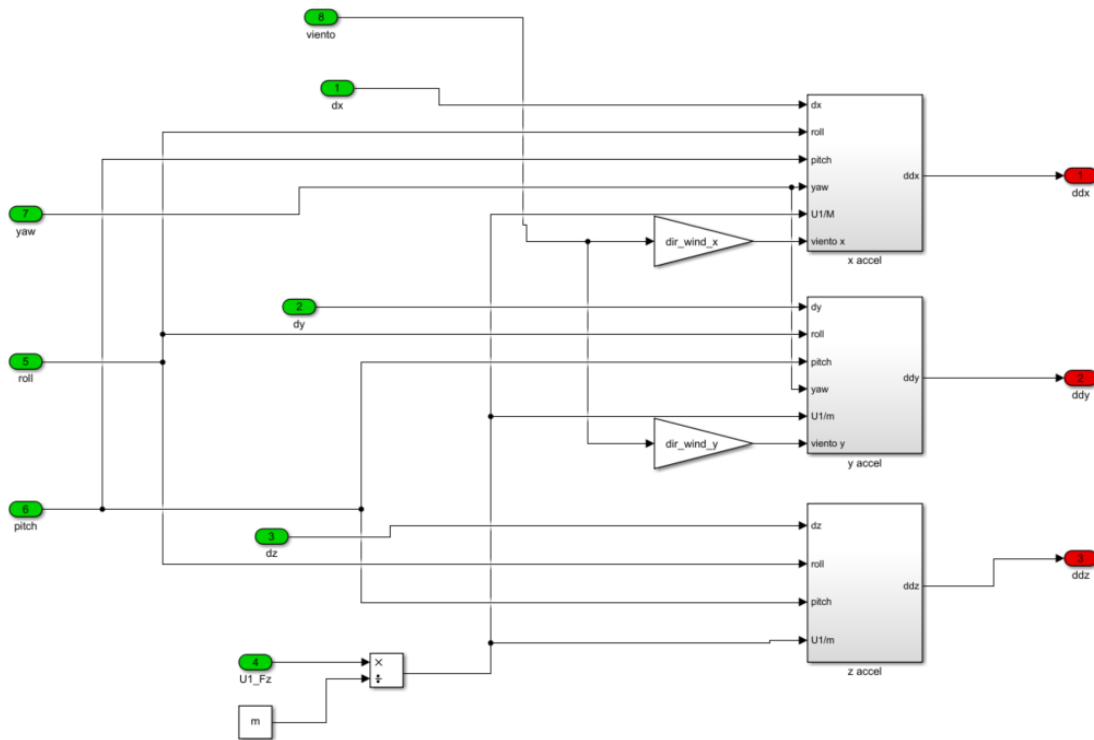


Ilustración 49- Subsistema de dinámica lineal.

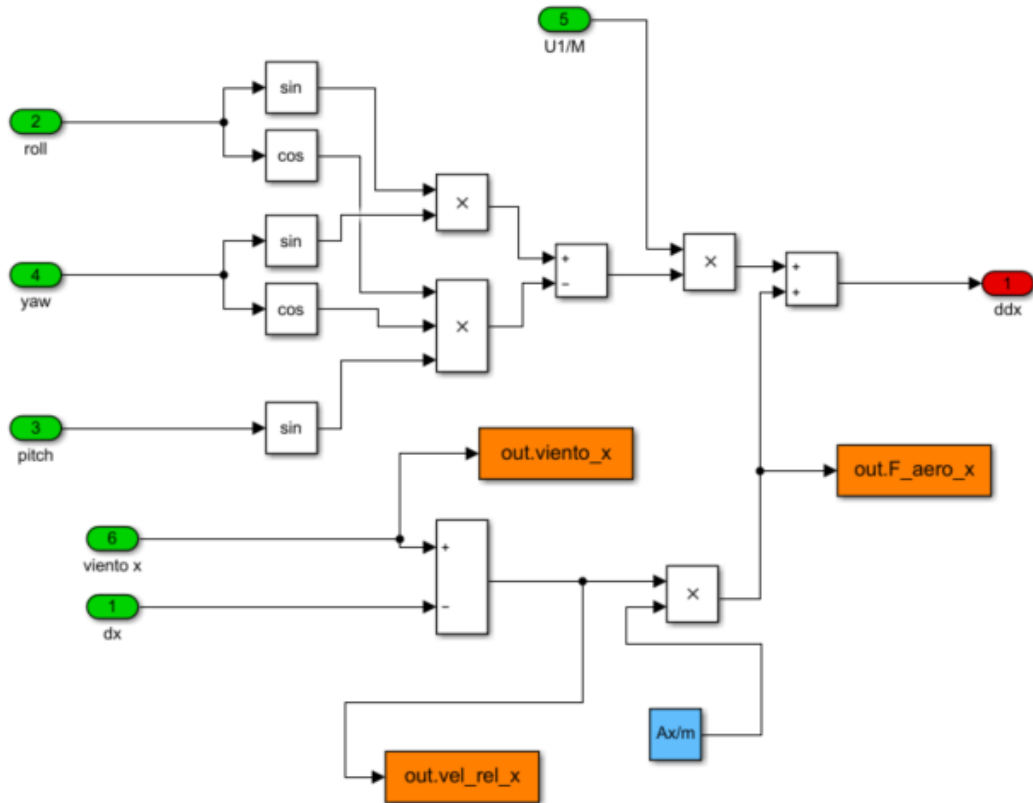


Ilustración 50- Ecuación de la aceleración lineal de X.

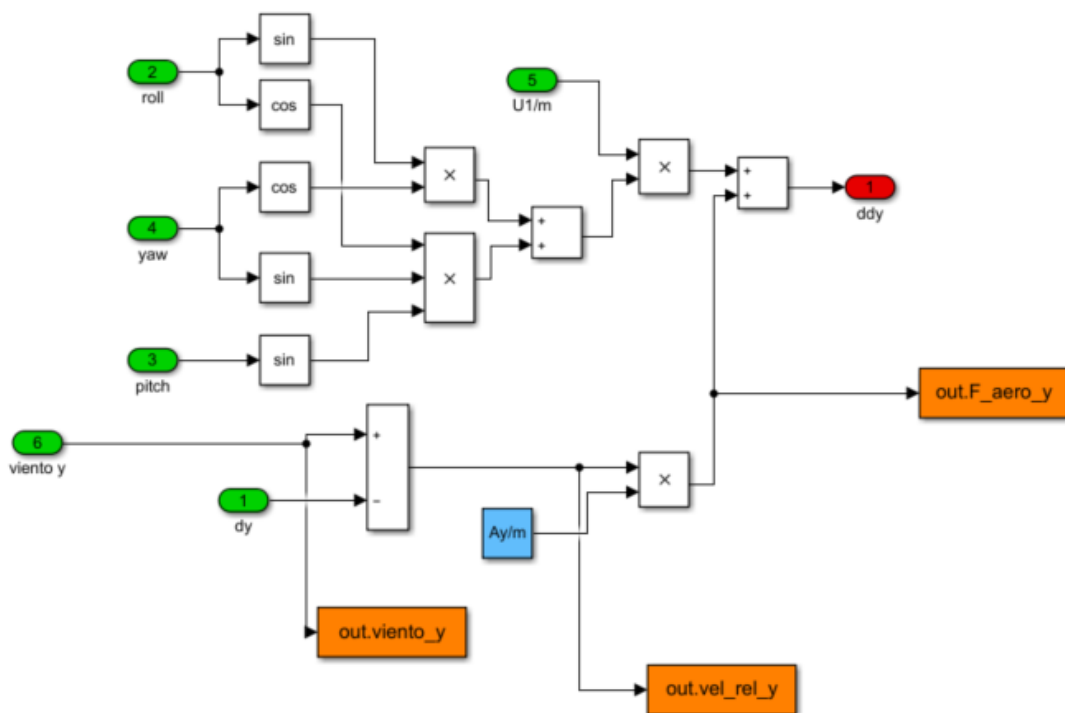


Ilustración 51- Ecuación de la aceleración lineal de Y.

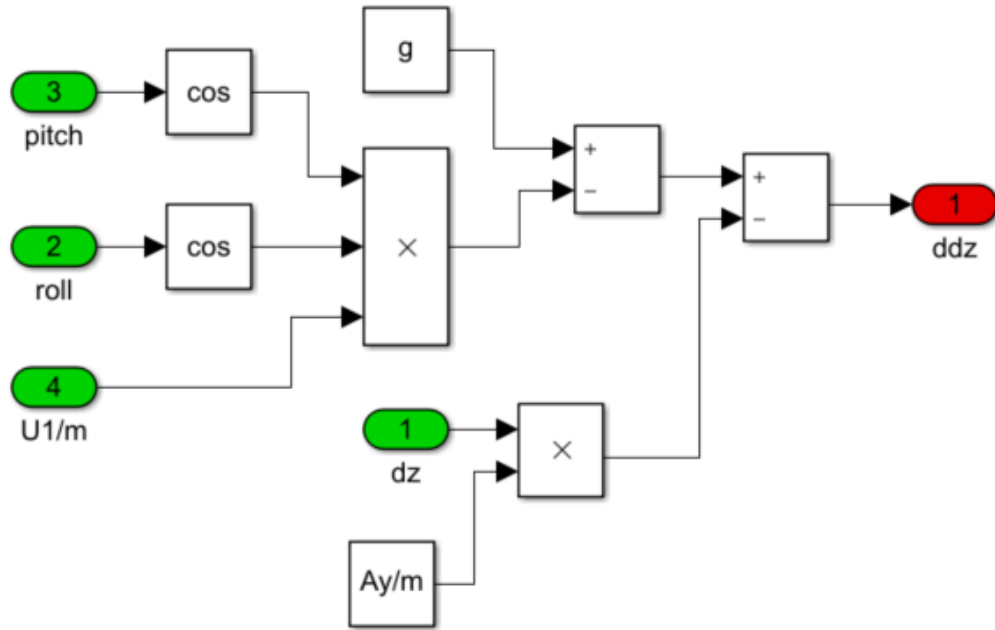


Ilustración 52-Ecuacion de la aceleración lineal en Z.

Una vez obtenidos los valores de las aceleraciones lineales habrá que integrarlas como en el caso de la dinámica rotacional para así obtener velocidad y posición.

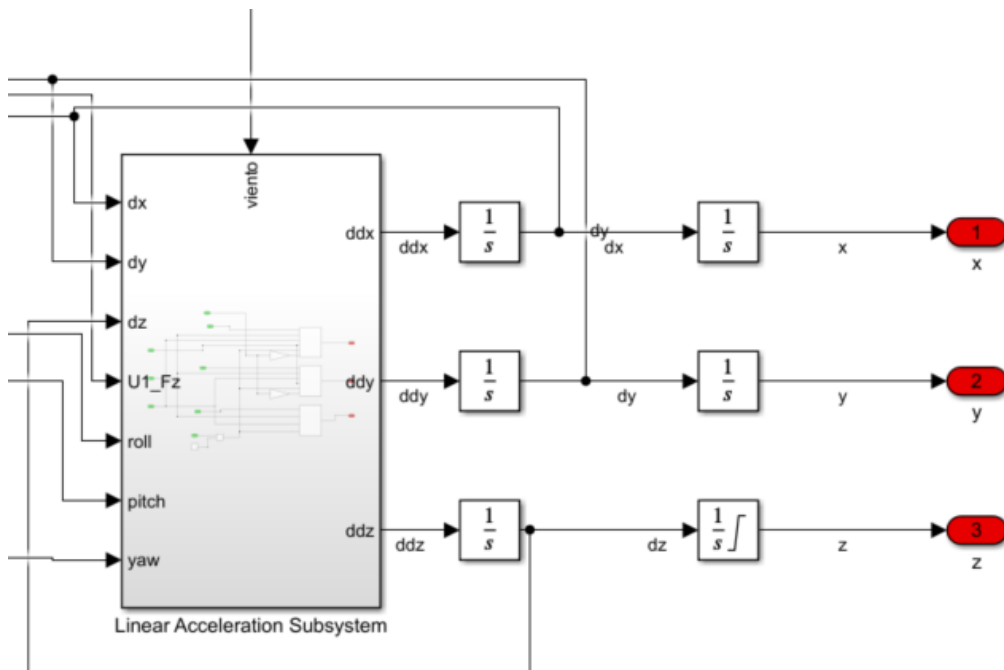


Ilustración 53- Subsistema de la dinámica lineal.

Por último, se ha creado un subsistema en el cual se modelan tres tipos de perturbaciones distintas. Se ha modelado un viento uniforme, un viento con carácter más aleatorio y una ráfaga. Según sea el caso que se desea simular habrá que escoger la fuente de viento deseada.

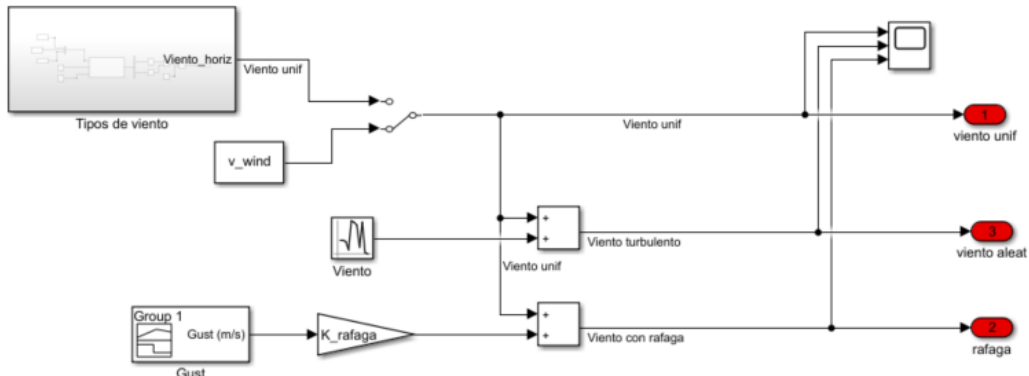


Ilustración 54- Subsistema del viento.

Finalmente, la planta tiene esta disposición.

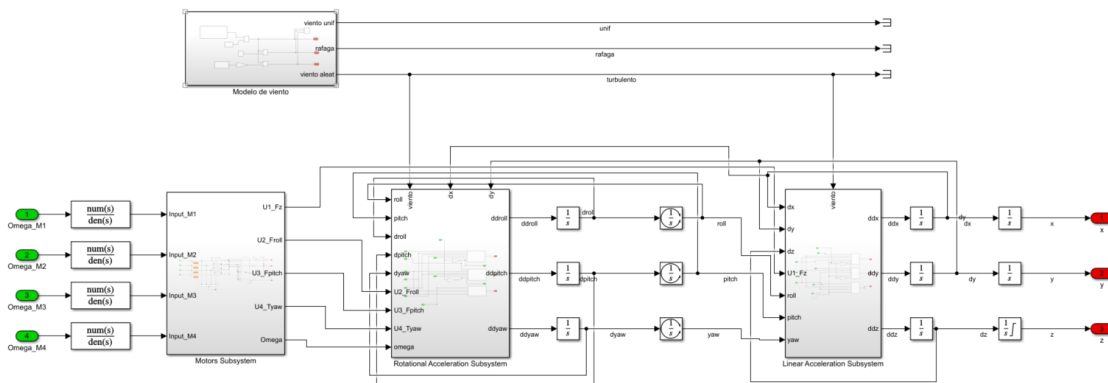


Ilustración 55- Planta del sistema.

7.1.7 LÍMITES MUNDO REAL

Una vez que la planta del modelo ha sido creada a partir de las ecuaciones dinámicas obtenidas, es importante recordar que estas ecuaciones no representan perfectamente el UAV. Por ejemplo, en muchos UAV los motores solo tienen una dirección de giro, o que la aeronave no puede alcanzar cotas positivas en el eje Z, si utilizamos el plano de referencia NED. Es por estas razones por las que hay que aplicar límites al modelo.

Primero habrá que considerar el plano de referencia en el cual se trabaja. En este modelo las direcciones X e Y quedaran libres, es decir sin acotar, mientras que la dirección Z quedara acotada entre cero y menos infinito. Esto se debe a que se utiliza el plano de referencia inercial NED, en el cual la dirección Z negativa es hacia arriba. Hay varias maneras de implementar estos límites en el modelo, como pueden ser bloques de saturación o límites de integración.

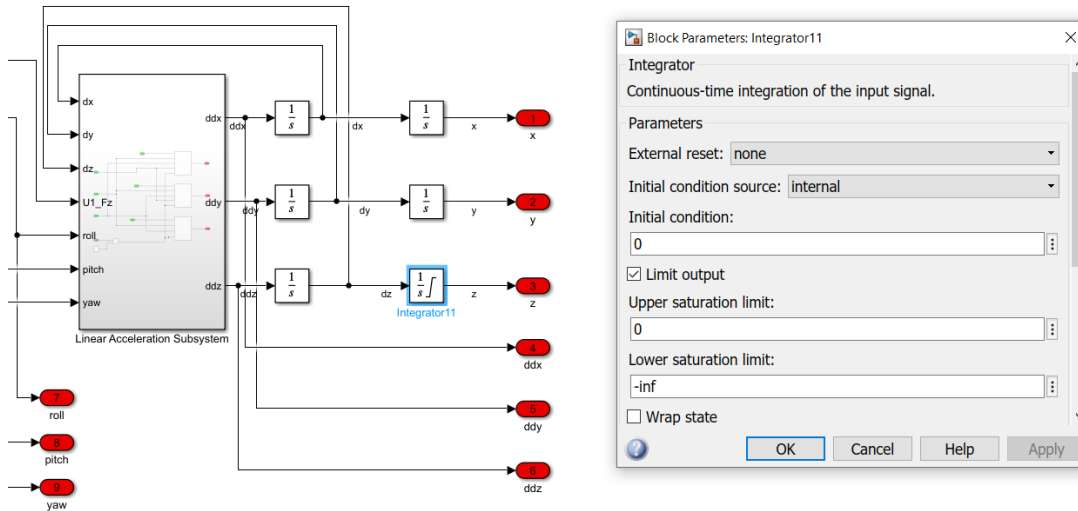


Ilustración 56- Limite del suelo.

Las siguientes limitaciones que se deberían implementar deberían ser las de las velocidades angulares de los motores. Algunos UAV tiene la capacidad de hacer girar sus motores en ambas direcciones en mitad del vuelo, sin embargo, el UAV con el que se trabaja solo tiene la capacidad de hacer girar los rotores en una sola dirección. Estos límites se deberían aplicar a las entradas de los motores de la planta. Se utilizan bloques de saturación, limitando desde cero hasta infinito. Se podría afinar más el modelo, si en vez de un límite superior de infinito se utilizará un valor debido a las limitaciones físicas del motor.

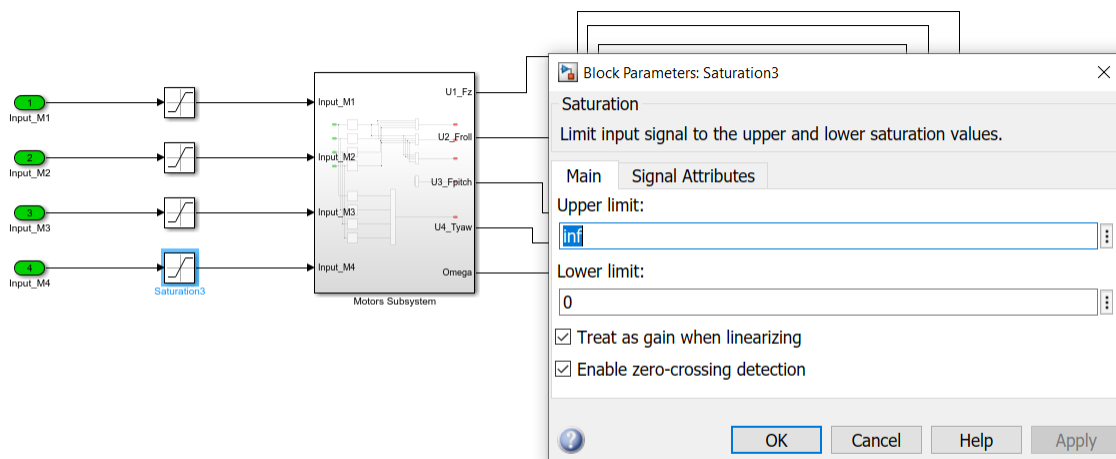


Ilustración 57- Limite sentido de giro de los motores.

Por último, los ángulos de Euler se pueden limitar entre los ángulos $-\pi$ y π . Estos límites se pueden introducir en el modelo en las integrales de las velocidades angulares.

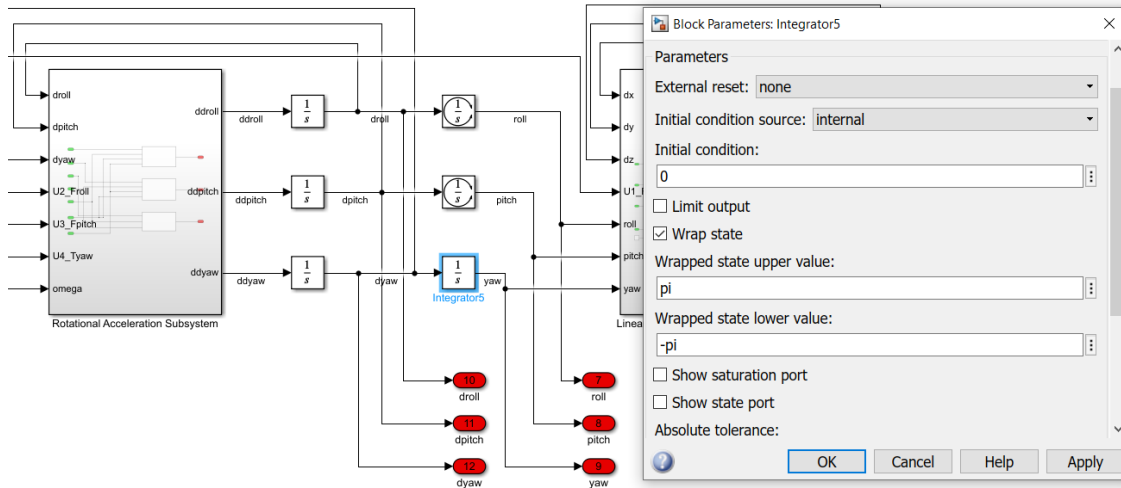


Ilustración 58- Delimitar el rango de los ángulos.

7.2 MODELO DEL CONTROLADOR

El controlador es una parte muy importante del sistema ya que sin él el sistema sería inestable por naturaleza. Existen distintos métodos para realizar el control del sistema, pero en este proyecto se van a utilizar PID's debido a que el software ArduCopter, el cual se quiere replicar, utiliza este método.

Los estados que han de ser controlados son altura (coordenada Z), coordenadas X e Y y el ángulo de Yaw. Para realizar la realimentación se obtienen los estados calculados en la planta del sistema y sirven de entrada para los distintos controladores que hay.

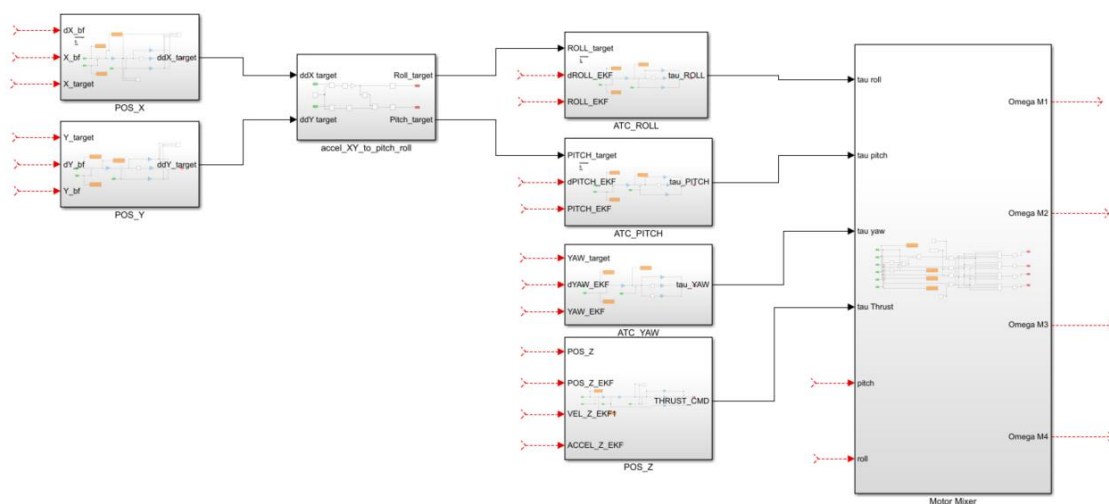


Ilustración 59- Estructura del controlador.

De forma sencilla el diagrama de un PID se puede asemejar a lo que se muestra en la siguiente ilustración.

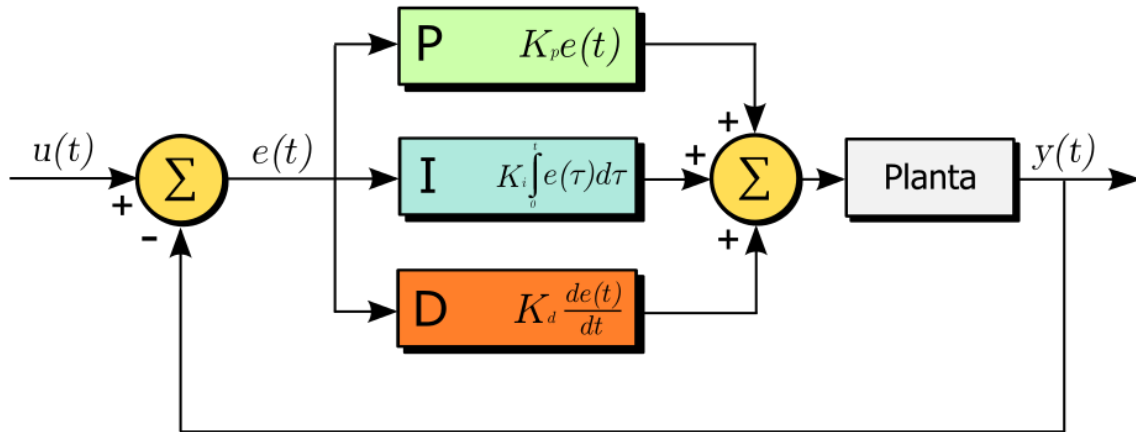


Ilustración 60- Estructura tradicional de PID.

La parte proporcional del controlador produce una ganancia general para el sistema. La parte integral produce una ganancia proporcional al error acumulado por la señal. La parte derivativa produce una ganancia relacionada con la pendiente del error, a mayor pendiente, mayor ganancia. Cada parte del controlador PID se suma para generar una entrada a la planta del sistema.

7.2.1 PID CONTROLLER

Los controladores PID se pueden representar en Simulink mediante ganancias o mediante una función de transferencia. Un controlador PID tiene una función de transferencia tal que:

$$G_c(s) = \frac{K_d s^2 + K_p s + K_i}{s}$$

En este proyecto se ha decidido modelar los PID como bloques de ganancias. El motivo es porque al autor le ha parecido una estructura más intuitiva y flexible ante cambios, más adelante se explicarán cuáles han sido estas modificaciones. La estructura de un PID convencional modelada mediante bloques de ganancias se muestra a continuación.

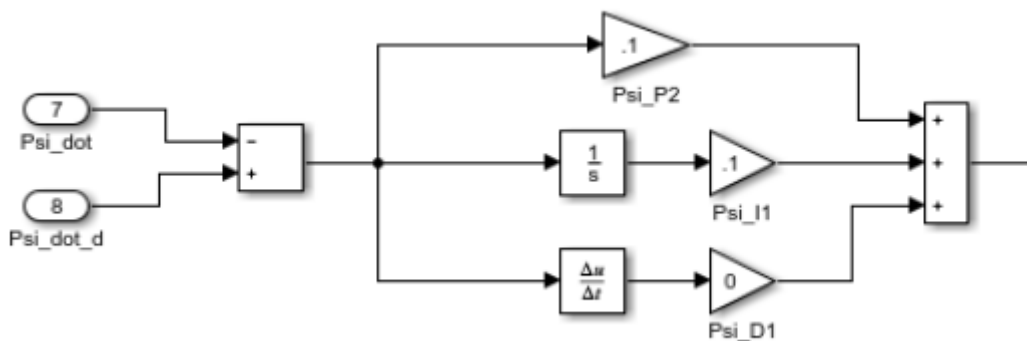


Ilustración 61- PID en Simulink.

Otro de los motivos por los cuales se ha decidido modelar los PID así es porque se ha considerado importante poder conocer la contribución y el comportamiento de cada una de las partes del PID.

Una de las señales de entrada más comunes a la hora de realizar el ajuste de los controladores e incluso a la hora de ejecutar una trayectoria, sería una entrada escalón. Uno de los problemas de trabajar con controladores PID y entradas escalón es que el termino derivativo puede causar impulsos en el sistema. Si por ejemplo para el instante de tiempo cero, se aplica un escalón a la altura, el error será de la magnitud del escalón, y la pendiente del error será vertical. Como el termino derivativo calcula la pendiente de la señal, esto provocara un impulso en el PID lo cual no es deseable. Por este motivo se utiliza una estructura PI-D, ya que esta estructura de controlador utiliza la pendiente del estado y no del error. Para introducir este controlador de forma adecuada, hay que procurar que estado deseado que se quiere alcanzar este expresado en las mismas unidades que los estados que salen de la planta.

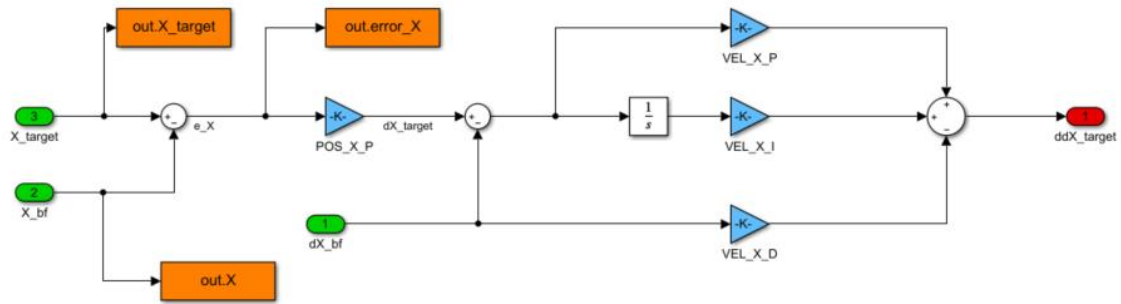


Ilustración 62- Controlador de posición X.

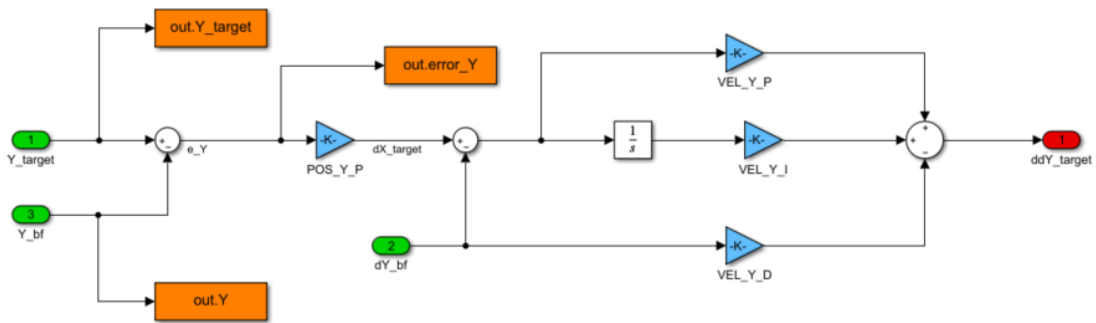


Ilustración 63- Controlador de posición Y.

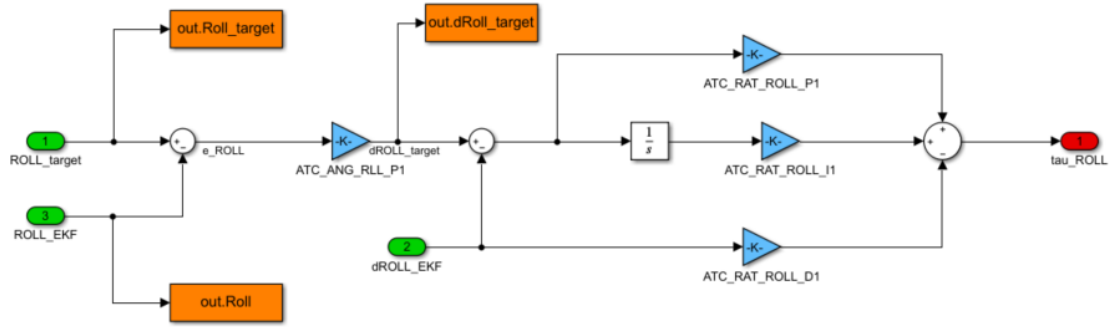


Ilustración 64-Controlador del roll.

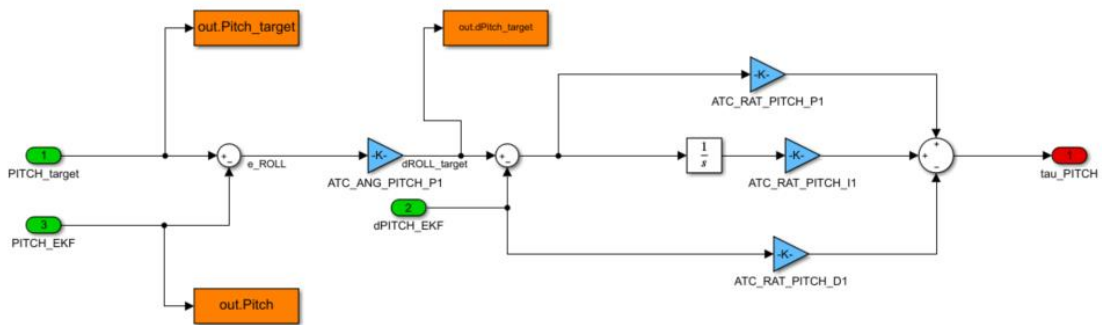


Ilustración 65-Controlador de pitch.

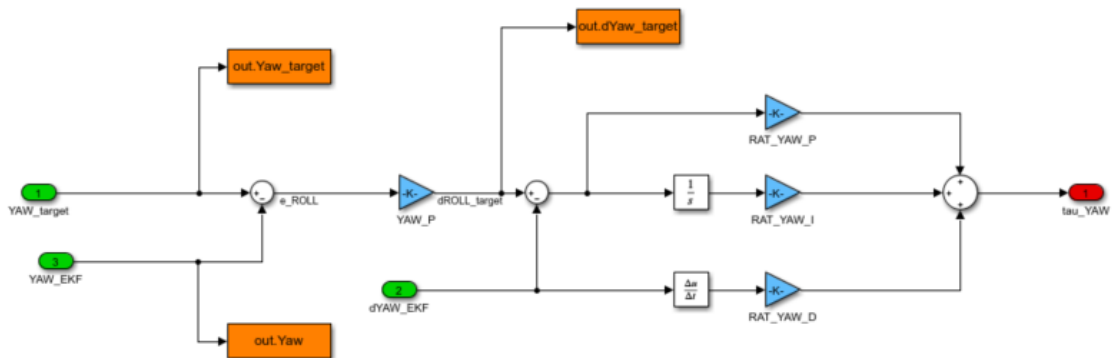


Ilustración 66-Controlador de yaw.

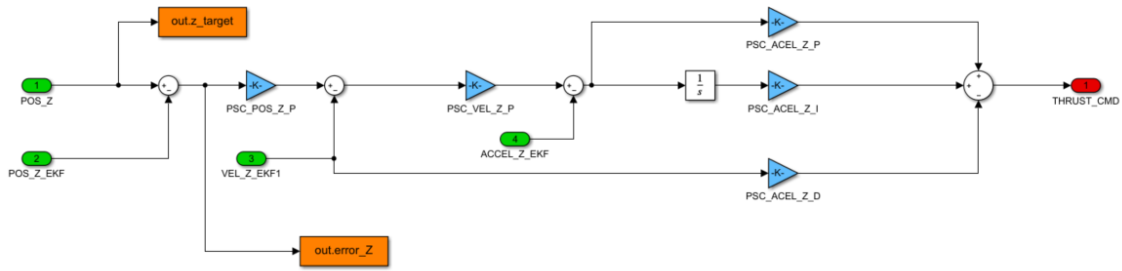


Ilustración 67- Controlador de altura.

Una vez que se han creado todos los controladores es necesario calcular las velocidades angulares que han de ser comandadas a cada motor. Para ello basta con reproducir las ecuaciones de las entradas de empuje de forma inversa.

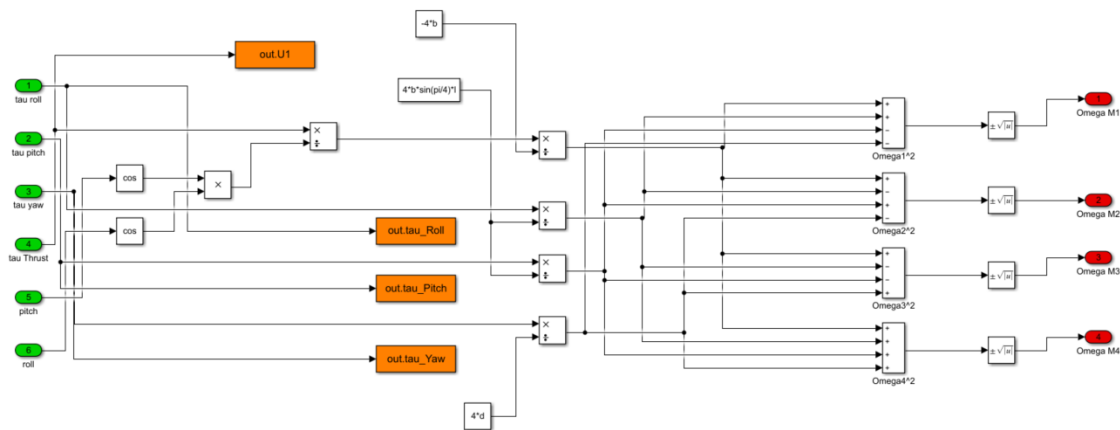


Ilustración 68- Subsistema Motor Mixer.

8 AJUSTE CONTROLADOR

Una vez desarrollado el modelo completo del sistema, se deberá realizar un ajuste de las ganancias de los controladores hasta conseguir la respuesta deseada en el sistema. Como se ha explicado antes, se ha decidido separar o agrupar ciertos controladores en función de las similitudes que tengan entre sí. El orden en el cual se ajustan los controladores no es arbitrario debido a la arquitectura en cascada del controlador. Se empezará por ajustar el controlador de la posición en Z ya que este tiene una dinámica independiente de la orientación. A continuación, se deberán ajustar los controladores del bucle interno, es decir, los de orientación y finalmente se ajustarán los controladores de posición en X e Y. Se hará un ajuste empírico y el objetivo será que la variable de estado siga lo mejor posible a la referencia sin preocuparnos de momento por si las ganancias son demasiado grandes o no, no existen unas especificaciones de tiempo de estabilización o sobreoscilación en concreto, simplemente se quiere ver como de bien se ajusta este controlador al sistema idealmente.

El proceso por el cual se ajustarán las ganancias será el siguiente:

- Simular una entrada escalón en la referencia del controlador.
- Ajustar las ganancias proporcionales
- Ajustar las ganancias derivativas
- Ajustar las ganancias integrales

8.1 CONTROLADOR POSICIÓN Z

Se comienza por introducir entrada escalón de un metro en la referencia del controlador del eje Z. Los valores que se han seleccionado tras un proceso de iteración de las ganancias manualmente son los que aparecen en la tabla 5.

Tabla 5- Ganancias controlador altura.

CONTROL POSICIÓN Z	
POS_Z_P	10
VEL_Z_P	10
ACCEL_Z_P	10
ACCEL_Z_I	0
ACCEL_Z_D	100

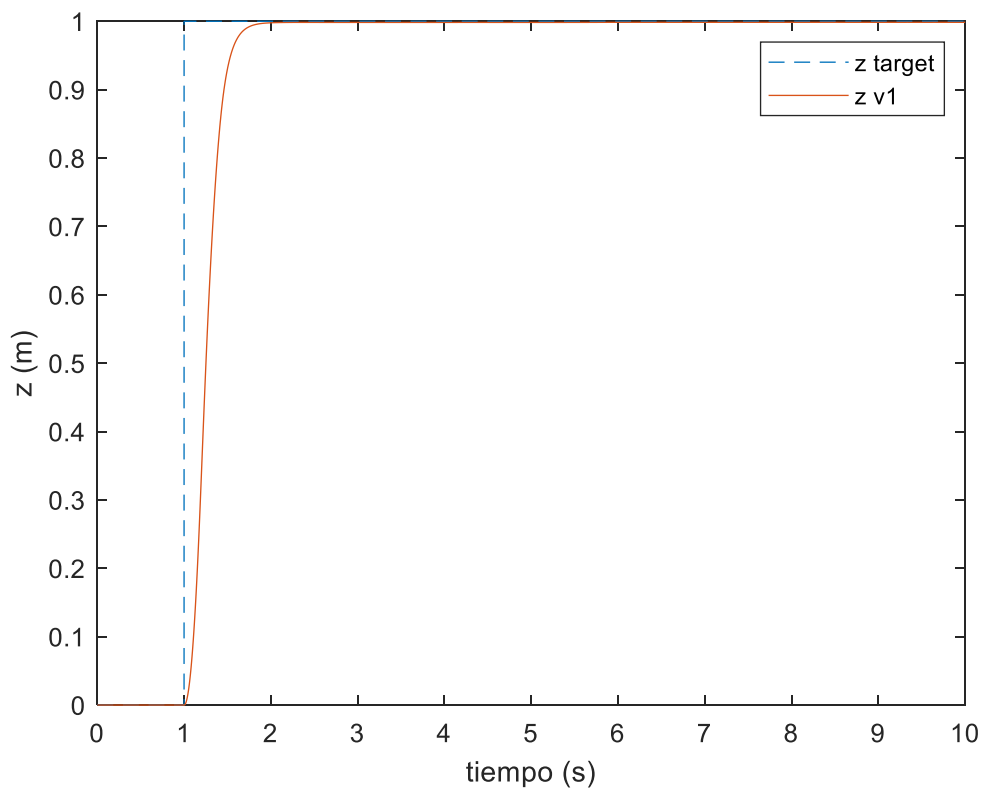


Ilustración 69- Respuesta ante escalón en Z.

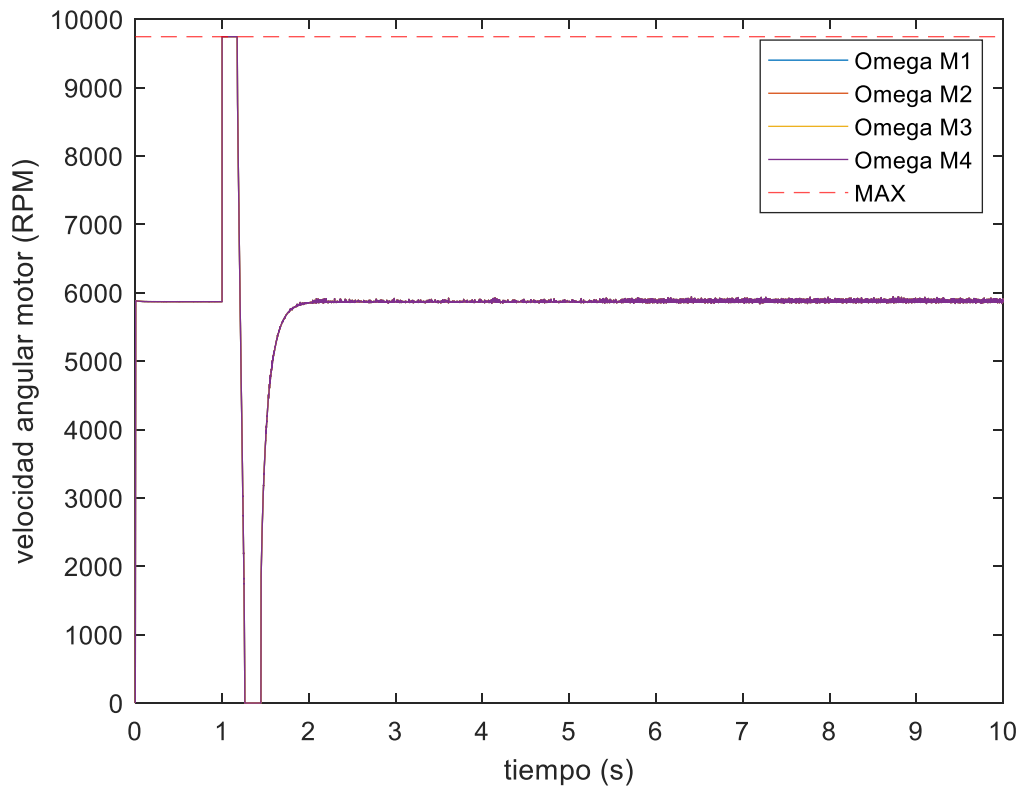


Ilustración 70-RPM de los motores.

Llama la atención que no hace falta hacer uso de la parte integral del controlador para conseguir un error en posición bajo. Aunque estas condiciones son ideales y seguramente habría que ajustar estas ganancias para la realidad. Pero el hecho de que no haga falta hacer uso de la parte integral habla bien de la adecuación de esta arquitectura de controlador.

8.2 CONTROLADOR YAW

Se sigue el mismo procedimiento para el controlador del Yaw, y se obtienen resultados igual de satisfactorios.

Tabla 6- Ganancias controlador de Yaw.

CONTROL YAW	
YAW_P	10
RAT_YAW_P	10
RAT_YAW_I	0
RAT_YAW_D	1

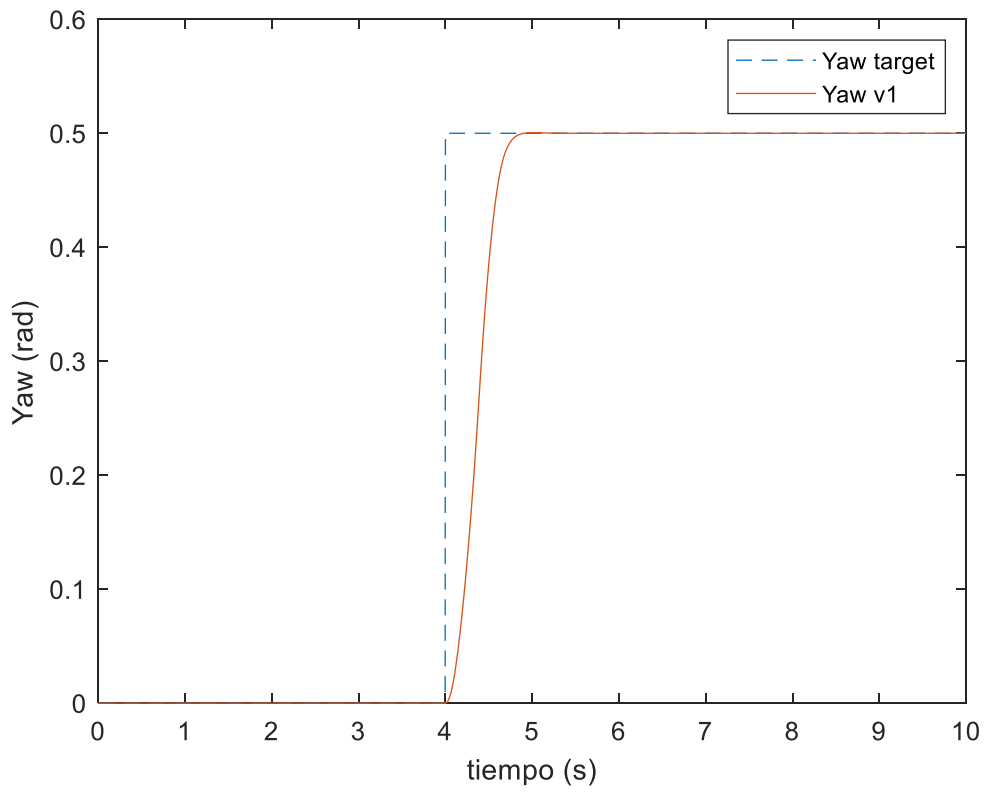


Ilustración 71- Respuesta ante escalón en Yaw.

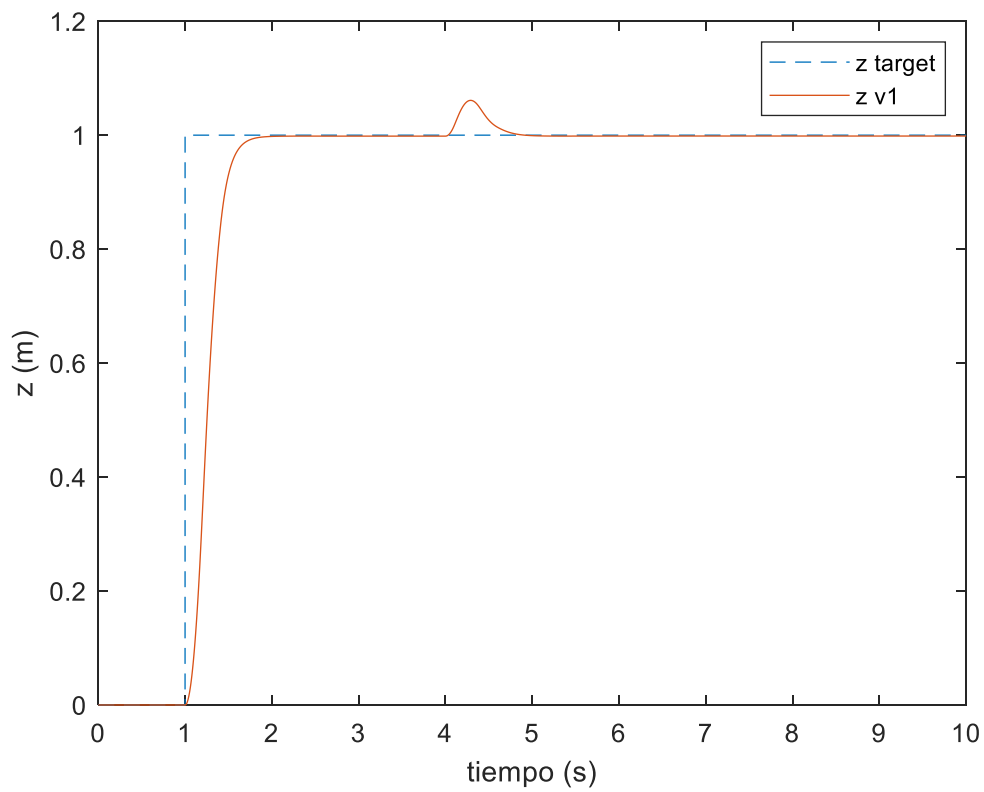


Ilustración 72-Respuesta de Z ante escalón en Yaw.

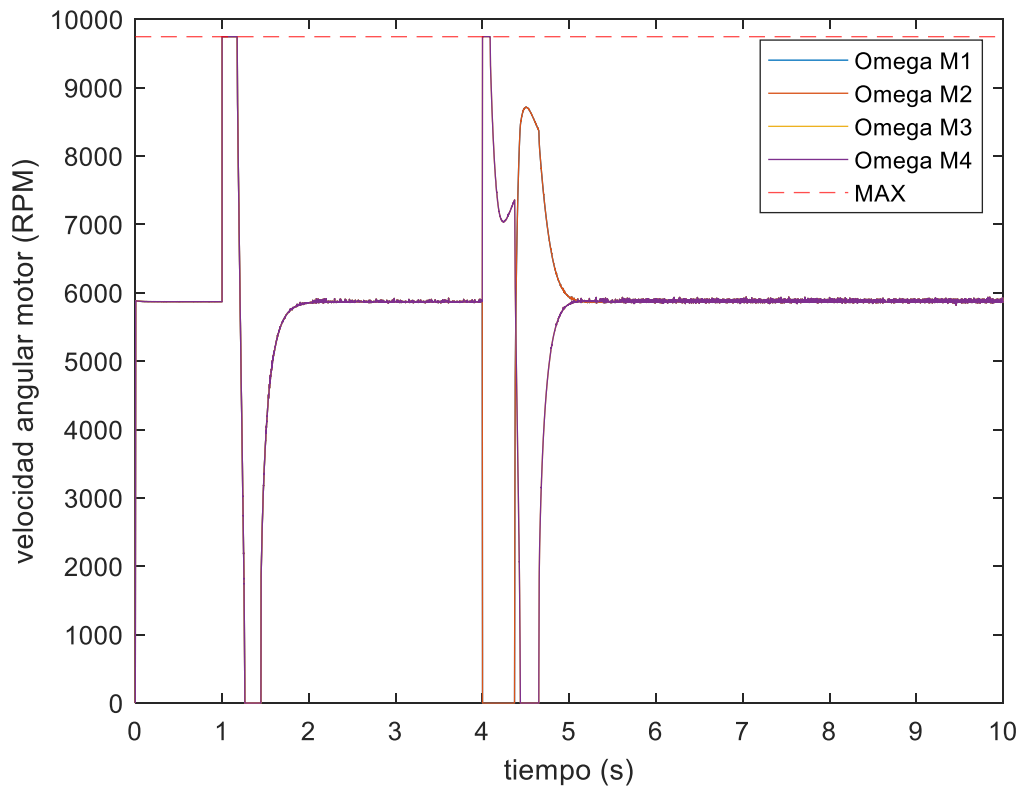


Ilustración 73-RPM de los motores.

8.3 CONTROLADOR DE ROLL Y PITCH

Una vez ajustado los controladores de altura y de yaw, se continúa ajustando el controlador de pitch y roll. Ambos controladores son similares y dado que el dron se considera simétrico, bastara con ajustar uno de los dos controladores e introducir el mismo valor de ganancias en el otro. En este caso se ha ajustado el de roll, pero se han comprobado ambos.

Tabla 7-Guanacias del controlador de Roll.

CONTROL ROLL	
ROLL_P	10
RAT_ROLL_P	10
RAT_ROLL_I	0
RAT_ROLL_D	1

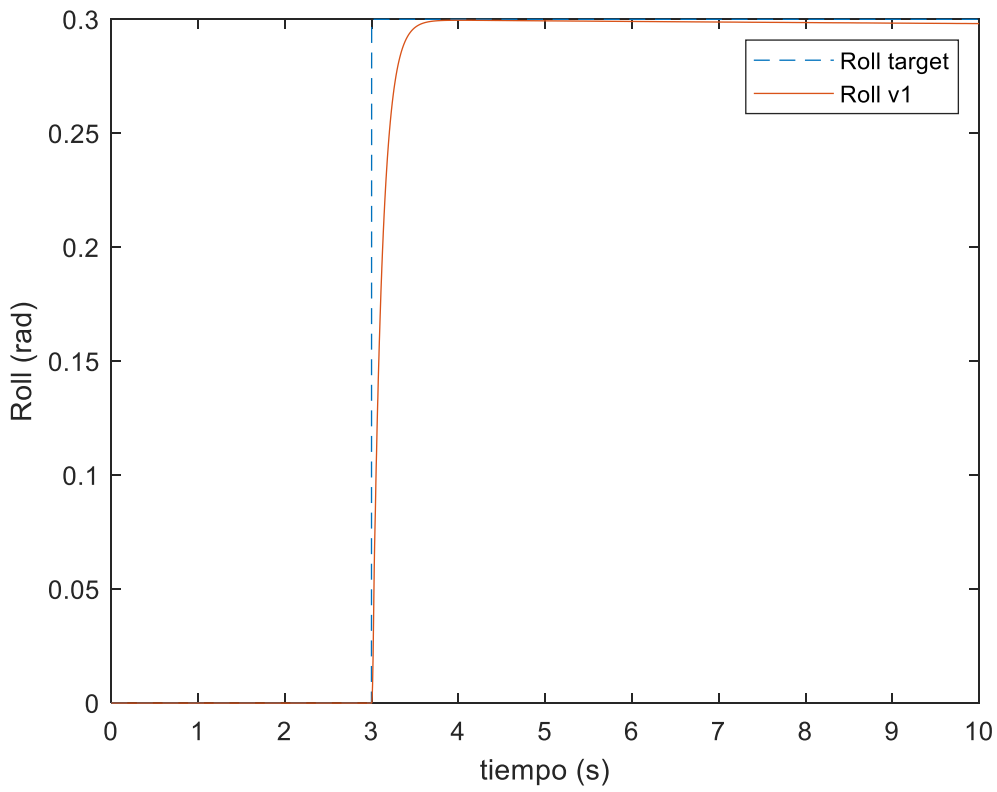


Ilustración 74- Respuesta ante escalón de roll.

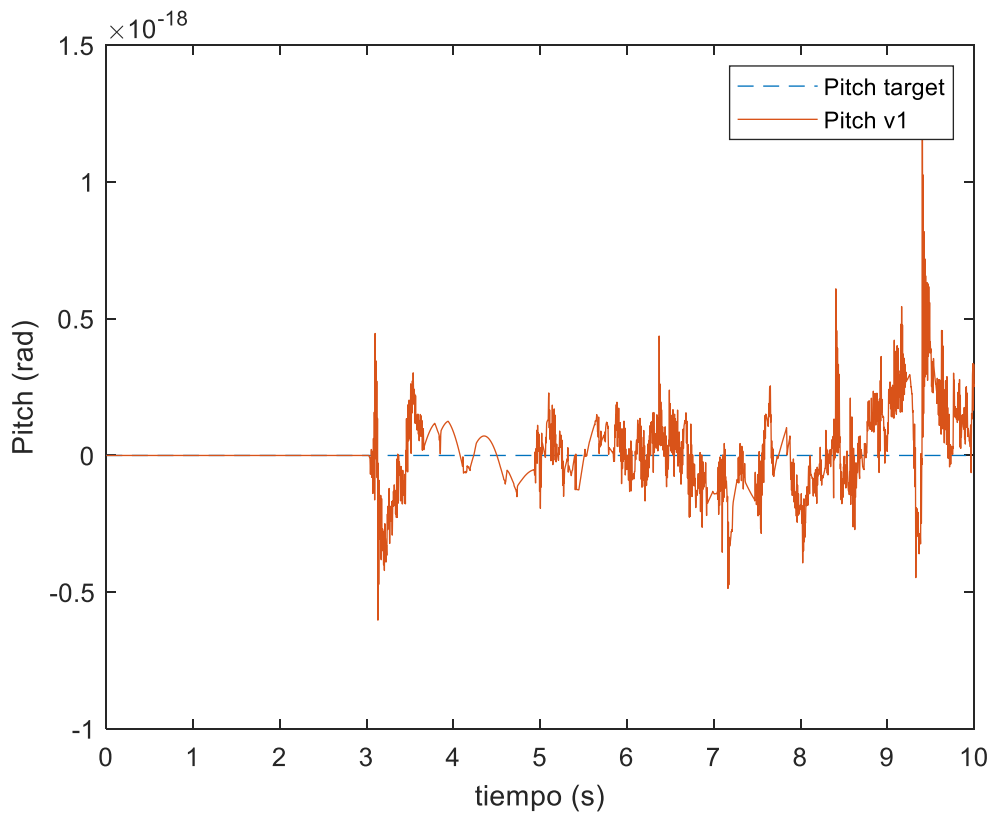


Ilustración 75- Respuesta pitch ante escalón pitch.

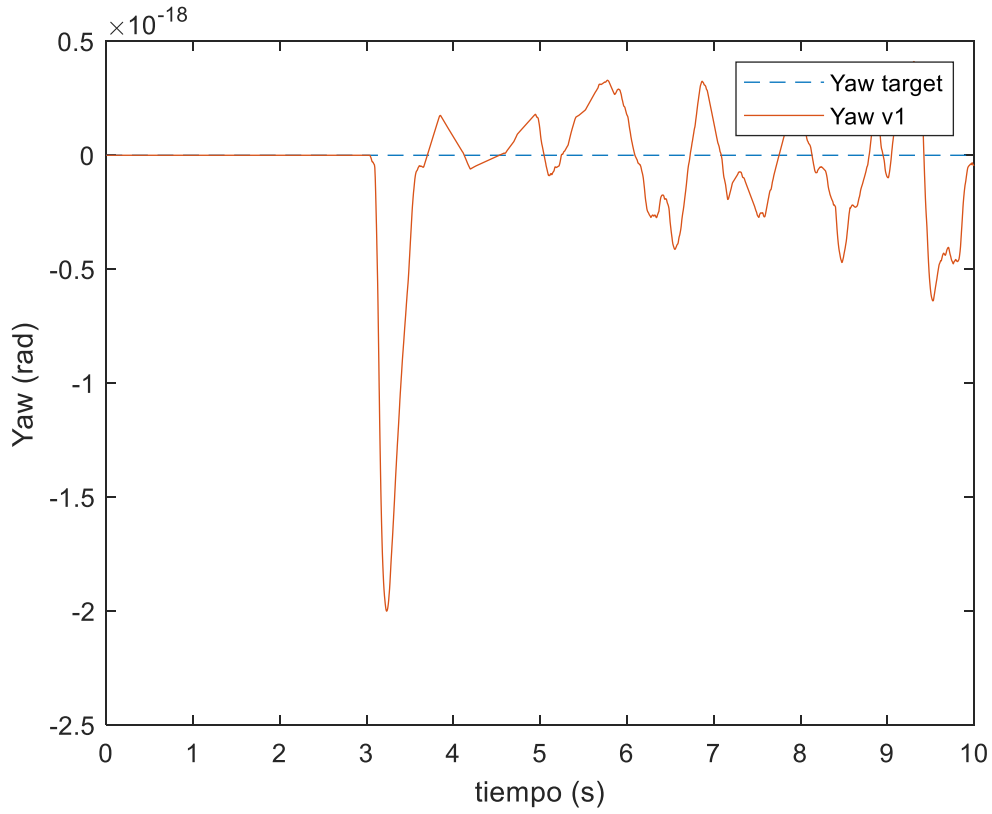


Ilustración 76-Respuesta de Yaw ante escalón de roll.

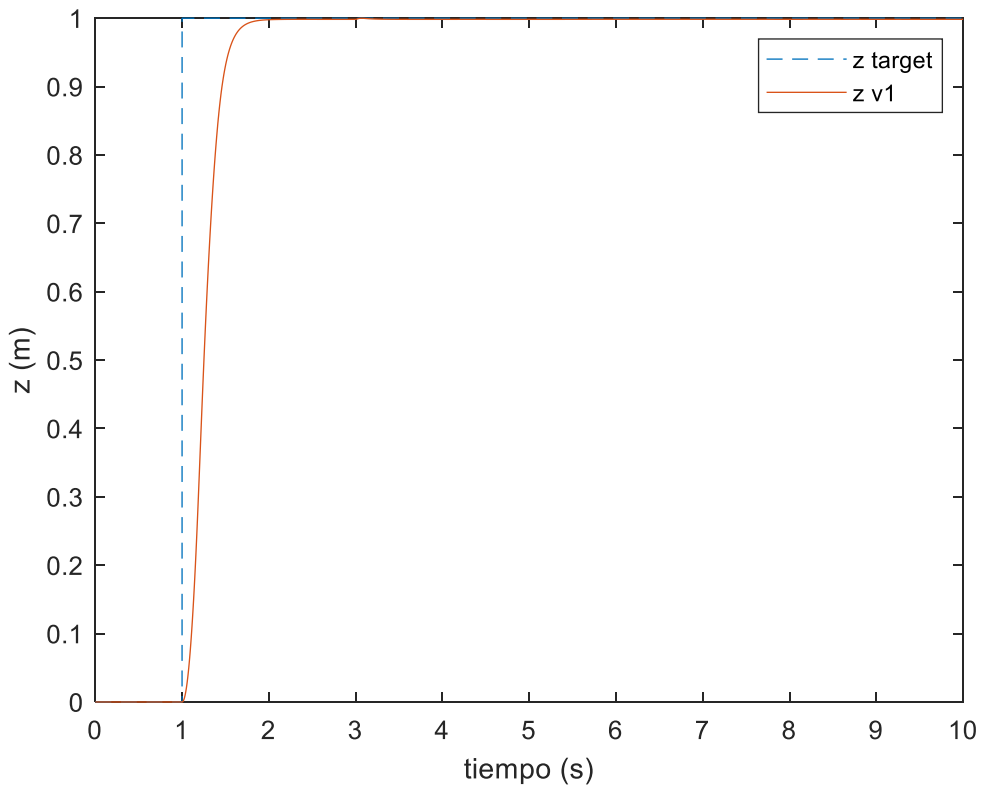


Ilustración 77-Respuesta de Z ante escalón de roll.

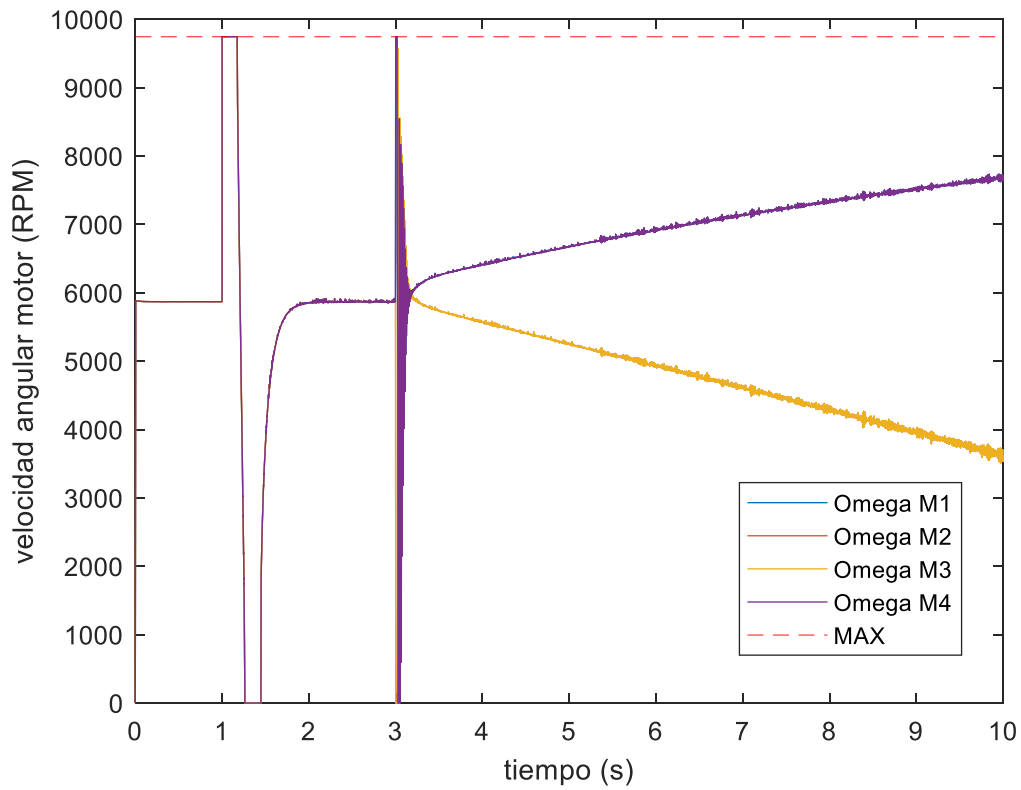


Ilustración 78-RPM de motores.

8.4 AJUSTE POSICIÓN HORIZONTAL

Una vez ajustado los controladores de altura y orientación, solo quedara ajustar los controladores de posición horizontal. Al igual que como sucede con el roll y el pitch, estos controladores de posición X y Y compartirán el valor de sus ganancias.

Tabla 8- Ganancias del controlador de posición horizontal.

CONTROL XY	
POS_XY_P	2
VEL_XY_P	4
VEL_XY_I	2
VEL_XY_D	1

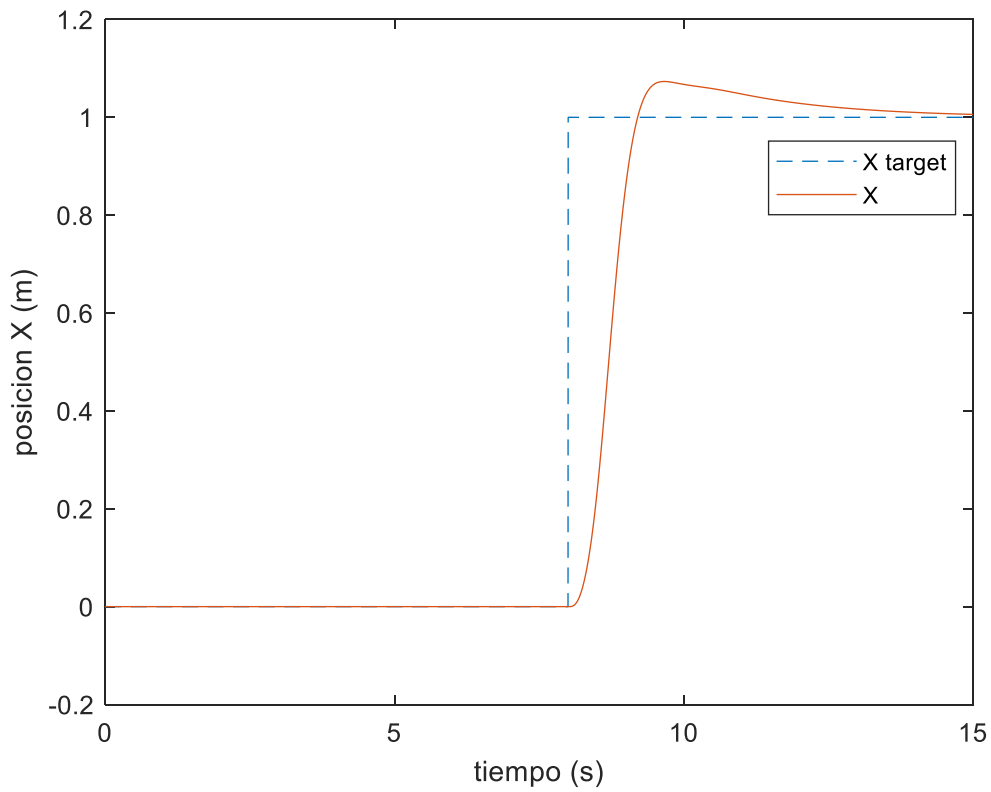


Ilustración 79- Respuesta ante escalón en X.

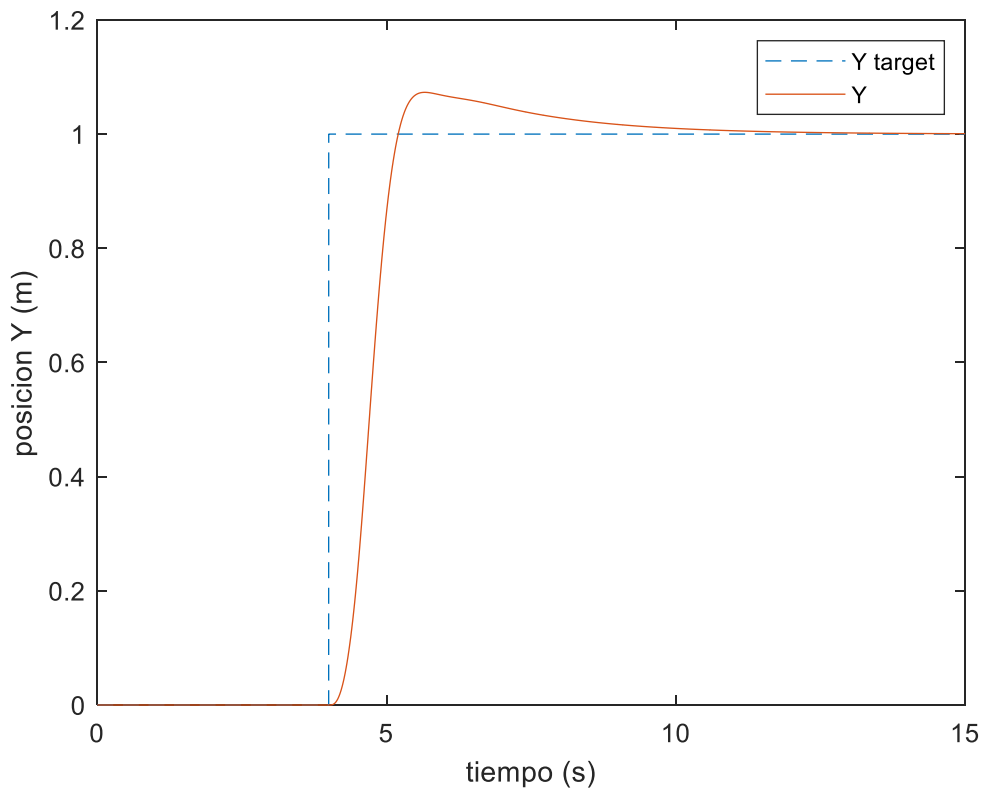


Ilustración 80-Respuesta ante escalón en Y.

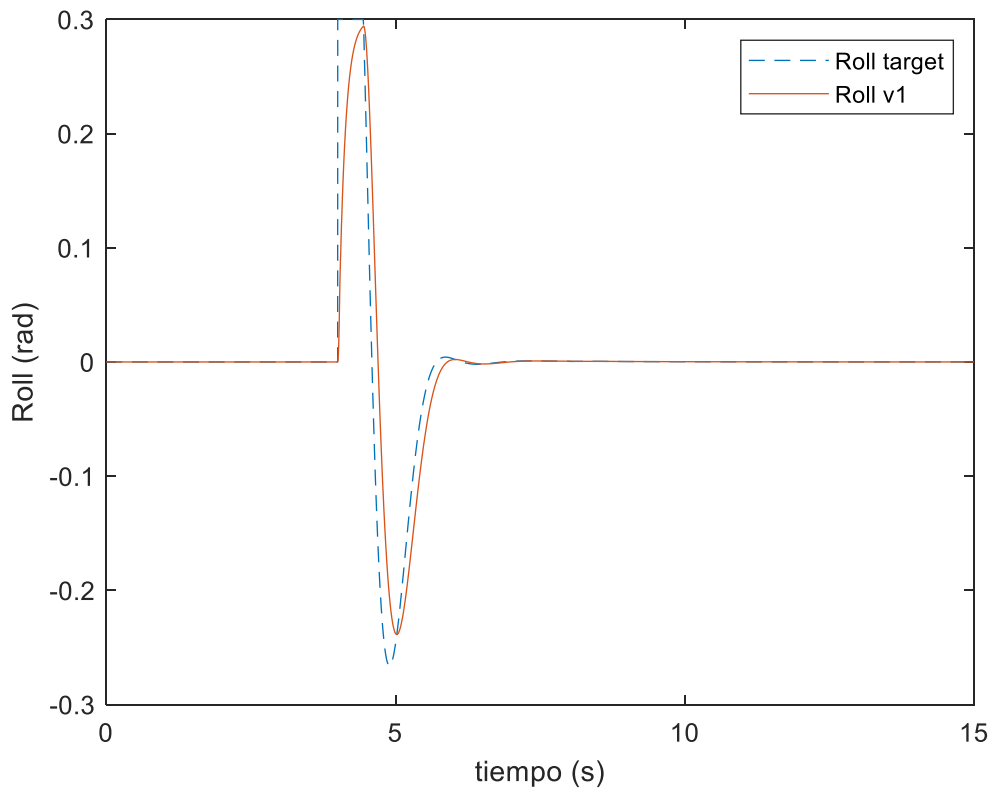


Ilustración 81- Respuesta del Roll.

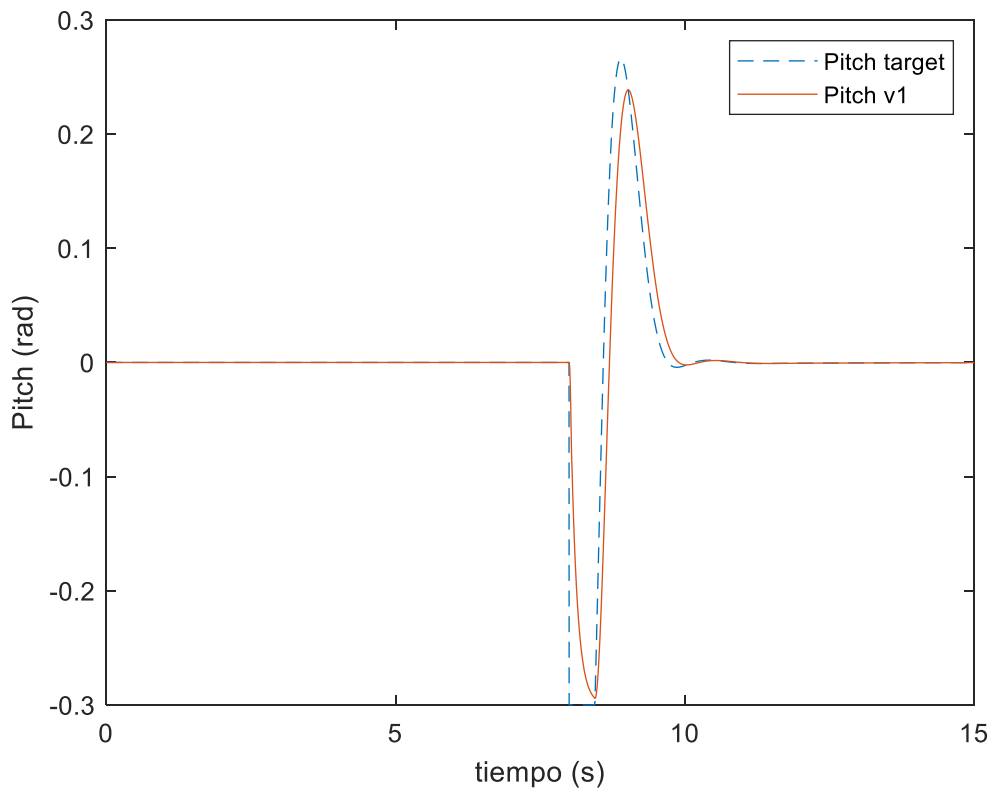


Ilustración 82-Respuesta de Pitch.

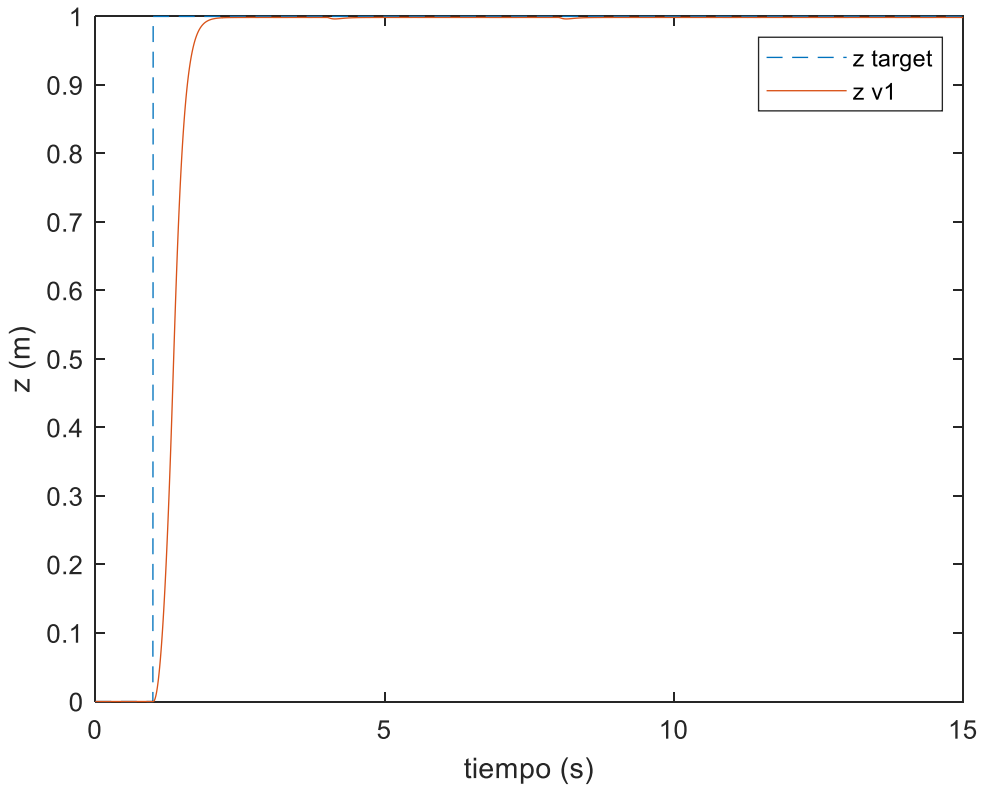


Ilustración 83-Respuesta de Z.

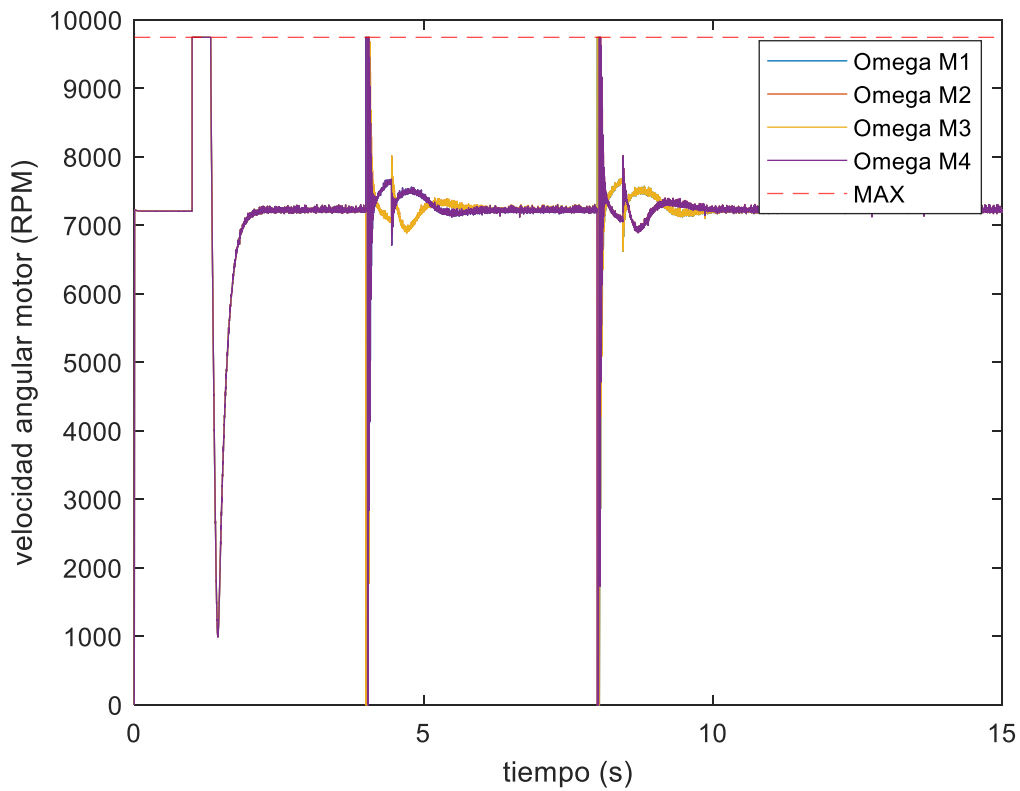


Ilustración 84-RPM motores.

Como se puede observar es más difícil ajustar la respuesta del sistema sobre la posición horizontal que sobre el resto de los controladores. Es por eso por lo que estos controladores serán los limitantes del sistema y tomara especial importancia la señal de entrada que tengan. Esto se explicará más adelante ya que la señal comandada influye mucho en el rendimiento del sistema de control.

8.5 TRAYECTORIA

Una vez ajustados los controladores ante entradas escalón se comprobará el rendimiento del sistema ante unas señales de entrada que se adecuan más a las señales comandadas en un UAV real. La única excepción será la señal comandada para la altura, ya que el excelente rendimiento del controlador de altura hará que no afecte tanto la entrada escalón.

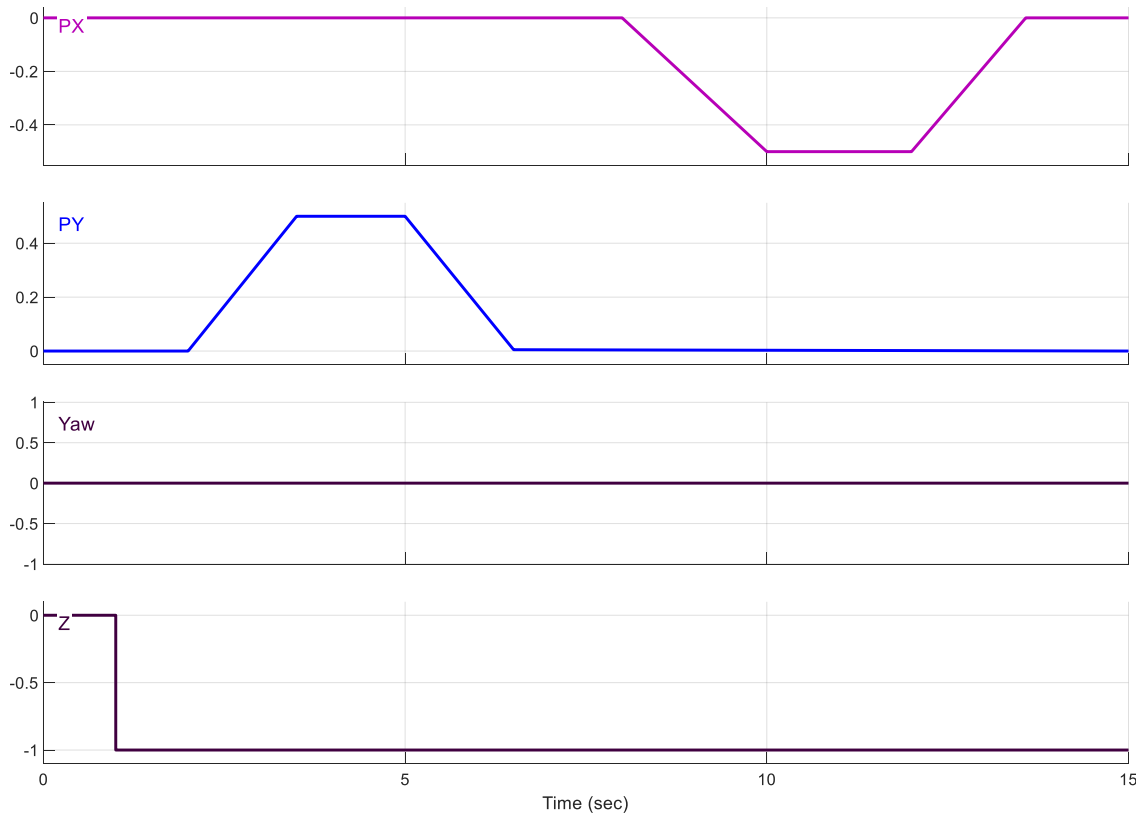


Ilustración 85-Trayectoria deseada.

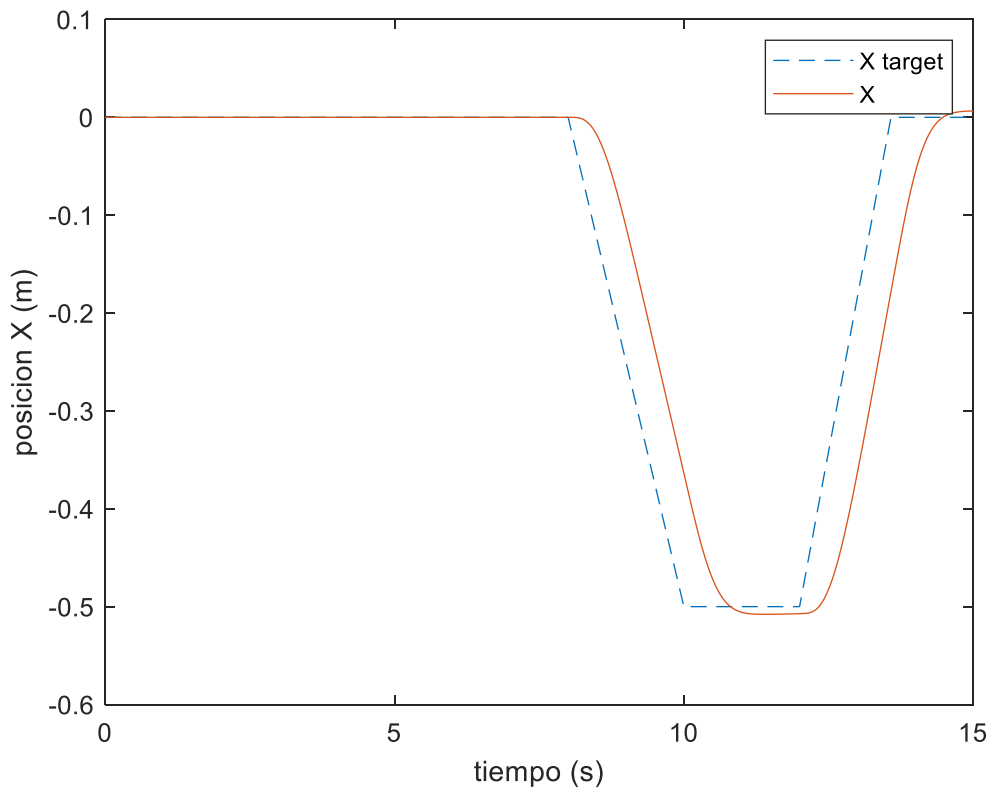


Ilustración 86-Respuesta en X.

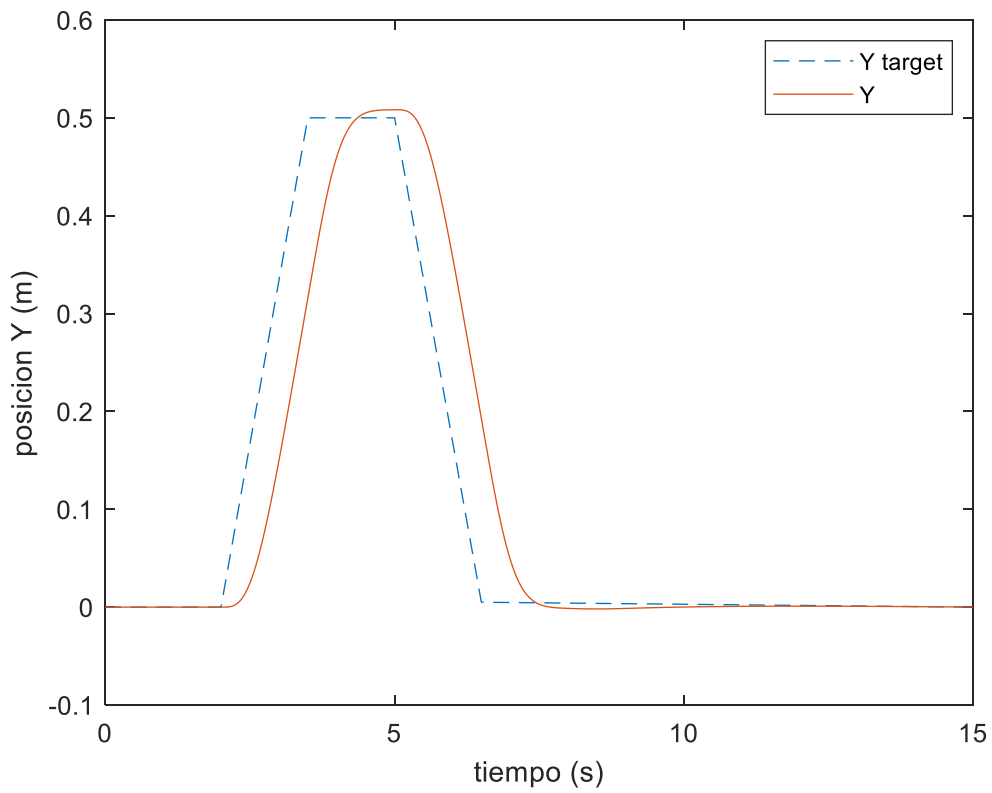


Ilustración 87-Respuesta en Y.

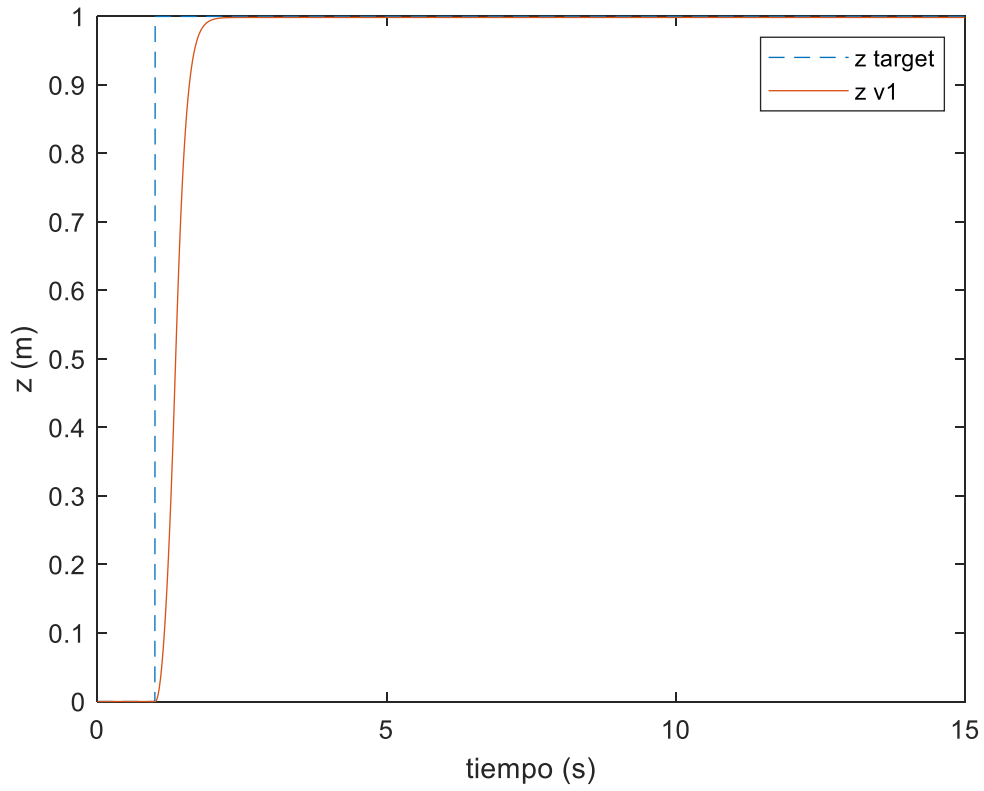


Ilustración 88-Respuesta en Z.

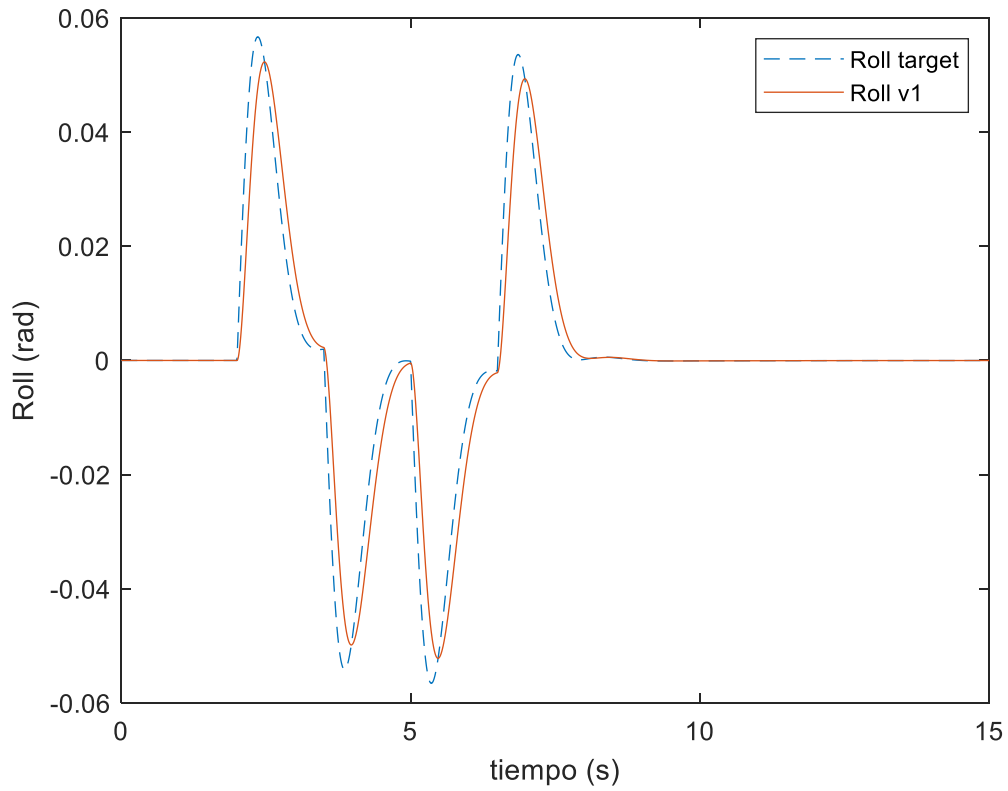


Ilustración 89-Respuesta de roll.

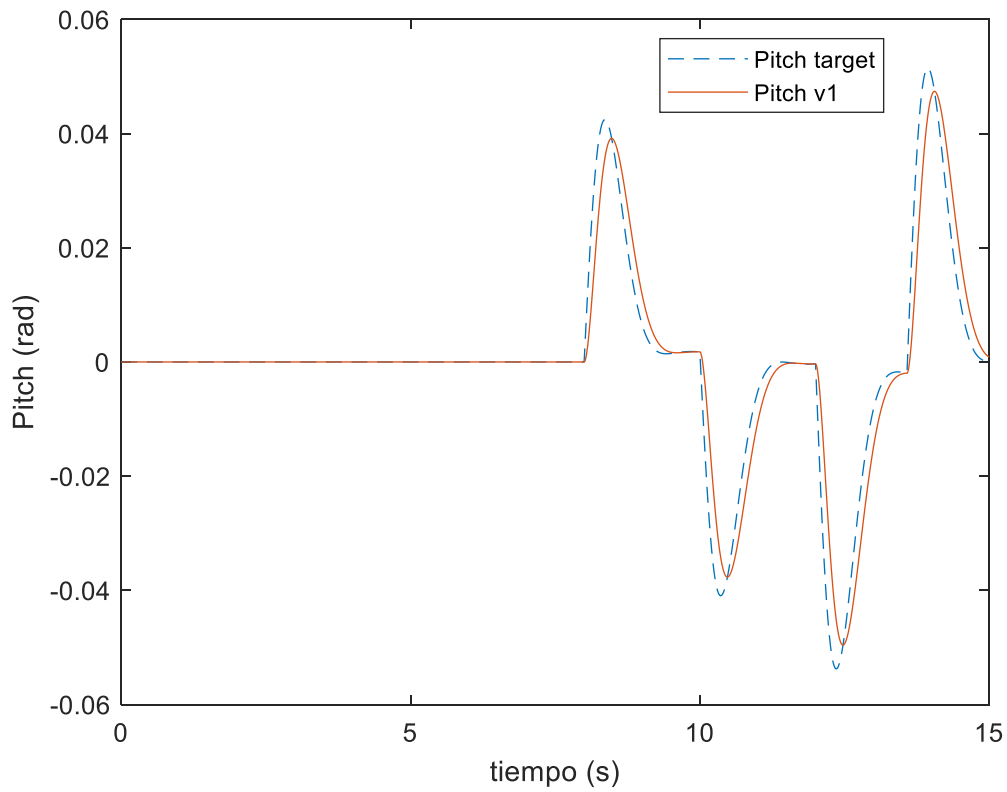


Ilustración 90-Respuesta de pitch.

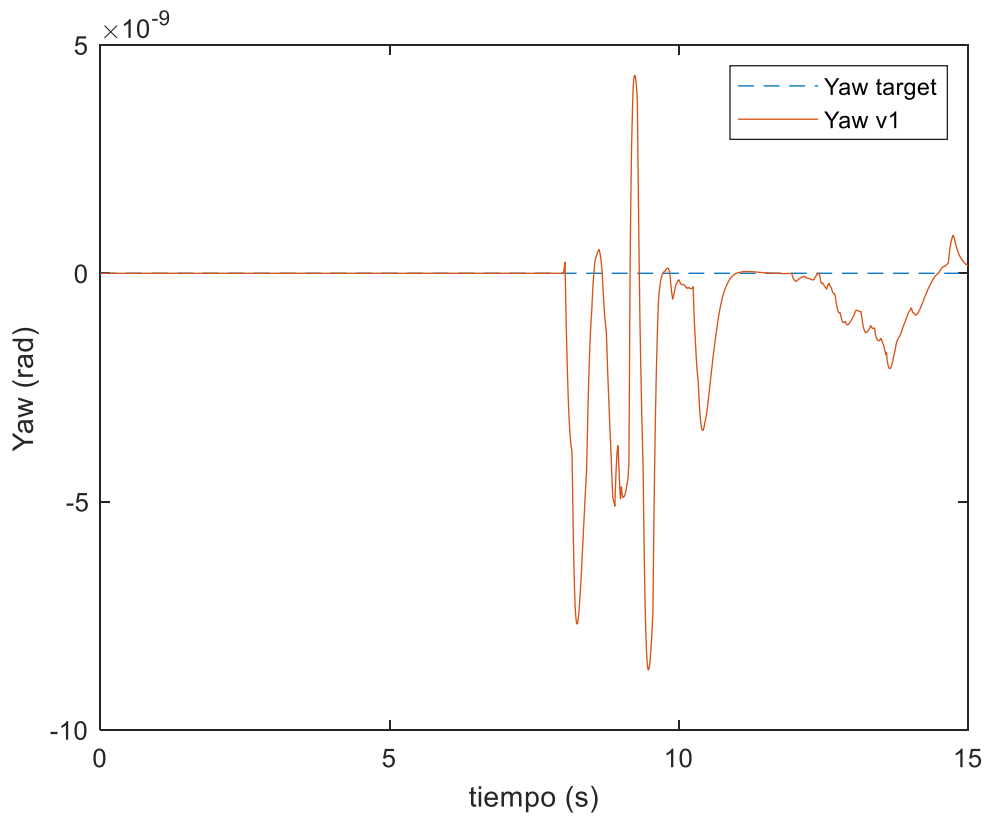


Ilustración 91-Respuesta de Yaw.

Se puede ver como los controladores de posición X e Y siguen a la señal comandada con un cierto retraso, pero aun así con una precisión más que aceptable. Por otro lado, el controlador de altura sigue la señal comandada de forma excelente y eso que se ha comandado una entrada escalón, la cual afecta negativamente al rendimiento. Por último, hay que comentar que los controladores de orientación siguen perfectamente las señales comandadas, y con un retraso mucho menor que los controladores de posición. En definitiva, se concluye que el sistema de control se ha ajustado de forma adecuada ya que en general todos sus controladores tienen un buen rendimiento al seguir las señales comandadas.

9 VALIDACIÓN

Una vez terminado el modelo y ajustado el controlador, el siguiente paso será comprobar como de buena es la simulación virtual con respecto a la realidad. Este es un paso fundamental ya que el objetivo de realizar el modelo del sistema completo es el de procurar representar con la mayor exactitud posible lo que ocurriría en un vuelo real.

Para la realización de la validación se ha seguido la siguiente metodología.

1. Se ha realizado una misión comandada de forma autónoma con el UAV real.
2. Se han obtenido los datos de la telemetría del piloto automático que hacen falta para recrear la trayectoria y para comparar los resultados de la simulación y la realidad. Estos datos son:
 - a. Trayectoria comandada que sale del módulo de navegación y que alimenta al sistema de control. Coordenadas X, Y, Z y ángulo de Yaw comandados.
 - b. Variables de estado que se desean comparar y son, coordenadas X, Y, Z reales y ángulos de Roll, Pitch y Yaw.
3. Se ha tenido que realizar operaciones de procesado de datos. Estas han sido fundamentales ya que las distintas variables de estado tienen distintas frecuencias de muestreo y es por eso por lo que se han tenido que juntar tablas en base al tiempo relativo de la operación y más tarde interpolar las series temporales para que no hubiera valores nulos.
4. Por último, se han incorporado las series temporales de las variables comandadas como señales dentro de un bloque de Simulink y así asegurar que tanto el UAV real como el virtual son comandados por la misma trayectoria.
5. Finalmente se han obtenido unos indicadores sobre la adecuación de la simulación y la realidad, que son los errores de posición de X, Y, y Z con respecto al vuelo real.

$$err_X^{MODELO} = \frac{|err_X^{SIM} - err_X^{real}|}{n}$$

$$err_Y^{MODELO} = \frac{|err_Y^{SIM} - err_Y^{real}|}{n}$$

$$err_Z^{MODELO} = \frac{|err_Z^{SIM} - err_Z^{real}|}{n}$$

Tabla 9- Parámetros del modelo.

Parámetro	Unidad	Valor	Descripción
Masa	kg	1,663	Masa del vehículo
L brazo	m	0,24	Distancia horizontal entre rotor y el centro de gravedad
Ixx	kg m ²	0,024	Momento inercia principal eje X
Iyy	kg m ²	0,024	Momento inercia principal eje Y
Izz	kg m ²	0,038	Momento inercia principal eje Z
A_xyz	kg/s	0,1 - 0,11	Resistencia traslacional aire
M_aero	kg m/s	0,1 - 0,15	Resistencia rotacional aire
b	kg s ²	7,29E-07	Coefficiente de Thrust
d	Nm s ²	7,2E-07	Coefficiente de Drag
Jr	kg m ²	6,5E-07	Inercia del rotor
RPM max	RPM	9746	Velocidad angular máxima de los rotores

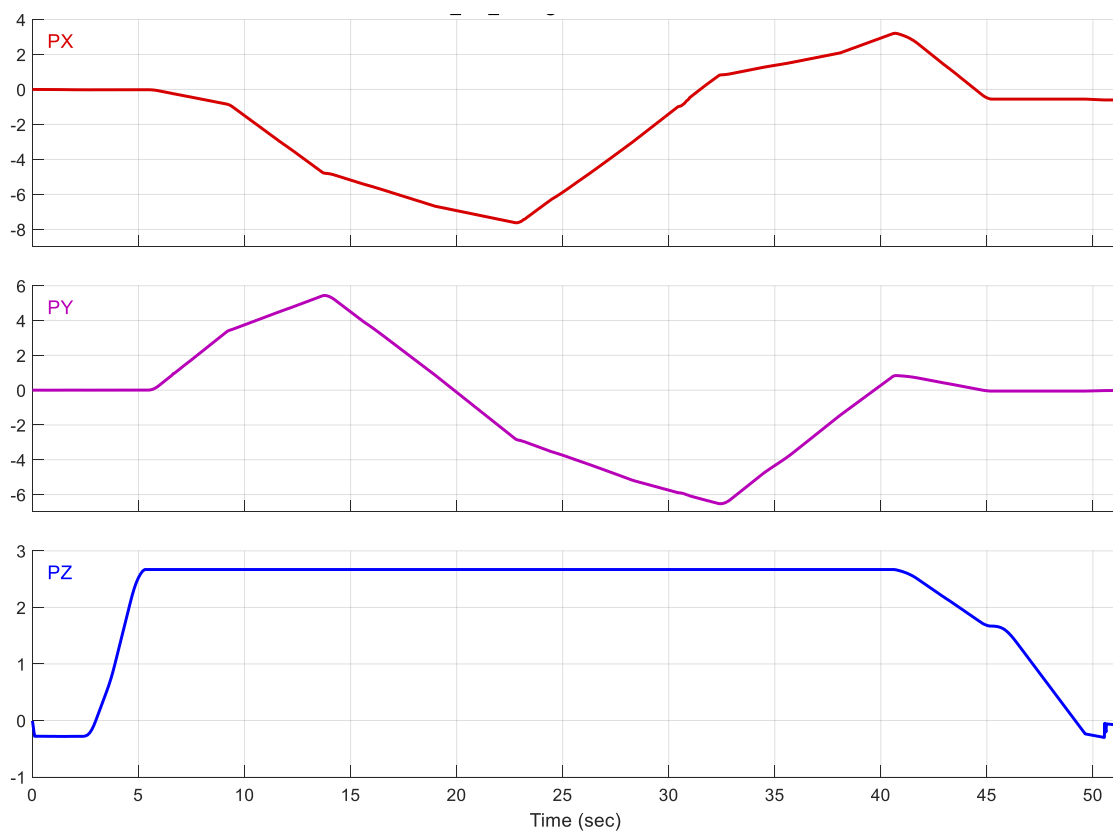


Ilustración 92-Trayectoria de validación obtenida de un vuelo real.

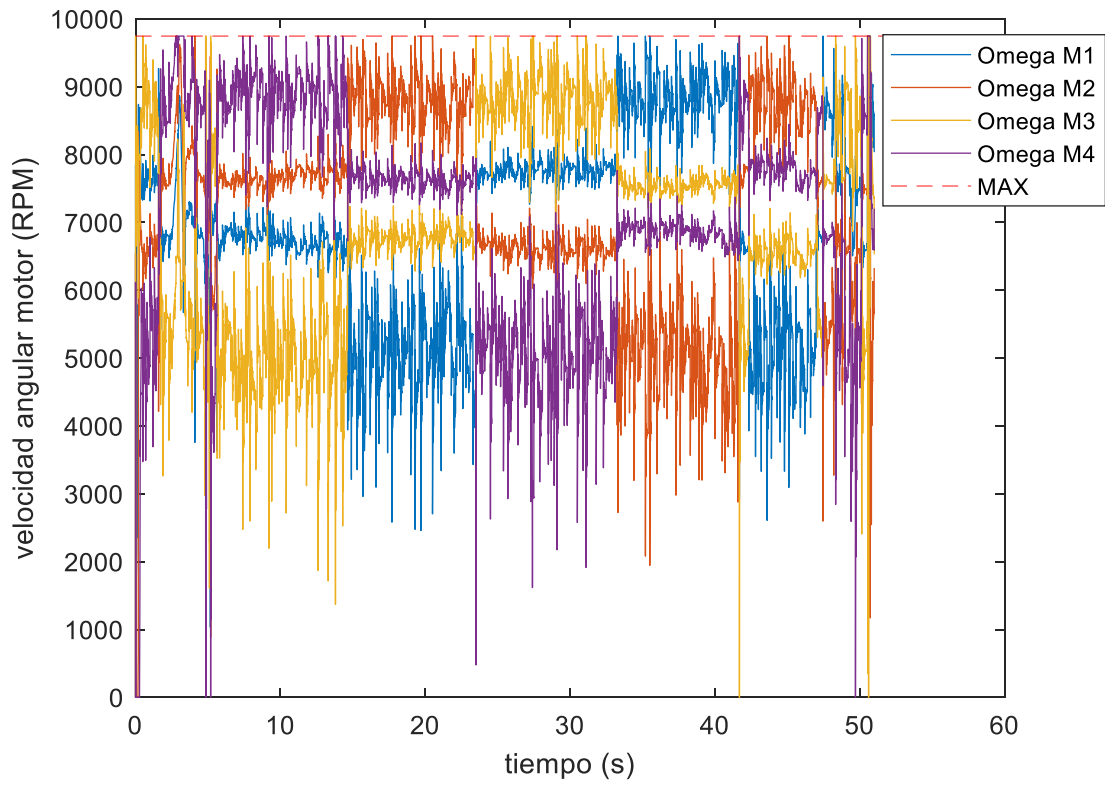


Ilustración 93-RPM motores.

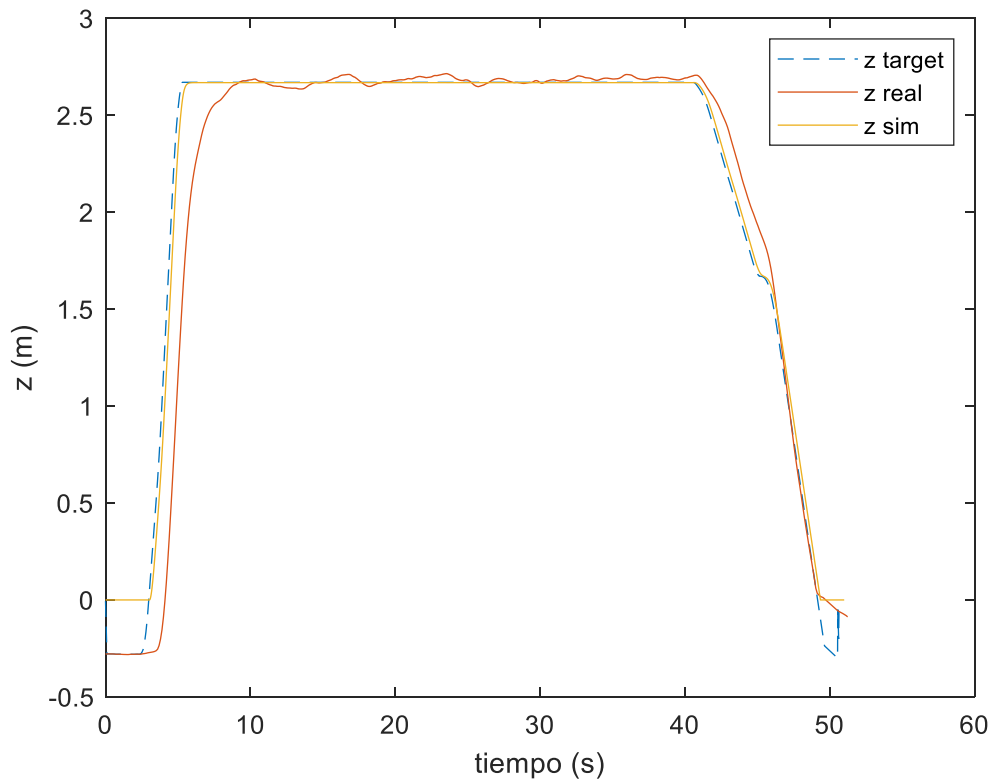


Ilustración 94-Coordenada Z.

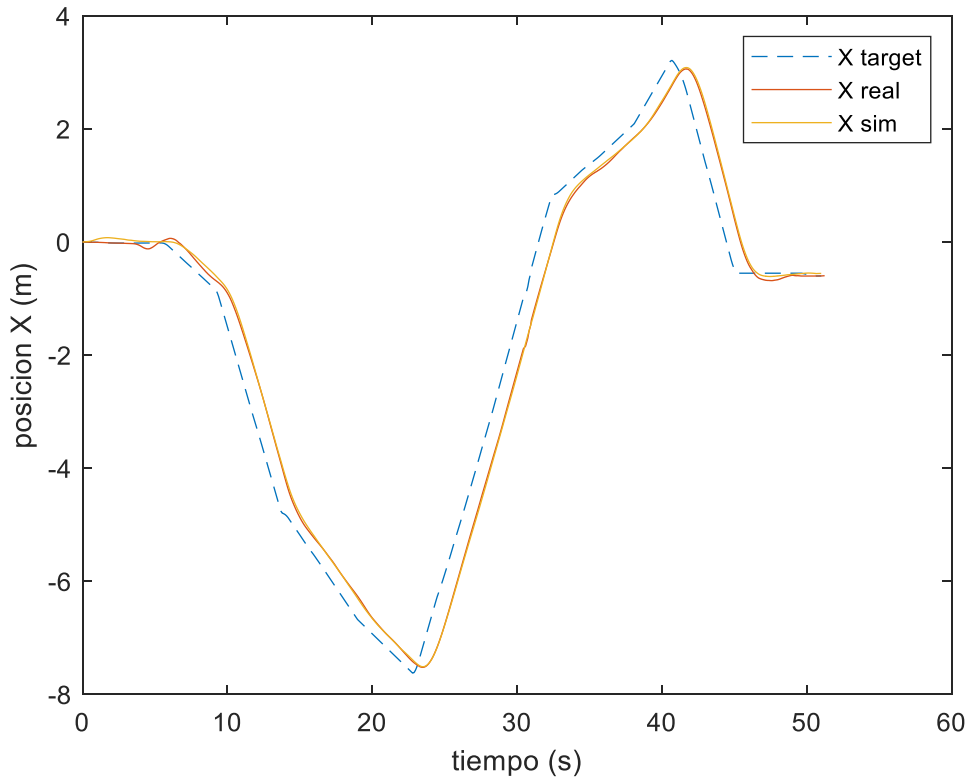


Ilustración 95-Coordenada X.

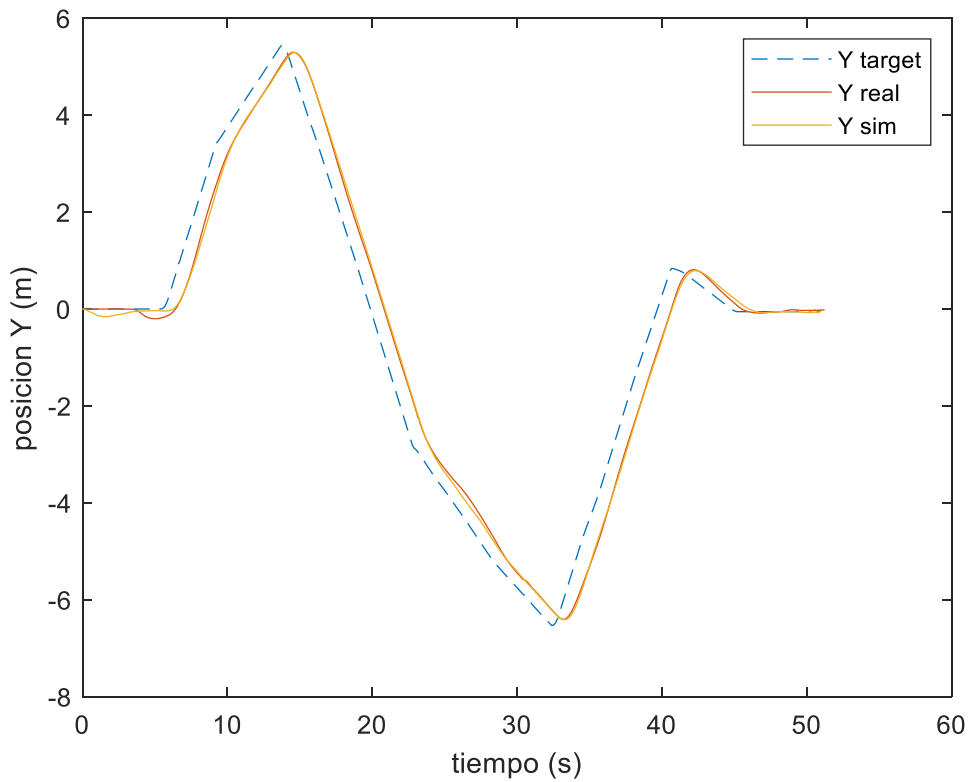


Ilustración 96-Coordenada Y.

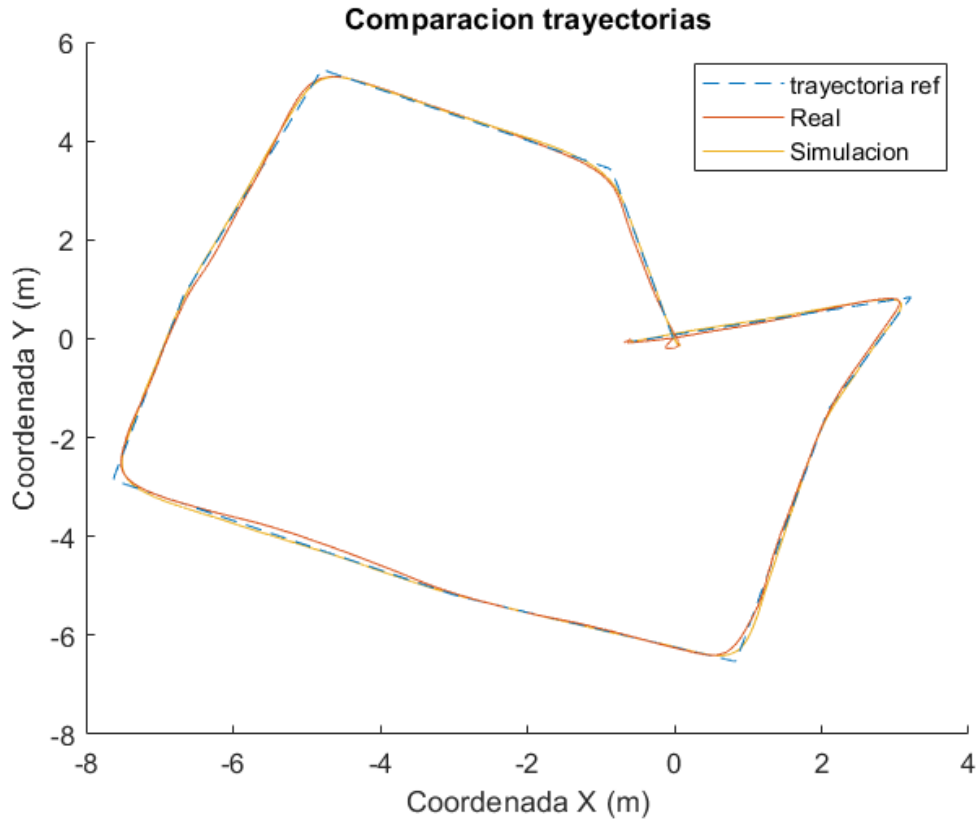


Ilustración 97-Trayectoria vista en XY.

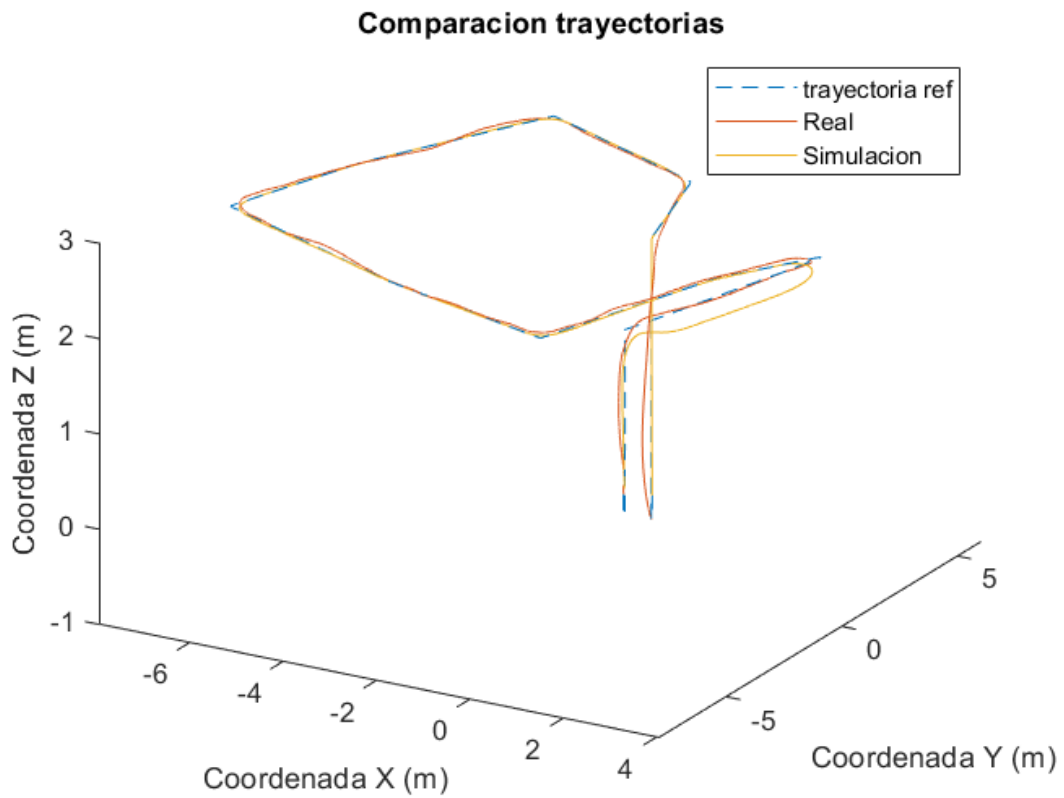


Ilustración 98-Trayectoria en XYZ.

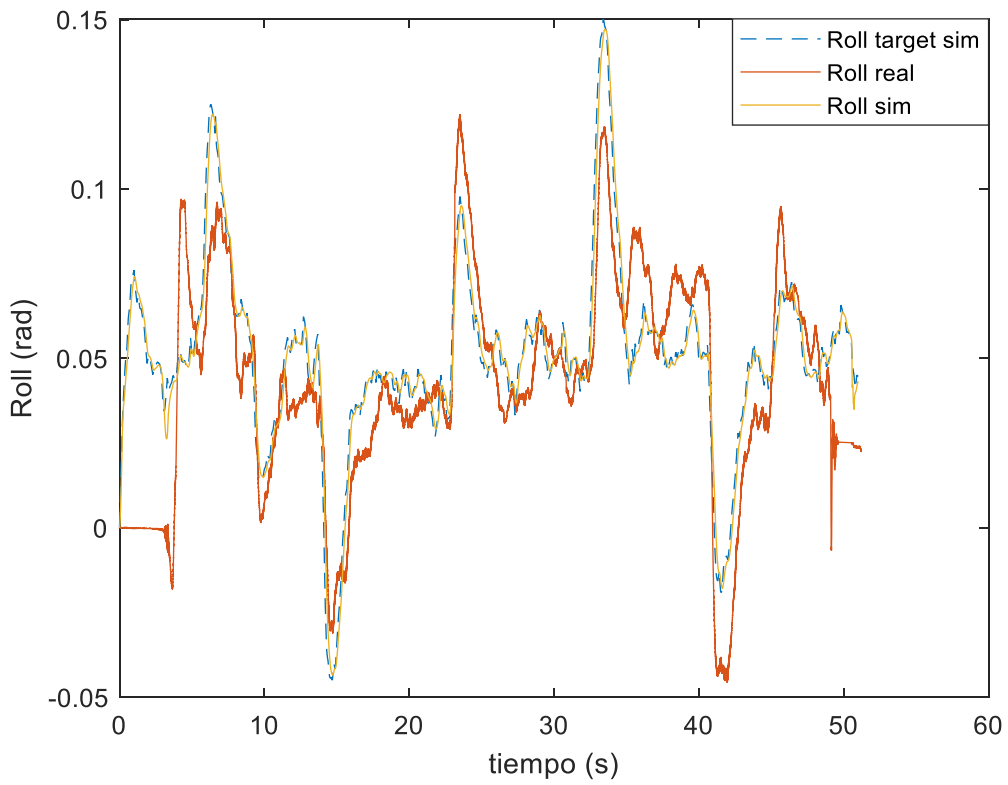


Ilustración 99- Roll validación.

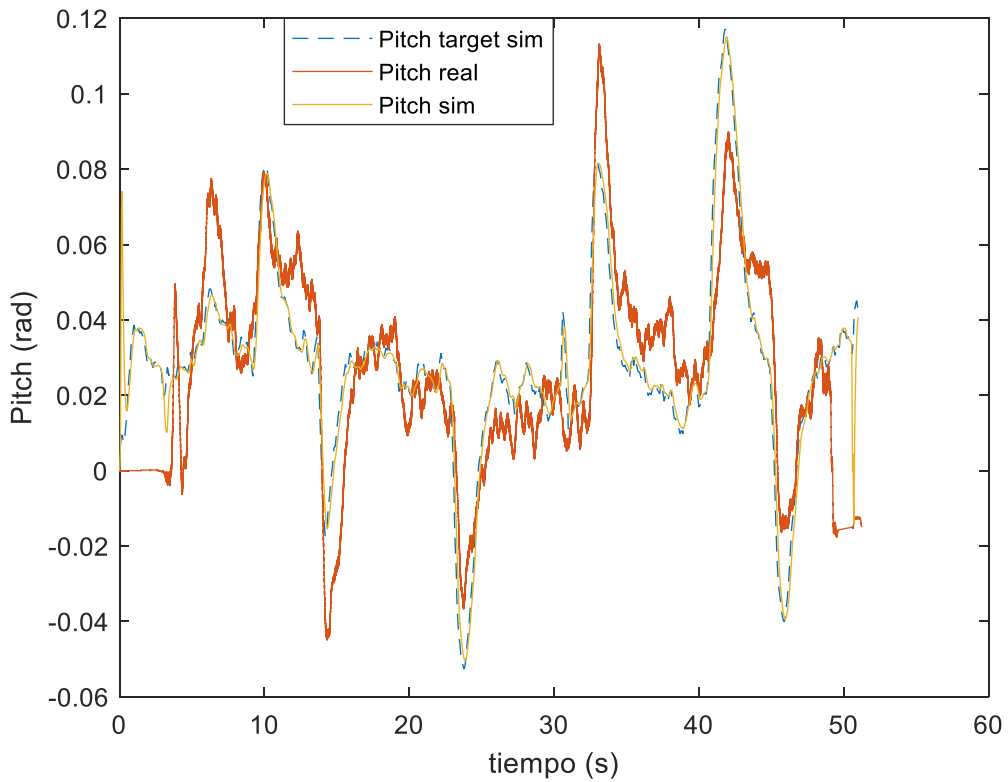


Ilustración 100-Pitch validación.

Si se observan los resultados de la simulación se ha conseguido un modelo que replica de forma bastante fiel la realidad. La precisión obtenida en la posición horizontal es llamativa ya que prácticamente ambas señales, la real y la simulada se calcan en todo momento. Se ha decidido expresar la precisión del modelo con un indicador conjunto para la coordenada X e Y debido a que los fabricantes de UAVs suelen expresarlo de esta forma.

$$error_{XY} = \pm 0.08 \text{ m}$$

La precisión del modelo para la altura es menor, aunque sigue siendo muy buena. El hecho de que la precisión en la coordenada Z sea menor se puede deber a varios factores, uno de ellos es que en este trabajo se ha recreado de forma simplificada el sistema de control de un piloto automático real, pero el sistema real cuenta con 'ayudas a la conducción' como puede ser que al despegar el sistema solo permite entregar el 80 % del empuje disponible por una cuestión de seguridad ya que no es bueno saturar los motores.

$$error_Z = \pm 0.11 \text{ m}$$

Además, hay que remarcar que los controladores de orientación reproducen de forma muy buena la realidad, lo que hace indicar que se ha modelado de forma correcta todo el sistema de control. Esto es así ya que estos controladores de pitch y roll se alimentan de la salida de los controladores de posición X e Y, no es que se hayan comandado los ángulos obtenidos del vuelo real.

10 CARACTERIZACIÓN UAV

Una vez que se ha comprobado que el modelo virtual se ajusta de forma correcta a la realidad se continua con un proceso de caracterización del sistema. Para ello se pretende llevar el UAV al 'límite', es decir, reproducir ciertas condiciones en las que el sistema se estrelle para así ver cuál es el rango de operación seguro del sistema. Esta caracterización es muy importante ya que nos dará una idea de qué tipo de trayectorias y en qué condiciones el sistema es capaz de realizar un vuelo controlado y seguro.

Se quieren definir los siguientes parámetros:

- Ángulos de inclinación máximos.
- Velocidad horizontal máxima.
- Velocidad de ascenso máxima.
- Velocidad de descenso máxima.
- Aceleración horizontal máxima.
- Aceleración de ascenso máxima.
- Aceleración de descenso máxima.
- Viento máximo con el cual es seguro operar.
- Carga de pago máxima.

10.1 ANGULOS MÁXIMOS

Para la obtención de los ángulos máximos de inclinación, es decir, de Pitch y Roll, se someterá el sistema a una perturbación con forma de ráfaga de viento. Se realiza así ya que este tipo de perturbación puede provocar que el sistema se aleje del punto de operación para el cual se ha diseñado el controlador y puede provocar que el sistema se vuelva inestable o que los motores saturen, ambos casos se traducirían en que el UAV se estrellara.

Se presentan los resultados de dos casos. El caso 1, será el último caso para el cual el UAV ha conseguido corregir su posición y estabilizarse. El caso 2, será el caso en el cual el UAV no logra estabilizarse y por lo tanto cae al suelo. Si comprobamos los ángulos de inclinación del caso 1 se pueden obtener los ángulos máximos de inclinación.

10.1.1 CASO 1

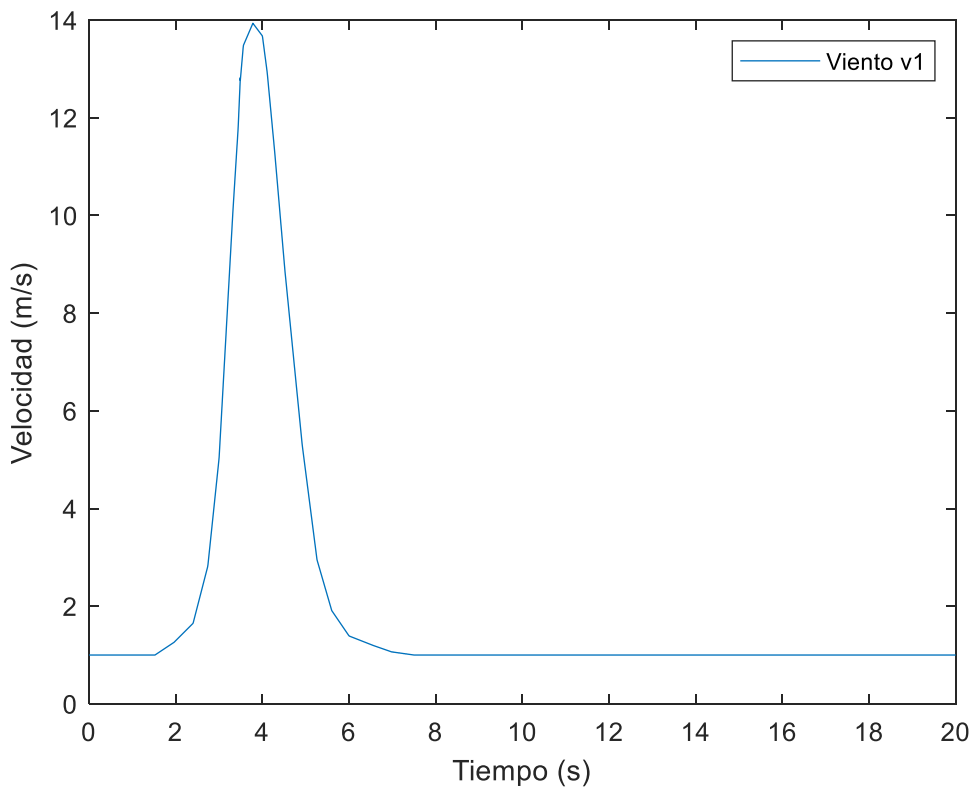


Ilustración 101- Perturbación de ráfaga de viento.

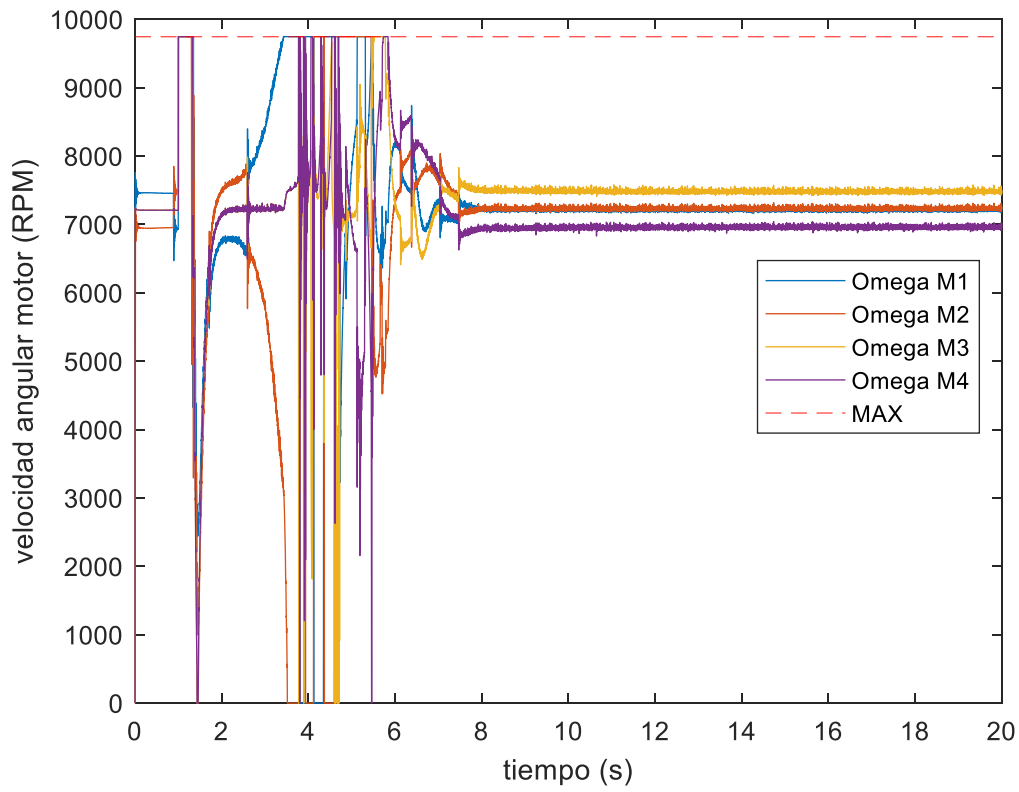


Ilustración 102-RPM motores.

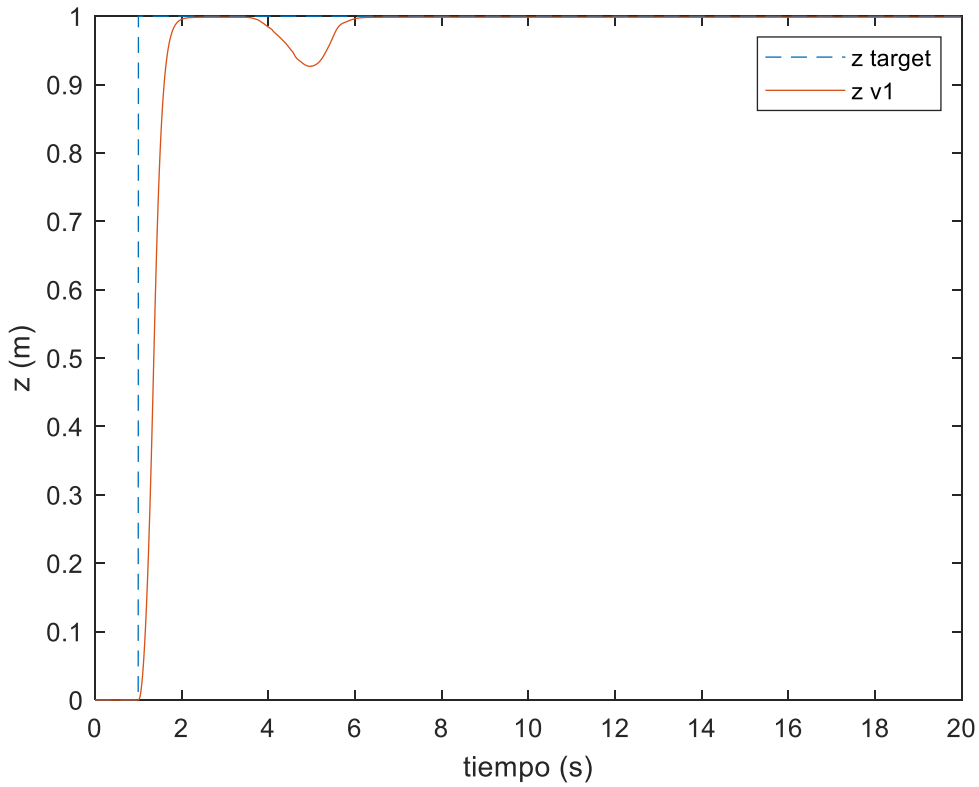


Ilustración 103-Coordenada Z.

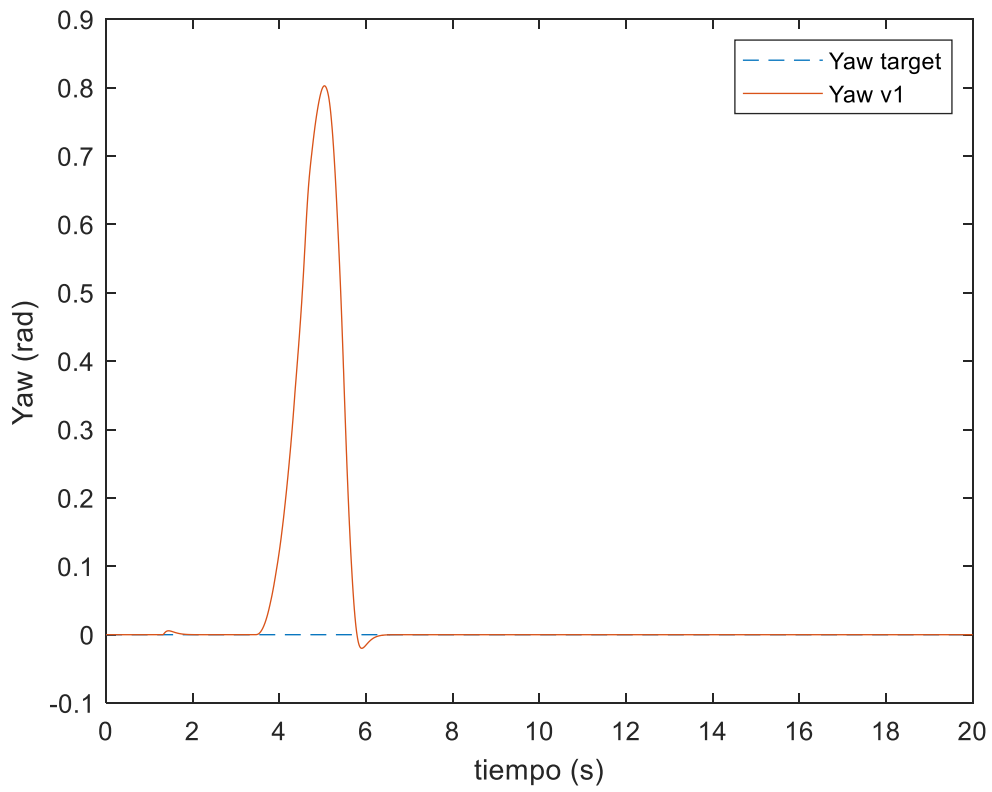


Ilustración 104- Angulo de Yaw.

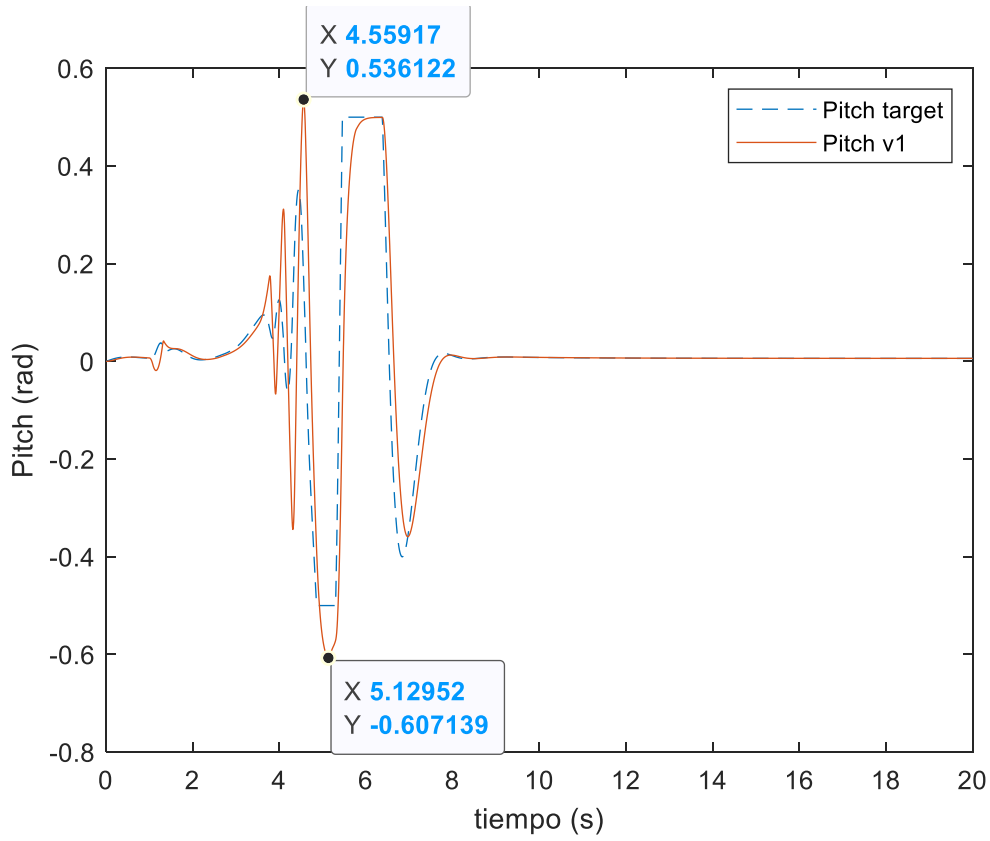


Ilustración 105-Angulo de Pitch.

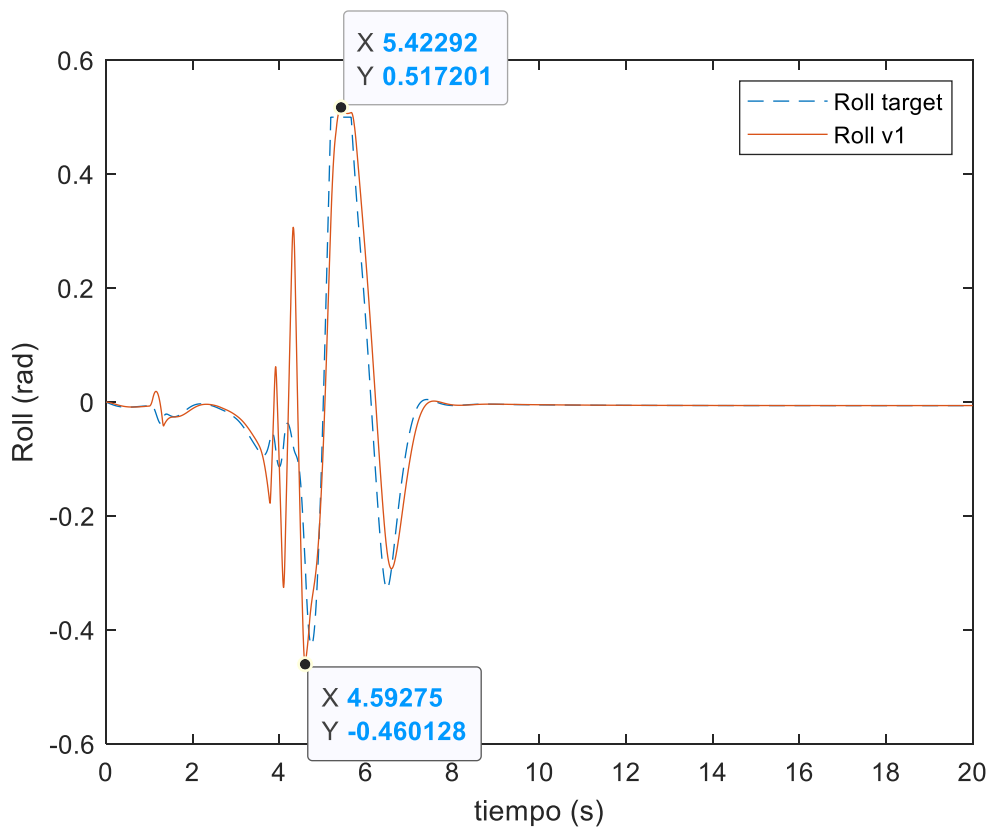


Ilustración 106-Angulo de Roll.

10.1.2 CASO 2

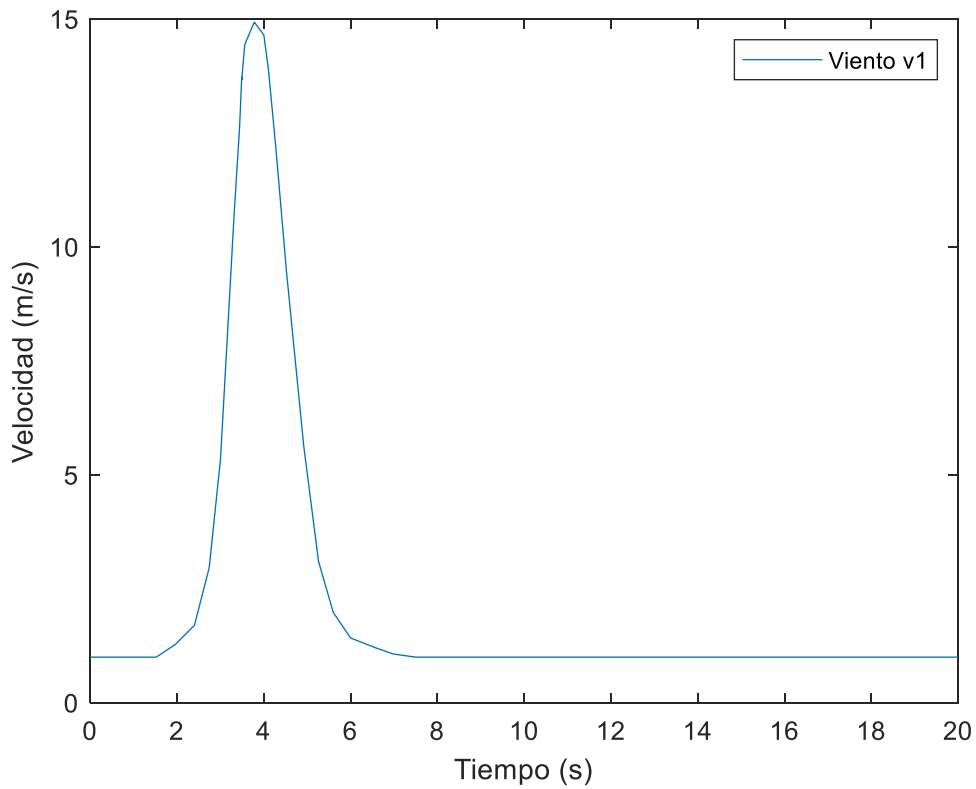


Ilustración 107-Perturbación de ráfaga de viento.

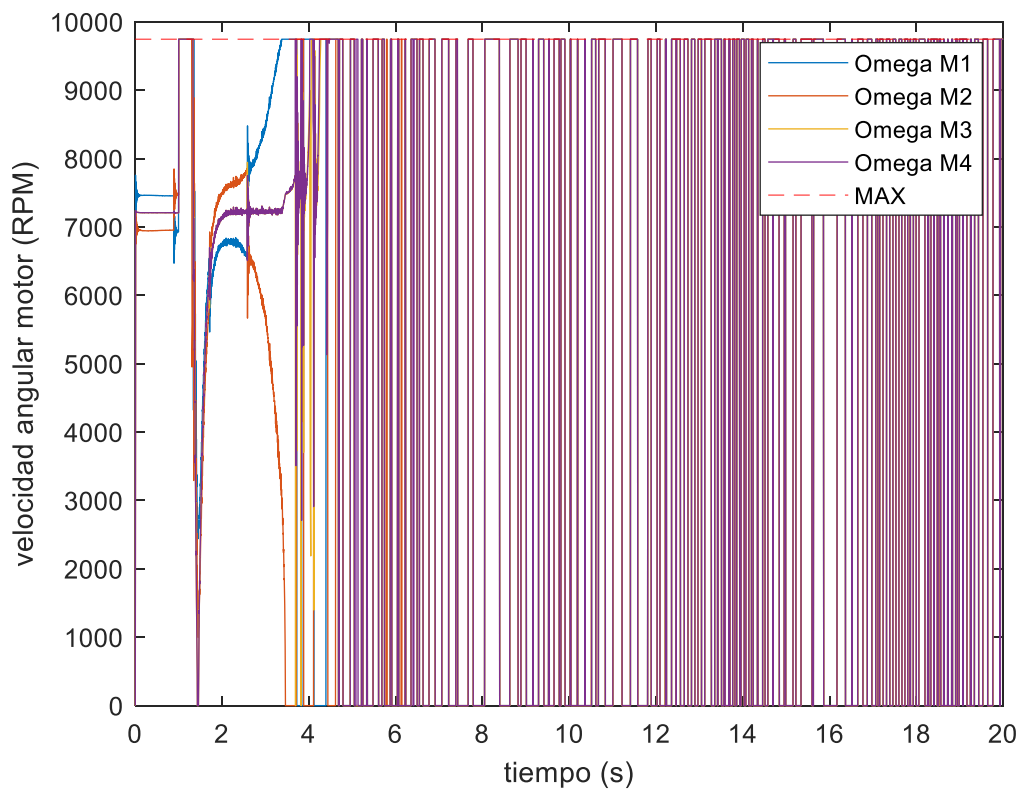


Ilustración 108-RPM motores.

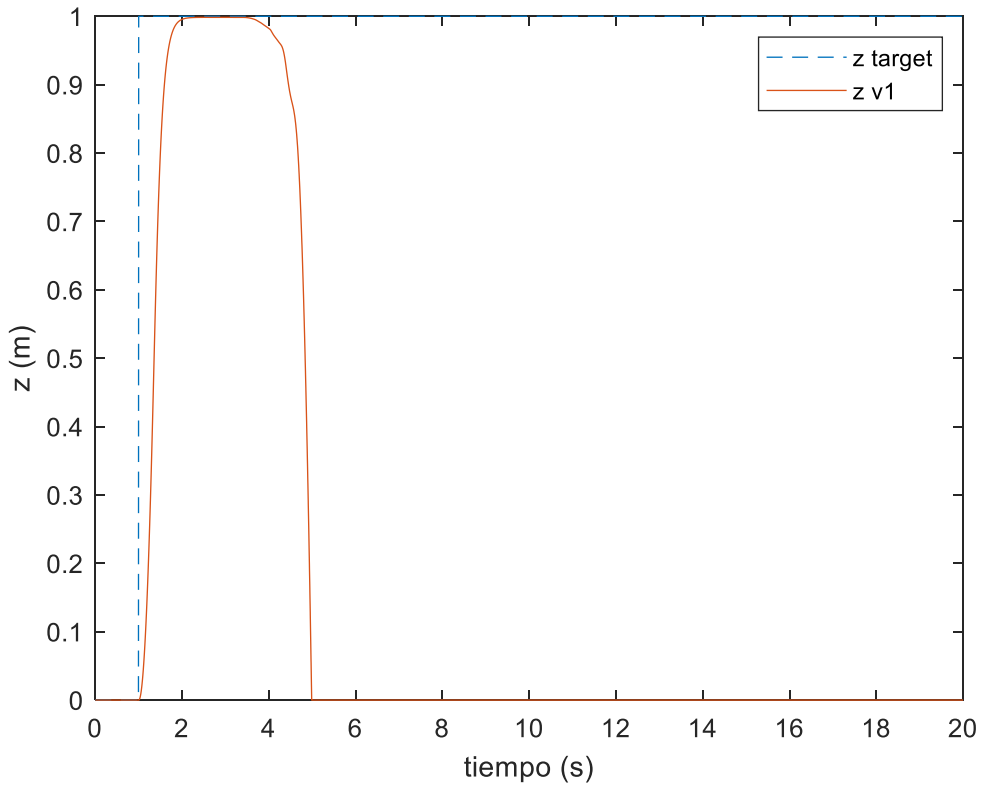


Ilustración 109-Coordenada Z.

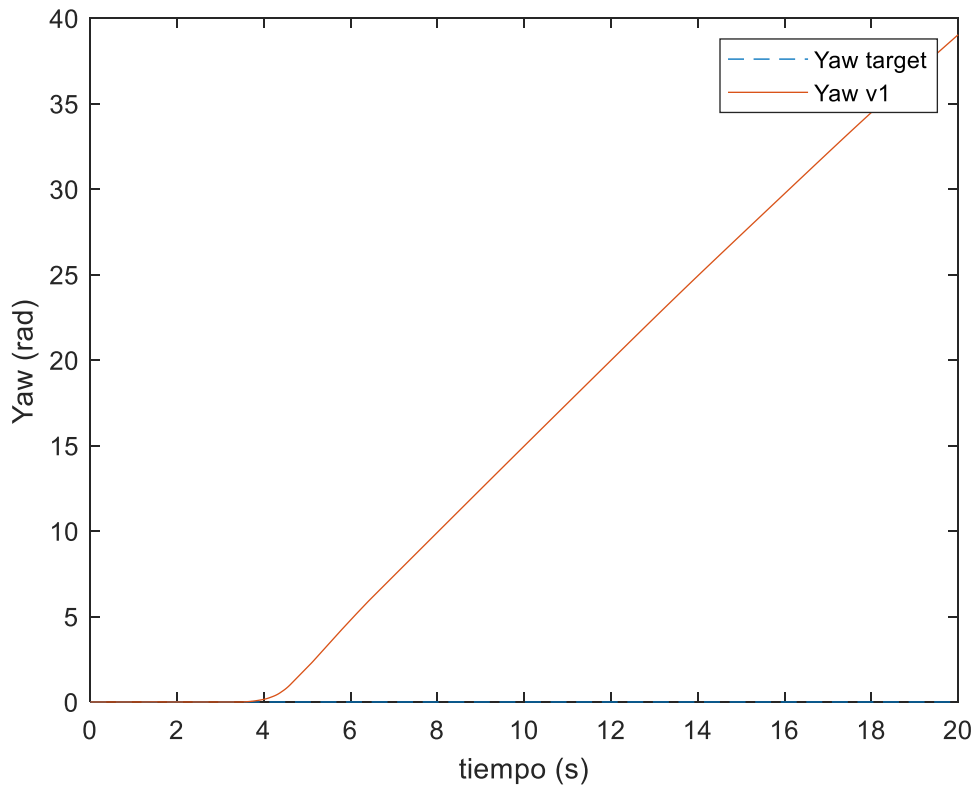


Ilustración 110- Angulo de Yaw.

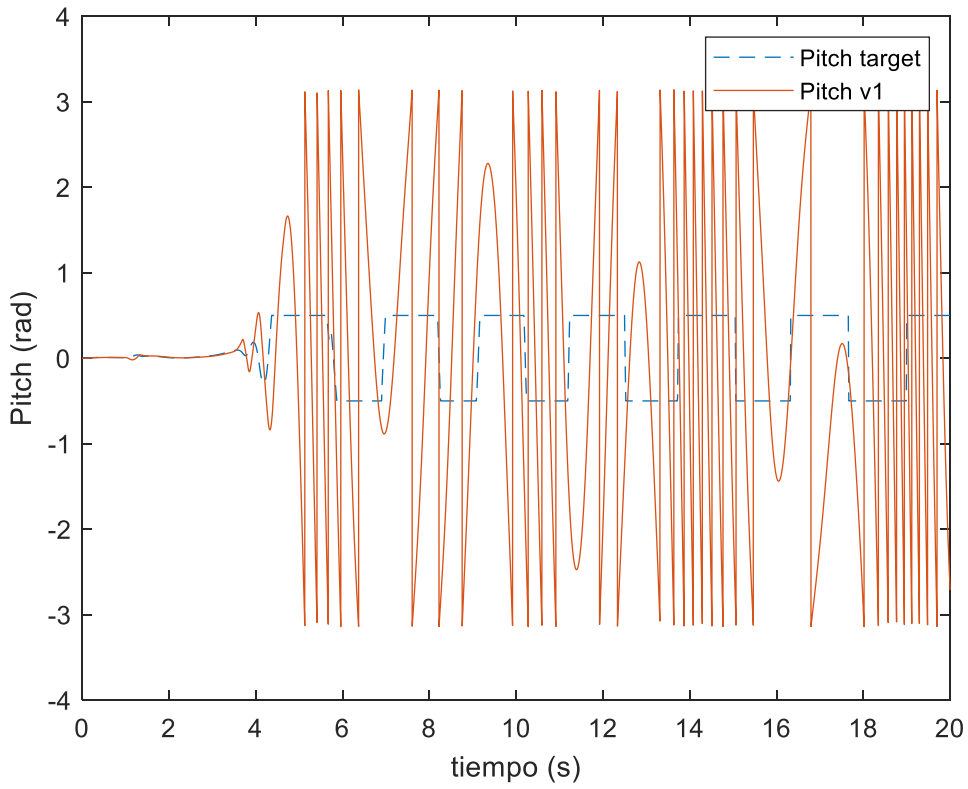


Ilustración 111-Angulo de Pitch.

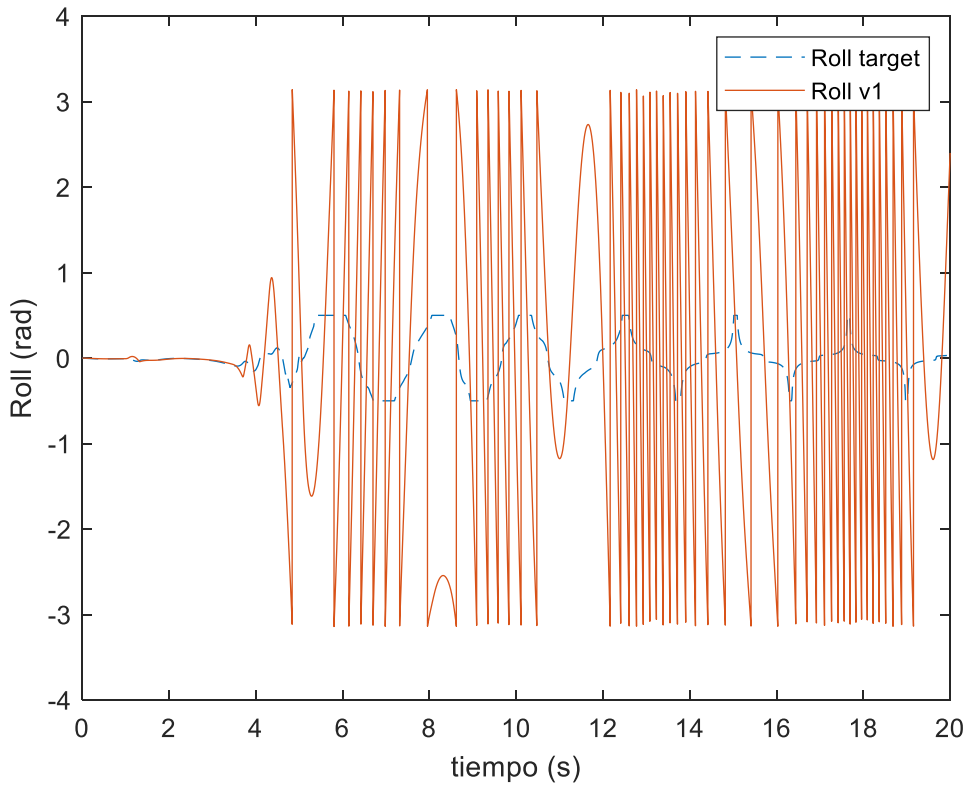


Ilustración 112-Angulo de Roll.

Los ángulos máximos obtenidos de la Ilustración 106 son los siguientes.

$$\theta_{MAX} = 0,6 \text{ rad} = 34^\circ$$

$$\phi_{MAX} = 0,6 \text{ rad} = 34^\circ$$

10.2 ACELERACIONES HORIZONTALES MÁXIMAS

Para la obtención de las aceleraciones horizontales máximas basta con aplicar las ecuaciones que se utilizan en el controlador para pasar del control de posición al de orientación. Estas ecuaciones lo que hacen es igualar el vector de aceleraciones deseado al vector del empuje del UAV al cual se le ha multiplicado la matriz de transformación de Euler para pasar de coordenadas en el cuerpo a coordenadas absolutas.

$$\theta_{MAX} = \tan^{-1}\left(\frac{\ddot{X}_{MAX}}{g}\right)$$

$$\phi_{MAX} = \tan^{-1}\left(\frac{\ddot{Y}_{MAX} \cos \theta}{g}\right)$$

$$\ddot{X}_{MAX} = 6,8 \text{ m/s}^2$$

$$\ddot{Y}_{MAX} = 6,8 \text{ m/s}^2$$

10.3 VELOCIDADES LINEALES MÁXIMAS

10.3.1 VELOCIDAD DE AVANCE

Para obtener la velocidad máxima de avance del sistema se comanda en posición una trayectoria de tipo rampa y se va modificando la pendiente de esta hasta alcanzar el límite del sistema. Como en los apartados anteriores se presentarán dos casos. El primero será el último caso en el cual el UAV consigue seguir la trayectoria comandada y el segundo caso será en el cual se produzca la caída del sistema.

10.3.1.1 CASO 1

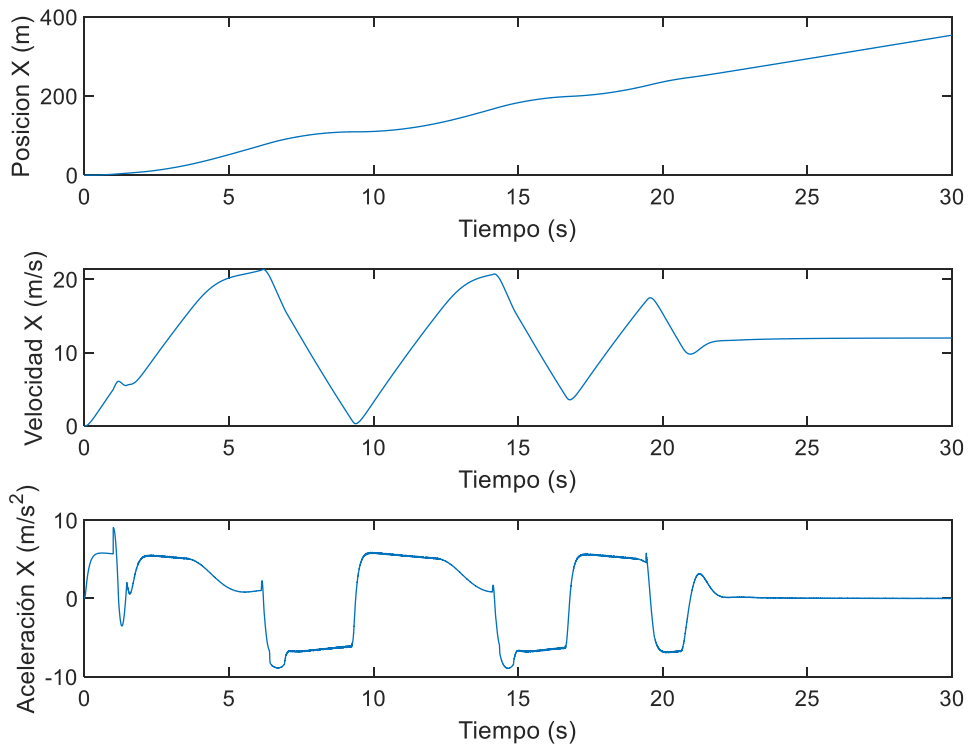


Ilustración 113- Posición, velocidad y aceleración en X.

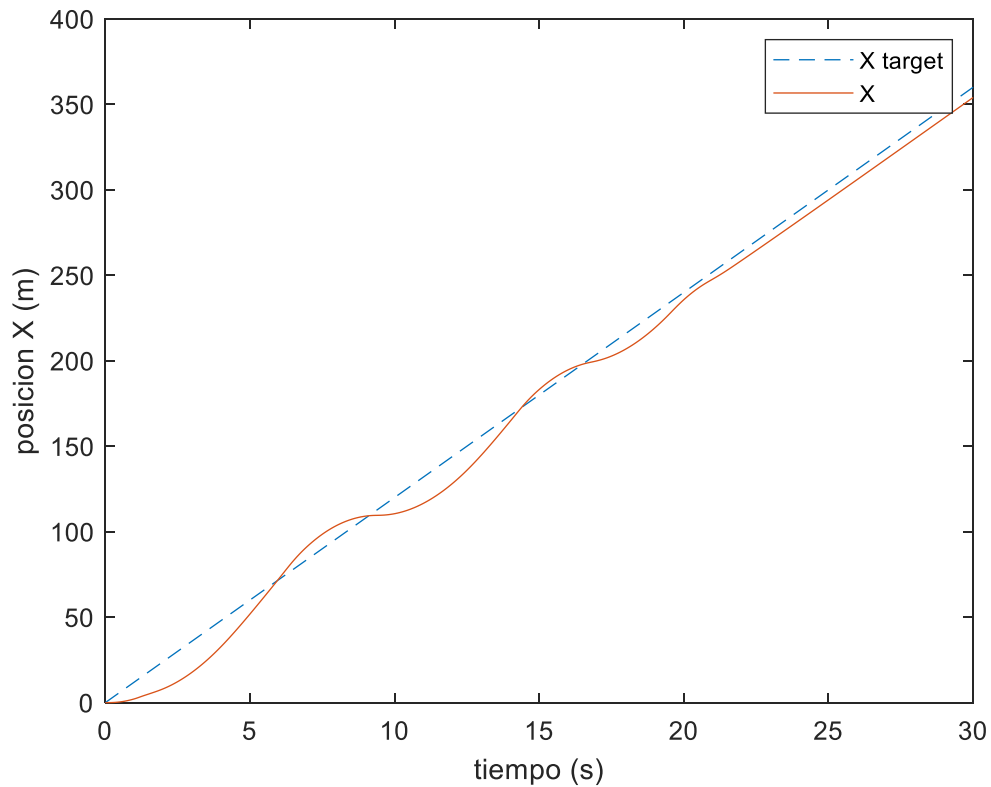


Ilustración 114-Respuesta de coordenada X.

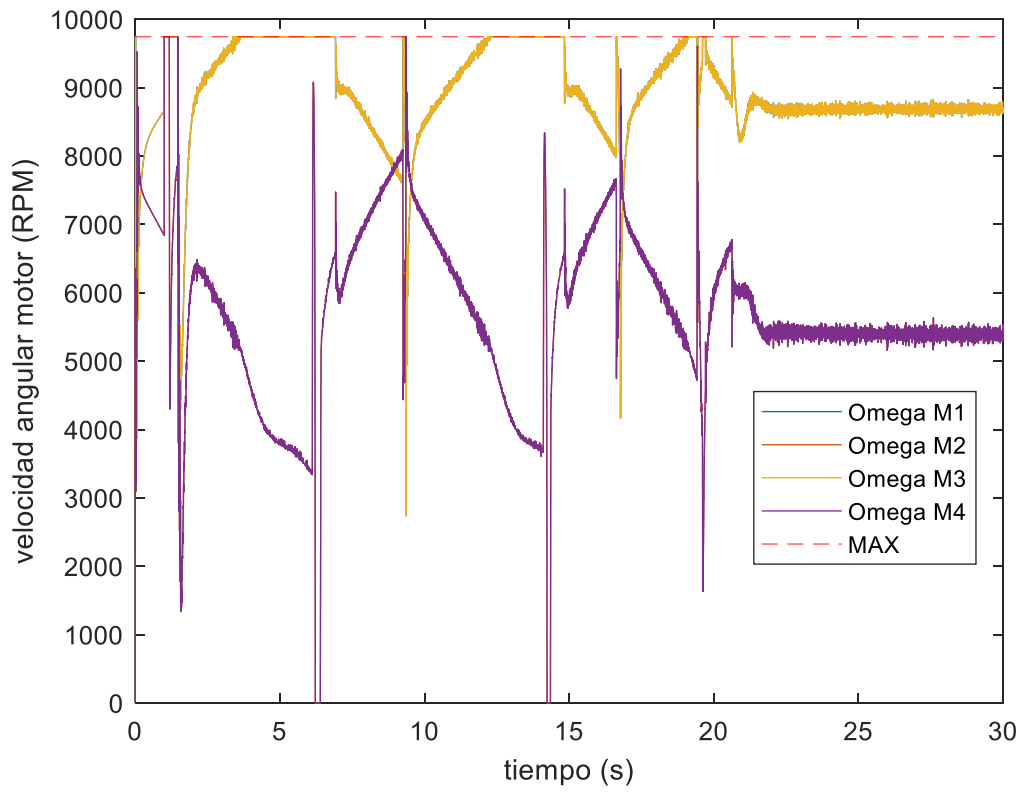


Ilustración 115-RPM motores.

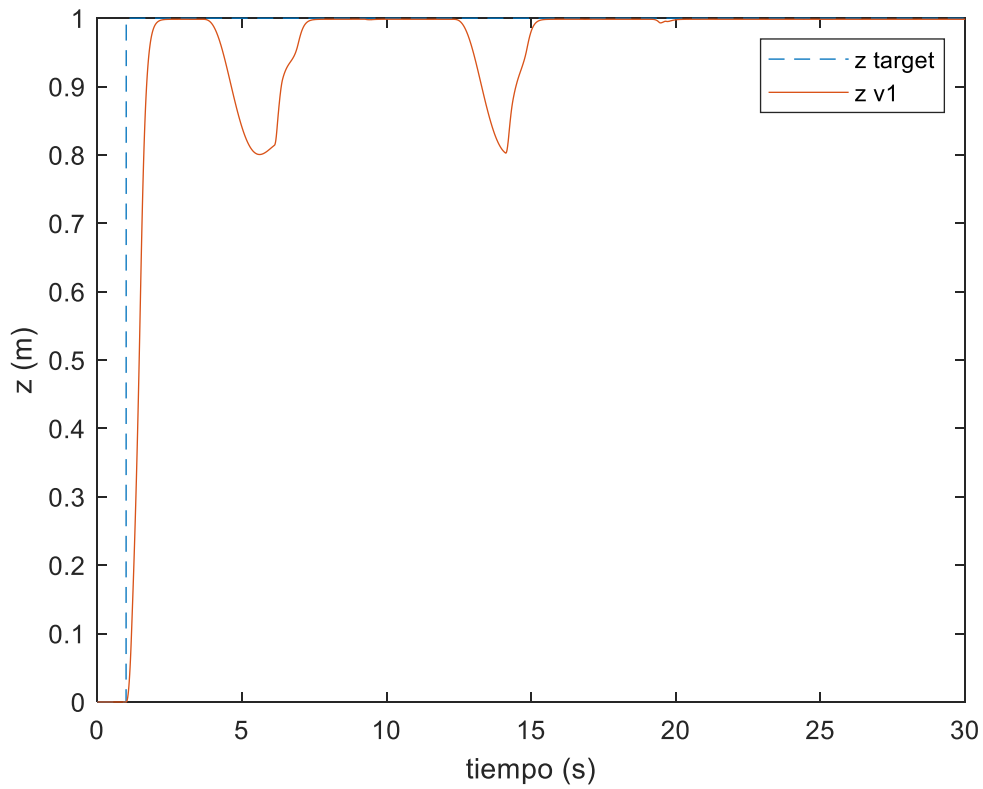


Ilustración 116- Coordenada Z.

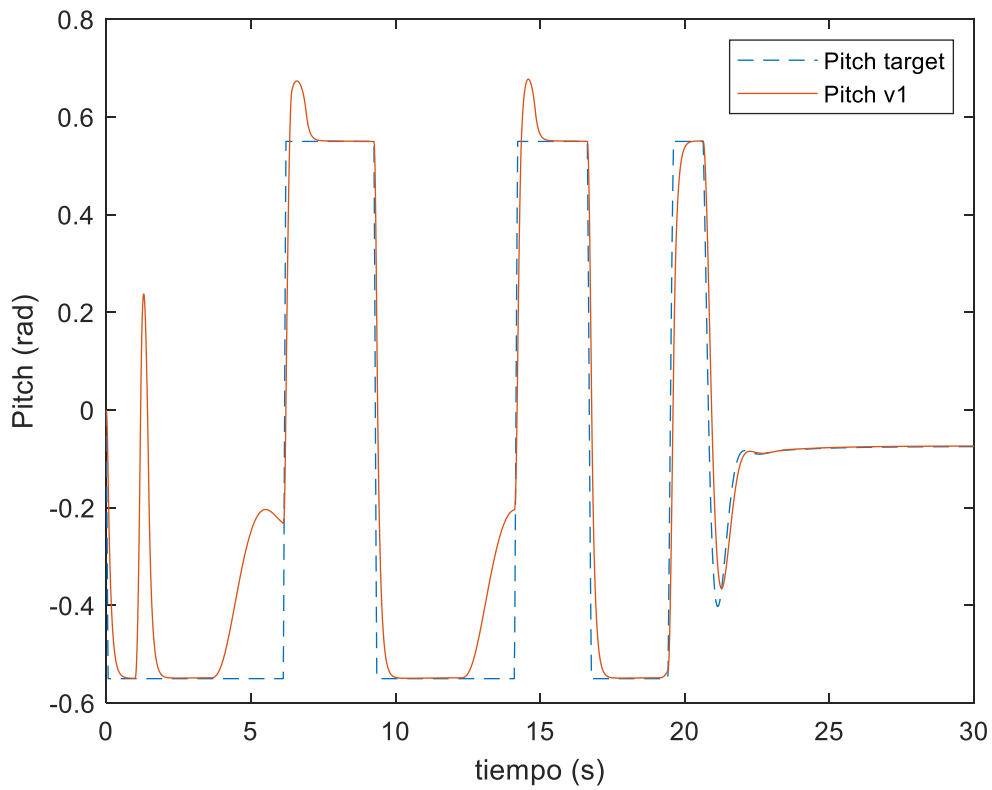


Ilustración 117- Angulo de Pitch.

10.3.1.2 CASO 2

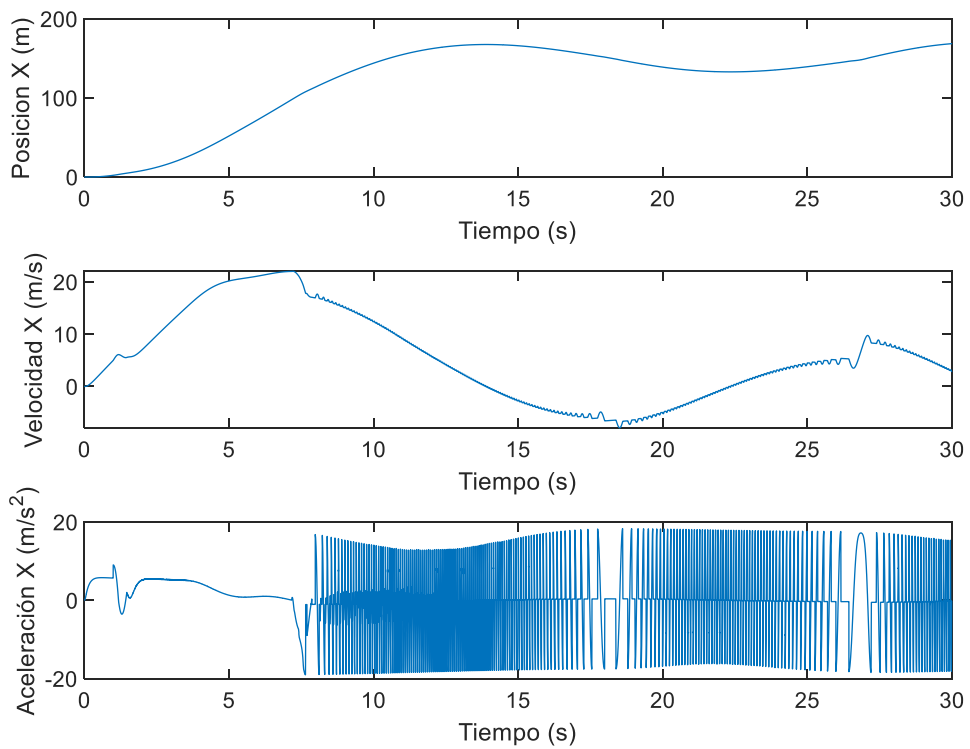


Ilustración 118-Posición, velocidad y aceleración de X.

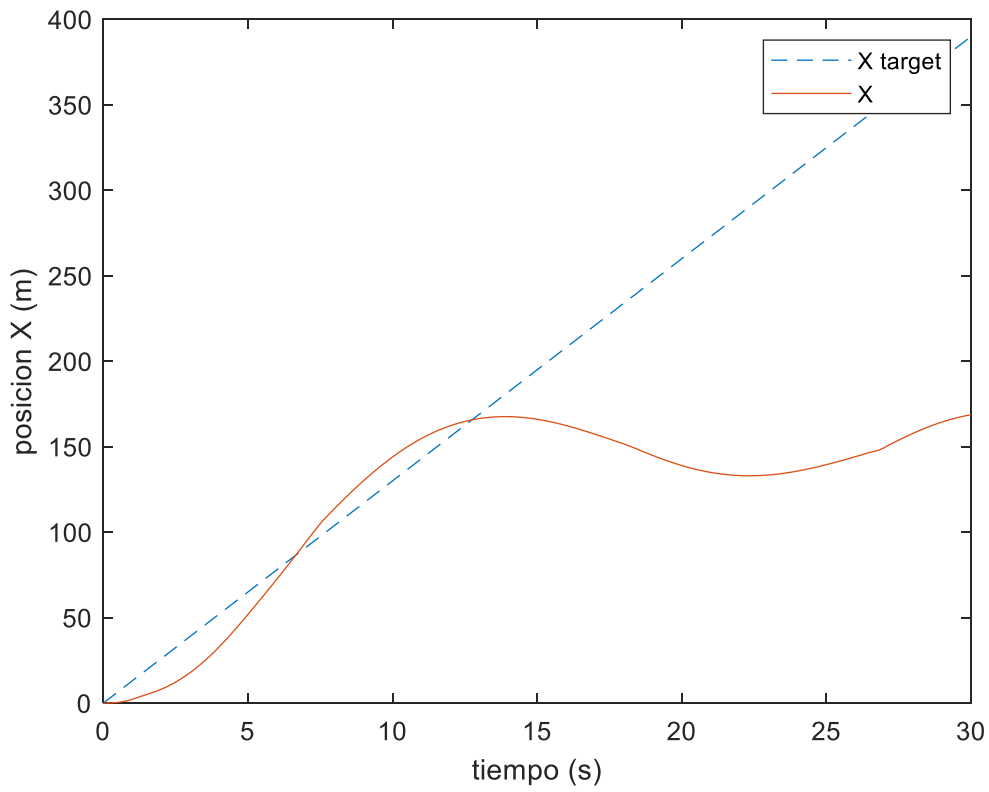


Ilustración 119-respuesta de coordenada X.

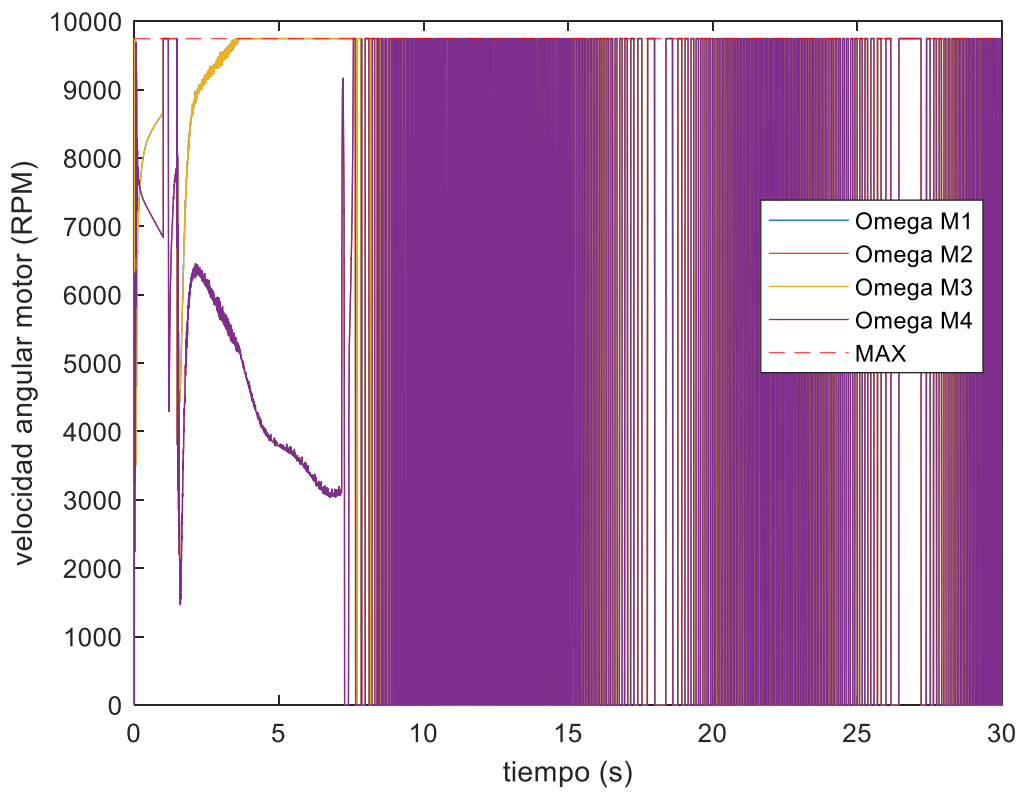


Ilustración 120-RPM de los motores.

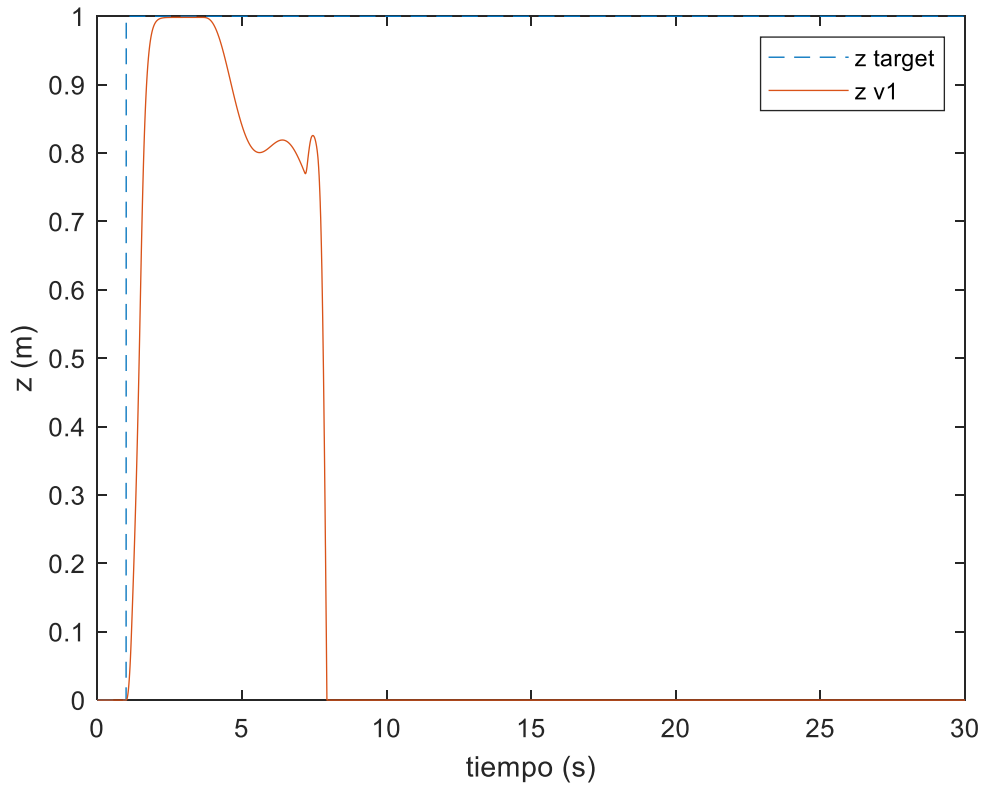


Ilustración 121-Coordenada Z.

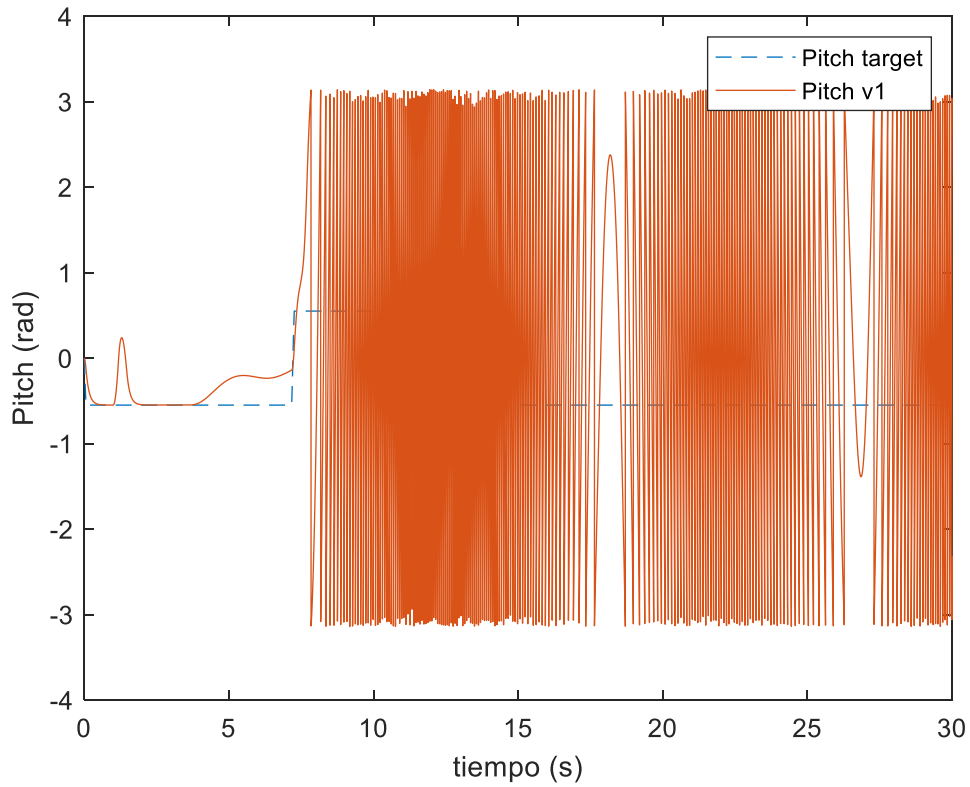


Ilustración 122-Angulo de Pitch.

La velocidad máxima de avance se obtiene de la Ilustración 114 y se tomara como velocidad máxima la velocidad estacionaria del caso 1.

$$V_{MAX}^{XY} = 12 \text{ m/s}$$

10.3.2 VELOCIDAD DE ASCENSO

Se sigue la misma metodología que para el cálculo de la velocidad de avance. Se comandan trayectorias verticales con una rampa de distintas pendientes hasta que se alcanza el límite del sistema.

10.3.2.1 CASO 1

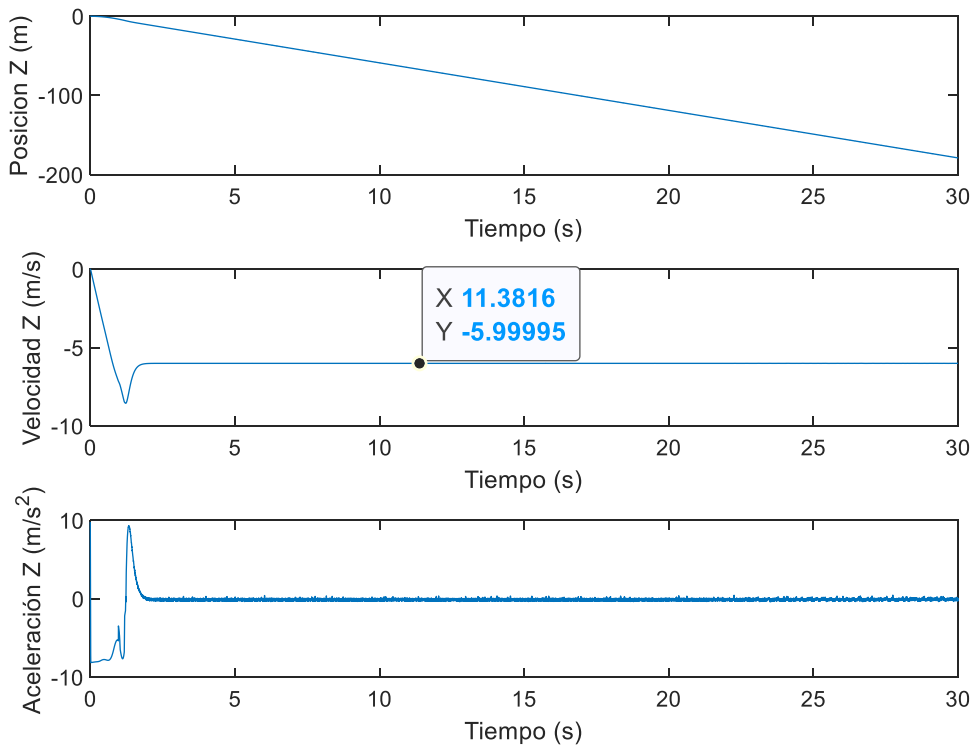


Ilustración 123-Posición, velocidad y aceleración en X.

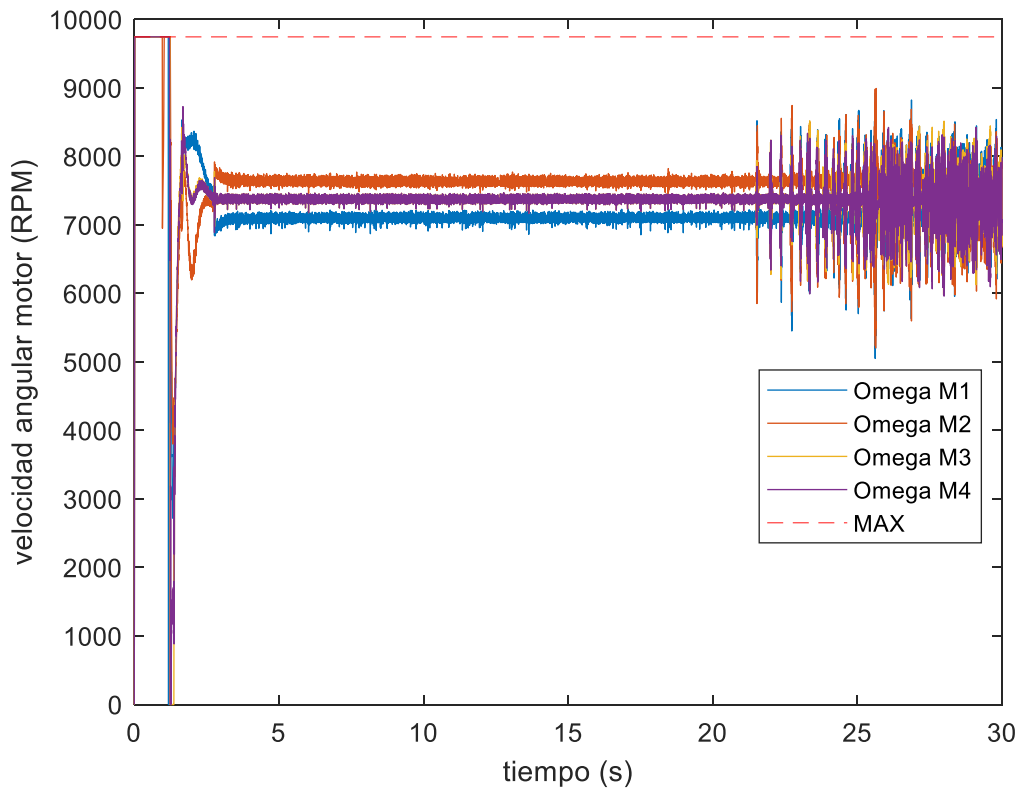


Ilustración 124-RPM de los motores.

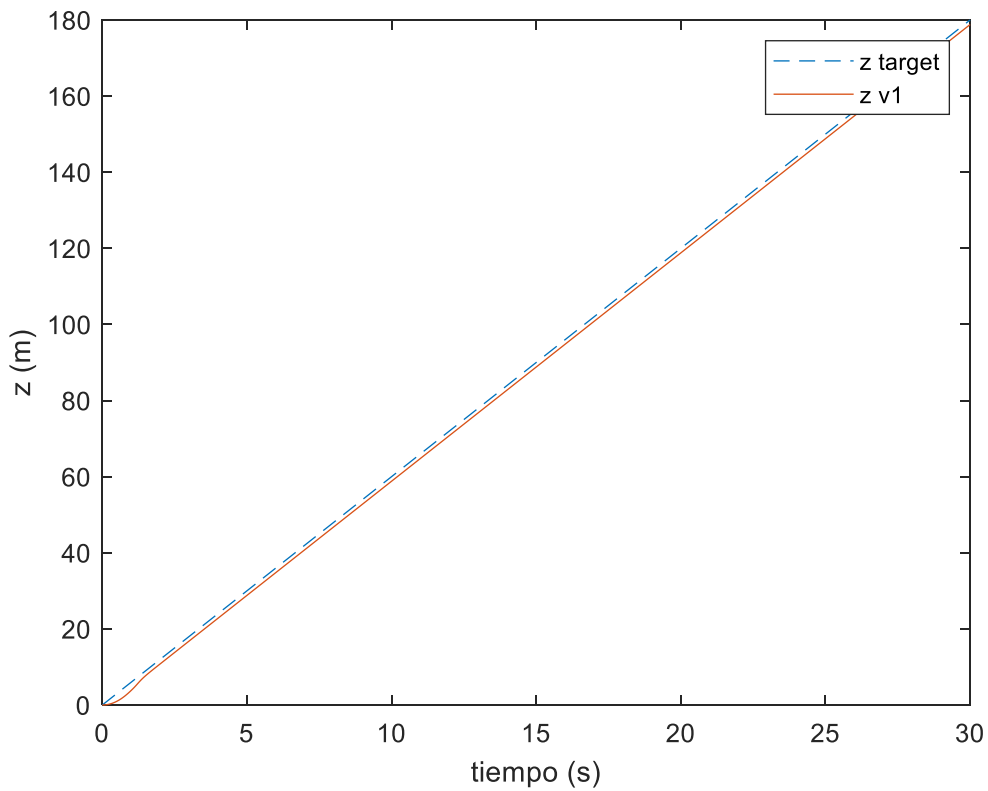


Ilustración 125- Respuesta de Z.

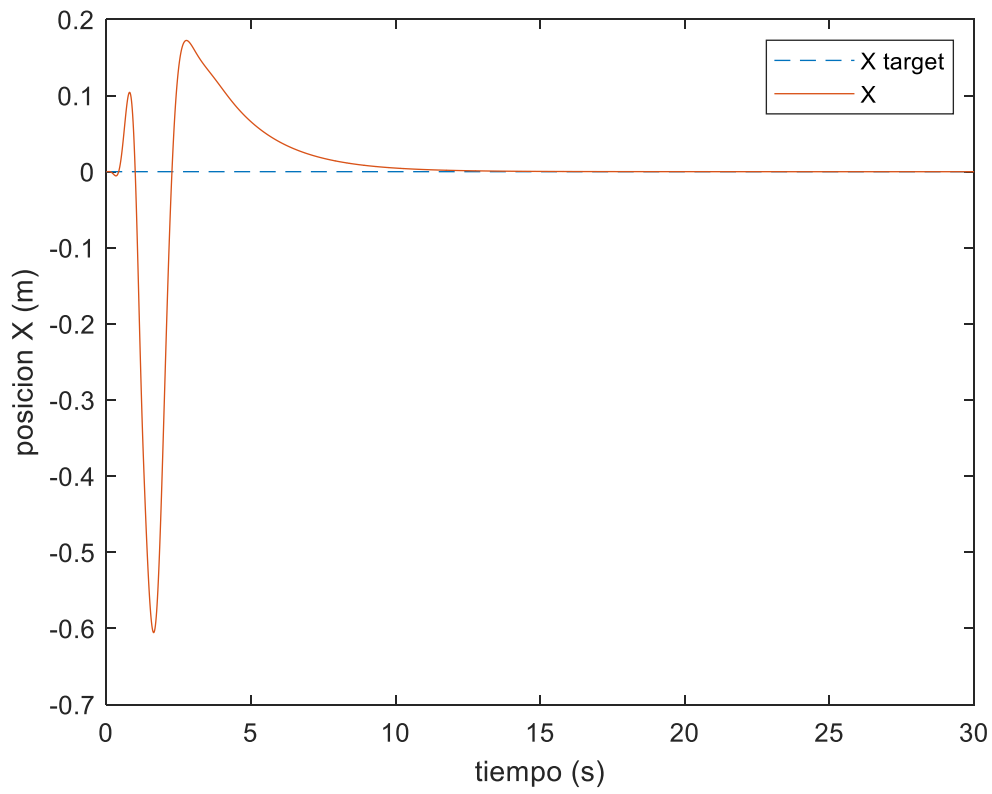


Ilustración 126-Coordenada X.

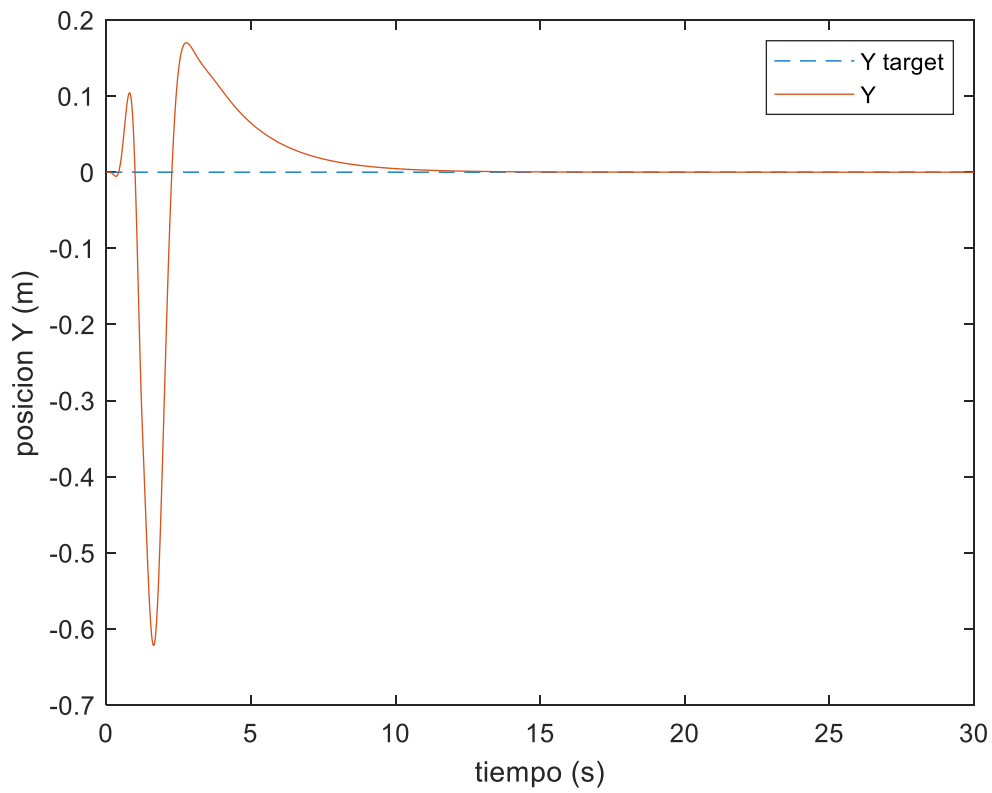


Ilustración 127-Coordenada Y.

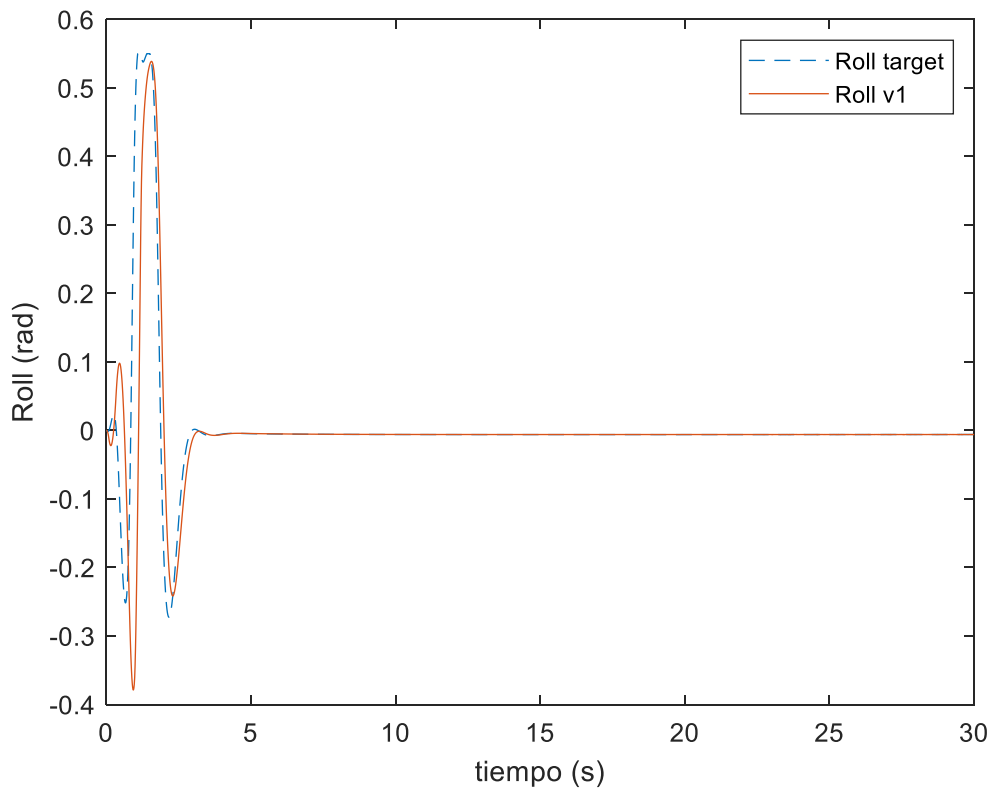


Ilustración 128-Angulo de Roll.

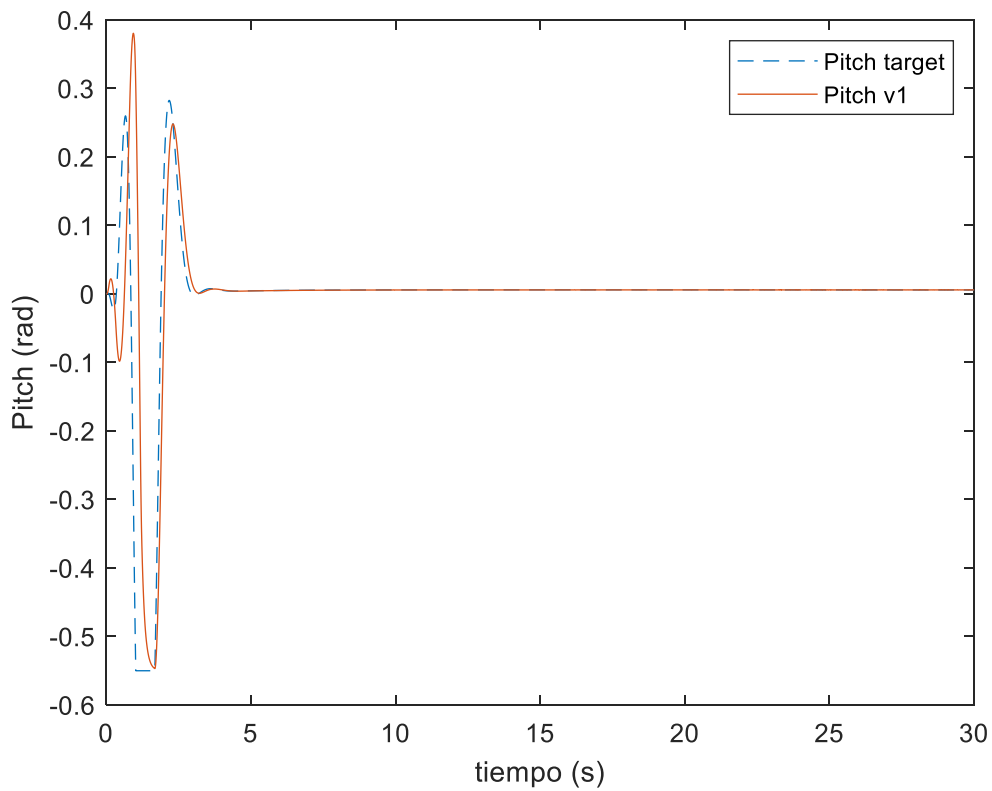


Ilustración 129-Angulo de Pitch.

10.3.2.2 CASO 2

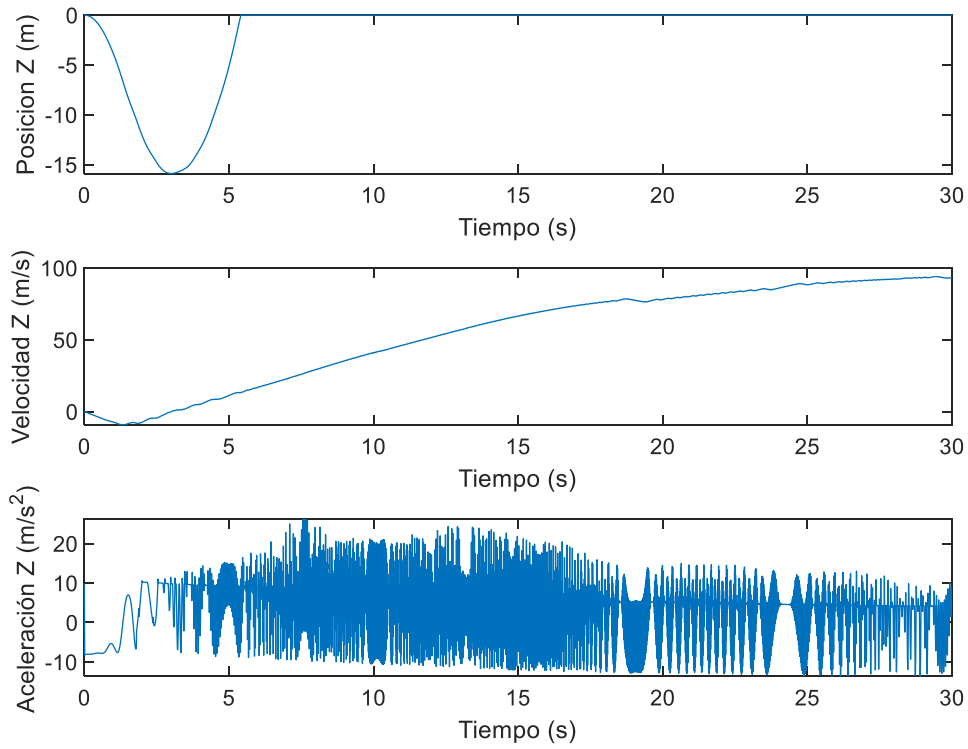


Ilustración 130-Posición, velocidad y aceleración en Z.

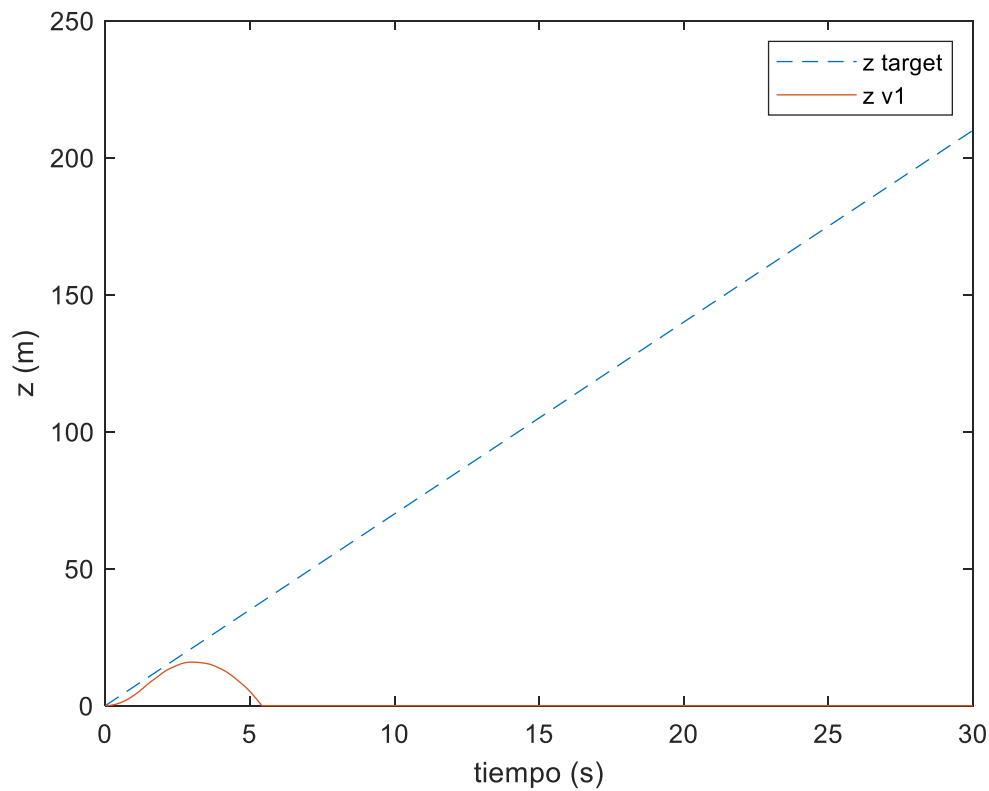


Ilustración 131-Respuesta en Z.

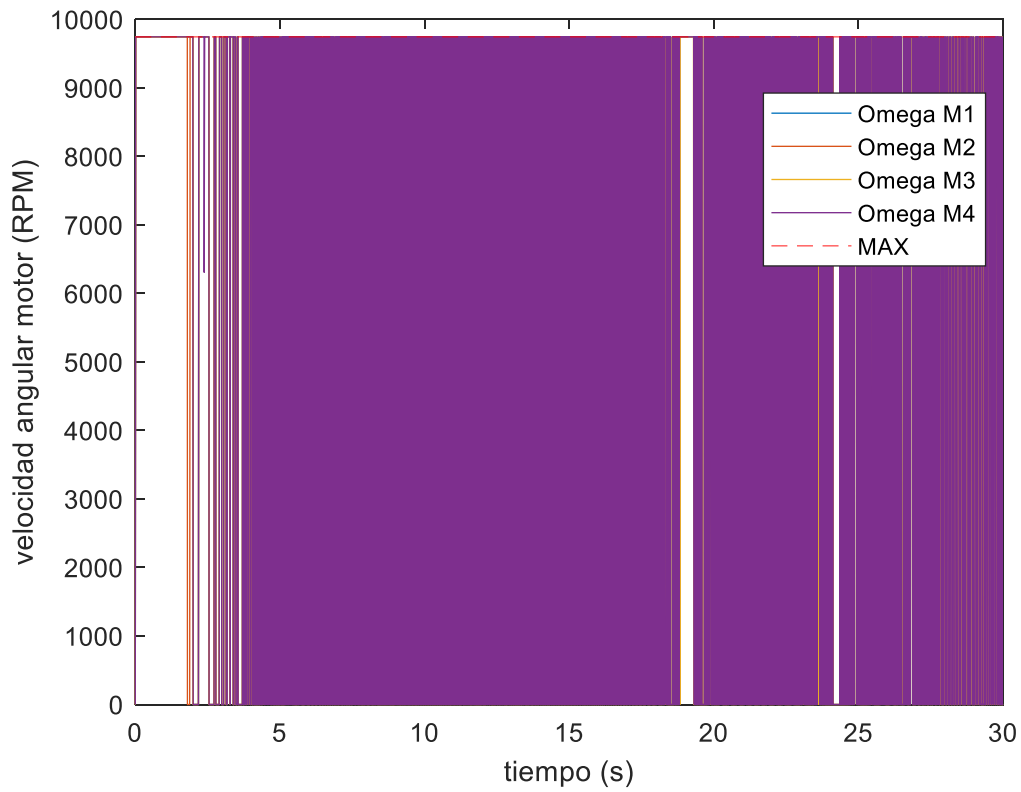


Ilustración 132-RPM de los motores.

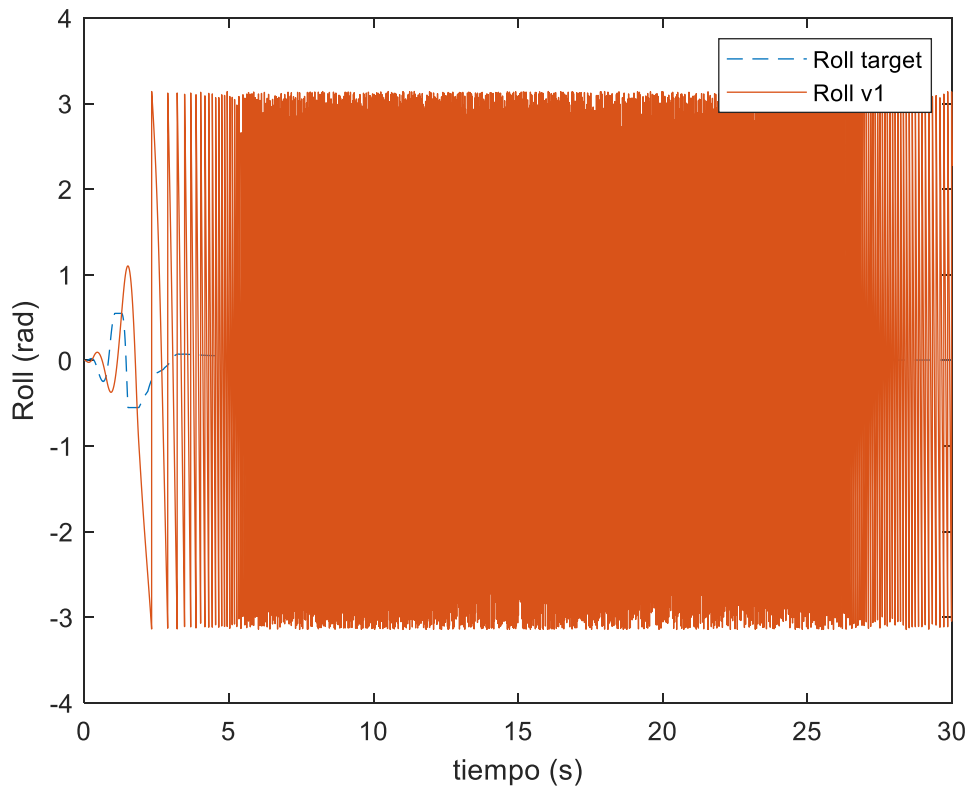


Ilustración 133- Angulo de Roll.

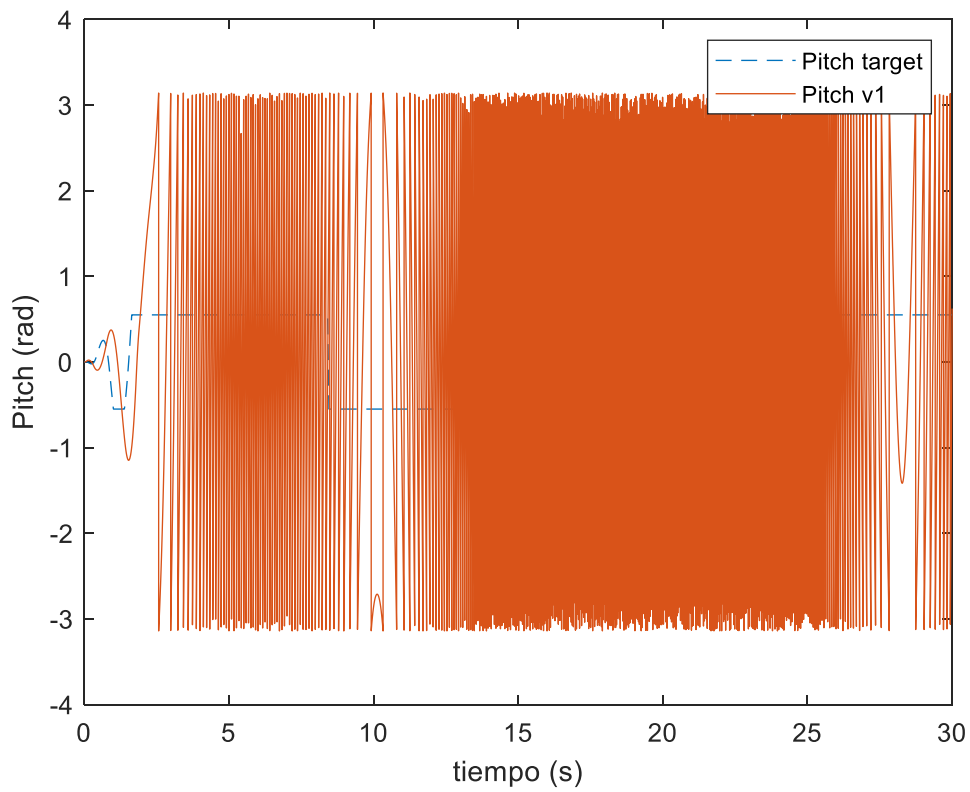


Ilustración 134- Angulo de Pitch.

Se obtiene la velocidad máxima de ascenso de la Ilustración 124.

$$V_{MAX}^{UP} = 6 \text{ m/s}$$

10.3.3 VELOCIDAD DE DESCENSO

10.3.3.1 CASO 1

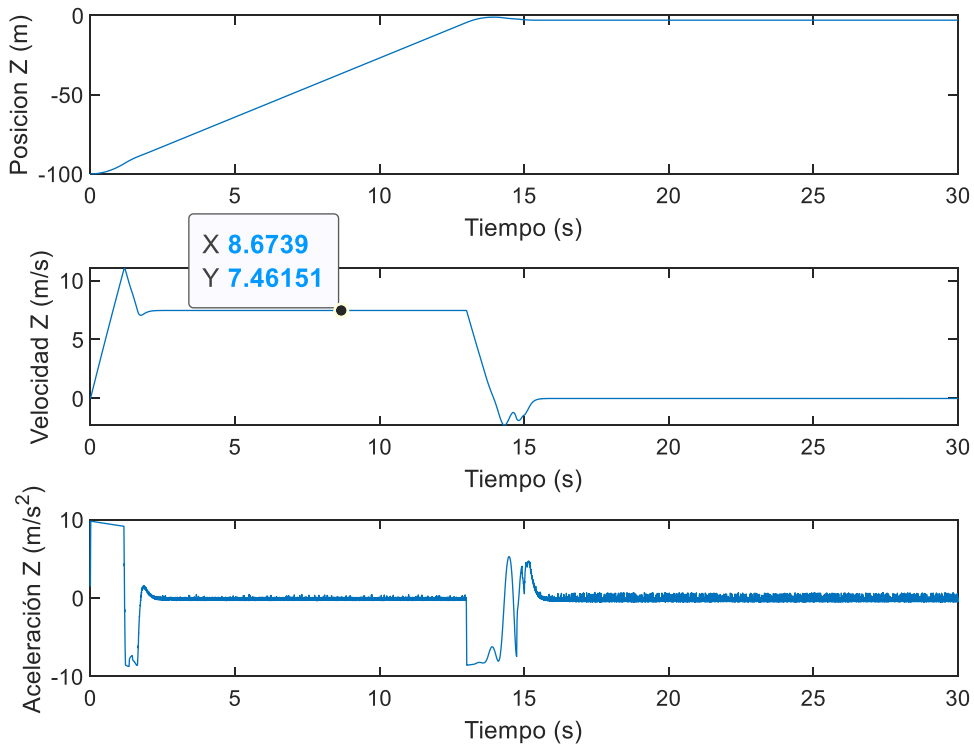


Ilustración 135-Posición, velocidad y aceleración en Z.

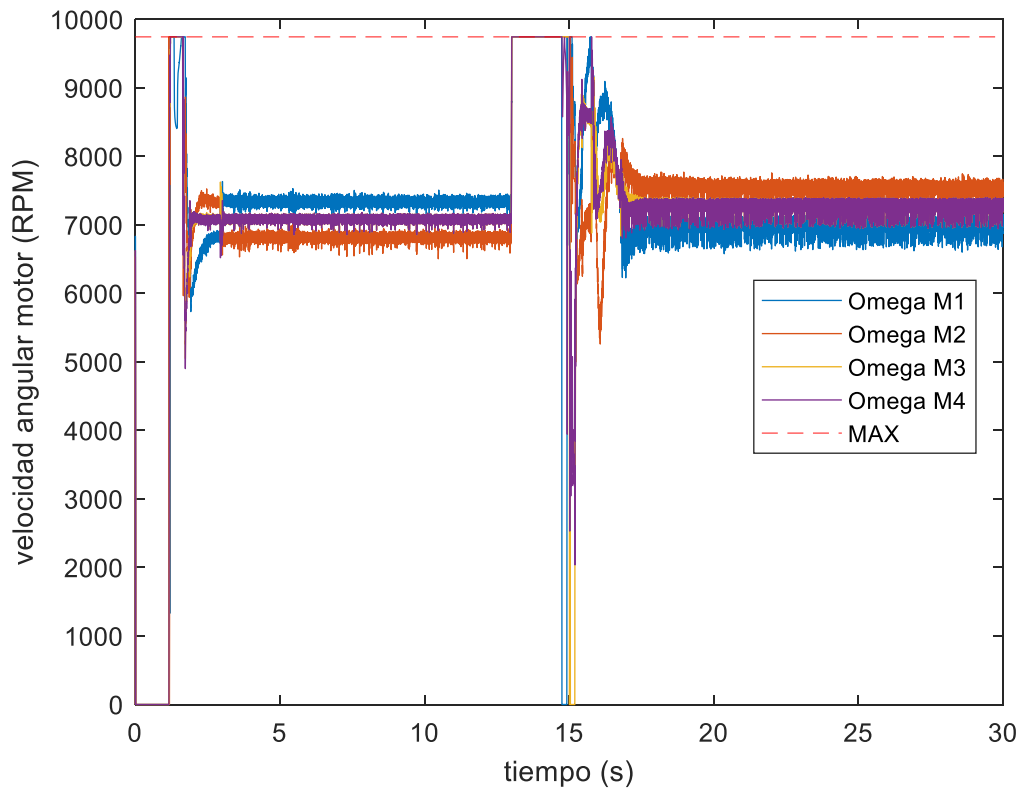


Ilustración 136-RPM de los motores.

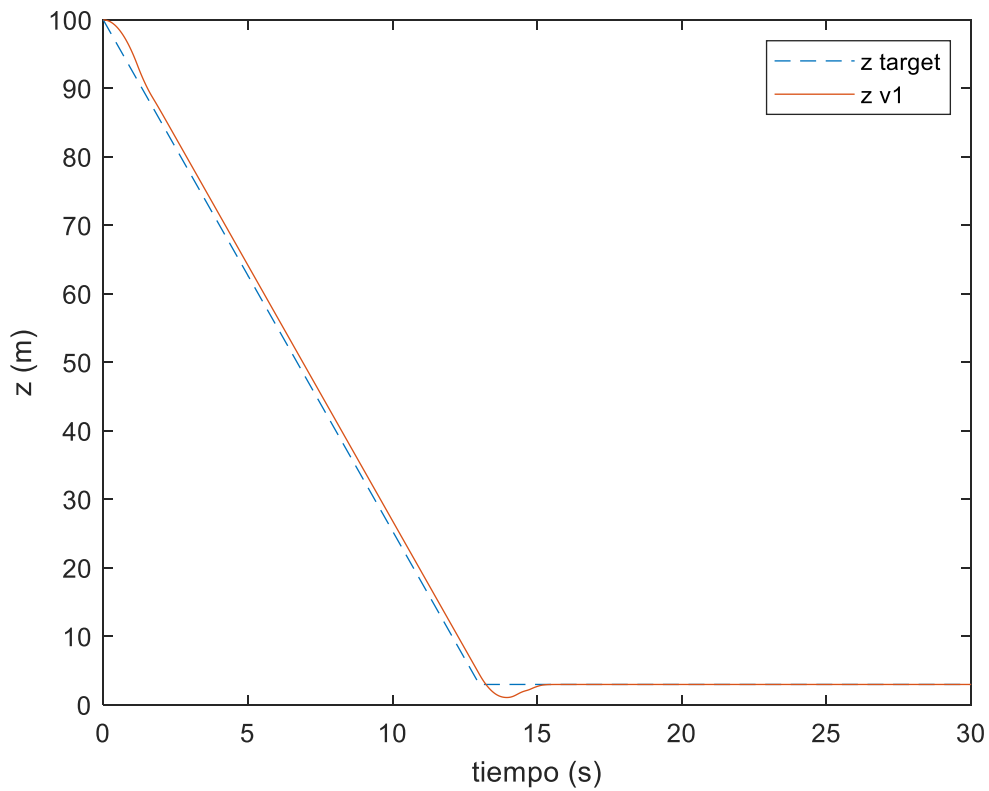


Ilustración 137- Respuesta en Z.

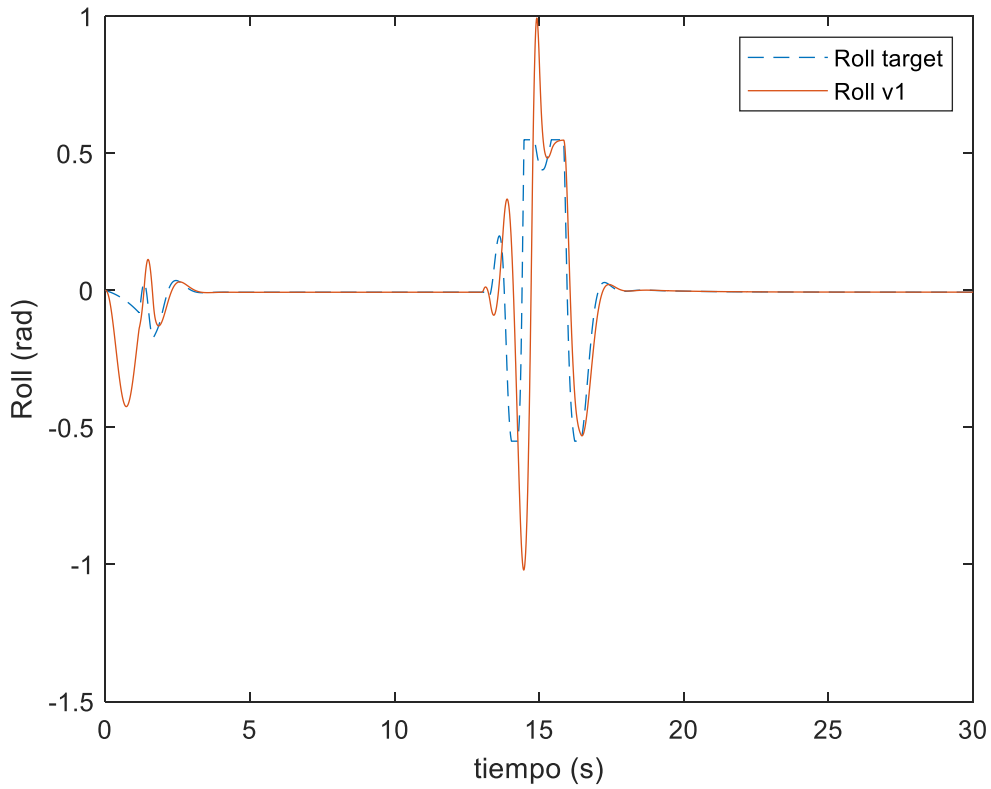


Ilustración 138- Angulo de roll.

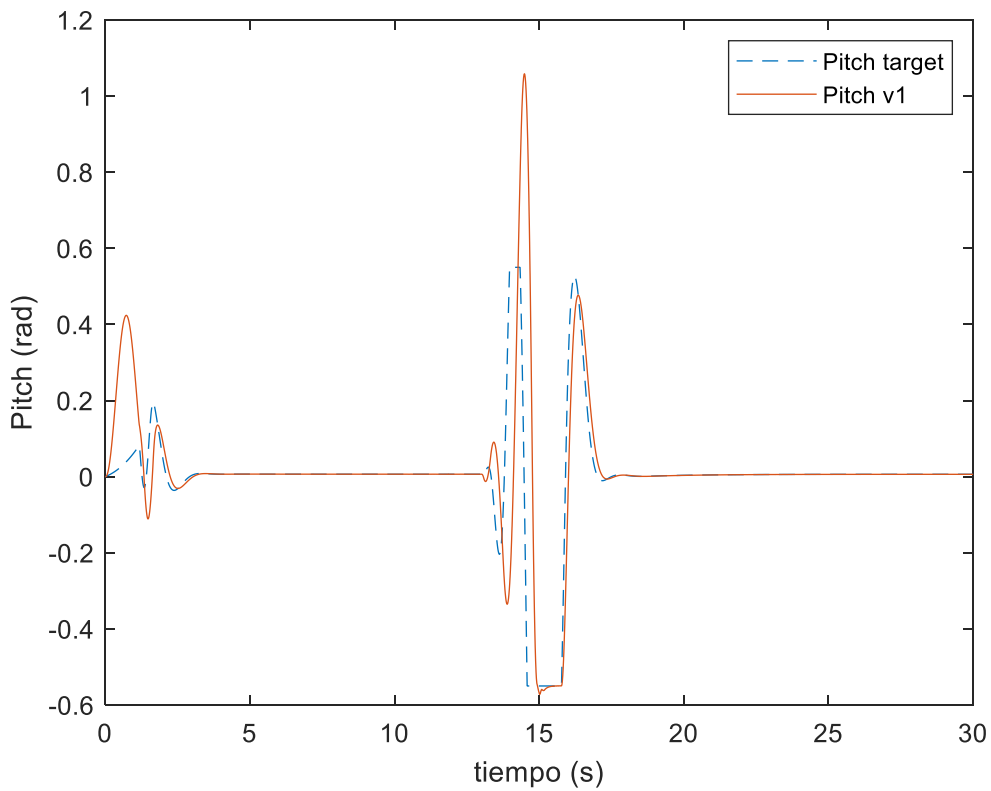


Ilustración 139- Angulo de Pitch.

10.3.3.2 CASO 2

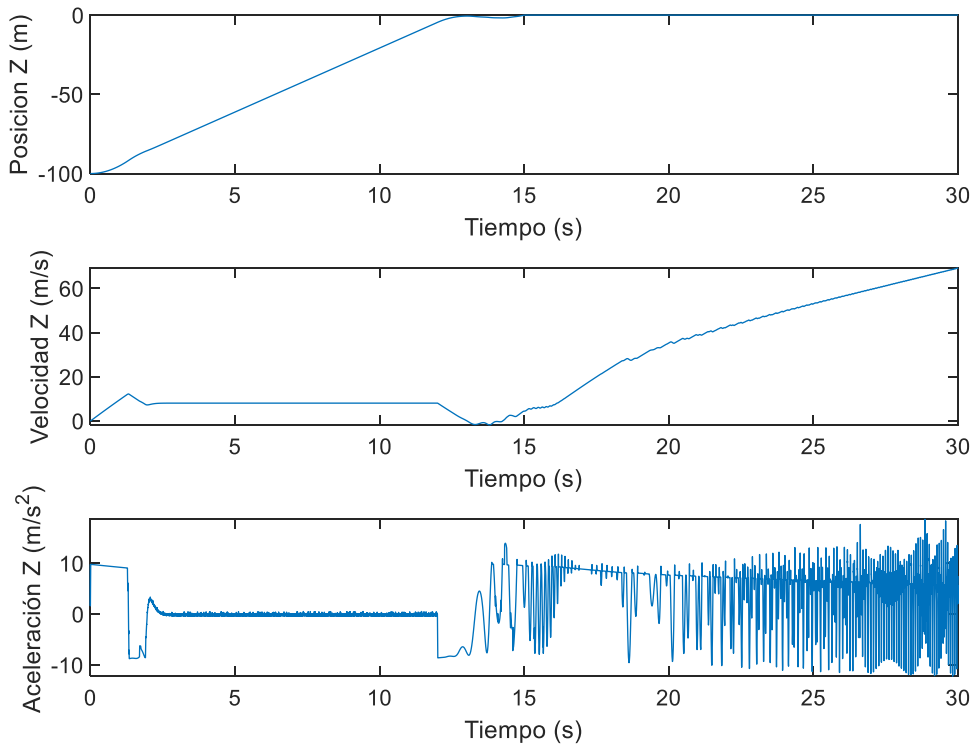


Ilustración 140- Posición, velocidad y aceleración en Z.

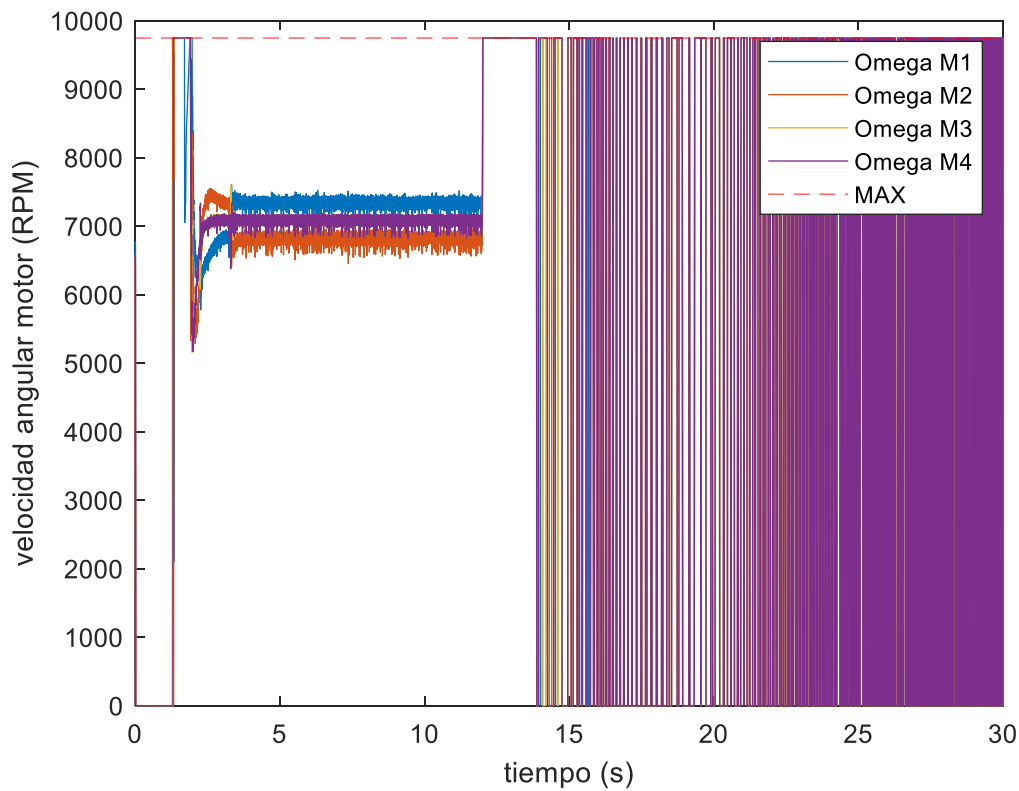


Ilustración 141- RPM de los motores.

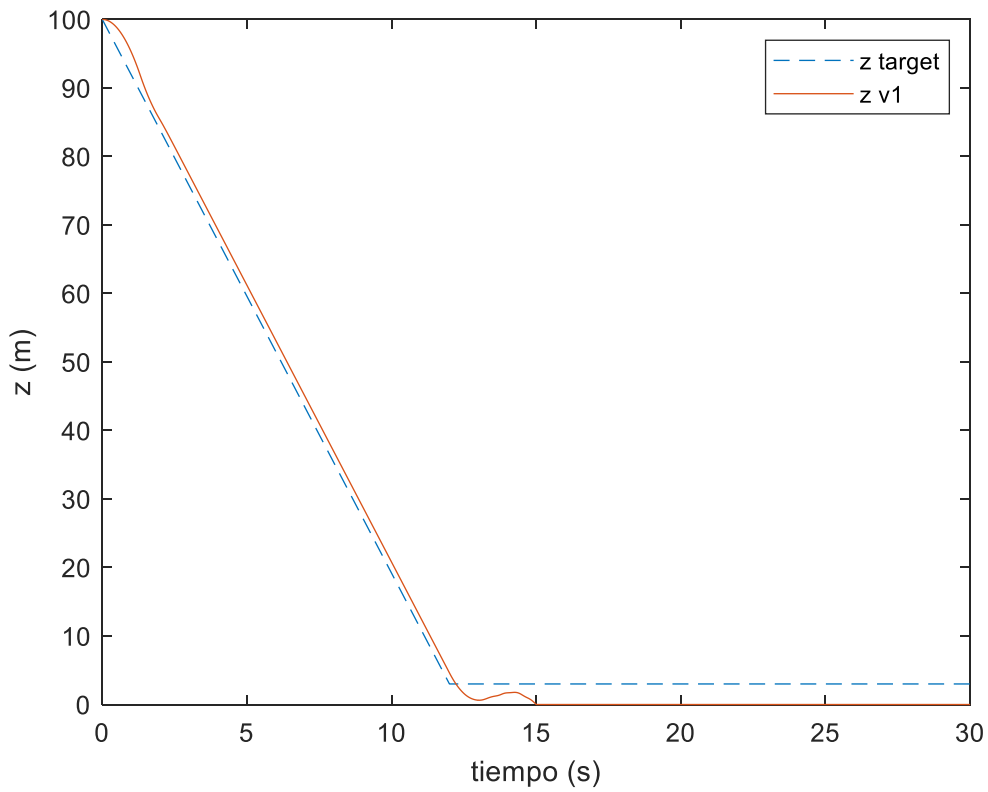


Ilustración 142- Respuesta en Z.

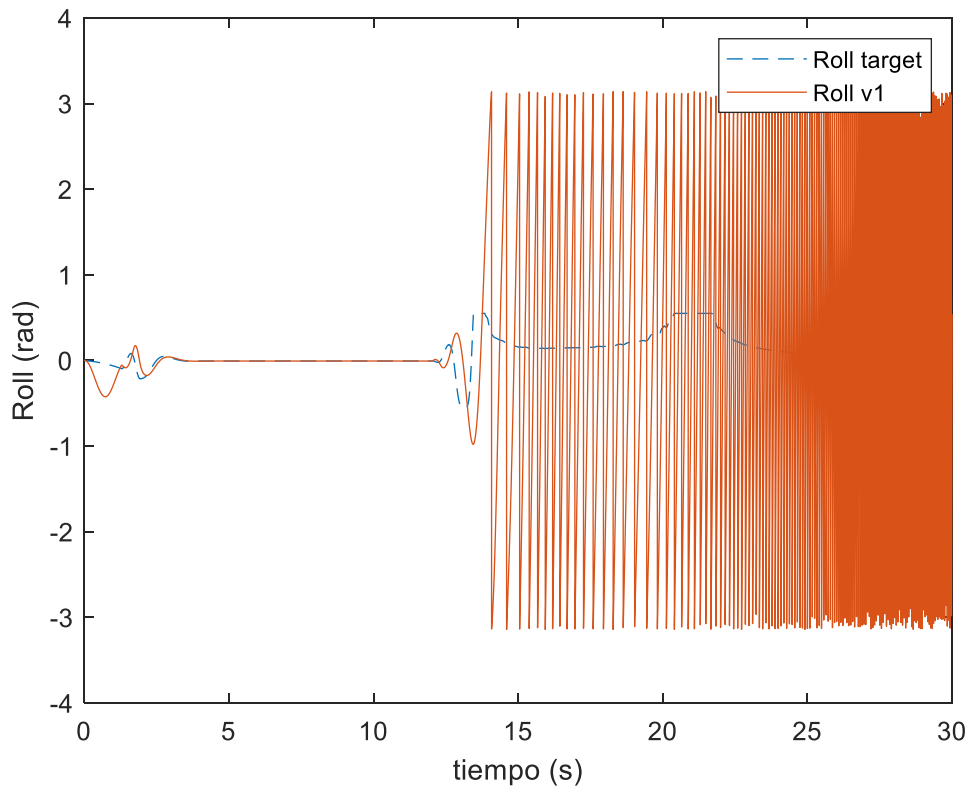


Ilustración 143- Angulo de Roll.

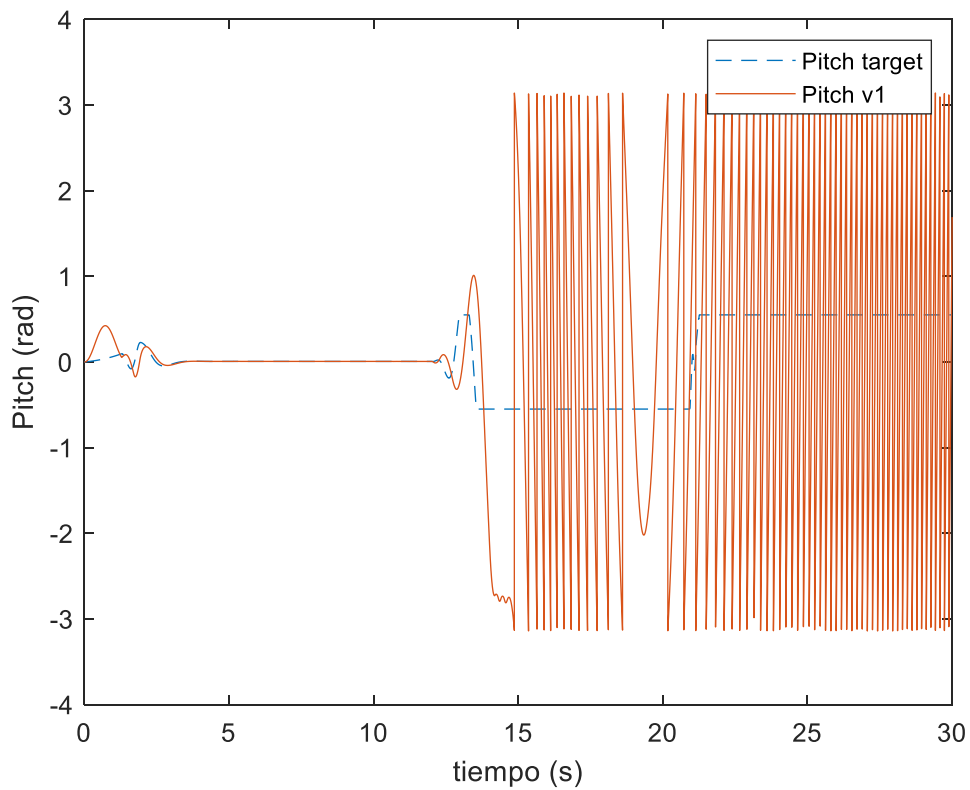


Ilustración 144- Angulo de Pitch.

Se obtiene el valor de la velocidad máxima de descenso de la Ilustración 136.

$$V_{MAX}^{DOWN} = 7,4 \text{ m/s}$$

10.4 ACELERACIÓN VERTICAL MAXIMA

10.4.1 ACELERACION ASCENSO

Para la obtención de la aceleración máxima de ascenso se comanda para la posición vertical una trayectoria con forma de exponencial y en instante determinado de la simulación el sistema alcanzará su valor de aceleración máxima y se caerá al suelo.

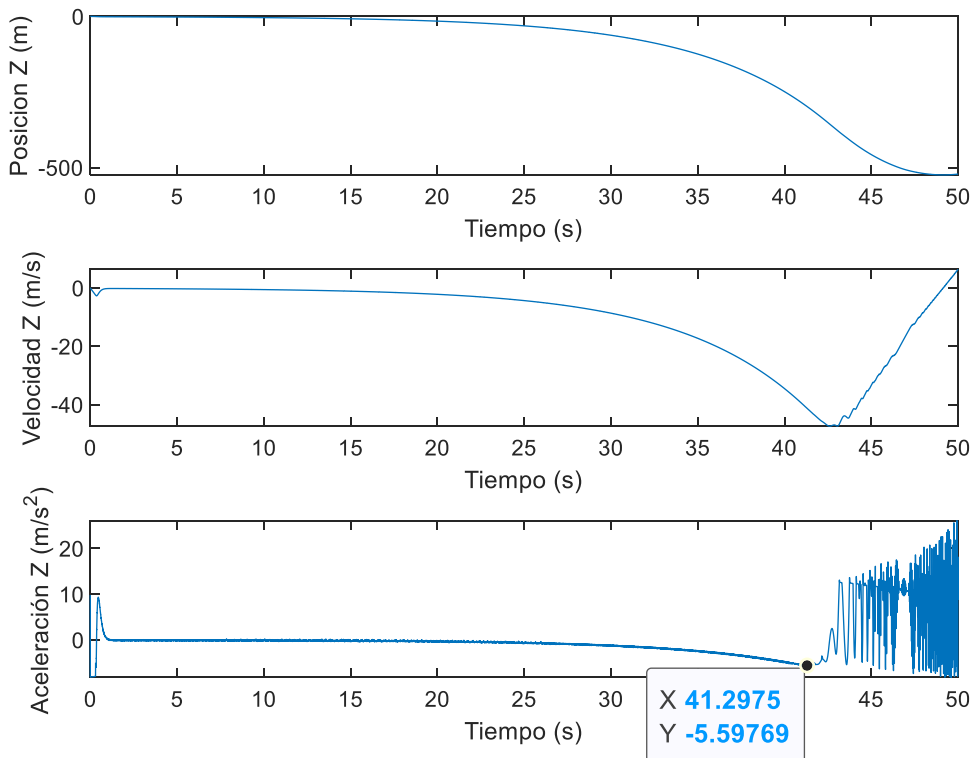


Ilustración 145- Posición, velocidad y aceleración en Z.

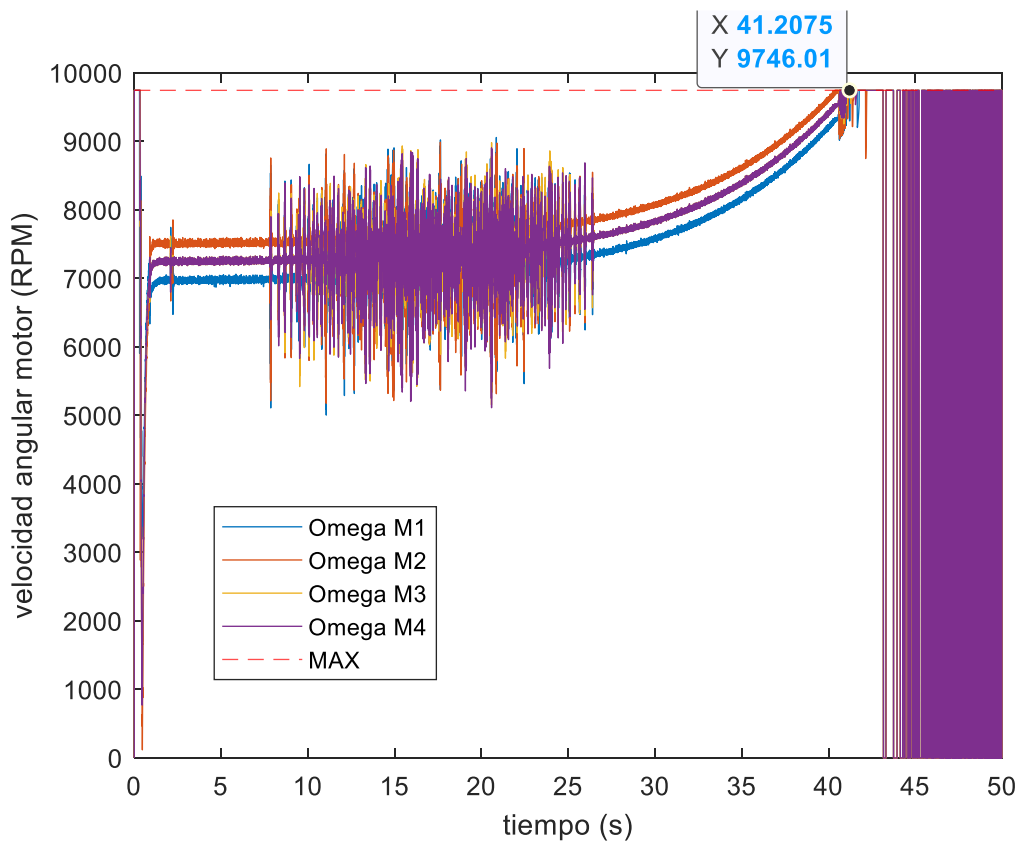


Ilustración 146- RPM de los motores.

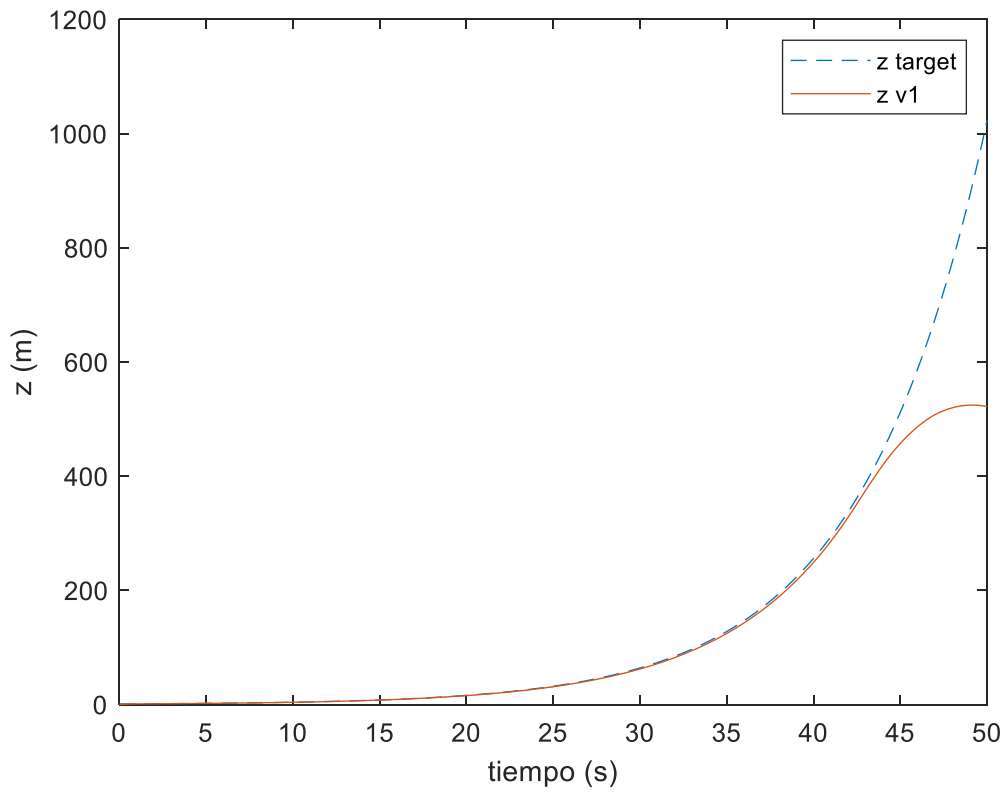


Ilustración 147- Respuesta en Z.

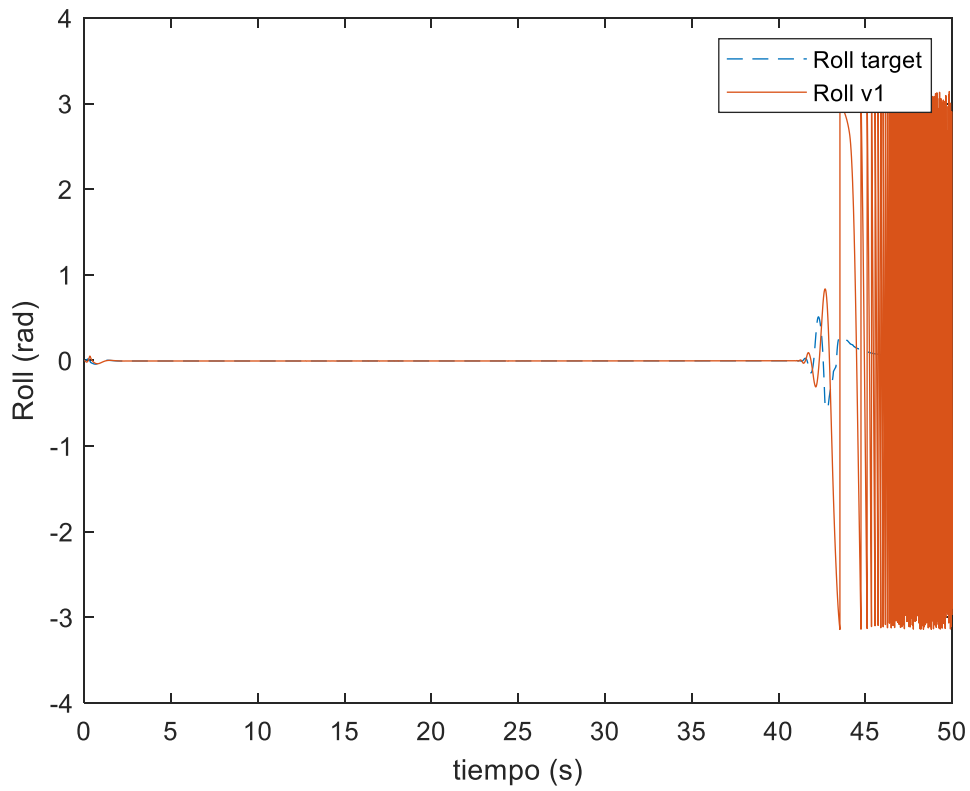


Ilustración 148- Angulo de Roll.

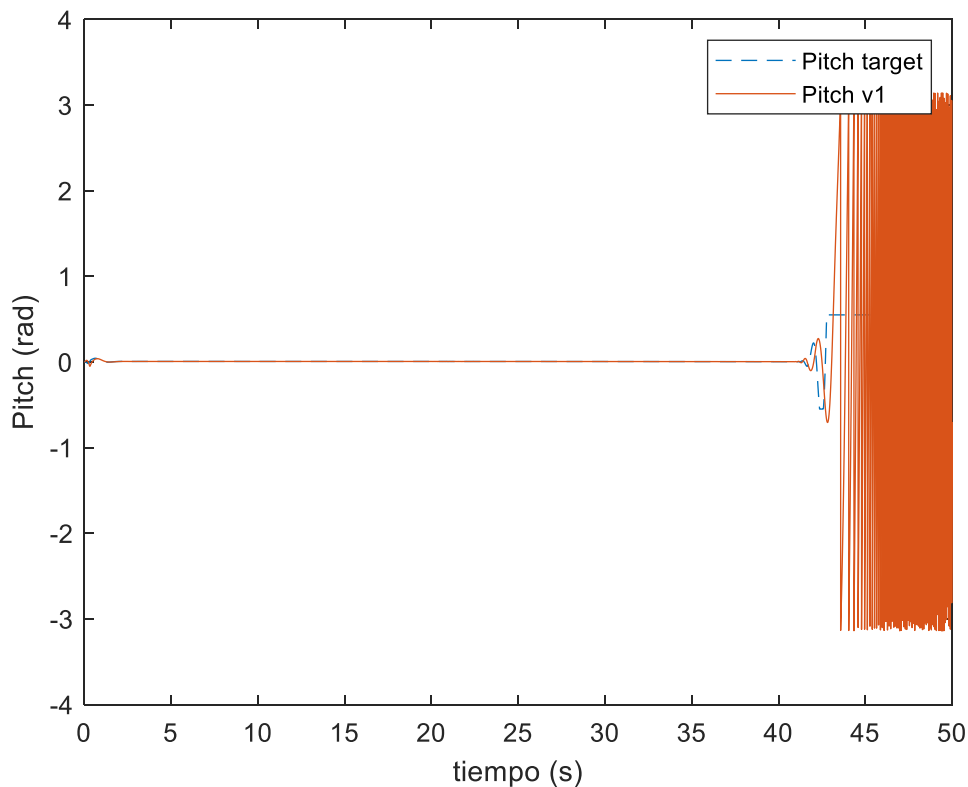


Ilustración 149- Angulo de Pitch.

Se obtiene el valor de la aceleración máxima de ascenso de la Ilustración 146.

$$A_{MAX}^{UP} = 5,6 \text{ m/s}^2$$

10.4.2 ACELERACIÓN DESCENSO

Este caso es un poco especial, debido a que no se va a utilizar la metodología del apartado anterior. Esto se debe a que, si se simula que el UAV parte inicialmente de una altura cualquiera y se comanda una trayectoria de descenso exponencial, el sistema lo que hace es no activar los motores hasta que la posición vertical queda por debajo de la referencia, por lo que nunca se activaran los motores. Para obtener este parámetro lo que se ha hecho es situar el dron a una altura inicial determinada y simular una entrada escalón de descenso lo suficientemente grande para que el dron no consiga corregir su posición y por lo tanto caerá al suelo. Como en apartados anteriores lo que se ha hecho es presentar los resultados del caso limite que consigue corregir y del caso en el cual el dron cae al suelo.

10.4.2.1 CASO 1

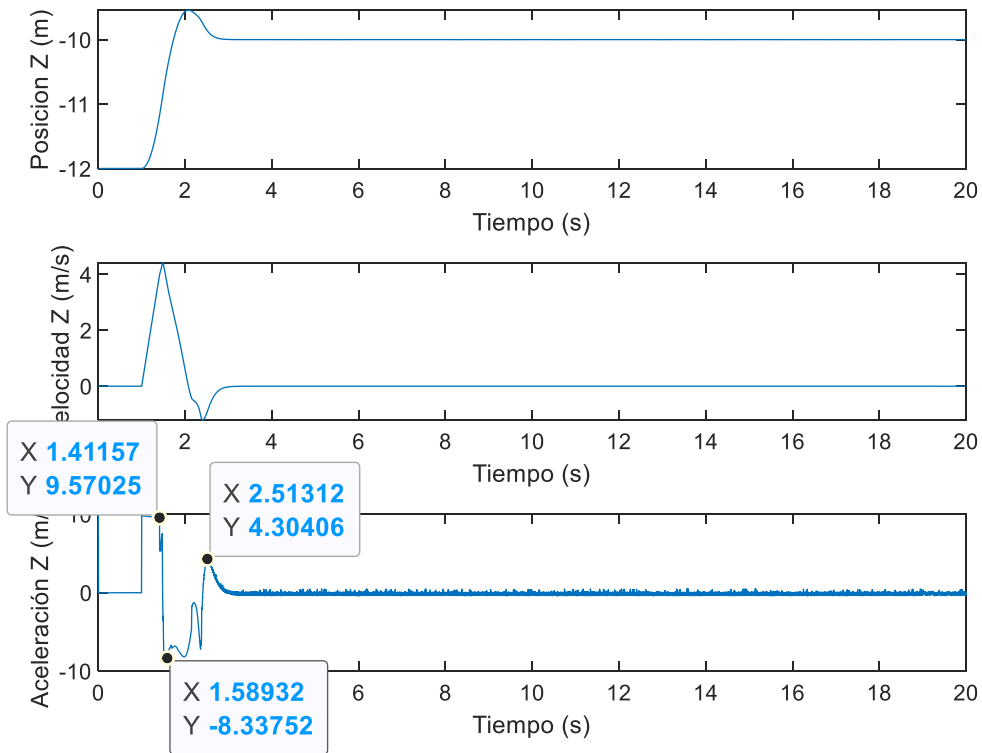


Ilustración 150- Posición, velocidad y aceleración en Z.

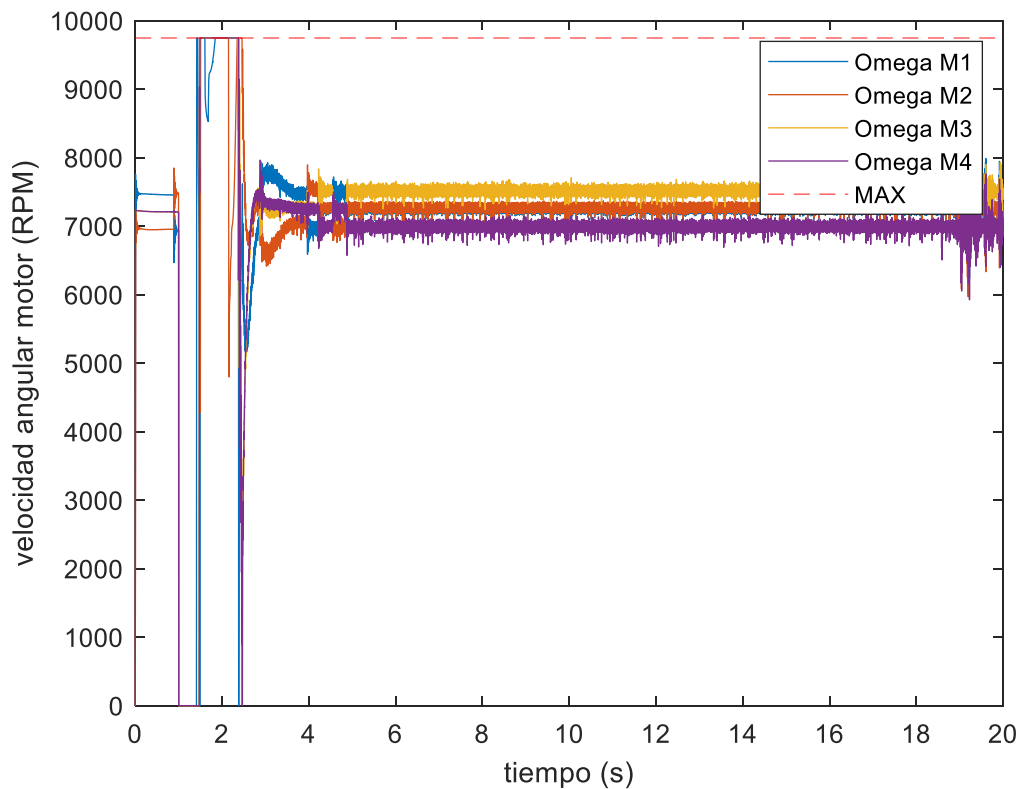


Ilustración 151- RPM de los motores.

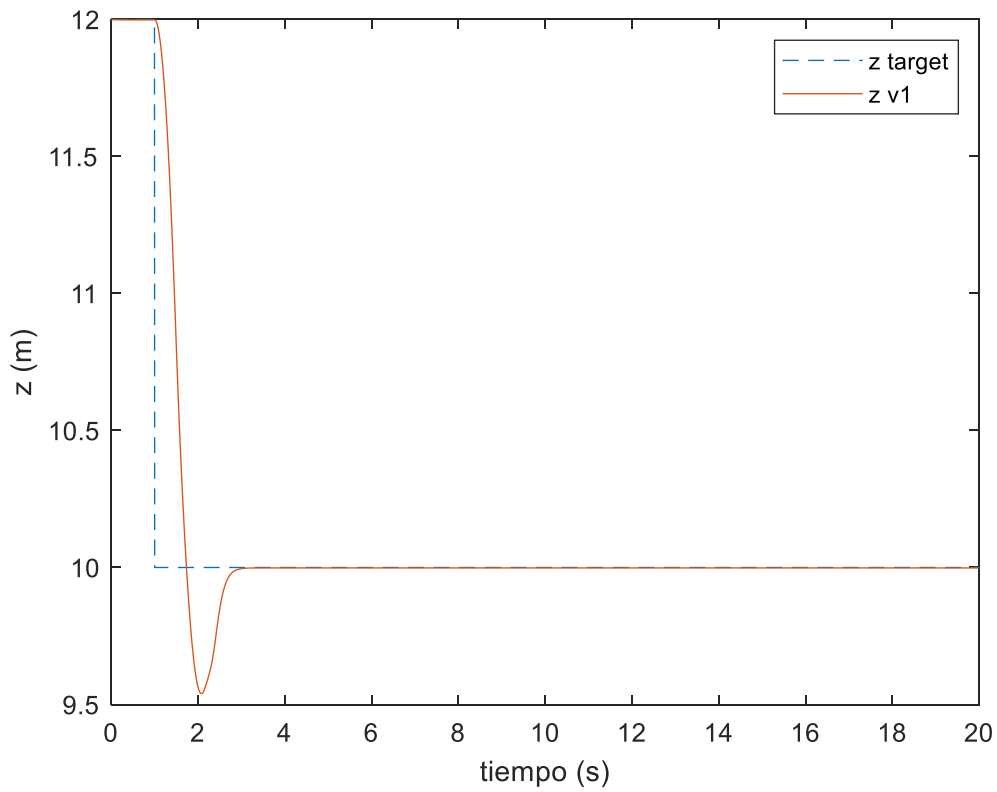


Ilustración 152- Respuesta en Z.

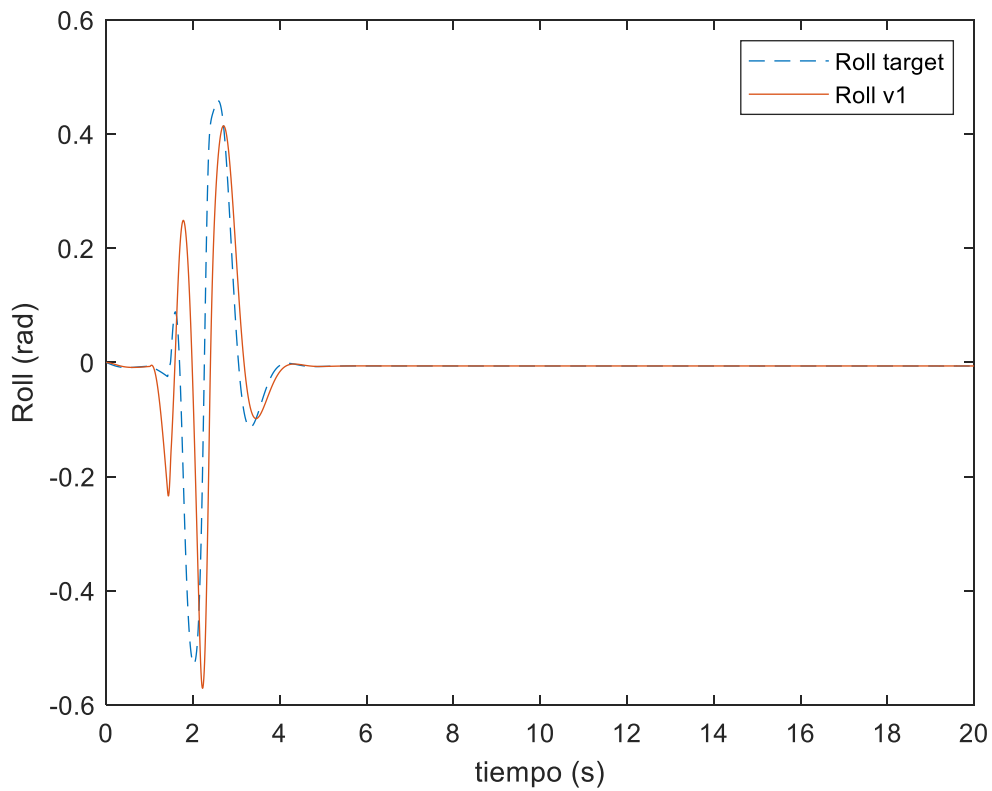


Ilustración 153- Angulo de Roll.

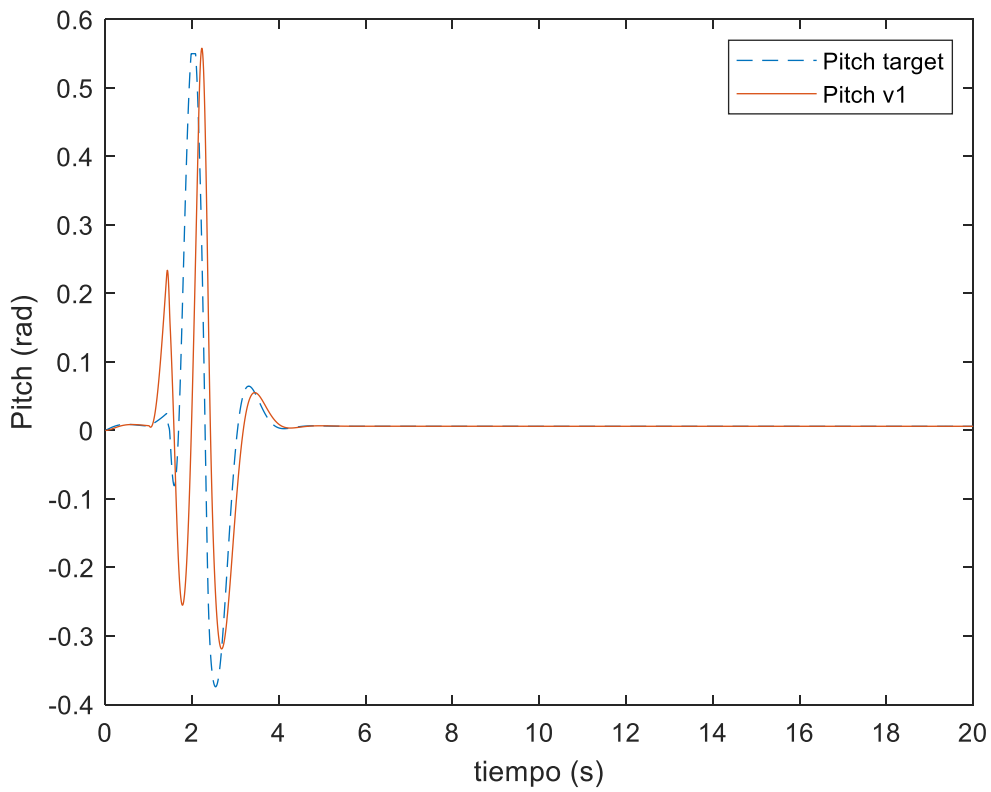


Ilustración 154- Angulo de Pitch.

10.4.2.2 CASO 2

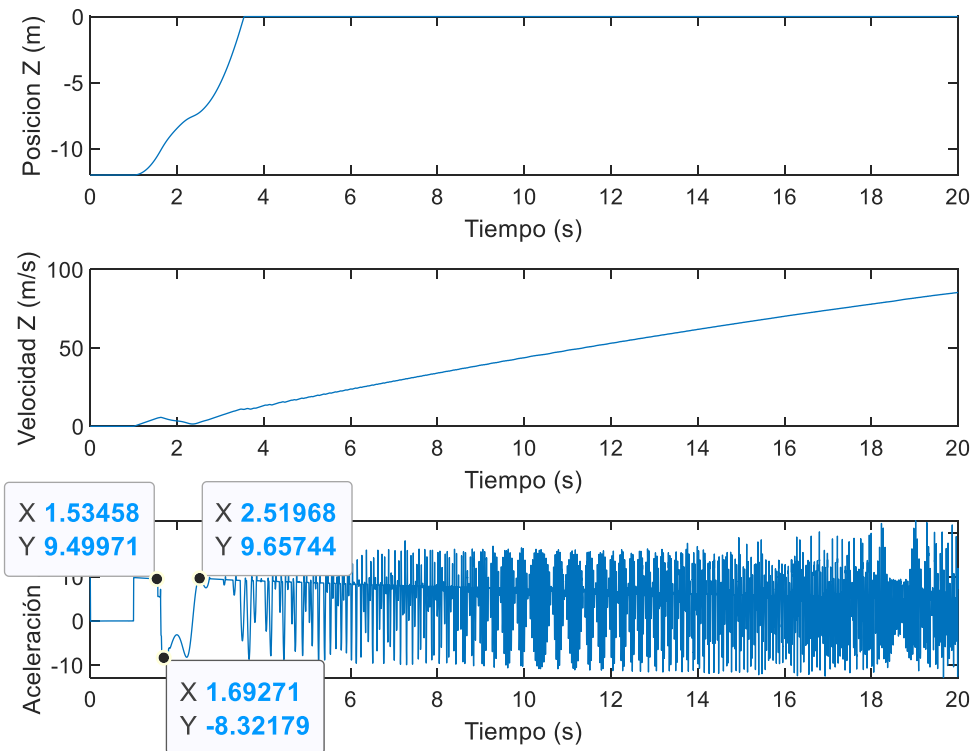


Ilustración 155- Posición, velocidad y aceleración en Z.

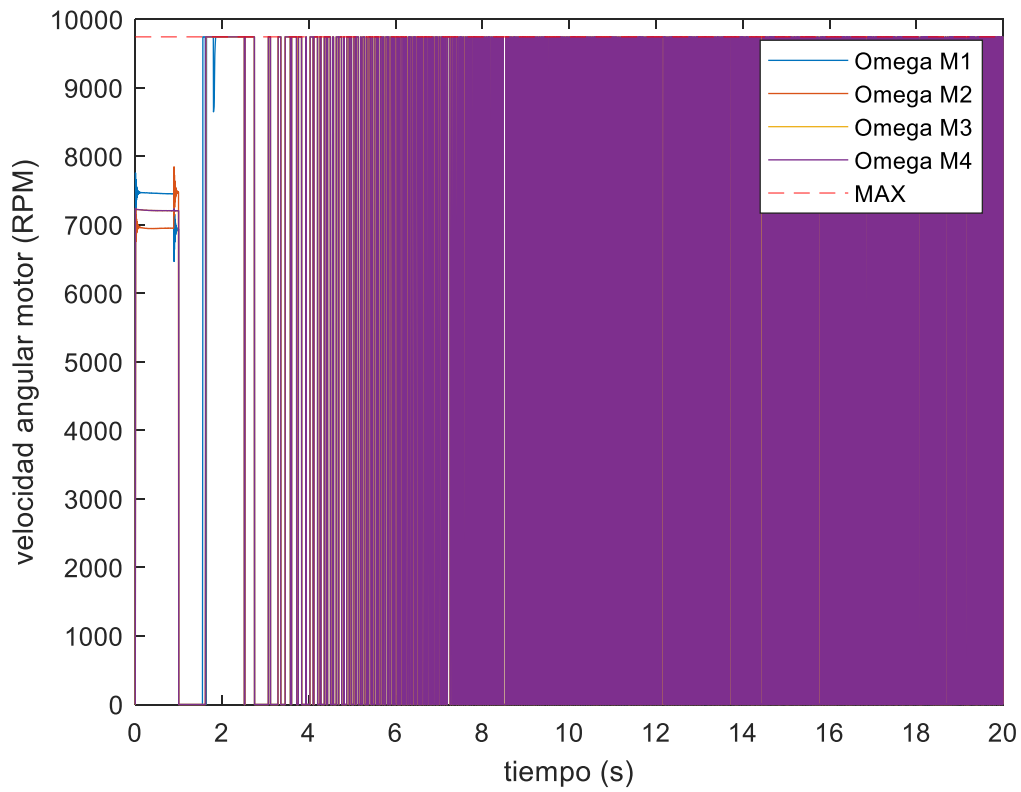


Ilustración 156- RPM de los motores.

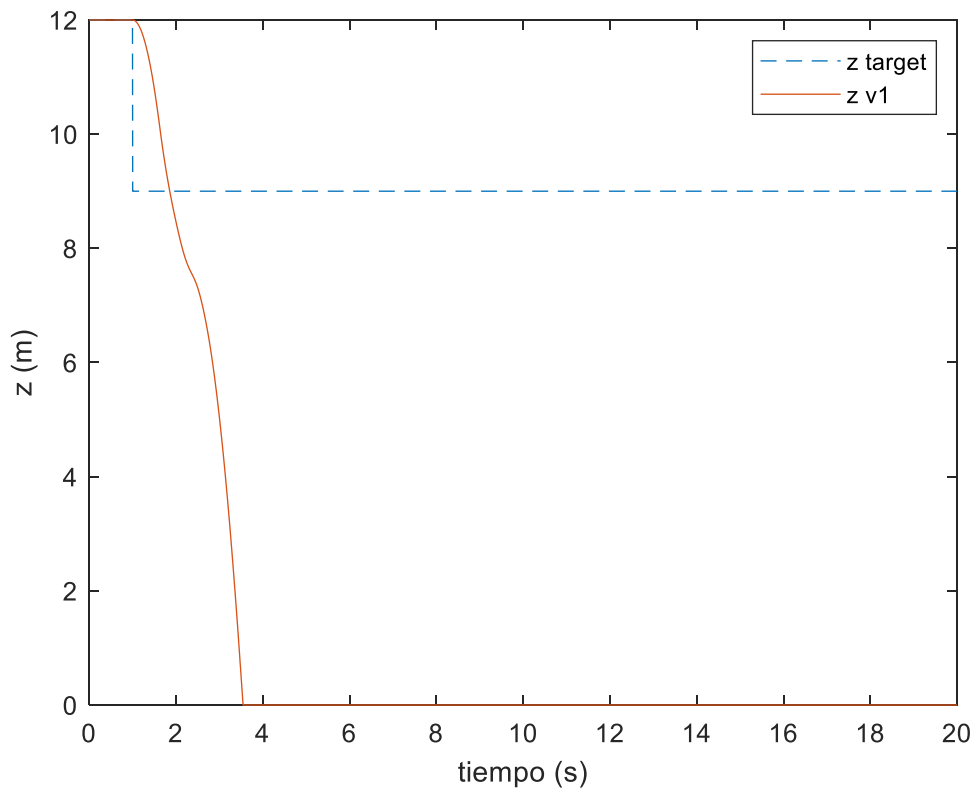


Ilustración 157- Respuesta en Z.

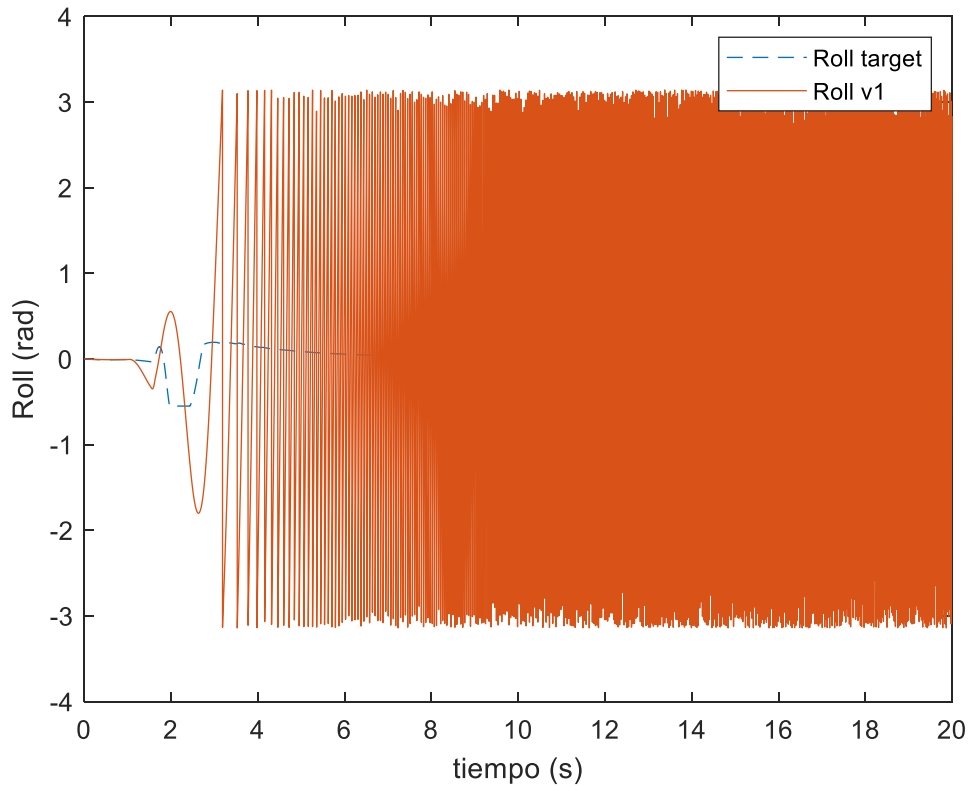


Ilustración 158- Angulo de Roll.

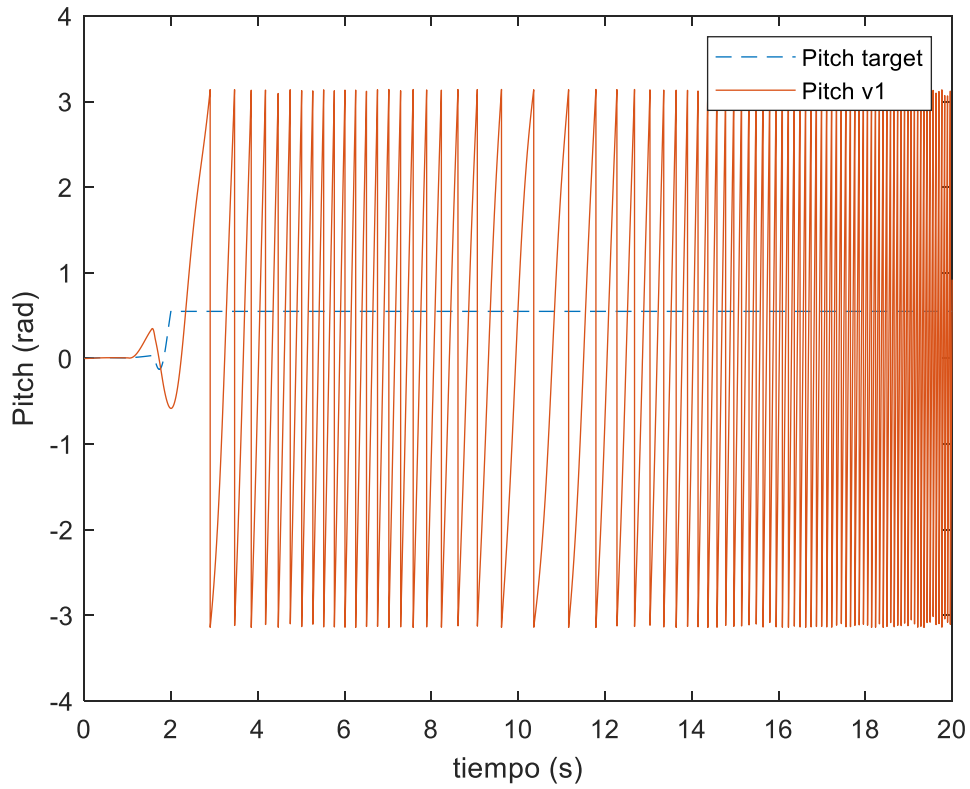


Ilustración 159 - Angulo de Pitch.

Se ha obtenido el valor de la aceleración máxima de descenso de la Ilustración 151.

$$A_{MAX}^{DOWN} = 4,3 \text{ m/s}^2$$

10.5 VIENTO MÁXIMO

Otro de los parámetros más interesantes que hay que definir es la cantidad de viento con la cual es seguro volar el dron. Este parámetro es de vital importancia ya que nos dará una indicación de cuando se debería abortar o aplazar una operación. Se han introducido dos tipos de viento: uno uniforme y otro turbulento. El viento turbulento se ha simulado como un viento uniforme al cual se le ha sumado un ruido el cual provoca cambios en la magnitud de la velocidad del viento. Tras realizar las simulaciones con ambos tipos de viento es curioso apreciar que para ambos modelos se alcanza el límite del sistema para el mismo valor de velocidad media. En los apartados siguientes se presentarán los resultados de las simulaciones.

10.5.1 VIENTO HORIZONTAL UNIFORME

10.5.1.1 Viento uniforme en dirección X e Y de 10 m/s

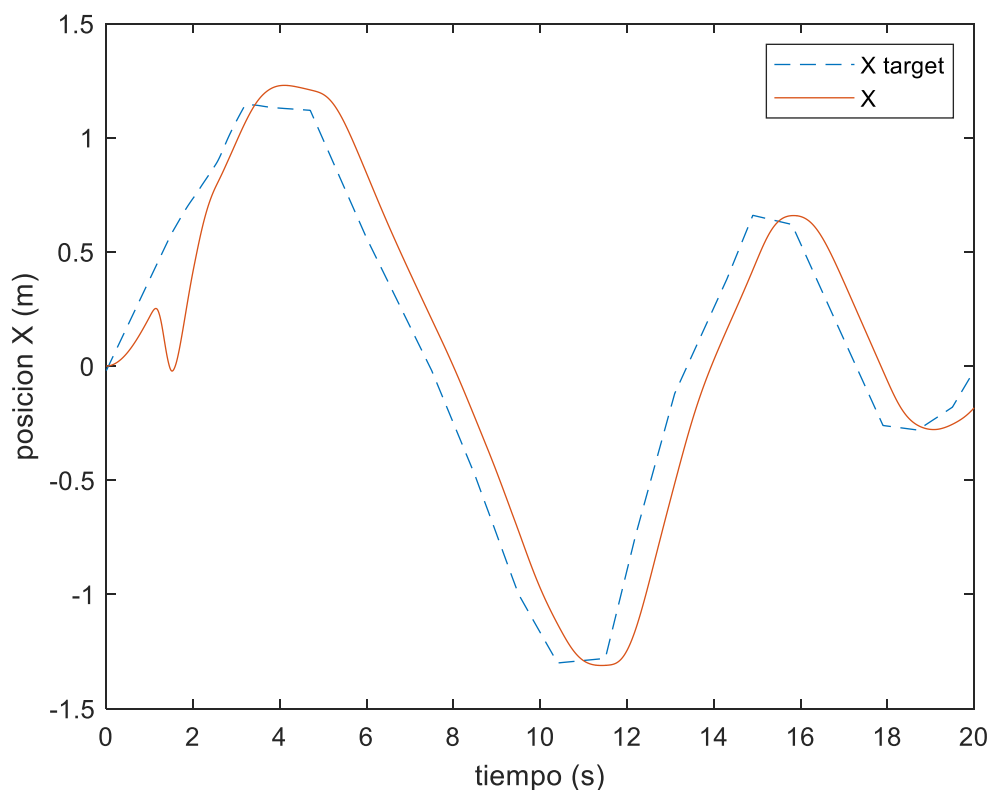


Ilustración 160- Coordenada X.

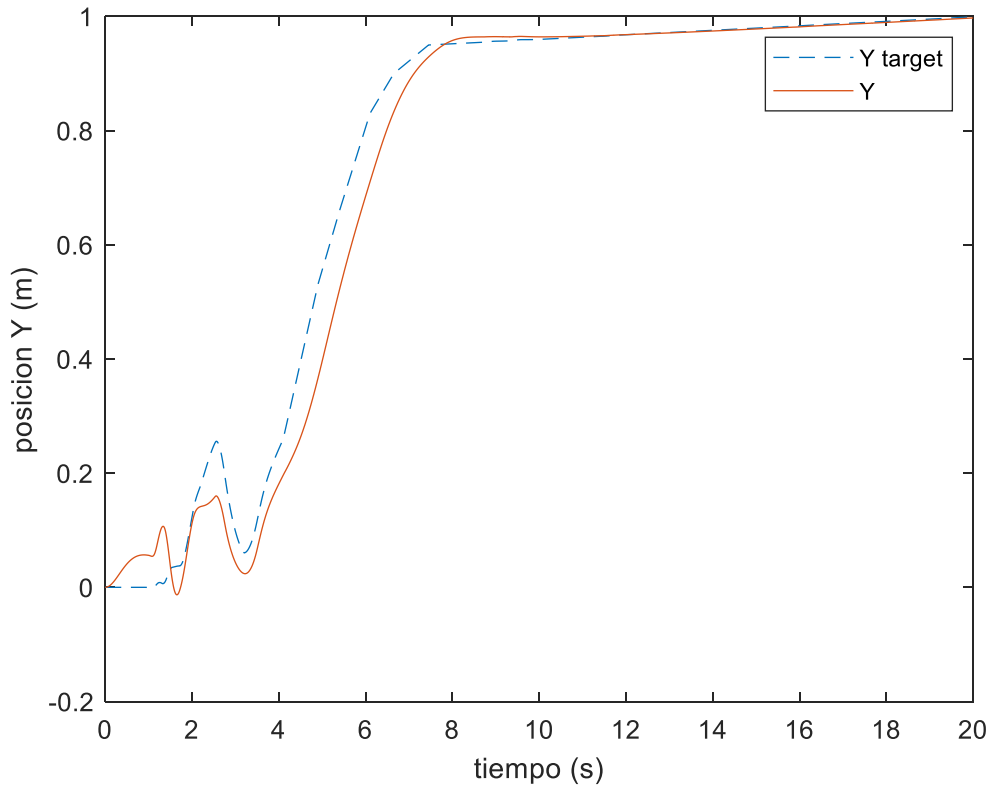


Ilustración 161- Coordenada Y.

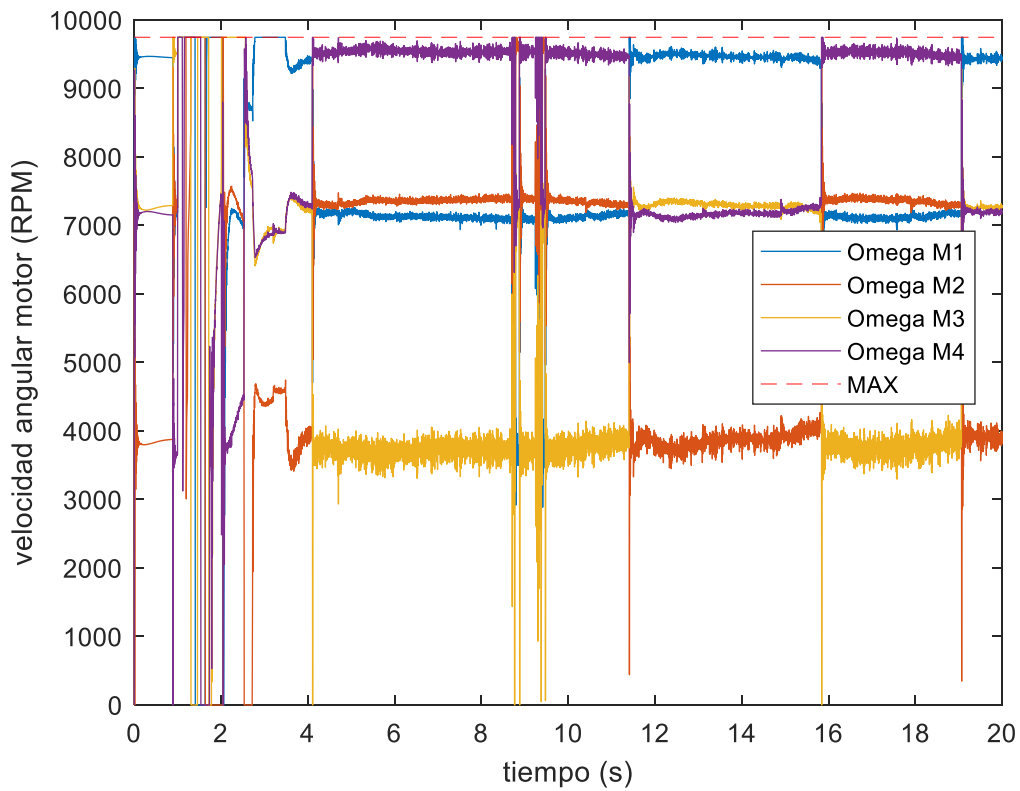


Ilustración 162- RPM de los motores.

10.5.1.2 Viento uniforme en dirección X e Y de 11m/s

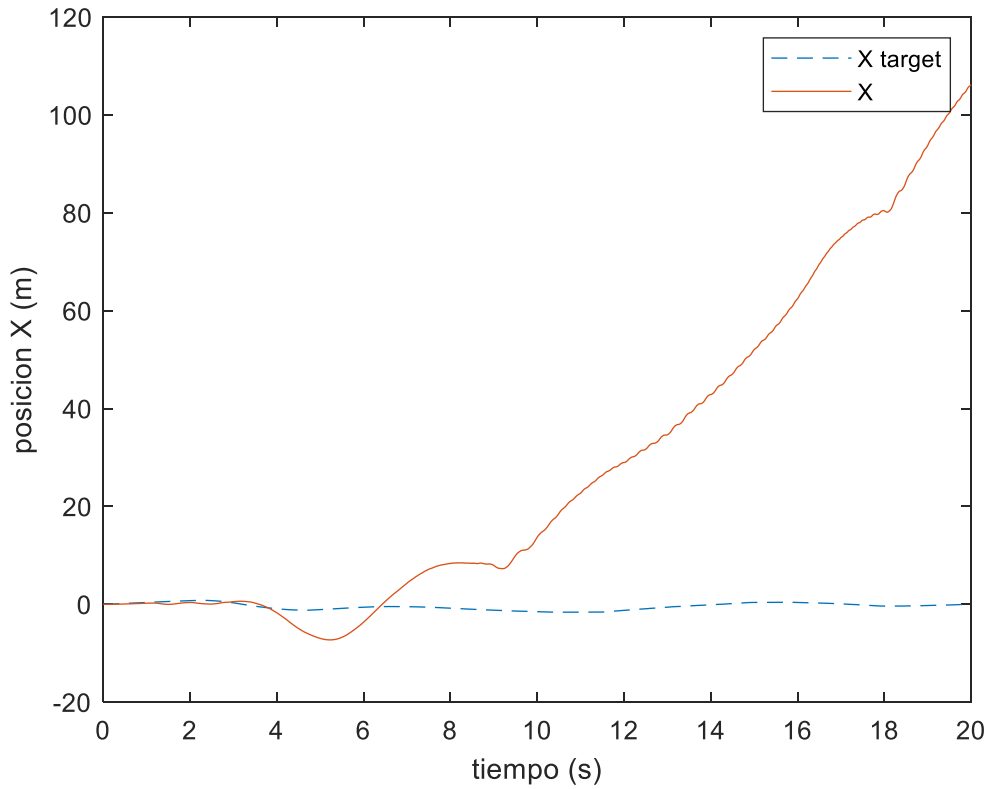


Ilustración 163- Coordenada X.

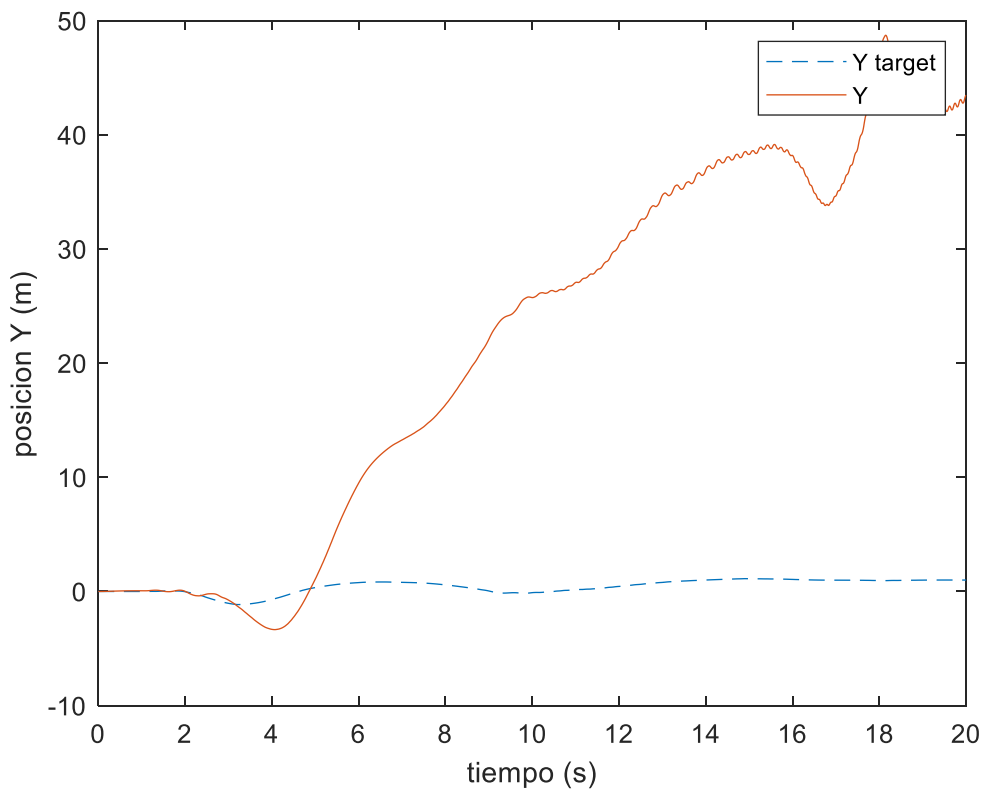


Ilustración 164- Coordenada Y.

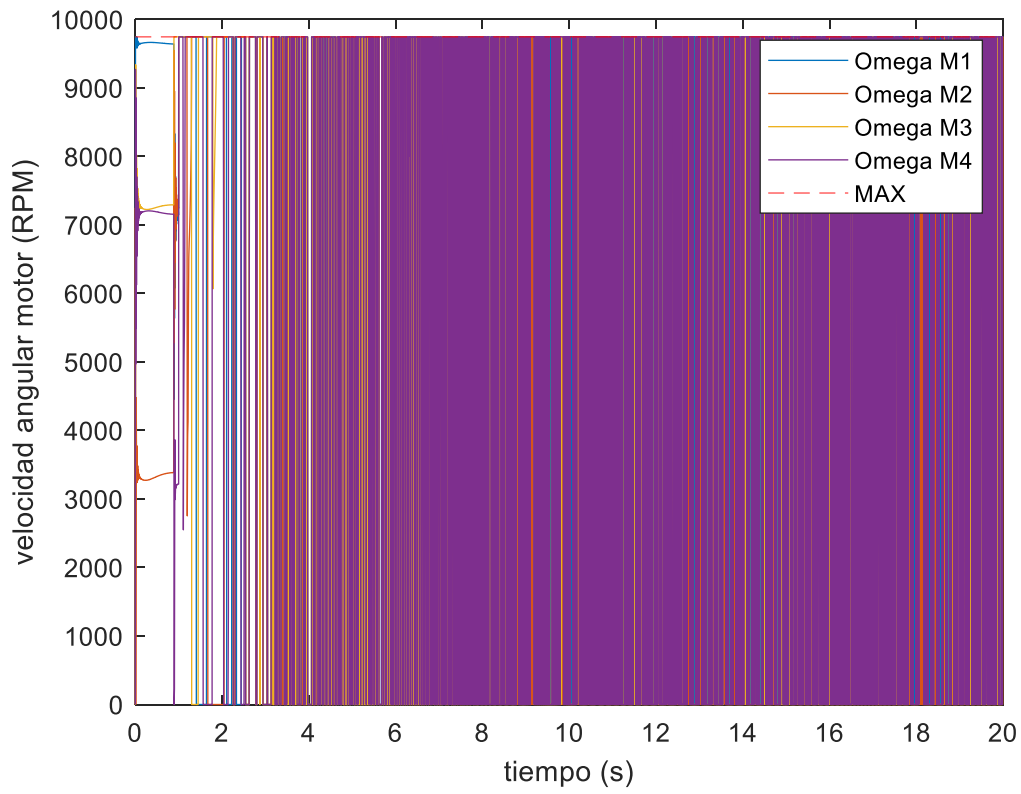


Ilustración 165- RPM de los motores.

10.5.2 VIENTO HORIZONTAL TURBULENTO

10.5.2.1 Velocidad horizontal media de 10 m/s

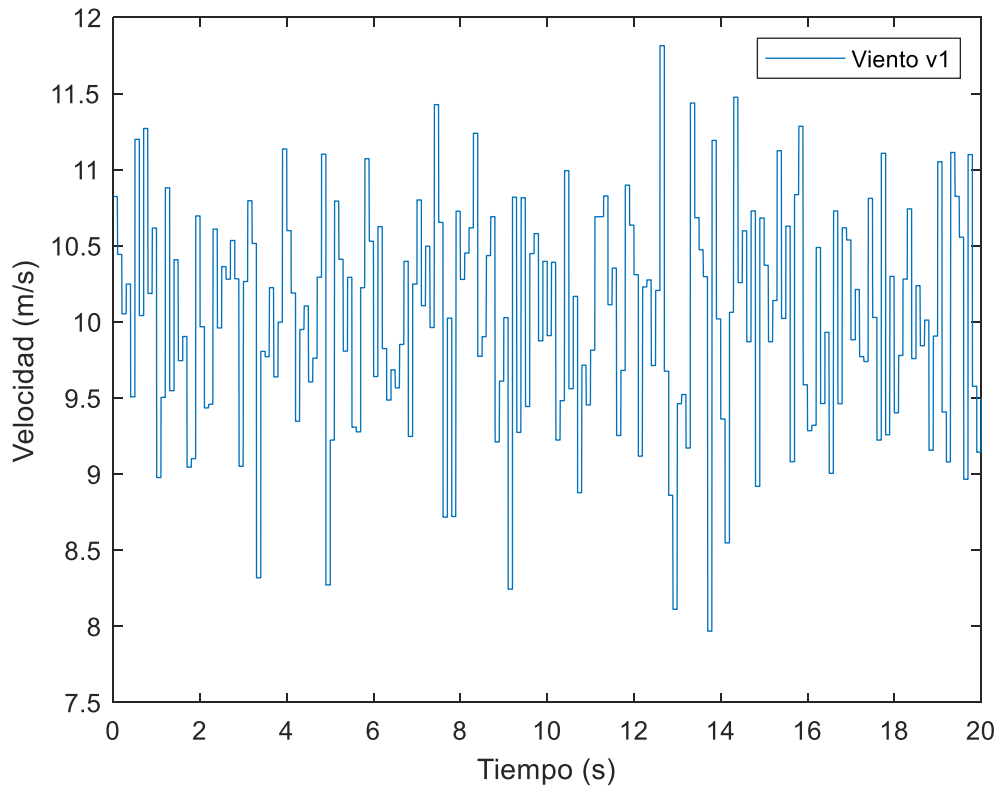


Ilustración 166- Perfil de viento horizontal.

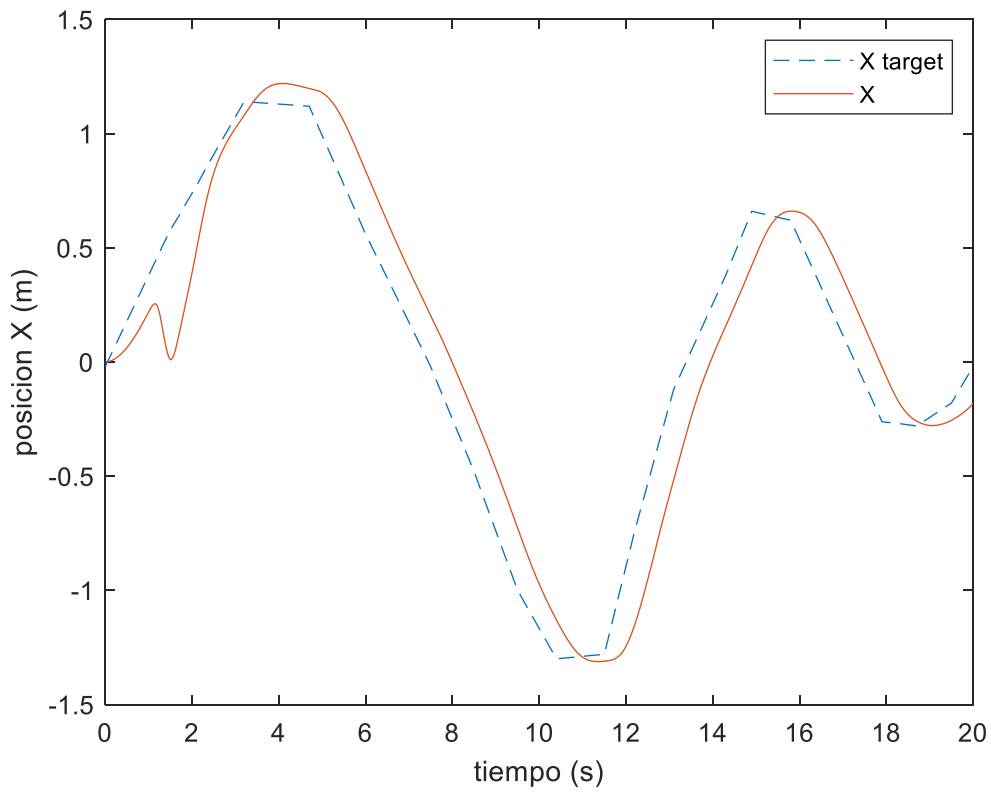


Ilustración 167- Coordenada X.

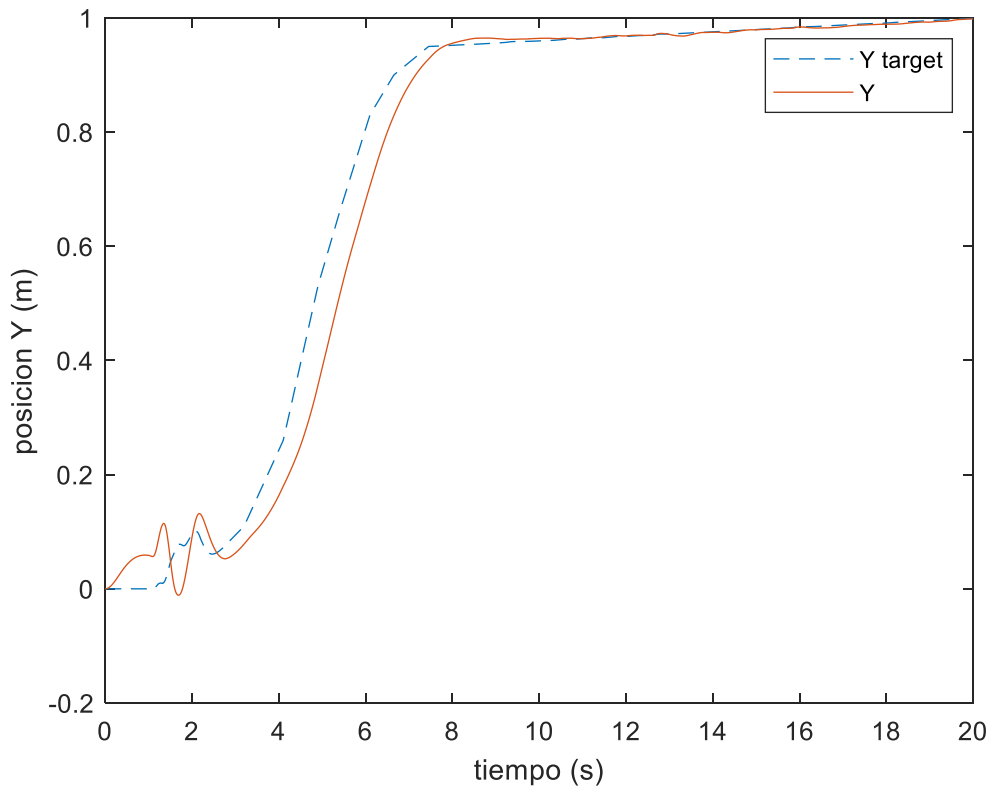


Ilustración 168 - Coordenada Y.

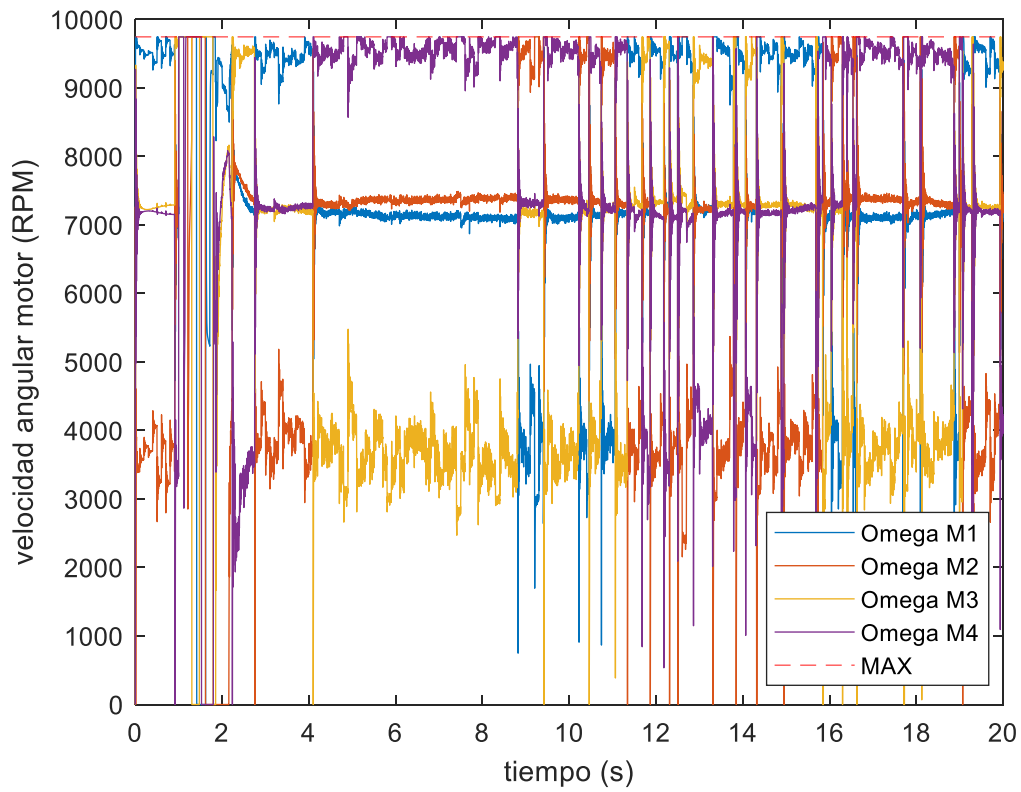


Ilustración 169- RPM de los motores.

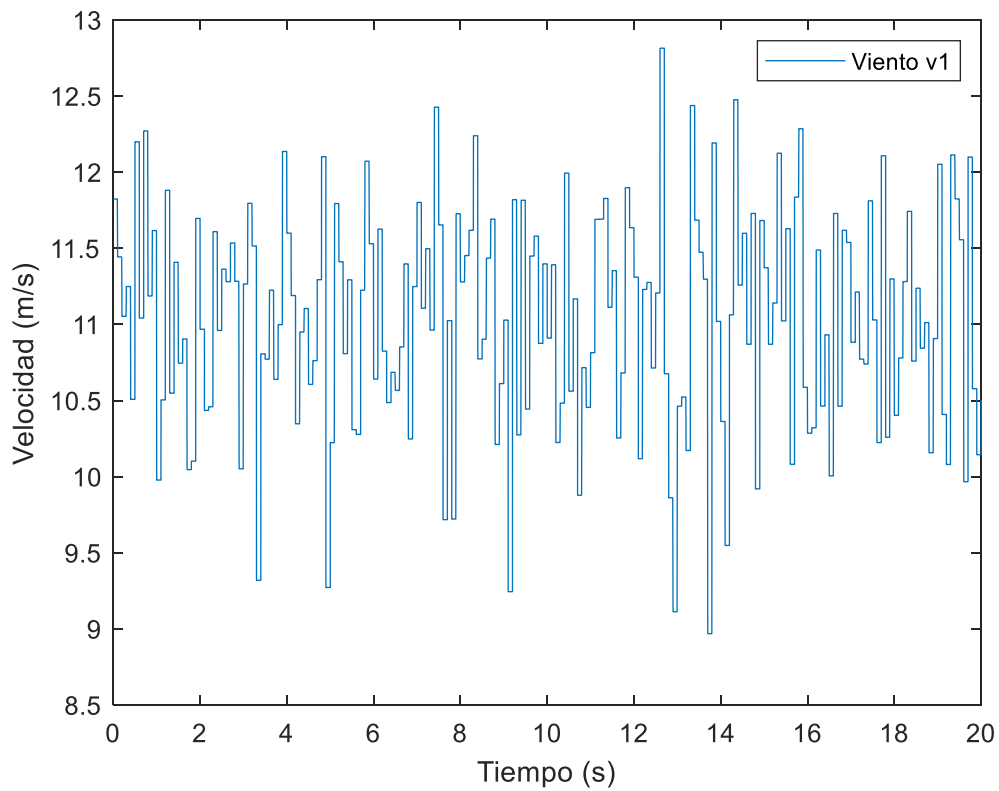


Ilustración 170- Perfil de viento horizontal.

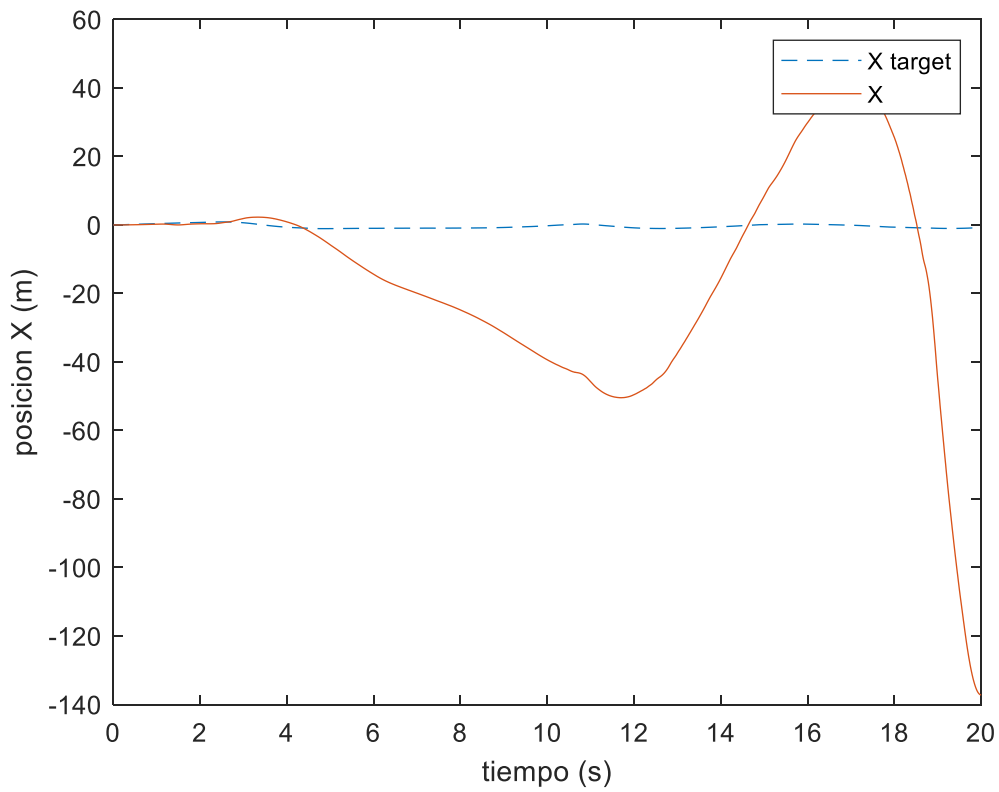


Ilustración 171- Coordenada X.

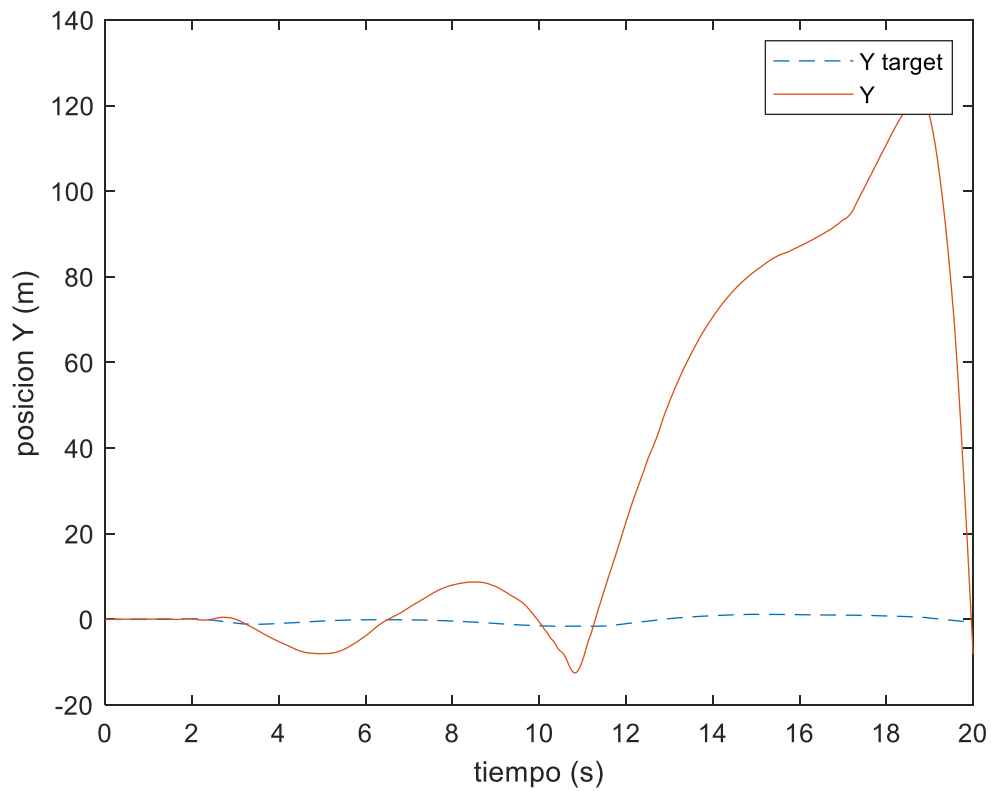


Ilustración 172- Coordenada Y.

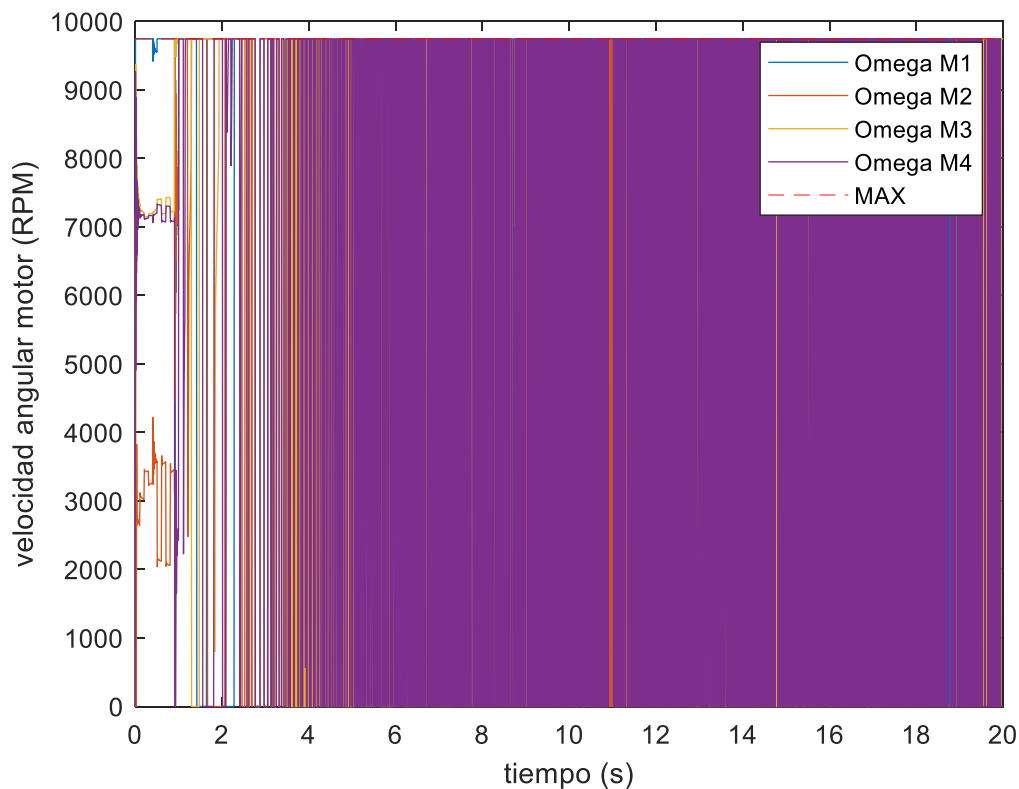


Ilustración 173- RPM de los motores.

Tras las simulaciones con ambos modelos de viento se puede concluir que la velocidad máxima a la cual es seguro volar es de:

$$V_{Viento}^{MAX} = 14 \text{ m/s}$$

Este valor se calcula obteniendo la hipotenusa del triángulo que forman los vectores de velocidad de viento en X e Y. Para velocidades de viento de 10 m/s en X e Y se obtiene una velocidad máxima absoluta de 14 m/s.

10.6 MÁXIMA CARGA EXTRA

Por último, se calculará el valor de la carga de pago máxima con la cual el sistema es capaz de realizar una trayectoria. Este valor es útil ya que nos da una aproximación de cuanta masa extra, como por ejemplo un paquete, el sistema es capaz de llevar. En el caso de este proyecto conocer este parámetro es útil debido a que uno de los objetivos es el de incorporar redundancia de sensorica para así realizar funciones inteligentes o mejorar el posicionamiento del sistema. El límite de incorporar sensores al sistema tiene más que ver con el peso de estos sensores que con la integración de estos en el piloto automático.

$$m_{MAX}^{EXTRA} = 2,3 \text{ kg}$$

10.6.1 CARGA DE PAGO 2.3 kg

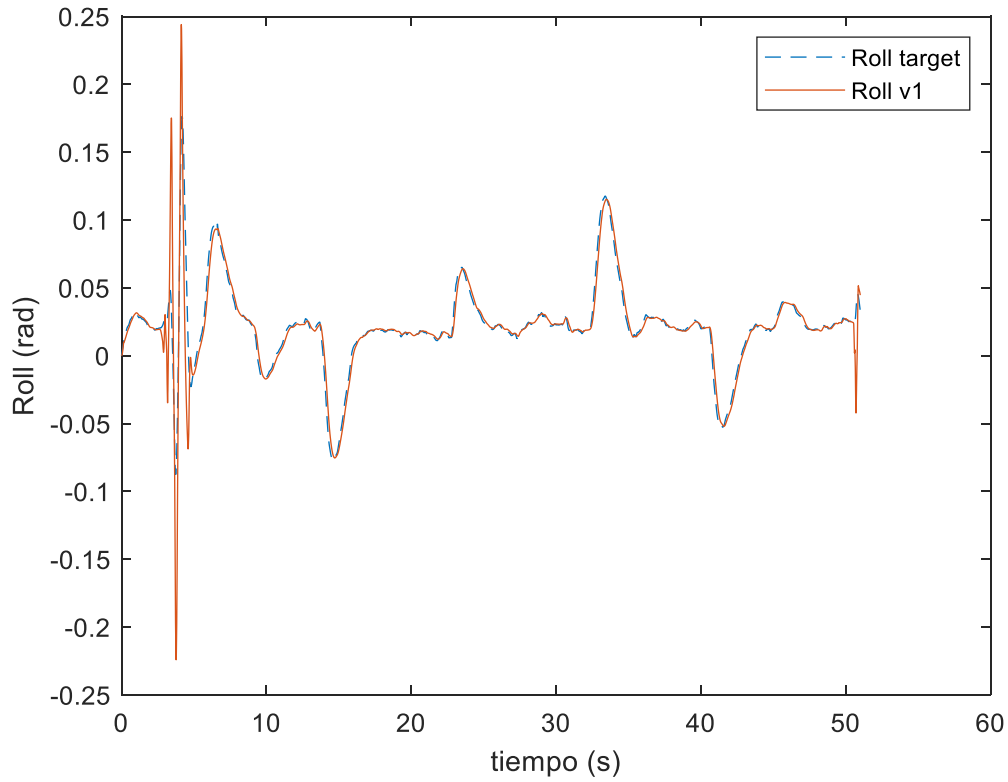


Ilustración 174- Angulo de Roll.

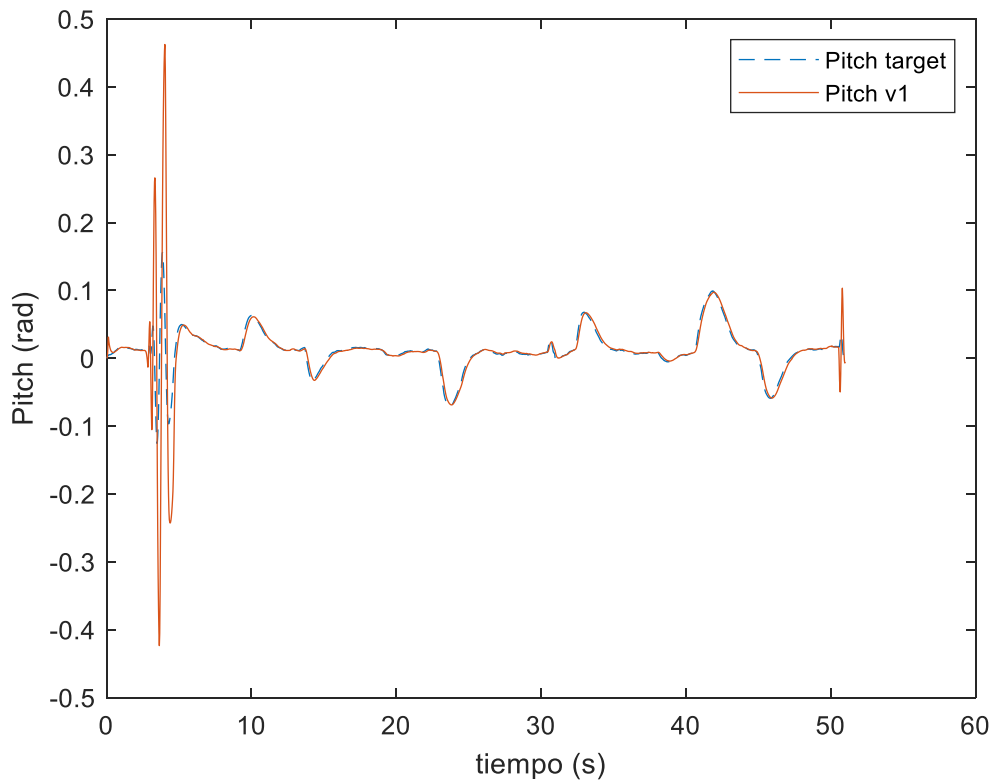


Ilustración 175- Angulo de Pitch.

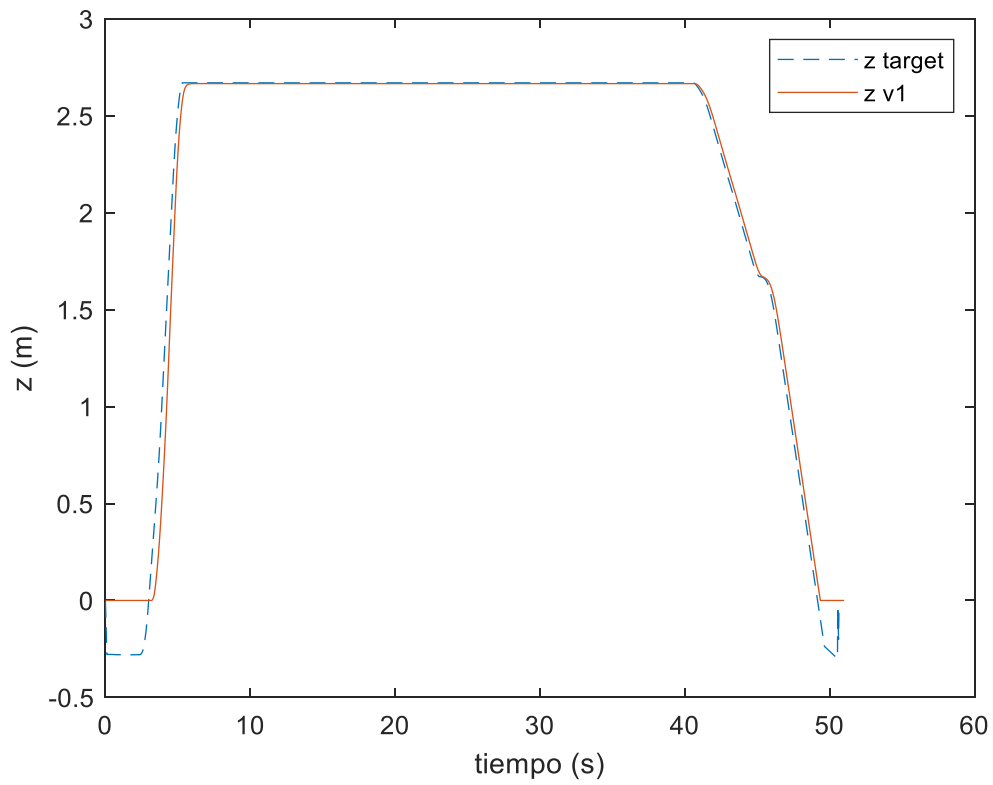


Ilustración 176- Coordenada Z.

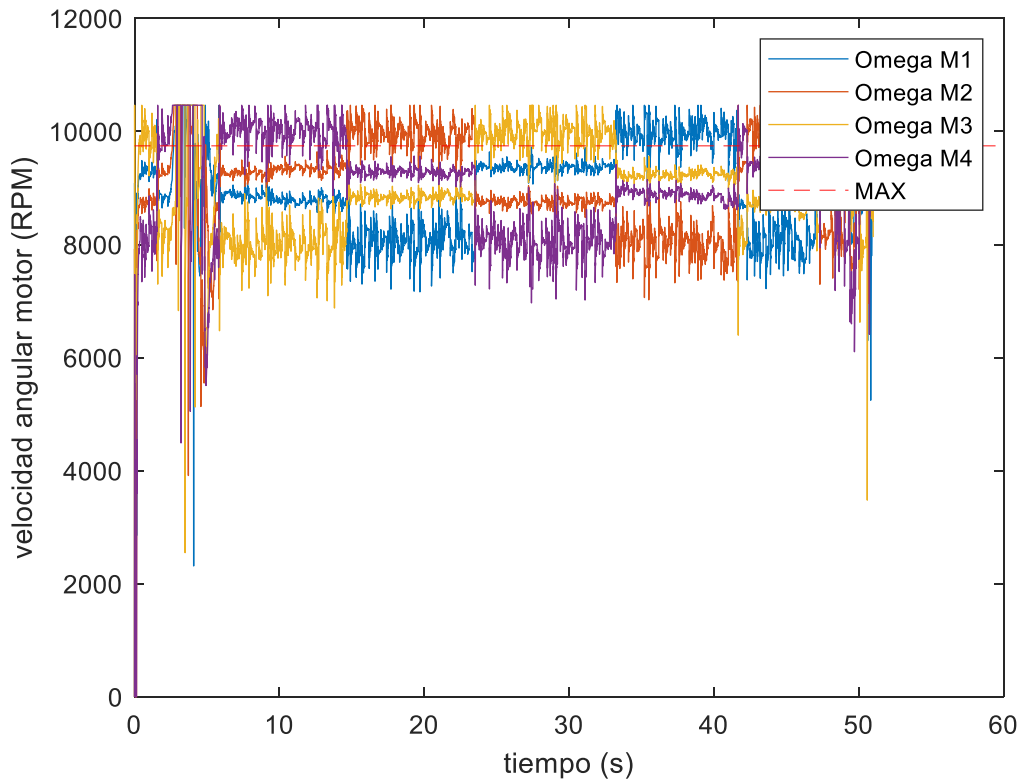


Ilustración 177- RPM de los motores.

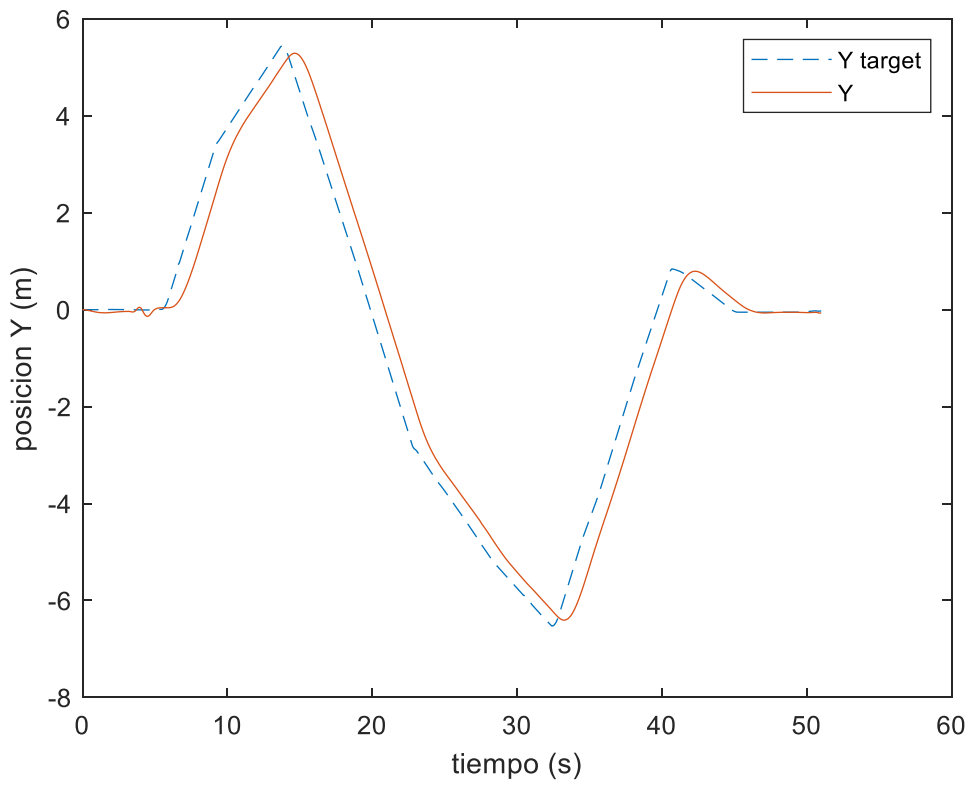


Ilustración 178- Coordenada Y.

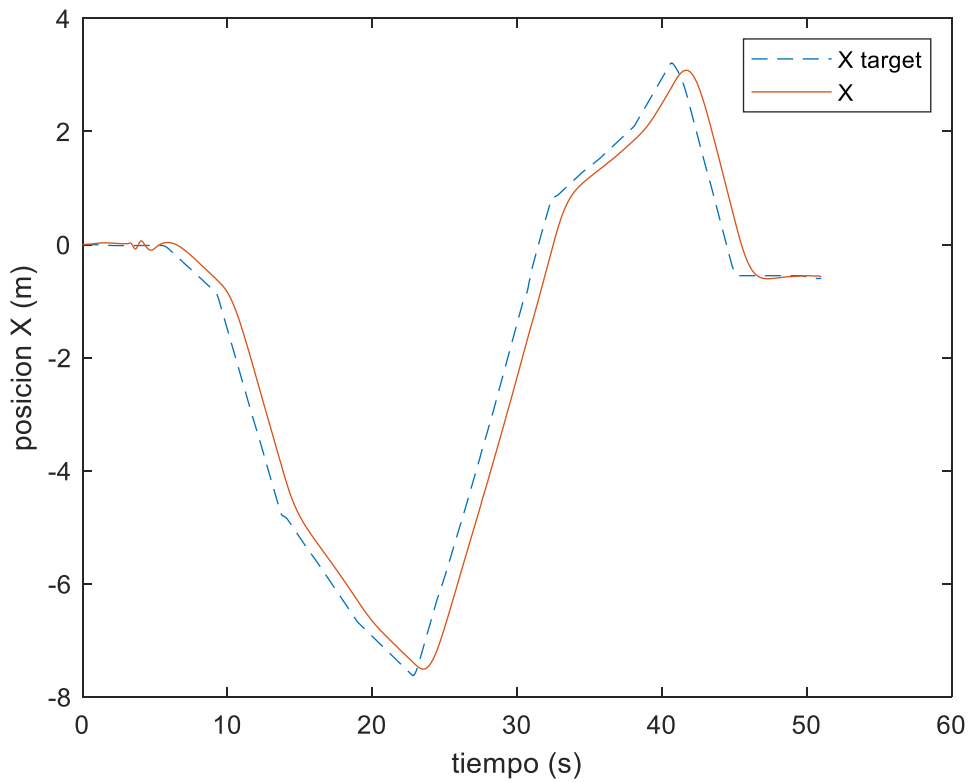


Ilustración 179- Coordenada X.

10.6.2 CARGA DE PAGO 2.4 KG

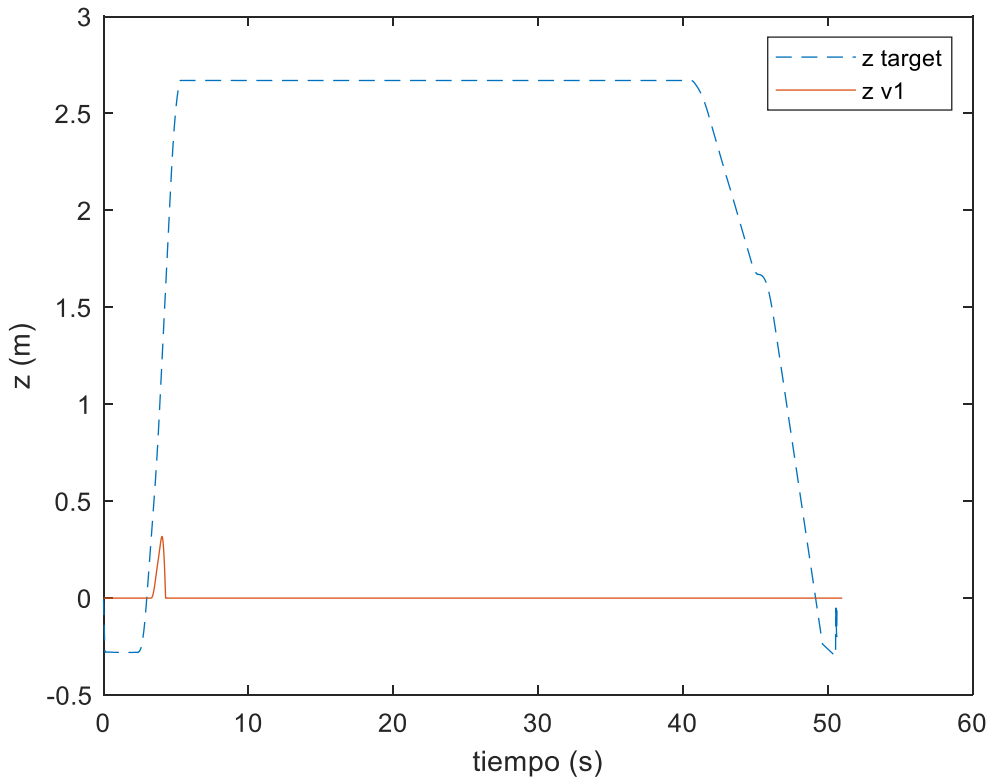


Ilustración 180- Coordenada Z.

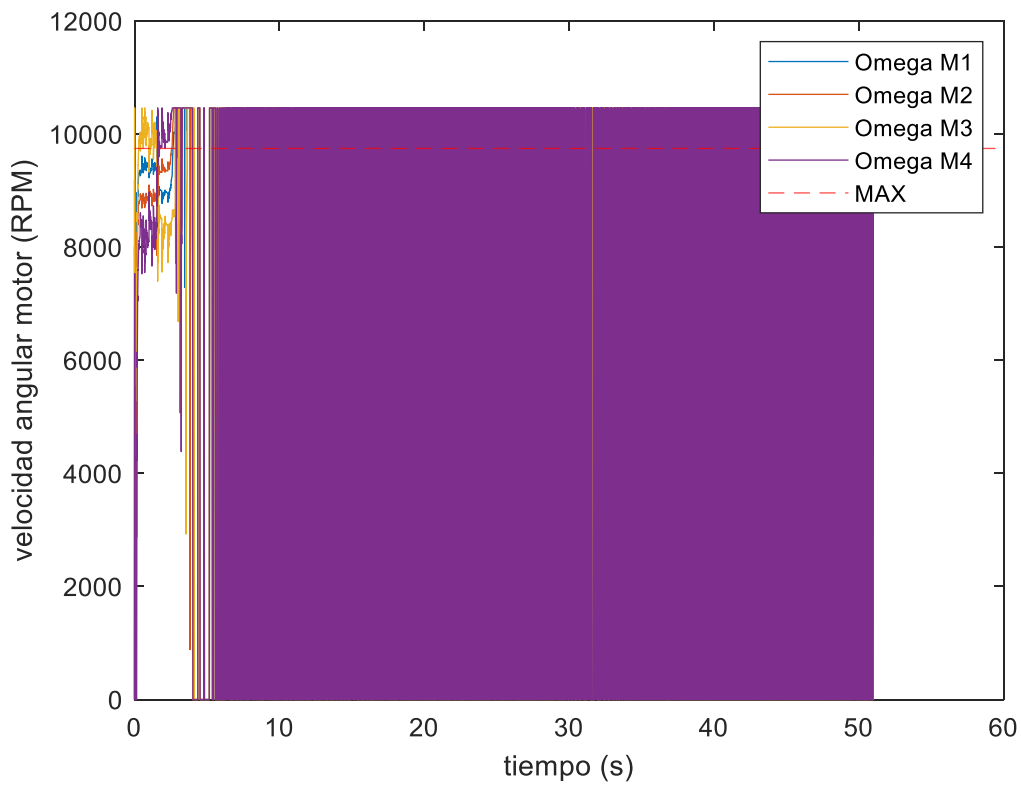


Ilustración 181- RPM de los motores.

10.7 RESUMEN DE LOS PARÁMETROS

Los parámetros presentados a continuación son importantes ya que a partir de estos parámetros se pueden definir las trayectorias que el sistema puede seguir y las condiciones en las que puede volar.

- Ángulos de inclinación máximos.

$$\theta_{MAX} = 0,6 \text{ rad} = 34^\circ$$

$$\phi_{MAX} = 0,6 \text{ rad} = 34^\circ$$

- Velocidad horizontal máxima.

$$V_{MAX}^{XY} = 12 \text{ m/s}$$

- Velocidad de ascenso máxima.

$$V_{MAX}^{UP} = 6 \text{ m/s}$$

- Velocidad de descenso máxima.

$$V_{MAX}^{DOWN} = 7,4 \text{ m/s}$$

- Aceleración horizontal máxima.

$$\ddot{X}_{MAX} = 6,8 \text{ m/s}^2$$

$$\ddot{Y}_{MAX} = 6,8 \text{ m/s}^2$$

- Aceleración de ascenso máxima.

$$A_{MAX}^{UP} = 5,6 \text{ m/s}^2$$

- Aceleración de descenso máxima.

$$A_{MAX}^{DOWN} = 4,3 \text{ m/s}^2$$

- Viento máximo con el cual es seguro operar.

$$V_{Viento}^{MAX} = 14 \text{ m/s}$$

- Carga de pago máxima.

$$m_{MAX}^{EXTRA} = 2,3 \text{ kg}$$

11 CONCLUSIONES

Se ha cumplido con el objetivo principal, desarrollar un modelo virtual que sea fiel al sistema real. Gracias a la elaboración y al ensayo del modelo mediante simulación se ha obtenido un amplio conocimiento sobre el funcionamiento del sistema y sus limitaciones. Además, este proyecto responde a una necesidad, que es la de realizar ensayos de casi cualquier tipo sin el inconveniente de estrellar el sistema real. La simulación es una herramienta muy valiosa a la hora de ensayar y hacer un primer diseño de sistemas de forma barata.

Otro de los beneficios de este proyecto es que permite ensayar distintos UAVs con distintas configuraciones simplemente ajustando los parámetros de la planta del modelo. Además, el sector de los sistemas aéreos autónomos está muy relacionado con la investigación y este modelo puede servir para evaluar la adecuación de algoritmos como pueden ser de fusión sensorial, navegación autónoma, etc.

Las conclusiones técnicas más importantes obtenidas tras la realización del proyecto son las siguientes:

- El aspecto más limitante de estos sistemas es la relación entre el peso y el empuje. Es por eso por lo que se debería trabajar de forma simultánea en el aligeramiento de estos vehículos y en la mejora de los sistemas de propulsión, los cuales cuestan poco dinero si se compara con la electrónica y la sensórica que suelen incorporar estos sistemas.
- Otra de las conclusiones más importantes y que hay que tener en cuenta es que el comportamiento de UAV depende de la trayectoria comandada al sistema de control. Para un mismo controlador, el dron puede tener errores de posición mayores o menores en función de la señal de entrada al controlador, por lo que si se desea un comportamiento muy preciso en posición se deberían comandar trayectorias con velocidades y aceleraciones bajas.
- El tipo de trayectorias que se puede comandar sin que el UAV se caiga dependerá de las propiedades físicas del vehículo y además del método de control elegido para el piloto automático. Es decir, depende del hardware y del software, por lo que aparte de trabajar en la relación peso- empuje, se debería seguir trabajando en distintas estrategias de control como pueden ser controladores no lineales o controladores predictivos según modelo.
- El error en posición se puede mejorar de forma muy sencilla y la cual no ha sido demasiado estudiada por lo visto en los artículos y referencias consultadas. Esta mejora se puede conseguir utilizando motores con sentido de giro reversibles, los cuales son soportados por el piloto automático ArduCopter en su versión actual.

11.1 LINEAS FUTURAS

Como se ha visto anteriormente en este mismo proyecto, el objetivo de desarrollar un modelo completo del sistema (Model in the Loop) es el de evaluar el rendimiento de distintas funcionalidades y algoritmos. Es por ello por lo que la forma más natural de seguir con este proyecto sería desarrollar ciertos algoritmos que interesaría implementar en el sistema.

El proceso que sigue cualquier sistema autónomo es el siguiente:

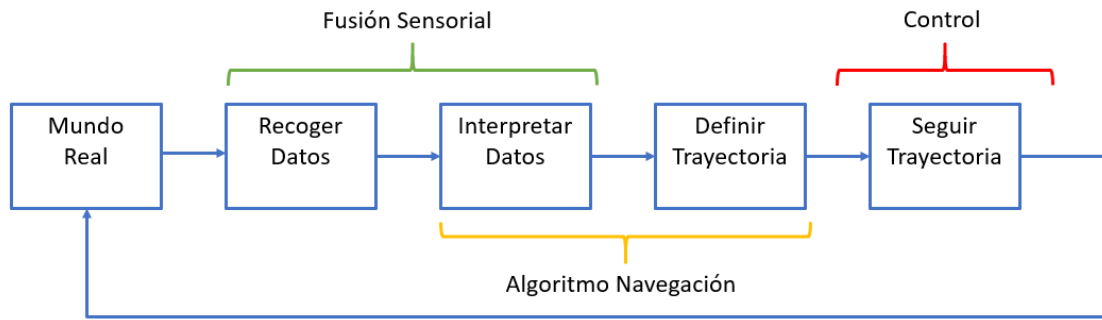


Ilustración 182- Proceso de un sistema autónomo.

En este proyecto se ha abarcado la parte del modelado y la parte del control del sistema. Pero para poder simular todo un sistema autónomo es necesario introducir dos partes fundamentales del proceso que son: la estrategia de fusión sensorial y el algoritmo de navegación.

- **Fusión sensorial:** Consiste en combinar los datos de distintos sensores para mejorar la estimación que se realiza sobre una determinada medida o para ser capaces de estimar medidas que sin esa combinación de sensores sería imposible estimar. Todos los sistemas autónomos disponen de un filtro, comúnmente un filtro de Kalman, que se utiliza para realizar la estimación de las variables que necesita el controlador. Otro tipo de aplicaciones comúnmente vistas en sistemas autónomos es la de tratar de fusionar las nubes de puntos que devuelven sensores como LiDAR, sonar, radar o cámaras.
- **Algoritmo de navegación:** Es el encargado de definir la trayectoria que debe seguir el sistema dadas unas coordenadas de interés. Además, este tipo de algoritmos sirven para dotar al sistema de un nivel de autonomía superior, ya que existe la posibilidad de que mientras el sistema sigue esa trayectoria se encuentre con un objeto inesperado y sea necesario redefinir dicha trayectoria.

12 BIBLIOGRAFÍA

- [1] ArduPilot, «GitHub,» 2021. [En línea]. Available: <https://github.com/ArduPilot/ardupilot>. [Último acceso: 2021].
- [2] Code world, «CodeTD,» 16 10 2020. [En línea]. Available: <https://www.codetd.com/es/article/11915300>. [Último acceso: 06 2021].
- [3] ArduPilot, 2021. [En línea]. Available: <https://ardupilot.org/dev/docs/learning-the-ardupilot-codebase.html>. [Último acceso: 2021].
- [4] M. Usman, «QuadcopterModelling and Control With MATLAB/Simulink Implementation,» 2020.
- [5] M. Gabriel Berrios, M. B. Tischler and T. Berger, "Turbulence Modeling of a Small Quadrotor UAS Using System Identification from," *ResearchGate*, p. 12, 2017.
- [6] Wikiwand, [En línea]. Available: https://www.wikiwand.com/en/Control_theory. [Último acceso: 2021].
- [7] T. Bresciani, «Modelling, Identification and Control of a Quadrotor Helicopter,» Department of Automatic Control, 2008.
- [8] S. Kapurch, NASA systems engineering handbook, 2010.
- [9] N. Ferry, «Quadcopter Plant Model and Control System Development With MATLAB/Simulink Implementation,» 2017.
- [10] J. D. P. Gene F. Franklyn, Feedback Control of Dynamic Systems, Pearson, 2018.
- [11] C. Tytler, «Autonomy in Motion,» Mayo 2020. [En línea]. Available: <https://charlestytler.com/modeling-vehicle-dynamics-euler-angles/>. [Último acceso: 2021].
- [12] A. Vergara, E. Salazar y O. J. Zapata, «Obtención de la función de transferencia de un motor de DC mediante el análisis de la curva de reacción,» *Revista de Aplicación Científica y Técnica*, vol. 3, nº 10, p. 10, 2017.
- [13] A. Romero Galán, «Revisión y Modificación del Firmware de Libre Acceso ArduCopter para su uso en el Proyecto AirWhale,» Universidad de Sevilla, 2015.
- [14] R. Luque Berraquero, «Simulación y control de un quadrotor,» Universidad de Sevilla, 2017.
- [15] R. Vega Astorga, «Simulation of a Quadrotor unmanned aerial vehicle,» Universidad Carlos III de Madrid, 2016.
- [16] MathWorks, «UAV Toolbox Support Package for PX4 User's Guide,» Matlab, 2020.

-
- [17] R. Felismina y M. Silva, «Study on the aerodynamic behavior of aUAV with an applied seeder for agricultural practices,» de *AIP Conference Proceedings*, 2017.
- [18] M. Berrios, T. Berger y M. Tischler, «Hover Flight Control Design for UAS Using Performance-based Disturbance Rejection Requirements,» de *AHS Annual Forum*, 2017.
- [19] L. De Frutos Bolzoni, «Desarrollo de algoritmos de control avanzado para dispositivos programables industriales,» Universidad de Valladolid, 2015.
- [20] O. A. García Campos, «Modelado, simulación y control de un quadrotor para transporte de carga colgante. Extensión al caso tridimensional,» Universidad de Sevilla, 2017.