

UNIVERSIDAD NACIONAL DE INGENIERIA
FACULTAD DE INGENIERIA MECANICA



**DESARROLLO DE UN ROBOT PROTOTIPO CON
TECNOLOGIA DE NAVEGACION AUTONOMA PARA
OPERACIÓN EN TERRENOS AGRICOLAS LLANOS**

INFORME DE TESIS

PARA OPTAR EL TITULO PROFESIONAL DE:

INGENIERO MECATRONICO

ELABORADO POR:
DIMEL ARTURO CONTRERAS MARTINEZ

ASESOR:
ING. RICARDO CASTAÑEDA DE LA ROSA

PROMOCION 2013- II

LIMA-PERÚ

2016

Dedicatoria

Dedico la presente tesis en primer lugar a Dios por iluminarme en las ideas y mantenerme constante. A mis padres y hermanos por su comprensión y ayuda. A mi abuelita Susana por sus consejos y dedicación en el hogar. A mis familiares por la confianza depositada en mí y a mis profesores de la UNI por brindarme los conocimientos necesarios para desarrollar la tesis.

Agradecimientos

Quiero agradecer a mi asesor el Ing. Ricardo Castañeda de la Rosa por su apoyo en el desarrollo de la presente tesis. También agradezco al Ing. Robinson Arévalo y la Srta. Jeymi por brindarme un espacio en el Laboratorio MT para el desarrollo de la Tesis. Gracias al INIFIM por financiar la implementación de la presente Tesis, al Ing. Manuel Luque Casanave por su gran asesoría y a mis compañeros estudiantes de la UNI Ryder Jhymsen y Gabriel García por el apoyo en el diseño y la implementación.

Contenido

Prólogo	1
Introducción	4
1.1. Formulación de la problemática	4
1.2. Objetivos	5
1.2.1. Objetivo principal.....	5
1.2.2. Objetivos Específicos.....	6
1.3. Estado del arte	6
1.4. Alcances y limitaciones.....	12
Marco teórico	13
2.1. Agricultura de precisión	13
2.1.1. Fumigación agrícola.....	13
2.2. Navegación autónoma	15
2.2.1. Sistema de control automático	16
2.2.2. Sistema de medición inercial (IMU)	18
2.2.3. Sensor Fusion	19
2.2.4. GNSS.....	20
2.3. SCADA	23
Modelamiento del problema.....	24
3.1. Modelo del problema	24
3.1.1. Estudio de campos agrícolas	27
3.1.2. Cultivo de palmeras.....	27
3.1.3. Cultivo de Olivos	30
3.2. Requerimientos.....	32
3.2.1. Capacidad de carga.....	32
3.2.2. Dimensiones del robot.....	33
3.2.3. Velocidad requerida	33
3.2.4. Autonomía energética	34
3.2.5. Alcance de las comunicaciones.....	35
Diseño del hardware.....	36
4.1. Sistema hidráulico para la fumigación	36
4.2. Sistema mecánico	40
4.2.1. Estructura	41

4.2.2.	Transmisión de movimiento	45
4.3.	Selección de actuadores	51
4.3.1.	Actuadores del sistema de navegación	51
4.3.2.	Actuadores del sistema de fumigación	52
4.4.	Sistema electrónico	52
4.4.1.	Selección de sensores navegación	53
4.4.2.	Microcontroladores	59
4.4.3.	Potencia	62
4.4.4.	Comunicaciones:	68
4.4.5.	Sensores para la adquisición de datos	69
4.4.6.	Sensores para el sistema de fumigación	71
4.5.	Sistema de energía.....	72
	Diseño de controladores y algoritmos	75
5.1.	Modelamiento del robot	75
5.2.	Algoritmo para estimación de la orientación	77
5.2.1.	Método de los Cosenos Directores (DCM)	78
5.3.	Controlador de navegación autónoma	83
5.3.1.	Control basado en puntos	83
5.3.2.	Controlador de orientación del robot.....	85
5.3.3.	Controlador de velocidad de los motores DC.....	87
5.3.4.	Algoritmos.....	101
5.4.	Controlador de tareas	103
5.4.1.	Controlador de presión de la fumigación	103
5.4.2.	Algoritmo para la fumigación	104
5.4.3.	Algoritmo para adquisición de datos.....	105
	SCADA	107
6.1.	Características del diseño	107
6.2.	Planificación de ruta.....	108
6.2.1.	Cálculo del área restringida.....	108
6.2.2.	Generación manual de la ruta	110
6.2.3.	Generación automática de la ruta – path planning	111
6.3.	Asignación de tareas.....	117
6.4.	Comandos remotos.....	119

6.5.	Monitoreo y supervisión.....	120
6.5.1.	Monitoreo en Tiempo Real.....	120
6.5.2.	Almacenamiento en base de datos.....	121
Implementación.....		124
7.1.	Implementación mecánica.....	124
7.2.	Implementación sistema hidráulico.....	128
7.3.	Implementación eléctrica – electrónica.....	132
7.3.1.	Controladores	132
7.3.2.	Driver de potencia de los motores.....	132
7.3.3.	Sensores del sistema de navegación.....	138
7.3.4.	Sistema de fumigación	139
7.3.5.	Sistema de adquisición de datos.....	141
7.3.6.	Comunicación del robot con el SCADA	143
7.4.	Costo de implementación.....	143
Pruebas y resultados.....		146
8.1.	Pruebas de comunicaciones del robot con el SCADA	146
8.2.	Prueba de movimientos del robot.....	146
8.2.1.	Prueba de control de velocidad del motor DC.....	147
8.3.	Prueba de control de orientación del robot.....	149
8.4.	Pruebas de navegación autónoma	151
8.4.1.	Pruebas del sistema de control y algoritmo	152
8.4.2.	Pruebas utilizando monitoreo remoto.....	154
8.5.	Prueba del módulo de pulverización	159
8.5.1.	Pruebas de la bomba DC	159
8.5.2.	Prueba de activación de válvulas y pulverización.....	160
8.6.	Prueba del módulo de adquisición de datos	161
8.6.1.	Muestra de los datos en tiempo real	161
8.6.2.	Consulta a la base de datos.....	161
Conclusiones		164
Recomendaciones.....		167
Bibliografía		169
Apéndice A.....		171
Planos del sistema mecánico		171

Apéndice B	173
Esquemas circuitos electrónicos.....	173
Apéndice C	176
Códigos de los controladores	176

Lista de figuras

Figura 1. 1: Trakur fumigando en una exhibición	7
Figura 1. 2: Robot Agribot de la empresa Jacto	8
Figura 1. 3 : Quadcopters y tractores operan de manera sincronizada	9
Figura 1. 4: Partes del prototipo del robot herbicida	10
Figura 1. 5: Robot Seekur	11
Figura 2. 1: Fumigación manual.....	14
Figura 2. 2 : Fumigación mecanizada.....	14
Figura 2. 3 : Fumigación utilizando avionetas	14
Figura 2. 4 : Cuidados durante la fumigación	15
Figura 2. 5 : Ruta asignada a un tractor autónomo agrícola	16
Figura 2. 6 : Lazo de control automático.....	17
Figura 2. 7 : Estructura de control en cascada	18
Figura 2. 8 : Composición de un IMU	18
Figura 2. 9 : Esquema de fusión de sensores	19
Figura 2. 10 : Procedimiento del método DCM para el IMU.....	20
Figura 2. 11 : Orbitas de los de las constelaciones de Satélites.....	21
Figura 2. 12 : Cálculo de la posición utilizando los Satélites.....	21
Figura 2. 13 : Funcionamiento del sistema RTK.....	22
Figura 2. 14 : Esquema del funcionamiento de un SCADA.....	23
Figura 3. 1 : Recorrido de las rutas del terreno agrícola por tractores	25
Figura 3. 2 : Mochila para fumigación	25
Figura 3. 3 : Flujo de procesos en el funcionamiento del Robot	26
Figura 3. 4 : Vista satelital del fundo de palmeras	28
Figura 3. 5 : Disposición de las palmeras en el terreno	28
Figura 3. 6 : Selección del sector de operación para las primeras pruebas del robot	29
Figura 3. 7 : Vista sobre tierra de las palmeras jóvenes	30
Figura 3. 8 : Vista satelital del fundo de olivos	30
Figura 3. 9 : Disposición de los árboles de olivo en el terreno.....	31
Figura 3. 10 : Selección del sector de operación para las pruebas del robot	32
Figura 3. 11 : Vista sobre tierra de los árboles de olivo	32
Figura 4. 1 : Esquema de elementos de la fumigación	36
Figura 4. 2 : División en 3 tramos de flujo del circuito hidráulico	38
Figura 4. 3 : Esquema del sistema de fumigación	40
Figura 4. 4 : Tubo cuadrado Acero.....	41
Figura 4. 5 : Estructura del robot.....	41
Figura 4. 6 : Propiedades físicas de la estructura del robot	42

Figura 4. 7 : Aplicación de la carga distribuida de 200 Kg.....	42
Figura 4. 8: Estructura mallada	43
Figura 4. 9 : Deformaciones por elementos finitos	43
Figura 4. 10 : Resultados del anáisis dinámico	44
Figura 4. 11 : Transmisión piñon – cadena	45
Figura 4. 12: Ubiación de los piñones en el diseño mecánico.....	46
Figura 4. 13: Diseño mecánico del Robot	47
Figura 4. 14: Modelo de bloque inclinado.....	48
Figura 4. 15 : Motor DC seleccionado	51
Figura 4. 16: Válvula solenoide.....	52
Figura 4. 17 : Diseño del sistema electrónico.....	53
Figura 4. 18: Componentes del kit RTK Piksi	54
Figura 4. 19: Características del RTK Piksi	54
Figura 4. 20: Conexión del kit RTK	55
Figura 4. 21: Diagrama de obtención de la posición utilizando el RTK	55
Figura 4. 22: IMU GY-85.....	56
Figura 4. 23 : Motor DC con eje directo al encoder	57
Figura 4. 24: Canales del encoder	57
Figura 4. 25: Funcionamiento del sensor ultrasónico.....	58
Figura 4. 26: Arduino Mega 2560 R3	60
Figura 4. 27: Conexiones del controlador C1	61
Figura 4. 28: Proceso del controlador C3	61
Figura 4. 29: Conexiones con el controlador de tareas.....	62
Figura 4. 30: Interfaces de potencia.....	63
Figura 4. 31 : Esquema de driver tipo puente H utilizando “full channel N Mosfet”	64
Figura 4. 32: Relés de activación de las válvulas solenoide.....	64
Figura 4. 33: Protecciones en un driver bidireccional Mosfet.....	67
Figura 4. 34: Módulo XBEE PRO S2	68
Figura 4. 35: Esquema de las comunicaciones	69
Figura 4. 36: Sensor de ultrasonido SRF05	71
Figura 4. 37: Transmisor de presión DANFOSS MBS 3000	71
Figura 4. 38: Conexión de las baterías	72
Figura 4. 39: Manejo del consumo instantáneo de energía	73
 Figura 5. 1 : Representación del robot y sus variables	75
Figura 5. 2 : Vector de corrección	81
Figura 5. 3 : Ángulos de Attitude del robot.....	83
Figura 5. 4 : Diagrama obtención del ángulo Yaw.....	83
Figura 5. 5: Navegación basada en puntos	84
Figura 5. 6 : Navegación basada en puntos	85
Figura 5. 7 : Diagrama del sistema de control de orientación	87

Figura 5. 8 : Motor DC y transmisión tornillo sínfín – corona.....	88
Figura 5.9: Modelamiento del motor con caja reductora.....	88
Figura 5. 10: Representación en bloques del modelo matemático del motor.....	88
Figura 5. 11: Señales de entrada y salida para la identificación del sistema.....	90
Figura 5. 12: Toolbox “ident” de Matlab	91
Figura 5. 13: Selección de la estructura de modelo no paramétrico	91
Figura 5. 14 : Resultados de la identificación	92
Figura 5. 15 : Respuesta de los modelos identificados ante un escalón unitario.....	93
Figura 5. 16 : Proceso de control en lazo cerrado	94
Figura 5. 17 : Diagrama de bloques.....	95
Figura 5. 18: Método de Routh Hurwitz para la estabilidad	96
Figura 5. 19: Esquema de control de velocidad del motor 2	98
Figura 5. 20: Configuración del bloque PIDz de Matlab para motor 2	98
Figura 5. 21 : Respuesta del controlador para referencia de 30RPM	99
Figura 5. 22: Respuesta del controlador para referencia de 30RPM	100
Figura 5. 23 : Configuración del bloque PIDz de Matlab para motor 4	100
Figura 5. 24 : Respuesta del controlador para referencia de 30RPM	101
Figura 5. 25 : Algoritmo para la navegación autónoma	102
Figura 5. 26 : Diagrama de flujo del controlador C1 y C3 para la navegación	103
Figura 5. 27 : Control de presión de la línea de presurizado	103
Figura 5. 28 : Lazo de control para la implementación del control de presión	104
Figura 5. 29 : Diagrama de bloques para la activación de las válvulas solenoides	104
Figura 5. 30 : Diagrama de flujo del proceso de fumigación	105
Figura 5. 31 : Proceso de adquisición de datos de los sensores	106
Figura 6. 1: Diagrama de diseño del sistema SCADA	108
Figura 6. 2 : Área restringida limitada por rectas	109
Figura 6. 3 : Ecuaciones de las rectas que limitan el área restringida	110
Figura 6. 4 : Diagrama de flujo para la formación de ruta	110
Figura 6. 5 : Ruta generada manualmente	111
Figura 6. 6 : Inicio y meta en un path planning	111
Figura 6. 7 : Caminos permitidos sobre el terreno, se muestran de color naranja.....	111
Figura 6. 8 : Mapa estructurado.....	112
Figura 6. 9 : Corners en el mapa.....	113
Figura 6. 10 : Búsqueda del corner más cercano al Pin.....	114
Figura 6. 11 : Ruta calculada	115
Figura 6. 12 : Gráfica del área restringida y corners en C#.....	116
Figura 6. 13 : Ruta calculada por el path planning en C#	117
Figura 6. 14 : Selección de la tarea.....	118
Figura 6. 15 : Asignación de la ruta deseada en modo autopath	118
Figura 6. 16 : Asignación de parámetros para la fumigación	118

Figura 6. 17 : Asignación de parámetros para la adquisición de datos	119
Figura 6. 18 : Asignación de parámetros para la adquisición de datos	119
Figura 6. 19 : Ventana para el control remoto del robot.....	120
Figura 6. 20 : Ventana para el monitoreo en tiempo real	121
Figura 6. 21 : Base de datos MSQl y tablas de las variables.....	122
Figura 6. 22 : Interfaz SCADA base de datos y gráficas.....	122
Figura 6. 23 : Interfaz SCADA Alarmas	123
Figura 6. 24 : Selección del puerto serial	123
Figura 7. 1: Ensamblado de la estructura del robot	124
Figura 7. 2 : Soporte vertical para fijar el motor	124
Figura 7. 3 : Ensamblado de las ruedas	125
Figura 7. 4: Transmisión de movimiento del motor a la rueda	125
Figura 7. 5 : Pruebas de transmisión de movimiento	126
Figura 7. 6: Porta baterías.....	126
Figura 7. 7 : Soporte para fijar el encoder al motor.....	126
Figura 7. 8 : Sistema mecánico del robot	127
Figura 7.9 : Plancha de aluminio como base del tanque de agroquímico	127
Figura 7. 10 : Barras de pulverización.....	128
Figura 7. 11 : Refuerzo triangular horizontal en las esquinas de la estructura.....	128
Figura 7. 12 : Motor DC y motobomba monofásica desarmada	129
Figura 7. 13 : Acoplamiento del motor DC con el propulsor	129
Figura 7. 14 : Motobomba conectada al tanque de agroquímico.....	129
Figura 7. 15 : Unión de componentes del circuito hidráulico	130
Figura 7. 16 : Boquilla sujetada a la barra de pulverización	130
Figura 7. 17 : Boquilla pulverizando el fluido.....	131
Figura 7. 18 : Módulo de pulverización en la parte posterior del Robot.....	131
Figura 7. 19 : Tarjeta electrónicas de los controladores y borneras de alimentación...	132
Figura 7. 20 : Driver de potencia implementado	133
Figura 7. 21 : Pruebas del driver de potencia	133
Figura 7. 22 : Conexión entre driver de potencia y motor.....	134
Figura 7. 23 : Pruebas en lazo abierto de los motores	134
Figura 7. 24 : Motor con encoder incorporado	135
Figura 7. 25 : Pruebas del control de velocidad en lazo cerrado del motor.....	135
Figura 7. 26 : Resultados del control en lazo cerrado para 2 motores en simultáneo ..	135
Figura 7. 27 : Prueba del circuito interconectado sobre la tabla de madera	136
Figura 7. 28 : Tablero eléctrico con todo el circuito incorporado	136
Figura 7. 29 : Parte frontal del tablero eléctrico	137
Figura 7. 30 : Prueba del control de los motores remotamente desde la laptop	137
Figura 7. 31 : Posición de la antena del GPS.....	138
Figura 7. 32 : Encoder acoplado al motor	139

Figura 7. 33 : Ubicación del IMU en el robot	139
Figura 7. 34 : Implementación del sistema de fumigación.....	140
Figura 7. 35 : Sensor ultrasónico para detección lateral de cultivos	141
Figura 7. 36 : Sensor Humedad del suelo, tipo resistivo	142
Figura 7. 37 : Integración de las conexiones de los sensores	142
Figura 7. 38 : Módulo de sensores.....	142
Figura 7. 39 : Disposición de los módulos XBEE PRO S2.....	143
Figura 8. 1: Pruebas de comunicaciones entre el robot y SCADA.....	146
Figura 8. 2 : Ventana del SCADA para pruebas.....	147
Figura 8. 3 : Pruebas de movimiento del robot en lazo abierto	147
Figura 8. 4 : Respuesta del controlador para referencia de 20RPM	148
Figura 8. 5 : Respuesta del controlador para una secuencia de pulsos a 20RPM.....	148
Figura 8. 6 : Pruebas de movimiento del robot en lazo cerrado	148
Figura 8. 7 : Pruebas de control de orientación con $Vl = 0$	149
Figura 8. 8 : Pruebas de control de orientación con $Vl=0.3m/s$	150
Figura 8. 9 : Control de la orientación para referencia 130° (Respecto al Norte)	150
Figura 8. 10 : Velocidades angulares (RPM) de cada motor calculadas por el control de orientación	150
Figura 8. 11: Estación base o fija, desde donde se envía tareas al robot.....	151
Figura 8. 12 : Visualización de la estación fija en el Mapa de GE.....	151
Figura 8. 13 : Coordenada de la estación fija	151
Figura 8. 14 : Orientación controlada para llegar al punto objetivo.....	153
Figura 8. 15 : Velocidades angulares de cada motor calculadas para la navegación ...	153
Figura 8. 16 : Error de posición durante la navegación.....	154
Figura 8. 17 : Asignando el punto objetivo desde la interfaz SCADA	154
Figura 8. 18 : Orientación y desplazamiento del robot hacia el punto objetivo	155
Figura 8. 19 : Evolución de las variables del control en la etapa de orientación	155
Figura 8. 20 : El robot llegando a su posición objetivo	156
Figura 8. 21 : Evolución de las variables del control cerca del objetivo	156
Figura 8. 22 : Trayectoria descrita por el Robot para llegar a su posición objetivo.....	157
Figura 8. 23 : Ruta asignada al Robot sobre la loza deportiva FIM	157
Figura 8. 24 : Desplazamiento del robot siguiendo la ruta	158
Figura 8. 25 : Ruta seguida por el robot	158
Figura 8. 26 : Pruebas de la motobomba DC.....	159
Figura 8. 27 : Determinación experimental del caudal que entrega la motobomba DC159	
Figura 8. 28 : Pruebas de la motobomba DC.....	160
Figura 8. 29 : Activación de las boquillas debido a sensor detección del ultrasónico .	160
Figura 8. 30 : Datos adquiridos en tiempo real	161
Figura 8. 31 : Consultas a la base de datos de los sensores agrícolas	161
Figura 8. 32 : Seleccionando los datos a graficar	162

Figura 8. 33 : Mediciones de radiación solar	162
Figura 8. 34 : Mediciones de humedad ambiental.....	162
Figura 8. 35 : Mediciones de temperatura ambiental	163

Lista de tablas

Tabla 3. 1 : Requerimientos para el robot en el fundo de palmeras	29
Tabla 3. 2 : Requerimientos para el robot en el fundo de olivos.....	31
Tabla 4. 1 : Requerimientos y consideraciones	37
Tabla 4. 2 : Constantes para el diseño	37
Tabla 4. 3 : Fórmulas utilizadas para el cálculo de la fumigación	39
Tabla 4. 4 : Resultados del diseño para el sistema de fumigación	39
Tabla 4. 5 : Modos, frecuencias naturales de la estructura del robot	44
Tabla 4. 6 : Características del piñón menor	45
Tabla 4. 7 : Características del piñón mayor	46
Tabla 4. 8 : Características de la cadena calculada.....	46
Tabla 4. 9 : Características de la cadena calculada.....	47
Tabla 4. 10 : Parámetros para el cálculo de torque necesario.....	48
Tabla 4. 11: Relacion de transmisión de movimiento	49
Tabla 4. 12 : Cálculo del torque necesario por motor	49
Tabla 4. 13 : Velocidad de desplazamiento del Robot con y sin carga	50
Tabla 4. 14 : Cálculo de la potencia requerida del motor.....	50
Tabla 4. 15 : Características del motor seleccionado	51
Tabla 4. 16 : Características de la válvula solenoide.....	52
Tabla 4. 17 : Funcionamiento del sensor ultrasónico	59
Tabla 4. 18 : Criterio para la elección de microcontroladores.....	60
Tabla 4. 19 : Actuadores y sus interfaces de potencia	65
Tabla 4. 20 : Características técnicas de los sensores de corriente y temperatura	67
Tabla 4. 21 : Características del módulo XBEE	68
Tabla 4. 22 : Características de los sensores del módulo	70
Tabla 4. 23 : Tipos de señales de los sensores para la fumigación.....	72
Tabla 4. 24 : Tipos de señales de los sensores para la fumigación.....	74
Tabla 5. 1: Ventajas y desventajas de los sensores del IMU.....	78
Tabla 5. 2 : Pasos del algoritmo de control PI discreto	86
Tabla 5. 3 : Parámetros del motor con caja reductora	89
Tabla 5. 4 : Modelos identificados de los motores	92
Tabla 5. 5 : Modelos de los motores seleccionados: derecho e izquierdo.....	93
Tabla 5. 6 : Pasos del algoritmo de control en tiempo discreto.....	98
Tabla 5. 7 : Variables medidas y sus características.....	106
Tabla 6. 1 : Parámetros para la asignación de tareas al Robot	117
Tabla 6. 2 Variables monitoreadas en Tiempo Real.....	121

Tabla 7. 1 : Controladores dentro del robot.....	132
Tabla 7. 2 : Lista de costo de la implementación	145

Prólogo

En la actualidad la automatización de procesos se ha desarrollado bastante, sin embargo en el rubro agrícola hay poco desarrollo de sistemas automatizados en comparación con áreas como la minería y fábricas. Existe una línea de automatización y tecnificación de procesos en agricultura llamado agricultura de precisión [1] (utiliza como una de sus herramientas la tecnología de la información y la robótica), en el cual se usa alta tecnología pero su costo es elevado para medianas y pequeñas empresas agrícolas.

En el desarrollo de la robótica de los últimos 10 años se ha dado cierto interés en dar soluciones a procesos repetitivos y de trabajo duro en la agricultura [2] , principalmente centros de investigación y empresas en países como EE.UU, Alemania, Australia y Brasil. Por otro lado en Argentina y Uruguay se ha adquirido bastante tecnología para su producción agrícola, principalmente para agricultura de precisión en grandes extensiones de terreno. Esto refleja que en Latinoamérica, como es muy común, usamos tecnología que los países desarrollados crean a excepción de Brasil que está incursionando en tecnología. En el caso de Perú no hay un gran índice de utilización de tecnología de agricultura de precisión y menos existe la iniciativa de crear tecnología en este ámbito. Es por ello que como uno de los primeros esfuerzos de iniciativa del desarrollo de tecnología robótica en la agricultura se realiza este trabajo de Tesis.

Las labores más comunes en agricultura son: la siembra, el cultivo, la cosecha, la fumigación, el análisis de suelos - plantas y el transporte de elementos (productos,

herramientas o equipos). Para las labores mencionadas existen actualmente tecnología de tractores desde medianos a grandes que poseen accesorios (aradoras, sembradoras, cosechadoras, barras fumigadoras, etc.) y sistemas autopilotos bastante sofisticados y caros. Es por ello que la mayoría de empresas pequeñas y medianas del Perú no conocen éstas tecnologías y otras la conocen pero saben que es muy cara y por ello no la utilizan. La presente Tesis se centra en el desarrollo de un prototipo de robot autopiloto para fumigación (ya que este es una tarea que pone en riesgo a las personas por los agroquímicos), adquisición de datos (muy importante para conocer el estado de los sembríos y tomar decisiones) y transporte de elementos en campos agrícolas.

En el presente trabajo se desarrolla un robot que posee 4 sistemas, los cuales son: sistema de navegación autónoma, adquisición de datos, SCADA y de ejecución de tareas sobre rutas (como fumigación, transporte de elementos y otros).

El desarrollo de la presente Tesis de divide en 8 Capítulos, los cuales son:

Capítulo 1: Se presenta la formulación de la problemática, los objetivos, el estado del arte del desarrollo de robots móviles autónomos para agricultura, y finalmente los alcances y limitaciones de la presente tesis.

Capítulo 2: Se presenta el marco teórico relacionado con agricultura de precisión, fumigación agrícola, navegación autónoma, sistemas de control automático, sistemas de navegación por satélite, fusión de sensores y finalmente el sistema SCADA.

Capítulo 3: Se desarrolla detalladamente el modelamiento del problema de la operación autónoma de los robots sobre unos campos agrícolas específicos y se obtiene los requerimientos para el diseño del Robot.

Capítulo 4: Se desarrolla el diseño del hardware del robot. Se presenta el diseño del sistema hidráulico para la fumigación, el sistema mecánico tanto en estructura como transmisión de movimiento, la selección de actuadores, el diseño de la electrónica de potencia, microcontroladores, sensores, el sistema de comunicaciones y el sistema energético.

Capítulo 5: Se desarrolla el diseño de los algoritmos de control y algoritmos lógicos para la navegación autónoma del Robot y para la ejecución de tareas como adquisición de datos y fumigación. Además para el diseño del control de navegación se realiza primero el modelamiento del Robot y la identificación de los motores DC luego se diseñan los controladores de orientación y velocidad de motores DC.

Capítulo 6: Se presenta el diseño del sistema de supervisión, control y adquisición de datos SCADA. Se desarrolla las funcionalidades de la Interfaz SCADA como son planificación de la ruta de navegación, asignación de tareas, comandos remotos, monitoreo en tiempo real y almacenamiento de la información en base de datos.

Capítulo 7: Se presenta el proceso de implementación del sistema mecánico, hidráulico, eléctrico y electrónico. Además se presenta el costo de la implementación.

Capítulo 8: Se presenta las pruebas y resultados del robot implementado. Se muestra las pruebas de comunicaciones entre el robot y SCADA, control de velocidad de los motores DC, control de orientación y control de la navegación autónoma. Además de las pruebas del sistema de fumigación y adquisición de datos.

Capítulo 1

Introducción

1.1. Formulación de la problemática

En la agricultura tradicional en el Perú no se utiliza significativamente la tecnología robótica. Los trabajos de movimiento de tierra se realizan con maquinaria pesada pero los trabajos rutinarios se ejecutan día a día por los mismos agricultores y sus trabajadores con herramientas y equipos básicos [3].

La tarea de fumigación se realiza por personas que en su mayoría no se protegen adecuadamente por lo que prácticamente llegan a estar en contacto de alguna manera con los agroquímicos utilizados. De ésta manera se puede afectar la salud de dicha persona. Ahora si consideramos que el terreno requiere una frecuencia de fumigación considerable y la extensión de los cultivos es grande, entonces hay mucha más probabilidad de que la persona que fumiga se enferme. Por otro lado las empresas agrícolas grandes poseen métodos para fumigar bastante agresivos, en el cual aplican una cantidad excesiva de agroquímicos con avionetas. La intención de usar estas avionetas para fumigar es hacerlo bastante rápido, pero no se toma en cuenta las consecuencias de la dispersión por el aire de los agroquímicos a los sembríos aledaños [4]. Es por ello que un robot que pueda realizar la tarea de fumigación aplicando la cantidad necesaria de agroquímicos y lo más cerca a los sembríos es mejor opción que los métodos tradicionales.

Además en el trabajo en campo agrícola se requiere muchas veces el apoyo en el transporte de elementos como herramientas, productos, etc. Esto conlleva a necesitar

de más mano de obra, es por ello que se propone que el **robot** pueda realizar el transporte de dichos elementos de manera autónoma entre diferentes posiciones del terreno, logrando así ahorro en costo de mano de obra, reducción de esfuerzo físico y optimización de tiempos de transporte de los elementos mencionados.

La ultima problemática analizada en la presente tesis es la falta de toma de datos de variables físico químicas (como humedad del suelo, humedad del ambiente, temperatura ambiental, radiación solar, pH, etc.) en el campo agrícola, ésto es muy común en pequeñas y medianas empresas agrícolas, es por ello que se limitan a no conocer en detalle el estado de su producción y menos aún poder proponer mejoras o soluciones a problemas existentes (como mejora de los riegos, fertilizaciones en ciertas regiones del terreno, mejor climatización, etc.). Con el robot que se propone se puede realizar toma de datos en diferentes posiciones y tiempos sobre el terreno agrícola, ésta información se guarda en una base de datos para ser analizado posteriormente y poder identificar los potencialidades y deficiencias a lo largo del terreno agrícola.

Por todo lo anterior mencionado, con éste trabajo de tesis se estaría dando los primeros pasos para el desarrollo de Tecnología robótica de automatización de algunos procesos de la agricultura en Perú.

1.2. Objetivos

1.2.1. Objetivo principal

Desarrollar un prototipo de robot autónomo para el proceso de fumigación, transporte de materiales y adquisición de datos en el terreno agrícola.

1.2.2. Objetivos Específicos

- a. Desarrollo del sistema de control para la navegación autónoma (autopiloto).
- b. Desarrollo de algoritmos para planificación de rutas del robot.
- c. Desarrollo del sistema SCADA del robot para visualización de las variables del robot, almacenamiento de datos de los sensores y asignación de rutas - tareas hacia el robot.
- d. Desarrollo del sistema electrónico que se compone de subsistemas embebidos (para el control y adquisición de datos), subsistema de comunicaciones, subsistema de potencia - energía y las protecciones tanto para el robot como para las personas que se encuentren en su entorno.
- e. Desarrollo de un módulo de pulverización para la fumigación.
- f. Desarrollo de la estructura mecánica del robot prototipo para realizar pruebas del diseño, algoritmos y control.

1.3. Estado del arte

A nivel mundial se han desarrollado 3 tipos de robots aplicados a la agricultura: Controlados por mandos, semiautónomos y totalmente autónomos (solo se le ingresa la tarea que debe cumplir). El caso de uno semiautónomo es el **robot Agrobot**, un robot desarrollado en España para cosechar fresas. Por otro lado los robots en agricultura más resaltantes, de los desarrollados en todo el mundo, que poseen la capacidad de desplazarse y realizar tareas de manera autónoma son:

I. Robots Desarrollados en América Latina:

a) Robot trakur

Es un robot desarrollado por el INTA (Instituto Nacional de Tecnología Agropecuaria) en Argentina [5]. Su principal función es la fumigación en invernaderos. Éste robot en su primera etapa era controlado por radiocontrol pero ahora en su última versión se desplaza autónomamente guiada por un cable (sensor invasivo), además utiliza unos imanes(otro sensor invasivo) en el terreno para saber en qué posiciones debe realizar la fumigación. La creación del robot, según sus creadores, fue debido a que en verano la alta temperatura dentro los invernaderos provocaba que muchos operarios no quieran usar los trajes que los protegen de los agroquímicos.



Figura 1. 1: Trakur fumigando en una exhibición

b) Agribot

Es un proyecto robótico desarrollado por la EESC – USP en Brasil del 2010 al 2012 [6]. El propósito de este proyecto fue desarrollar una plataforma robótica modular capaz de moverse en ambientes típicos de la zona agrícola con el fin de adquisición de datos y la investigación de nuevas tecnologías para la teledetección en

la agricultura. Sus principales características son robustez, movilidad, bajo peso, alta capacidad operativa y autonomía consistentes con las necesidades agrícolas.

Agribot posee un sistema de navegación autónoma, visión y comunicaciones. Se desarrolló en cooperación de la empresa de máquinas agrícolas Jacto. El desarrollo de éste robot le permitió a la empresa Jacto ver una línea de comercio de robots fumigadores, en la cual actualmente trabaja.



Figura 1. 2: Robot Agribot de la empresa Jacto

II. Robots Desarrollados en EE.UU y otros continentes:

c) RHEA Project

Es un proyecto iniciado por la Unión Europea que se basa en la investigación y desarrollo de un sistema robótico autónomo que incorpora tractores y drones para múltiples trabajos en agricultura. El proyecto RHEA (Robot Fleets for Highly Efficient Agriculture and Forestry Management) se inició el 2009 y sigue en desarrollo [7].

El grupo desarrollador reúne la experiencia y el know-how de 19 grupos de trabajos pertenecientes a 15 organizaciones de 8 países europeos con una gran experiencia en temas que abarcan conocimientos específicos en la robótica,

agronomía y fabricación de equipos agrícolas. Debido a su gran ambición tecnológica, este proyecto tiene un gran presupuesto para su desarrollo.

El proyecto incluye el desarrollo de robots aspersores de agroquímicos, eliminadores de hierbas malas y para adquisición de datos físico-químicos del campo, es por ello que éste es el proyecto que más se asemeja al proyecto que se va a desarrollar. Además también trabajan con una estación base que intercambia datos con los robots.

Una diferencia sustancial del proyecto RHEA con otros proyectos, es que integra el uso de drones para identificar las zonas específicas donde el robot tiene que eliminar las malezas y/o plagas y a la vez mapea la ruta para el robot tractor [8].



Figura 1. 3 : Quadcopters y tractores operan de manera sincronizada

El robot tractor se compone de los siguientes elementos:

- Sensores de guiado : GPS y cámara
- Sistema mecánico
- Sensores de percepción: 2 cámaras RGB
- Sistema de inyección de herbicidas
- Sistema de control : Computadora + Software avanzado

- Almacenadores: 2 tanques de agua y 1 tanque de herbicida.
- Boquillas Solenoide para la aspersión.

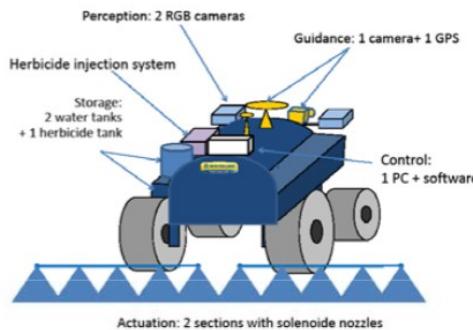


Figura 1. 4: Partes del prototipo del robot herbicida

d) ArduPilot - ArduRover

Es una plataforma de software y electrónica de bajo costo, muy utilizada en quadcopters pero también tiene la opción de ser utilizada para robots móviles del tipo Ackerman [9]. Sin embargo se limita mucho a sus modelos predeterminados (el robot en diseño es de tipo diferencial), y por ejemplo resultaría tedioso añadir la funcionalidad de evasión de obstáculos y realización de tareas como las que pretende la Tesis.

e) ClearpathRobotics

Es una empresa que provee una plataforma robótica completa que incluye la mecánica, electrónica y software [10]. Es bastante completa y avanzada la tecnología que usa para trabajos de autopilotos. El problema es que tiene demasiados recursos que vuelven caro a sus productos. Además la mecánica que ellos proporcionan es muy general como para el de un terreno agrícola que requiere un diseño mecánico particular.

f) Mobile Robotics

Empresa que se encuentra en EE.UU y que desarrolla sistemas robóticos móviles. Es el mismo caso que ClearpathRobotics, ya que son productos sofisticados y por ende caros, sin embargo su mecánica no es la más adecuada para terrenos agrícolas particulares. El robot móvil más avanzado que disponen es el Robot Seekur [11].



Figura 1. 5: Robot Seekur

Luego de este análisis de tecnología ya desarrollada (que ahora se ha vuelto algo comercial) y la que está en desarrollo, se llega a la conclusión que:

- No se ha desarrollado un robot de bajo costo que cumpla la función de fumigación, adquisición - almacenamiento de datos, y tareas de transporte de materiales en campos agrícolas (el más cercano es el FITOROBOT pero éste se diseñó para Invernaderos es decir para interiores y por ello no utiliza GPS sino SONAR, lo cual hace que no sea útil para terrenos a campo abierto).
- Las plataformas comerciales que nos brindan soporte tecnológico integral para la navegación autónoma no son las más óptimas, unas por ser limitadas y otras por ser de alto costo con recursos innecesarios y plataforma mecánica no adecuada para las tareas proyectadas.

- Los mayores avances en este tipo de robots se han realizado en Europa ya que es donde mejor practican la Agricultura. En América el más avanzado en desarrollo es Brasil.
- En Europa muchos países siguen investigando independiente y conjuntamente la tecnología robótica para agricultura, esto nos da una idea de que es un tema muy amplio y factible. Es por ello que en el Perú se debe imitar dichos pasos y comenzar con un robot que se asemeje y tenga mejores expectativas de diseño pero siempre cuidando el bajo coste para que pueda ser adquirido por los agricultores.

1.4. Alcances y limitaciones

- **Alcances**

En el presente trabajo de tesis se desarrolla un robot prototipo que se puede adaptar a múltiples tareas en el campo.

El sistema de navegación autónoma y SCADA desarrollado puede utilizarse en un robot de mayor escala u otras aplicaciones.

- **Limitaciones**

El sistema de navegación desarrollado funciona para ambientes abiertos (outdoor) debido a que se basa en la correcta medición del sensor GPS, por ello no es posible aplicarlo para ambientes cerrados.

Debido a que la implementación del robot en una escala similar a la de un tractor comercial, tanto en tamaño como potencia, resultaría demasiado caro se partió en el diseño de características mecánicas y eléctricas aceptables para un prototipo.

Capítulo 2

Marco teórico

2.1. Agricultura de precisión

El concepto sobre el que se basa la agricultura de precisión es aplicar la cantidad correcta de insumos, en el momento adecuado y en el lugar exacto [1]. La agricultura de precisión (AP) involucra el uso de sistemas de posicionamiento global (GPS) y de otros medios electrónicos para obtener datos del cultivo. La información recolectada puede ser empleada para evaluar con mayor precisión la densidad óptima de siembra, estimar la cantidad adecuada de fertilizantes o de otros insumos necesarios, y predecir con más exactitud el rendimiento y la producción de los cultivos. Las tecnologías de la agricultura de precisión permiten satisfacer una de las exigencias de la agricultura moderna: el manejo óptimo de grandes extensiones.

2.1.1. Fumigación agrícola

La fumigación consiste en esparcir un compuesto plaguicida(agroquímicos) de diferentes formas, para proteger un cultivo o un terreno de agentes nocivos para el mismo. Para fumigar se utiliza una bomba de fumigación y una manguera o en algunos casos barras pulverizadoras. El compuesto plaguicida puede ser de diversos tipos, según su destino, su acción específica, la forma de presentación, la composición química o el grado de peligrosidad [12].

Los métodos de fumigación son los siguientes:

- **Fumigación manual**

Es muy común en campos agrícolas pequeños y la persona se expone directamente al agroquímico.



Figura 2. 1: Fumigación manual

- **Fumigación con Tractores**

Se utiliza para campos agrícolas de mediana o gran extensión y que poseen caminos por donde se puede desplazar el tractor.



Figura 2. 2 : Fumigación mecanizada

- **Fumigación con Avionetas**

Se utiliza cuando la extensión del campo agrícola es bastante grande y no existen caminos por donde pueda pasar la maquinaria, por eso se realiza de manera aérea.



Figura 2. 3 : Fumigación utilizando avionetas

Existen diversos riesgos que puede ocasionar la mala ejecución de una fumigación [4], algunos de estos son:

- La falta de implementos de protección de la persona que ejecuta el proceso de fumigación.
- La emisión excesiva del plaguicida sobre el campo agrícola.
- La mala preparación del líquido a esparcir, causado por muy alta concentración de químicos o la utilización de químicos inapropiados.



Figura 2. 4 : Cuidados durante la fumigación

2.2. Navegación autónoma

La navegación autónoma es el desplazamiento de un robot sobre un ambiente determinado sin intervención de un piloto o control remoto [13]. Existen diversos tipos de navegación autónoma, sin embargo una forma de clasificarlos es la siguiente:

- Navegación autónoma en interiores
- Navegación autónoma en exteriores

En la presente tesis se estudia la navegación autónoma en exteriores ya que el ambiente de trabajo es el campo agrícola que se encuentra al aire libre (Outdoor),

Para que el robot posea navegación autónoma debe de poseer la instrumentación necesaria y un sistema de control automático. En la instrumentación se consideran principalmente los actuadores necesarios (motores) y los siguientes sensores:

- Sensor de geoposicionamiento
- Sensor de medición inercial
- Sensor de velocidad ruedas
- Sensor de detección de obstáculos

Por otro lado es imprescindible en la navegación autónoma un sistema de comunicaciones para que el usuario pueda asignar las rutas al robot y también monitorearlo.

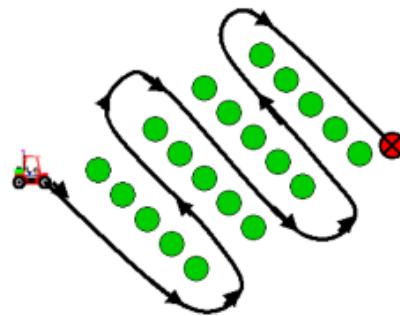


Figura 2. 5 : Ruta asignada a un tractor autónomo agrícola

2.2.1. Sistema de control automático

El control automático es el mantenimiento de un valor deseado para una cantidad o condición física, midiendo su valor actual, comparándolo con el valor referencia, y utilizando la diferencia para proceder a reducirla mediante una acción correctiva. En consecuencia, el control automático exige un lazo cerrado de acción y reacción que funcione sin intervención humana [14].

El elemento más importante de cualquier sistema de control automático es el lazo de control realimentado, que no es más que una trayectoria cerrada formada por un sensor, un controlador y un elemento final de control.

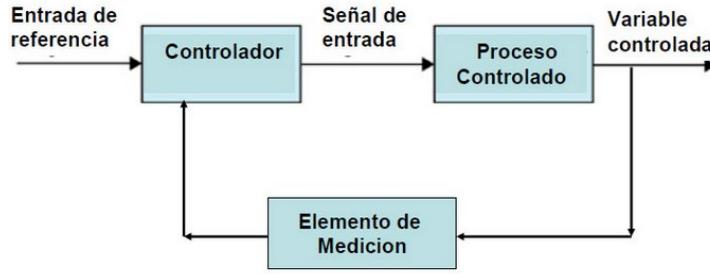


Figura 2. 6 : Lazo de control automático

En la navegación autónoma el sistema de control automático sigue continuamente las referencias necesarias para que el robot siga la ruta deseada.

La estrategia de control define la estructura o método que utiliza el sistema de control automático. Las estrategias de control dependen del proceso a controlar, las más utilizadas son las siguientes [15]:

- Control clásico y digital PID
- Control feedforward
- Controlador adaptativo
- Controlador optimo
- Control no lineal
- Controlador predictivo
- Control adaptativo
- Controlador basado en inteligencia artificial
- Controlador en cascada
- Controlador de relación
- Controlador de rango dividido

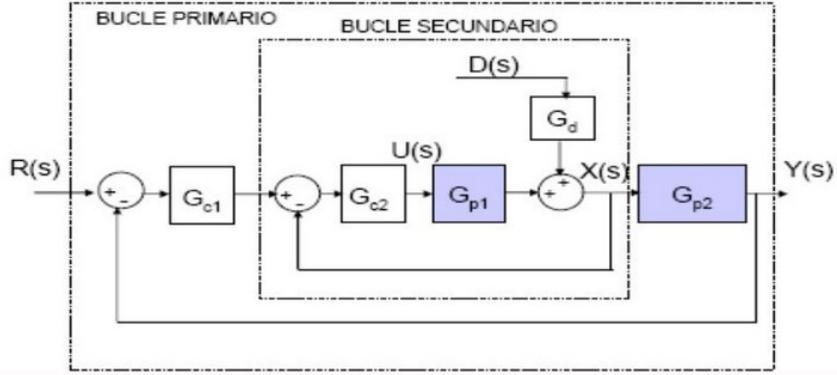


Figura 2. 7 : Estructura de control en cascada

2.2.2. Sistema de medición inercial (IMU)

Una unidad de medición inercial o IMU (inertial measurement unit), es un dispositivo electrónico que mide e informa acerca de la velocidad, orientación y fuerzas gravitacionales de un aparato, usando una combinación de acelerómetros y giróscopos [16] .

El término IMU se utiliza ampliamente para referirse a una caja que contiene tres acelerómetros y tres giroscopios dispuestos en forma ortogonal para cada eje, sin embargo en la actualidad la mayoría de los fabricantes también incluyen tres magnetómetros y 1 barómetro para mejorar las mediciones [16].

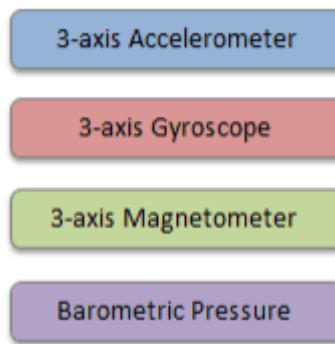


Figura 2. 8 : Composición de un IMU

2.2.3. Sensor Fusion

Sensor Fusion es un término amplio que se utiliza para cualquier proceso que involucre combinar información de múltiples sensores en una única medición con el fin de mejorar una medida [17]. Existen tres estrategias básicas para combinar sensores: redundante, complementario y coordinado.

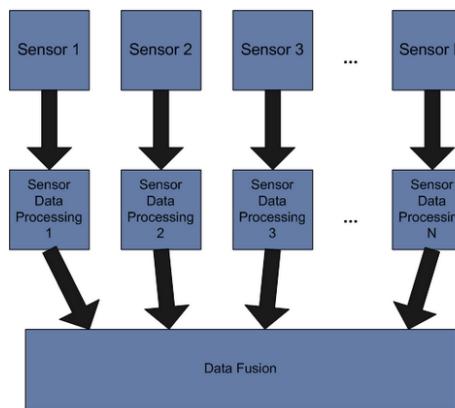


Figura 2. 9 : Esquema de fusión de sensores

En algunos casos se utilizan múltiples sensores cuando un sensor en particular es muy impreciso o ruidoso como para dar datos confiables. Al agregar un segundo sensor, se puede otorgar más confiabilidad. Cuando dos sensores retornan una misma percepción, éstos son considerados redundantes. Los sensores pueden tener además redundancia lógica, cuando sensan el mismo objetivo, pero se utilizan distintos algoritmos para procesar los datos sensados.

Métodos de fusión de sensores más utilizados son [18]:

- Filtro complementario
- Filtro de Kalman
- Método de la matriz de cosenos directores (DCM)
- Quaterniones

Método DCM:

Es un método que se basa en las ecuaciones cinemáticas de cuerpo rígido y en una matriz de rotación R que tiene que ser actualizada continuamente. Ésta diseñada principalmente para sistemas de medición inercial IMU y de navegación inercial INS [19]. El procedimiento que se sigue se muestra en el esquema (ver Figura 2.10):

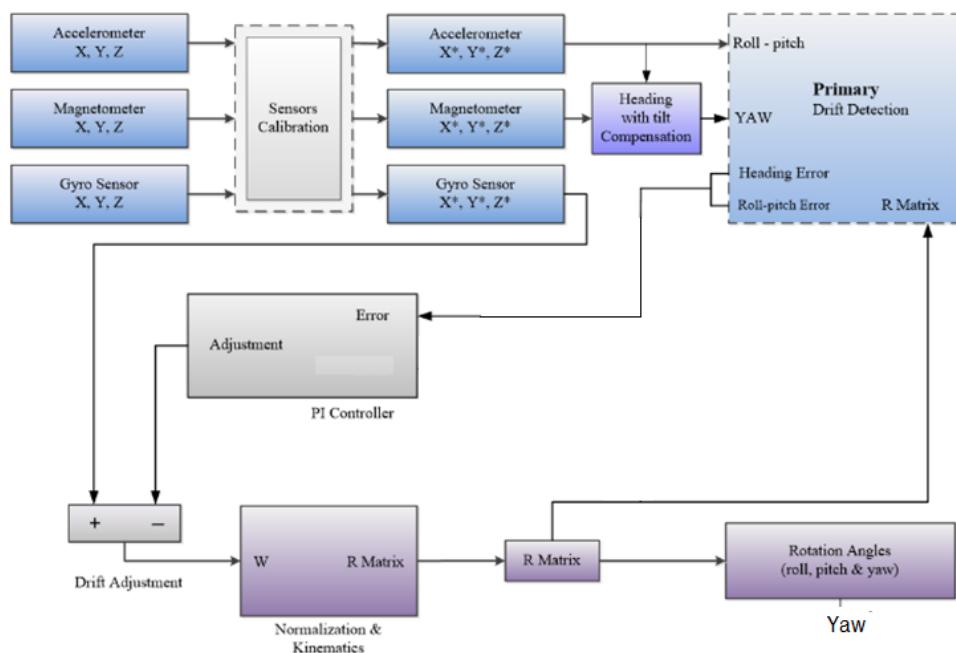


Figura 2. 10 : Procedimiento del método DCM para el IMU

2.2.4. GNSS

El GNSS (sistema global de navegación por satélite) es una constelación de satélites que transmiten señales al globo terrestre. Éstos permiten determinar las coordenadas geográficas y la altitud de un punto dado para fines de navegación, transporte, geodésicos, hidrográficos, agrícolas, y otras actividades afines [20].

Actualmente hay 4 GNSS principales los cuales son:

- GPS (Estados Unidos , 31 satélites en órbita)
- GLONASS(Rusia, 29 satélites en órbita)

- Galileo (Unión Europea , 14 satélites en órbita)
- BeiDou (China, 21 satélites en órbita)

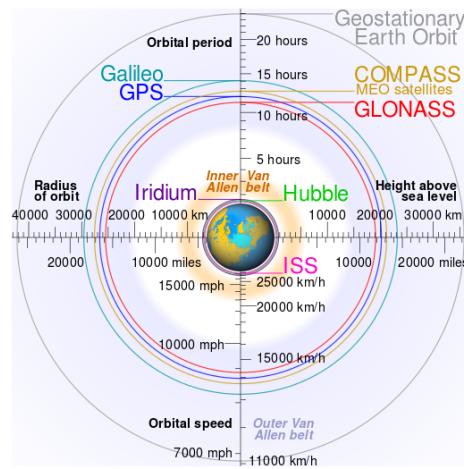


Figura 2. 11 : Orbitas de los de las constelaciones de Satélites

Para calcular la posición de un objeto sobre la superficie terrestre se necesita de por lo menos las señales de 4 satélites (ver Figura 2. 12), ya que 3 de ellos se utilizan para calcular la posición en 3 dimensiones y el cuarto permite eliminar los errores de sincronismo.

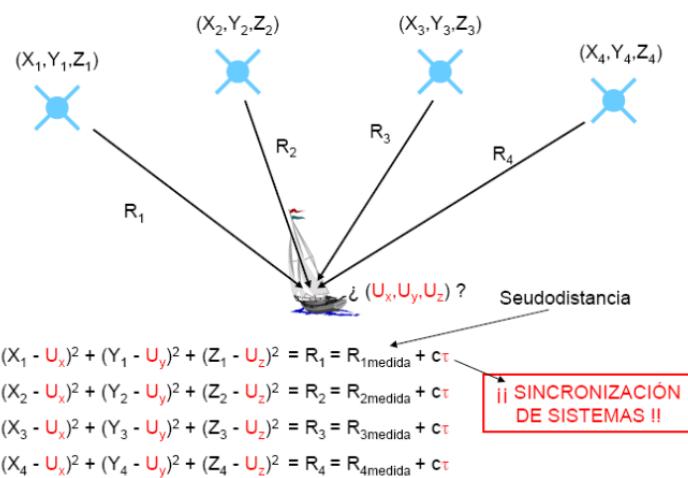


Figura 2. 12 : Cálculo de la posición utilizando los Satélites

La exactitud estándar de posicionamiento utilizando el GNSS civil actual es de 5m y si se mantiene estático puede llegar a una exactitud dentro de un círculo de 2m de radio. Sin embargo dicha exactitud no nos sirve para el proyecto en desarrollo, ya que los terrenos agrícolas no poseen caminos tan sobredimensionados.

Es por ello que realizando una investigación se encontró los diferentes métodos para mejorar la exactitud del sistema GNSS:

- a. GNSS diferencial
- b. GNSS RTK
- c. Métodos de sensor fusión

Siendo la que otorga mayor exactitud el método RTK.

Sistema RTK:

El sistema RTK(Real time Kinematics) se basa en la corrección de errores propios del módulo GNSS utilizando una estación fija con coordenadas fijas que compense los errores en tiempo real mientras el robot se mueve [21].

Para ello se hace uso de una comunicación inalámbrica adicional para el envío de las correcciones desde la estación fija o base hacia el robot.

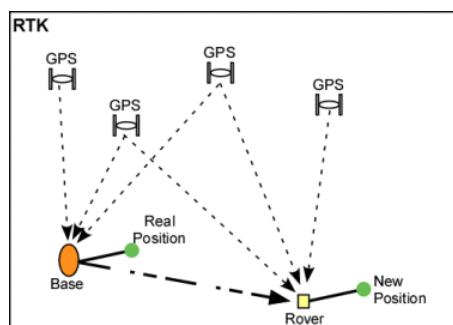


Figura 2. 13 : Funcionamiento del sistema RTK

2.3. SCADA

SCADA proviene de las siglas de Supervisory Control And Data Acquisition (Adquisición de datos y supervisión de control). Es un software de control y monitoreo de un proceso, que se comunica con los dispositivos de campo y controla variables desde la pantalla del ordenador. Proporciona información del proceso a diversos usuarios: operadores, supervisores de control de calidad, supervisión, mantenimiento, etc. [22].

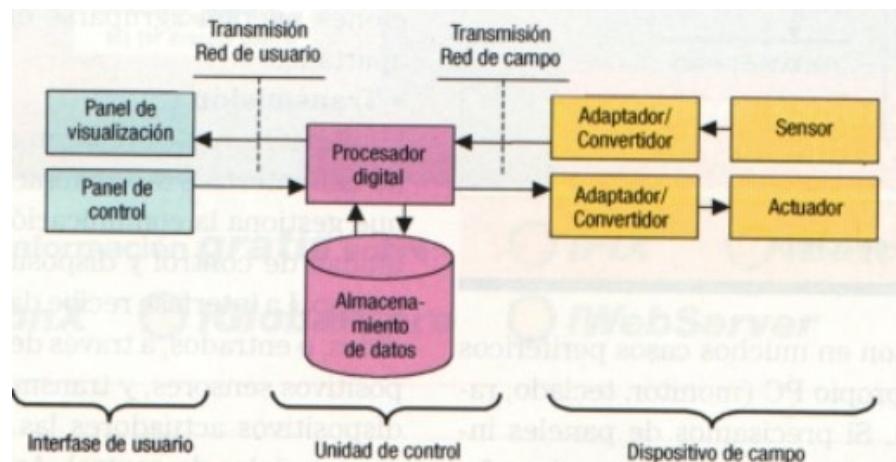


Figura 2. 14 : Esquema del funcionamiento de un SCADA

Capítulo 3

Modelamiento del problema

3.1. Modelo del problema

Para modelar el problema se ha tomado como punto de partida las tareas que se requiere que realice el robot. Las tareas propuestas para el robot son:

- Eliminación de plagas por fumigación, utilizando método de pulverización de agroquímicos.
- Sensado de variables y adquisición de datos del ambiente y suelo del terreno agrícola.
- Traslado de objetos relativamente pesados como apoyo a los agricultores en el movimiento de insumos, herramientas y productos.

A partir de las tareas propuestas se determina la necesidad de utilizar un robot móvil que pueda desplazarse en las rutas permitidas en el terreno agrícola, para ello se requiere que sea comandado por un operador mediante radio frecuencia, esta condición no representaría un proceso óptimo por el costo adicional de contar con un operador. La alternativa es que el robot sea autónomo en la navegación, para lo cual se ha decidido desarrollar un sistema autopiloto, de forma que se pueda asignar a priori las rutas por las que operará el robot. El funcionamiento sería similar a la forma de operar los tractores en el terreno agrícola (ver Figura 3. 1).

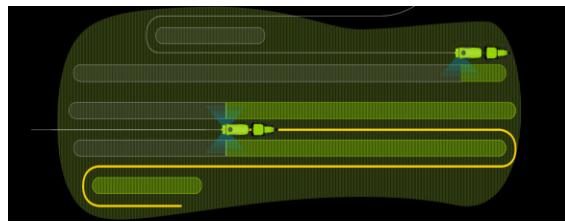


Figura 3. 1 : Recorrido de las rutas del terreno agrícola por tractores

El robot contará con sensores para la adquisición de datos y un módulo de pulverización para realizar la fumigación.

Se han considerado los siguientes sensores para la adquisición de datos :

- Humedad del suelo
- pH del suelo
- Temperatura del ambiente
- Humedad del ambiente
- Radiación solar

La operación del módulo de pulverización será similar al de las mochilas de pulverización (ver Figura 3.2) que comúnmente se utilizan en las faenas agrícolas. Sin embargo para el robot se ha considerado un sistema eléctrico que incorpora una bomba de pulverización, sensores y válvulas para la pulverización del agroquímico.



Figura 3. 2 : Mochila para fumigación

El módulo de pulverización comprende los siguientes elementos principales:

- Bomba de presurizado
- Tanque de agroquímico
- Electroválvulas
- Sensor de presión
- Boquillas de pulverización
- Regulador de presión

Se presenta el diagrama de procesos que debe seguir el robot junto con su sistema SCADA (ver Figura 3.3).

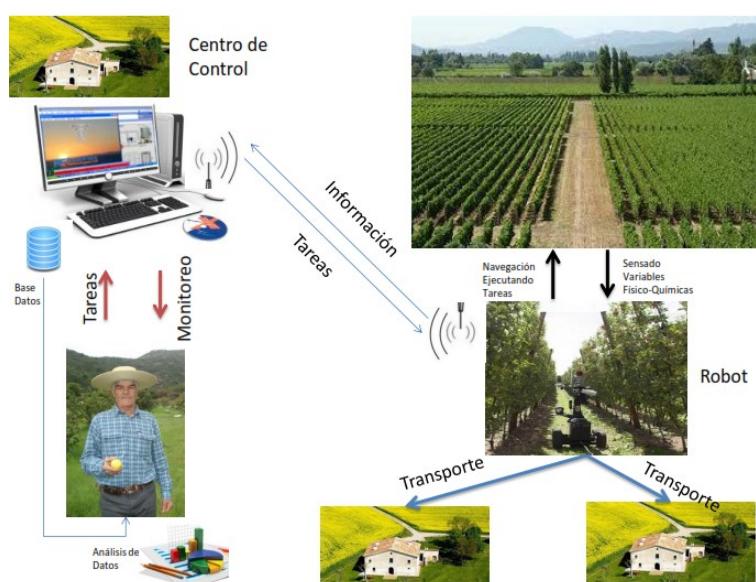


Figura 3. 3 : Flujo de procesos en el funcionamiento del Robot

El orden de los procesos es el siguiente:

- a. El operador enciende el robot y espera que éste reciba correctamente la señal del GPS para la navegación. Luego abre el programa SCADA en la computadora y comprueba la comunicación con el robot.

- b. El operador asigna la ruta de trabajo para el robot y además selecciona el tipo de tarea con sus parámetros correspondientes. Luego lo envía mediante el programa SCADA utilizando un emisor inalámbrico de comunicación.
- c. El robot recibe la tarea, por medio de su receptor, y la ejecuta de acuerdo a su tipo y recorriendo la ruta asignada. Asimismo envía continuamente datos sobre su estado (posición, estado energético, etc) y datos de los sensores agrícolas si fuera el caso.
- d. El programa SCADA recibe continuamente los datos enviados por el robot y los muestra en su interfaz visual, además almacena en una base datos algunas variables importantes y data de los sensores agrícolas.
- e. Finalmente el operador puede observar la interfaz del SCADA para saber el estado del robot y de su tarea asignada, así como también puede volver a asignar otra tarea por una ruta diferente.

3.1.1. Estudio de campos agrícolas

Para la aplicación de la operación del robot en el campo agrícola se ha considerado dos tipos de cultivos de tallo alto.

3.1.2. Cultivo de palmeras

Ubicado en un fundo en el distrito de Paracas, provincia de Ica, en las coordenadas -13.894086° N , -76.125432°S .El terreno posee una extensión aproximada de 213 m x 515 m. En la Figura 3.4 se muestra la vista satelital del mismo.



Figura 3. 4 : Vista satelital del fundo de palmeras

Los cultivos son de palmeras jóvenes con una altura promedio de 0.6m y máxima de 0.8m. La disposición a lo largo del terreno se muestra (ver Figura 3.5).

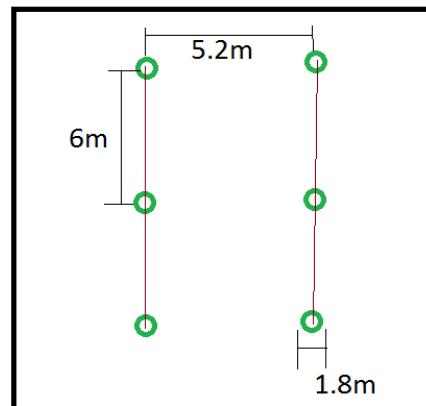


Figura 3. 5 : Disposición de las palmeras en el terreno

En la Tabla 3.1 se presentan los requerimientos considerados para el diseño del robot.

Tipo de terreno	Suelo duro y arenoso en algunas partes
Pendiente máxima	5°
Altura máxima de obstrucción en el suelo	10cm

Tareas de interés por realizar	Fumigación y monitoreo de las palmeras por adquisición de datos
Tarea complementaria	Transporte de cargas <=100 Kg
Terreno de operación	100mx100m
Agroquímico	TRIUNFO – FARMEX

Tabla 3. 1 : Requerimientos para el robot en el fundo de palmeras

Al ser muy extenso el terreno, se ha considerado un sector de 1ha aproximadamente (ver Figura 3.6).

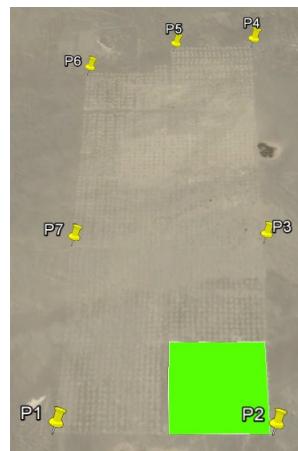


Figura 3. 6 : Selección del sector de operación para las primeras pruebas del robot

Se muestra el terreno seleccionado resaltado en color verde, al cual lo llamaremos **terreno de operación**, y será el sector donde se realizarán todas las pruebas del robot. Cabe resaltar que el terreno de operación posee *270 palmeras jóvenes* (18 surcos de 15 palmeras c/u) (ver Figura 3.7).



Figura 3. 7 : Vista sobre tierra de las palmeras jóvenes

3.1.3. Cultivo de Olivos

Ubicado en un fundo en el distrito de Chincha Baja , provincia de Ica, en las coordenadas -13.479422° N , -76.174920° S.

Este terreno posee una extensión aproximada de de 400 m x 400 m. En la Figura 3. 8 se muestra una vista satelital del terreno.



Figura 3. 8 : Vista satelital del fundo de olivos

Los cultivos son de palmeras jóvenes de una altura promedio de 2.4 m y máxima de 2.6m. La disposición a lo largo del terreno se aprecia en la Figura 3.9.

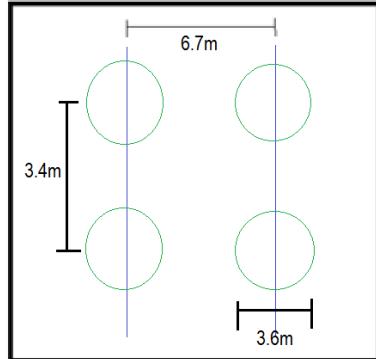


Figura 3. 9 : Disposición de los árboles de olivo en el terreno

En la Tabla 3.2 se presentan los requerimientos considerados para el diseño.

Tipo de terreno	Suelo arcilloso
Pendiente máxima	5°
Altura máxima de obstrucción en el suelo	8 cm
Tareas de interés por realizar	Fumigación y monitoreo de los olivos por adquisición de datos
Tarea complementaria	Transporte de cargas ≤ 100 Kg
Terreno de operación	100mx100m
Agroquímico	LANNATE LV – FARMEX

Tabla 3. 2 : Requerimientos para el robot en el fundo de olivos

Al ser muy extenso el terreno, se ha considerado un sector de 1ha aproximadamente en el que las operaciones de prueba del robot sean más ventajosas. Para ello se trabajará con el sector que posee más amplitud de camino para el robot (ver Figura 3.10).



Figura 3. 10 : Selección del sector de operación para las pruebas del robot

Se muestra el terreno seleccionado resaltado en color verde, al cual lo llamaremos ***terreno de operación***, y será el sector donde se realizarán todas las pruebas del robot. Cabe resaltar que el terreno de operación posee *540 árboles de olivo* (18 surcos de 30 palmeras c/u) (ver Figura 3.11).



Figura 3. 11 : Vista sobre tierra de los árboles de olivo

3.2. Requerimientos

3.2.1. Capacidad de carga

El robot está pensado para el transporte de cargas livianas o pulverización, pero en ningún caso ambos. El punto de partida en éste aspecto es considerar una capacidad de carga de 100Kg, la cual representaría lo siguiente: en el caso de transporte, la carga sería elementos como sacos de abono, herramientas, productos de

los sembríos, etc. Para la fumigación, la carga sería el tanque de agroquímico junto con su sistema de pulverización.

3.2.2. Dimensiones del robot

Las dimensiones del robot dependen del tipo de fundo y los sembríos que poseen. Para los fundos considerados, se tiene rutas permitidas con un ancho mínimo de 3m y además los sembríos (árboles o palmeras) tienen un alto máximo de 2,6m.

Es por ello que se considera las siguientes dimensiones para el robot:

- a. Ancho máximo del robot: 2m
- b. Largo máximo del robot : 1.5m (criterio largo debe ser menor que el ancho)
- c. Alto máximo del robot : 1m (por criterio de centro de gravedad)
- d. Dimensiones mínimas: Depende de las dimensiones del espacio para cargas y componentes internos del robot.
- e. Alto de barras de fumigación (desde el suelo) : 2,6m

3.2.3. Velocidad requerida

De acuerdo a referencias sobre el desarrollo de robots autónomos para el campo agrícola, se observó que la velocidad usual de desplazamiento es relativamente lenta.

Las principales razones son las siguientes:

- a. Debido a que la información de posicionamiento que es otorgada por el GPS, posee actualización lenta (entre 0.1 a 1 seg.).
- b. La controlabilidad del robot a bajas velocidades es mucho más fácil y segura.
- c. Las tareas sobre el campo agrícola (como adquisición de datos, fumigación) tiene que realizarse lentamente.

- d. El robot que es de dimensiones y peso considerables, por seguridad debe tener la capacidad de esquivar obstáculos y no provocar daños a objetos o personas que puedan entrar en su ruta.

La velocidad considerada para el robot es:

Velocidad máxima sin carga = 1m/s

Velocidad máxima con carga = 0.5m/s

3.2.4. Autonomía energética

Para analizar la autonomía se ha partido de las siguientes premisas:

- a. El robot debe poseer un diseño con actuadores netamente eléctricos.
- b. Según el estado del arte de robots autónomos eléctricos, la autonomía usual es de 1 a 2 horas con un paquete grande baterías.
- c. Se considera en el diseño, la disposición de baterías de reemplazo que siempre estarán cargadas (es decir que cuando el paquete de baterías se descargue, ésta se cambie y se deje cargando inmediatamente, de manera que se conserve la disponibilidad).
- d. Se ha considerado un tiempo de cargado de baterías de 4 horas.

De acuerdo a las premisas se ha considerado el siguiente periodo de autonomía energética:

Tiempo autonomía con carga: 1 hr.

Tiempo autonomía sin carga: 2 hr.

3.2.5. Alcance de las comunicaciones

El alcance que tendrán los módulos de comunicación (Transceivers) estará en función de la extensión de los fundos donde se va a operar.

La distancia máxima es la distancia diagonal del terreno, calculado por el teorema de Pitágoras con el ancho y largo.

Fundo 1 (213m x 515m)

$$dmáx = \sqrt{213^2 + 515^2} = 557 \text{ m} \quad (3.1)$$

Fundo 2 (400m x 400m)

$$dmáx = \sqrt{400^2 + 400^2} = 565 \text{ m} \quad (3.2)$$

Se aprecia que la distancia máxima no supera los 600m, sin embargo por problemas ambientales e interferencias en el campo, la potencia de los módulos de comunicación se reduce. Es por ello que necesitamos un módulo transceiver con alcance de 1Km como mínimo.

Capítulo 4

Diseño del hardware

4.1. Sistema hidráulico para la fumigación

Se diseñará el sistema de fumigación en base al campo de aplicación más crítico, es decir el campo que posee los cultivos más altos y que requieren mayor presión de bombeo. En éste caso se está considerando los árboles de olivo.

A continuación se muestra el esquema del sistema de fumigación propuesto, con todos sus elementos:

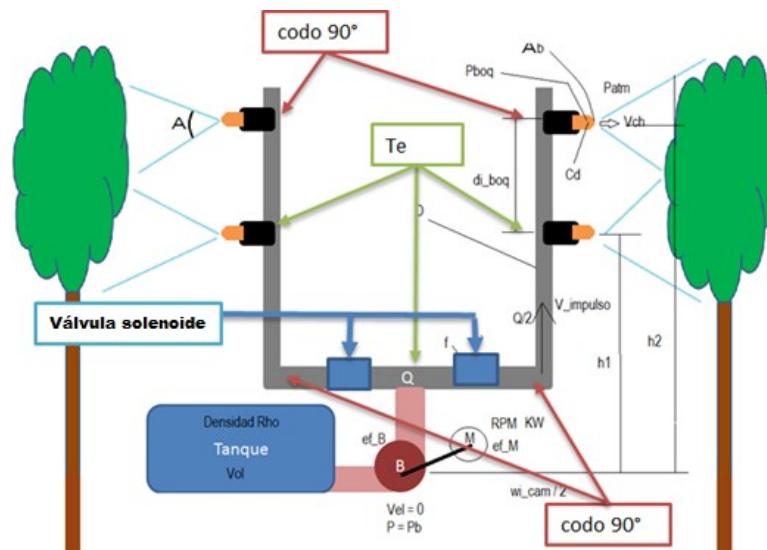


Figura 4. 1 : Esquema de elementos de la fumigación

El cálculo para el dimensionamiento consta en determinar la potencia y presión necesaria que debe de tener la motobomba y además el diámetro de la manguera para su selección.

El sistema de fumigación se diseñará de acuerdo a los siguientes requerimientos y consideraciones de partida:

Requerimiento			Consideraciones de partida		
Caudal requerido	q_{req}	10 L/min	Presión mínima de boquilla	$P_{boquilla}$	3 Bar
Ancho del camino	wi_{cam}	2 m	Velocidad de avance del robot	Ve_{ava}	4 Km/hr
Altura de la vegetación	he_{veg}	2.5 m	Distancia entre boquillas	$di_{boquillas}$	0.4 m
Ancho de la vegetación	wi_{veg}	2.5 m	Caída vertical del chorro	z_2	0.2 m
Total hectáreas	wi_{veg}	10 ha	Altura entre la bomba y boquilla más alta	h_2	1.5 m
Tiempo duración del tanque (continuo)	T_{tan}	20 min	Ancho del robot	wi_{robot}	1.5 m

Tabla 4. 1 : Requerimientos y consideraciones

Constantes:

Área de salida de la boquilla	ar_boq	0.000009 m^2
Número de boquillas	N_{boq}	4
Eficiencia del motor	ef_mot	0.7
Eficiencia del propulsor de la bomba	ef_bom	0.8
Densidad líquido agroquímico	ρ	1200 Kg/m^3
Coeficiente de descarga	Cd	0.8
Coeficiente de fricción	f	0.02

Tabla 4. 2 : Constantes para el diseño

- Cálculos**

Para los cálculos se divide en 3 tramos toda la vía del agroquímico, como se muestra a continuación:

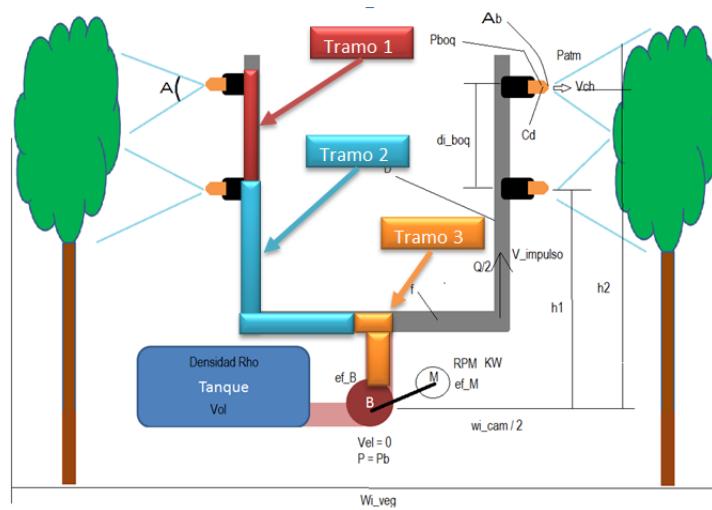


Figura 4. 2 : División en 3 tramos de flujo del circuito hidráulico

Las fórmulas utilizadas son:

Variable calculada	Fórmula
Caudal por boquilla (L/min) <i>[El estimado según requerimiento]</i>	$q_{boq} = \frac{q_{req}}{N_{boq}}$
Volumen necesario del tanque (L)	$V_{tanq} = q_{req} T_{tanq}$
Longitud de la barra de fumigación (m)	$Lo_{barra} = h_{veg} - h_{bomba}$
Volumen de agroquímico por esparcir (L/ha)	$V_{esp} = \frac{600 q_{req}}{w_{i_{veg}} v_{e_{ava}}}$
Velocidad de salida de chorro (m/s)	$v_{e_{cho}} = \frac{(w_{i_{veg}} - w_{robot})}{2} \sqrt{\frac{g}{2 z_2}}$
Diámetro de manguera (pulg)	$dia_{manguera} = \sqrt{ar_{boq} 10^6 \frac{v_{e_{cho}}}{vel_impul_{in}}}$

Caudal por boquilla (l/min)	$q_{cal_{boq}} = \frac{ar_{boq} ve_{cho}}{Cd}$
Caudal calculado (l/min)	$q_{cal} = q_{cal_{boq}} n_{boq}$
Pérdida de presión en el recorrido del flujo (m)	$H_{perdidas} = \frac{f Leq v^2}{2 dia_{mangue} g}$ $Leq = Lo_{flujo} + Lacc$ <p>Para 2 codos:</p> $Lacc = 2 Lo_{flujo} dia_{mangue}$ $Lo_{flujo} = h_2 + \frac{w_{robot}}{2}$
Presión de bombeo(Bar)	$P_{cal} = P_{atm} + P_{boquilla} + \rho g (h_2 + H_{perdidas} + \frac{ve_{cho}^2}{2 g})$
Potencia requerida de la bomba (KW)	<p>Potencia hidráulica ideal</p> $Pot_h = \frac{q_{cal} P_{cal}}{600}$ <p>Potencia Motor</p> $Pot_{motor} = \frac{Pot_h}{ef_{bom} ef_{mot}}$

Tabla 4. 3 : Fórmulas utilizadas para el cálculo de la fumigación

• Resultados

D	Diámetro Interno	0.3125	pulg
Pent	Presión de bombeo	10.5	bar
Q	Caudal	10	l/min
Pot_{motor}	Potencia de Bomba	175.7	W

Tabla 4. 4 : Resultados del diseño para el sistema de fumigación

Esquema propuesto:

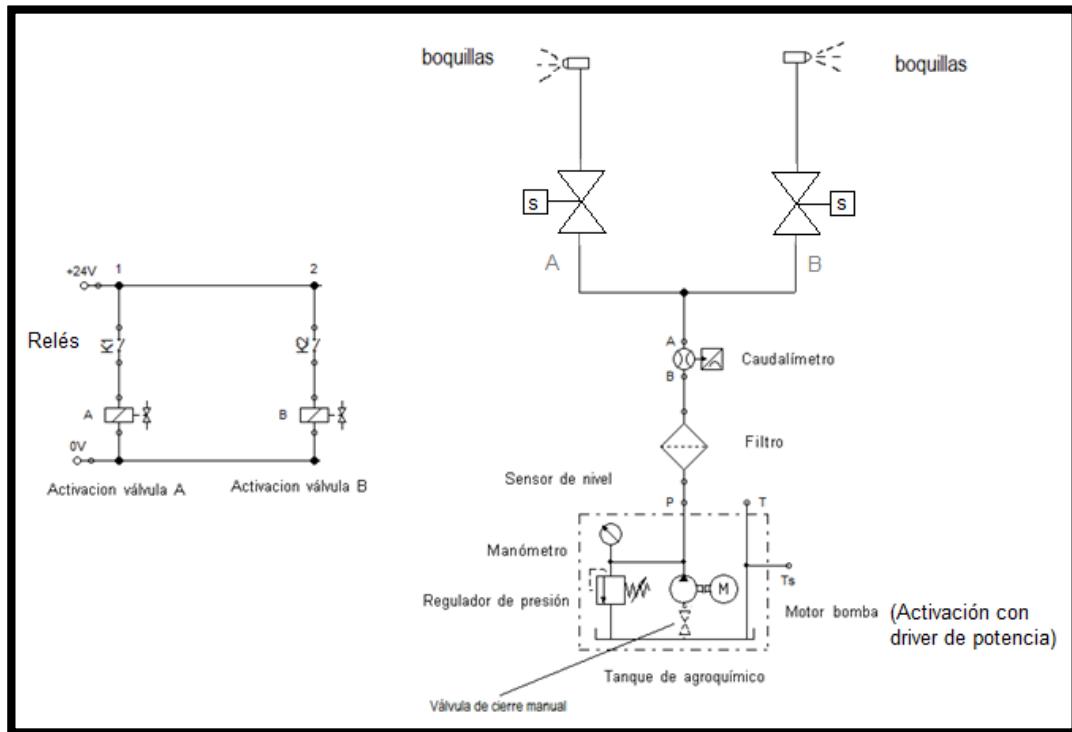


Figura 4. 3 : Esquema del sistema de fumigación

4.2. Sistema mecánico

La arquitectura del robot se seleccionó del tipo diferencial luego de comparar con otras arquitecturas existentes (omnidireccional, ackerman), debido a su facilidad de control y ensamblado. Por otro lado se utilizan ruedas comerciales de radio R = 17.5, éste dato sirve para el diseño del robot.

La carga útil que el robot debe transportar es de alrededor 100Kg, es por ello que la estructura debe ser capaz de soportar la carga de su propio peso más la carga útil (total aproximado = 280Kg).

El diseño mecánico se divide en la parte estructural y de transmisión de movimiento que se detallan a continuación:

4.2.1. Estructura

Se eligió como material base los tubos cuadrados de acero inoxidable de 5/4" y 2mm de espesor (ver Figura 4.4), además de un planchas de fierro de 5mm de espesor para los soportes inferiores de la estructura.



Figura 4. 4 : Tubo cuadrado Acero

Para el diseño de la estructura se utilizó el software SolidWorks 2014, considerando los criterios de simplicidad de fabricación, robustez y bajo costo:

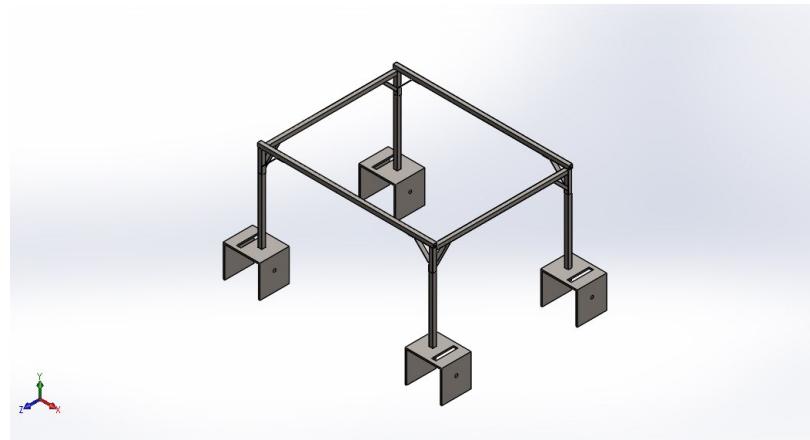


Figura 4. 5 : Estructura del robot

Las propiedades físicas de la estructura como son el peso, centro de gravedad y otros se muestran a continuación:

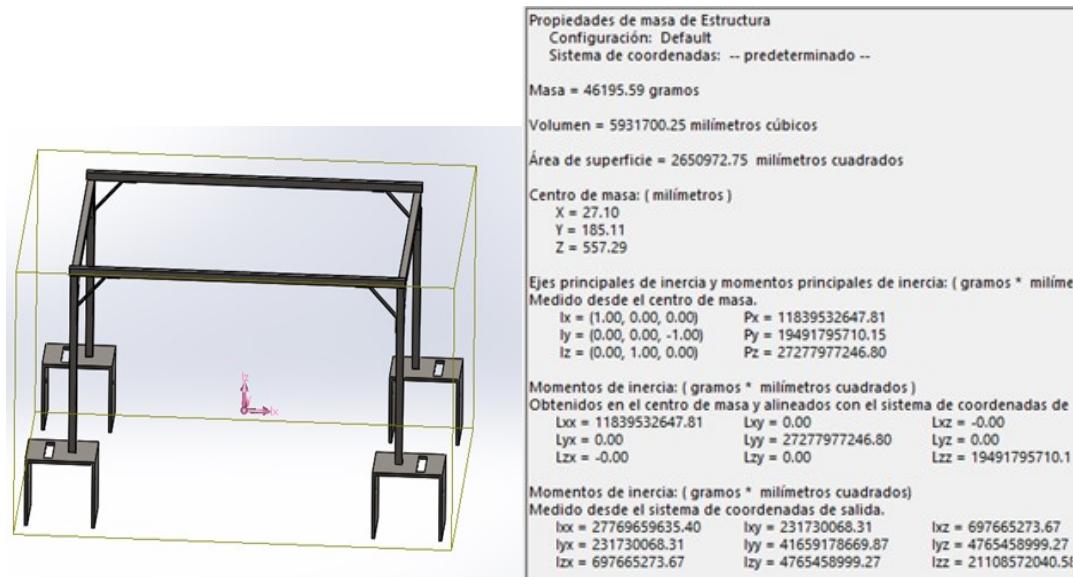


Figura 4. 6 : Propiedades físicas de la estructura del robot

Se realizó un análisis estático y de frecuencias con 280Kg de carga sobre la estructura, utilizando el modulo de simulación por elementos finitos de Solidwork.

Análisis estático :

- Aplicación de la carga:

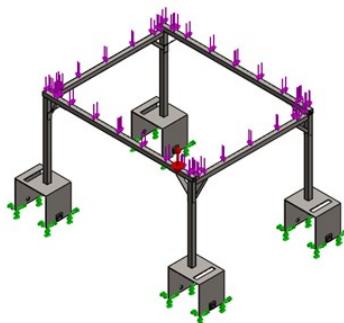


Figura 4. 7 : Aplicación de la carga distribuida de 280 Kg.

- Mallado:

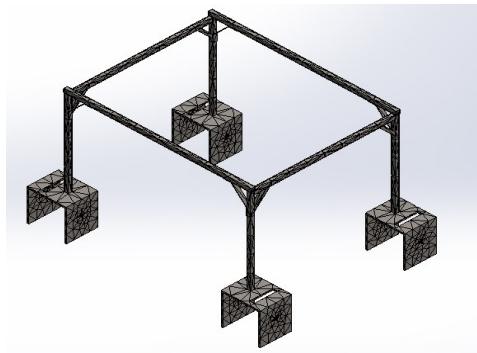


Figura 4. 8: Estructura mallada

Se obtuvo el siguiente resultado:

- Deformaciones :

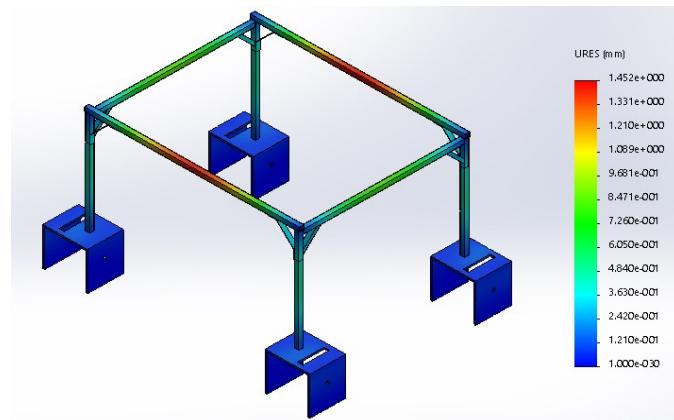


Figura 4. 9 : Deformaciones por elementos finitos

Se puede apreciar una deformación máxima en el tubo horizontal de magnitud 1.45mm, éste valor es aceptable.

Análisis de frecuencias

- Lista de modos:

Número de frecuencias	Rad/seg	Hertz	Segundos
1	260.35	41.436	0.024134
2	307.06	48.87	0.020462
3	359.56	57.225	0.017475

Número de frecuencias	Rad/seg	Hertz	Segundos
4	428.32	68.17	0.014669
5	581.18	92.497	0.010811

Tabla 4. 5 : Modos, frecuencias naturales de la estructura del robot

- Resultados:

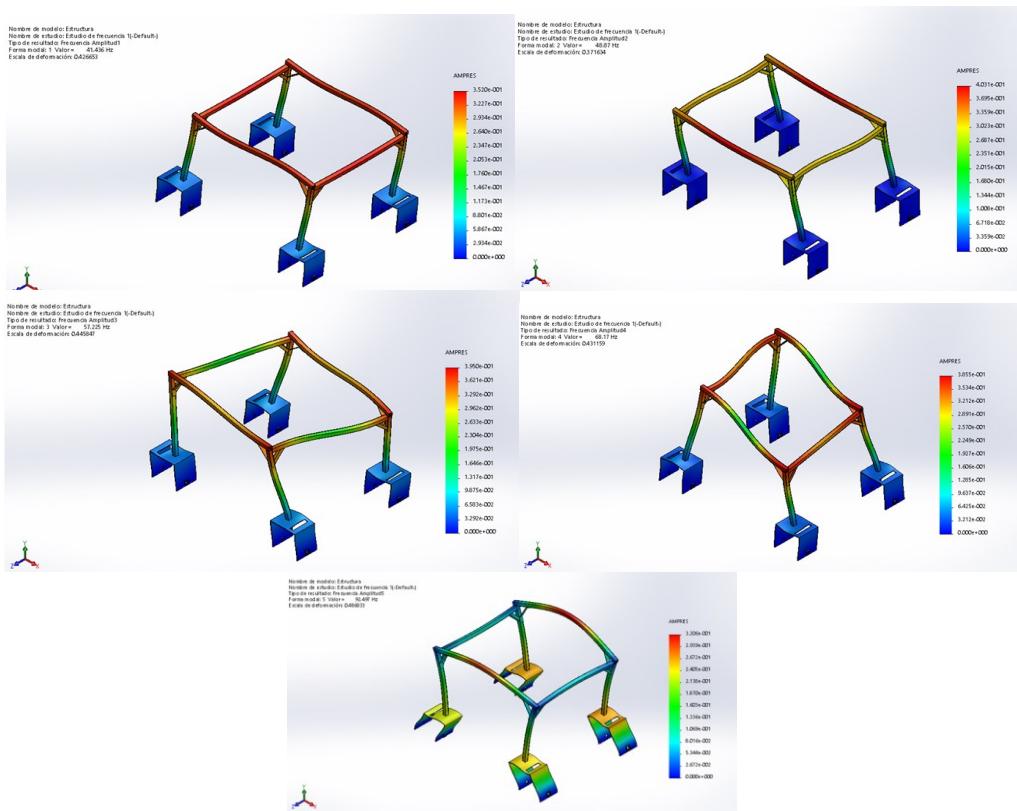


Figura 4. 10 : Resultados del análisis dinámico

Los valores de frecuencias naturales obtenidos permiten prevenir posibles fallos por resonancia cuando la vibración externa coincide en frecuencia con los modos. Un posterior análisis llevó a la conclusión de que debido a que el robot no se desplaza muy rápidamente sobre un terreno accidentado sobre el cual se podrían producir frecuencias de vibración iguales a las frecuencias naturales de la estructura, entonces se concluye que el diseño no fallará por resonancia.

4.2.2. Transmisión de movimiento

Para transmitir el movimiento se utiliza el mecanismo de piñón – cadena, elegido ya que puede transmitir el giro del motor al eje de la rueda estando alejado de ella. La idea de tener al motor separado del eje de la rueda es porque de dicha manera se protege al motor de posibles obstrucciones cerca del suelo y además de contaminación por charcos o polvo.

Por otro lado, es posible hacer cambios mecánicos de relación de transmisión cambiando los piñones del eje del motor como de la rueda, pudiendo lograr así mayor torque a costa de menor velocidad o viceversa.



Figura 4. 11 : Transmisión piñon – cadena

Los piñones calculados son:

Piñón	Símbolo	Valor	Unidad
Menor (base cálculo)			
Paso	p	9.65	mm
Diámetro primitivo	d1	33.8	mm
N. dientes	z1	11	
Peso	mg1	0.11	Kg
Diámetro eje	Deje1	14	mm

Tabla 4. 6 : Características del piñón menor

Mayor			
Paso	p	9.65	mm
Diámetro primitivo	d2	67.6	mm
N. dientes	z2	22	
Peso	mg2	0.43	Kg
Diámetro eje	Deje2	20	mm

Tabla 4. 7 : Características del piñón mayor

La cadena calculada es:

Cadena	Símbolo	Valor	Unidad
Tipo	Simple hilera	06B-1	
Paso	p	9.65	mm
Numero eslabones	L/p	171.9	
Longitud	L	1659.6	mm
Peso x metro	W/m	0.41	Kg/m
Peso total	W	0.68	Kg

Tabla 4. 8 : Características de la cadena calculada

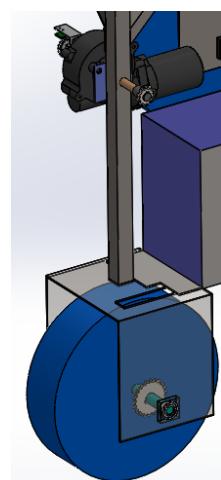


Figura 4. 12: Ubicación de los piñones en el diseño mecánico

A continuación se muestra el diseño mecánico del robot con sus componentes:

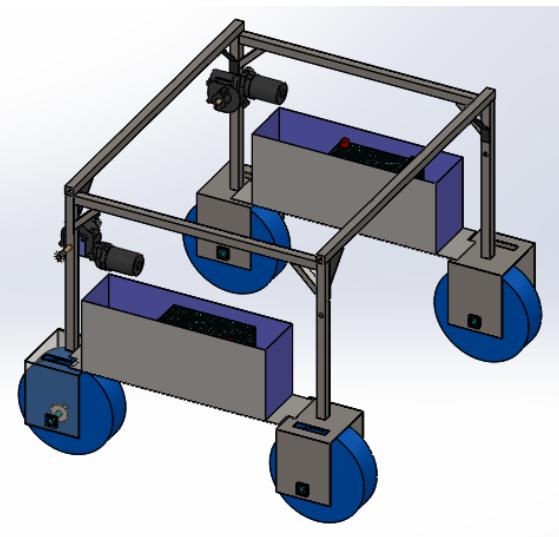


Figura 4. 13: Diseño mecánico del Robot

Cálculo de la masa total del robot:

Elemento	Masa unitario (kg)	Cantidad	Masa(kg)
Tablero y electrónica	20	1	20
Baterías	17	2	34
Motores ruedas	4	2	8
Ruedas	2	4	8
Estructura	50	1	50
Planchas	10	2	20
Piñones, cremalleras, acoples, otros	3	2	6
Tanque fumigación	100	1	100
Motor bomba	4	1	4
Barras fumigacion y mangueras	5	1	5
Otros	40	1	20
Total			275

Tabla 4. 9 : Características de la cadena calculada

Torque necesario:

Utilizando el modelo de bloque inclinado para simular el movimiento del robot sobre una pendiente:

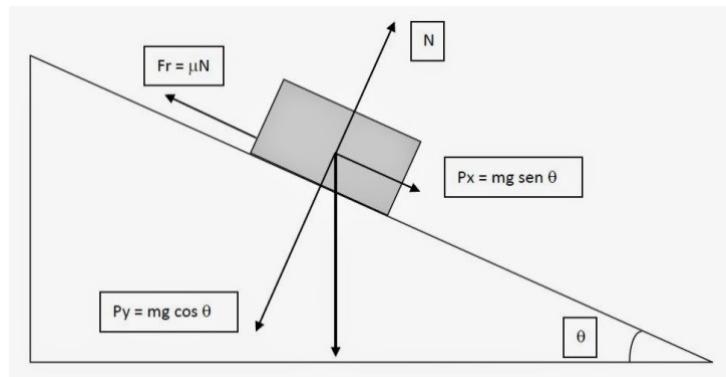


Figura 4. 14: Modelo de bloque inclinado

Considerando las constantes:

Parámetro	Notación	Valor	Unidad
Coeficiente fricción estática	μ_s	0.6	
Coeficiente fricción dinámica	μ_d	0.4	
gravedad	g	9.81	m/s ²
pendiente máxima	θ	10	grados
pendiente máxima	θ	0.174	rad
radio rueda	R	0.175	m
Masa total	M	275	kg

Tabla 4. 10 : Parámetros para el cálculo de torque necesario

La relación de transmisión de movimiento es:

En velocidades angulares	En torque
$\frac{n_{rueda}}{n_{motor}} = 0.5$	$\frac{T_{rueda}}{T_{motor}} = 2$

Tabla 4. 11: Relacion de transmisión de movimiento

Variable	Valor	Unidad	Fórmula
Fuerza normal total horizontal	2697.75	N	Mg
Fuerza normal total pendiente máxima	2656.76	N	$Mg \cos \theta$
Fuerza normal por rueda	664.19	N	$\frac{Mg}{4} \cos \theta$
Fuerza peso por inclinación	468.46	N	$Mg \sin \theta$
Fuerza peso por inclinación por rueda	117.11	N	$\frac{Mg \sin \theta}{4}$
Fuerza por fricción	265.67	N	$\frac{Mg}{4} \cos \theta \mu_d$
Fuerza total por vecer por cada rueda (Ft)	382.79	N	$\frac{Mg \sin \theta}{4} + \frac{Mg}{4} \cos \theta \mu_d$
Torque necesario por rueda	66.98	N.m	$F_t R$
Torque necesario por motor	33.49	N.m	$\frac{F_t R}{2}$

Tabla 4. 12 : Cálculo del torque necesario por motor

Velocidad de desplazamiento:

Utilizando el radio y relación de transmisión mencionados se analiza dos modos de operación del robot:

Modo	Masa aproximada (Kg)	Velocidad máxima desplazamiento (m/s)	Velocidad del motor (RPM)
Sin carga	175	1	108
Con carga	275	0.5	54

Tabla 4. 13 : Velocidad de desplazamiento del Robot con y sin carga

Potencia del motor:

Con los valores de torque y velocidad angular requeridos, se calcula la potencia que debe tener el motor. Además se consideran eficiencias mecánica y eléctrica.

Variable	Valor	Unidad	Fórmula
Eficiencia eléctrica	0.8		
Eficiencia mecánica	0.8		
Tensión de trabajo	24	V	
Velocidad motor con carga (w)	54	RPM	
Torque necesario con carga (T)	33.49	N.m	
Potencia ideal cada motor	191.39	Watts	$T \cdot w$
Potencia real cada motor	299.05	Watts	

Tabla 4. 14 : Cálculo de la potencia requerida del motor

4.3. Selección de actuadores

4.3.1. Actuadores del sistema de navegación

Se requieren 2 motores DC que cumplan los requisitos de potencia, torque y velocidad calculos en el diseño mecánico, se selecciona el siguiente modelo de motor:

Característica	Valor	Unidad
Voltaje nominal	24	V
Potencia nominal	350	Watts
Velocidad máxima	200	RPM
Corriente en vacío	4	A
Corriente nominal	15	A
Corriente pico	50	A

Tabla 4. 15 : Características del motor seleccionado

Cabe comentar que el motor DC seleccionado proviene de la dirección de los autos de marca Mitsubishi y es de bajo costo respecto a motores importados de la misma característica.



Figura 4. 15 : Motor DC seleccionado

4.3.2. Actuadores del sistema de fumigación

Se utilizan 2 válvulas solenoides para poder fumigar por ambos lados del robot.

Las características son:

Característica	Valor
Voltaje activación	24V
Corriente bobina	1A
Estado	Normalmente cerrada
Presión máxima	20 Bar
Diámetro	3/8"

Tabla 4. 16 : Características de la válvula solenoide

El material de la válvula es de bronce lo cual lo hace resistente a la oxidación.



Figura 4. 16: Válvula solenoide

4.4. Sistema electrónico

El sistema electrónico diseñado se presenta en el siguiente esquema:

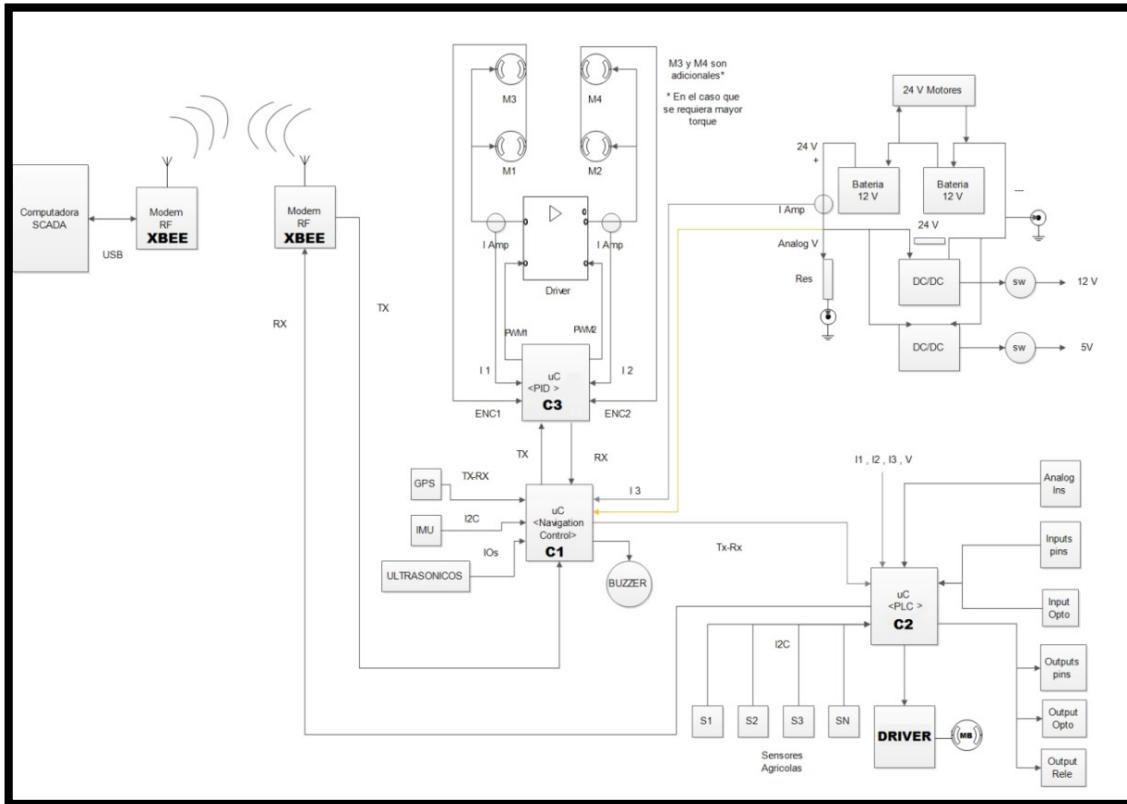


Figura 4. 17 : Diseño del sistema electrónico

4.4.1. Selección de sensores navegación

Para la navegación autónoma el sistema de control del robot requiere información sobre sus variables de posición, velocidad y orientación; para ello se seleccionó los siguientes sensores:

Sensor de posicionamiento global

Comúnmente se nombra a los sistemas GNSS como simplemente GPS, sin embargo como se mencionó en el marco teórico el GNSS es el conjunto de satélites en diferentes órbitas. Por practicidad en el transcurso de la presente tesis se nombrará como GPS al sistema de navegación por satélite GNSS.

Para lograr la mayor exactitud en el movimiento del robot se eligió como GPS al sistema RTK de la empresa Estadounidense Swift Navigation Inc.

- **RTK Piksi**

El RTK Piksi es un sistema RTK de bajo costo en comparación a otros en el mercado, diseñado para desarrolladores de robots, drones, etc. El kit se compone de 2 GPS y 1 par de Transceivers 3DR 433MHz (ver Figura 4. 18).



Figura 4. 18: Componentes del kit RTK Piksi

Características de funcionamiento:

- Centimeter-accurate relative positioning
(Carrier phase RTK)
- 10 Hz position/velocity/time solutions
- Open-source software and board design
- Low power consumption - 500mW typical
- Small form factor - 53x53mm
- USB and dual UART connectivity
- External antenna input
- Full-rate raw sample pass-through over USB

Figura 4. 19: Características del RTK Piksi

Según su descripción, sobre tierra el RTK Piksi puede alcanzar una exactitud de centímetros (radio de 10cm), con lo cual es posible el desarrollo del robot.

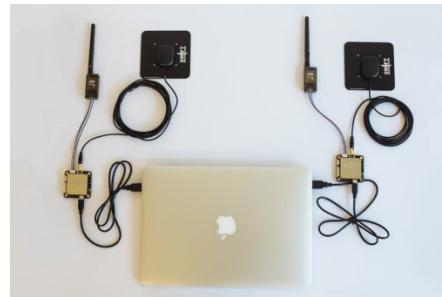


Figura 4. 20: Conexión del kit RTK

Cada módulo GPS del kit puede entregar información mediante el protocolo NMEA, sin embargo en modo RTK la información útil no son las tramas que se entrega por NMEA porque en ésta no se realiza las correcciones, sino las tramas que envía el GPS mediante protocolo SBP (propio de la marca Piksi). Cabe mencionar que los datos obtenidos por SBP son mediciones relativas y no absolutas, es por ello que si se necesita trabajar con coordenadas absolutas se debe de realizar la transformación respectiva a partir del punto fijo.

A continuación se presenta el diagrama de obtención de la posición a partir del sistema RTK:

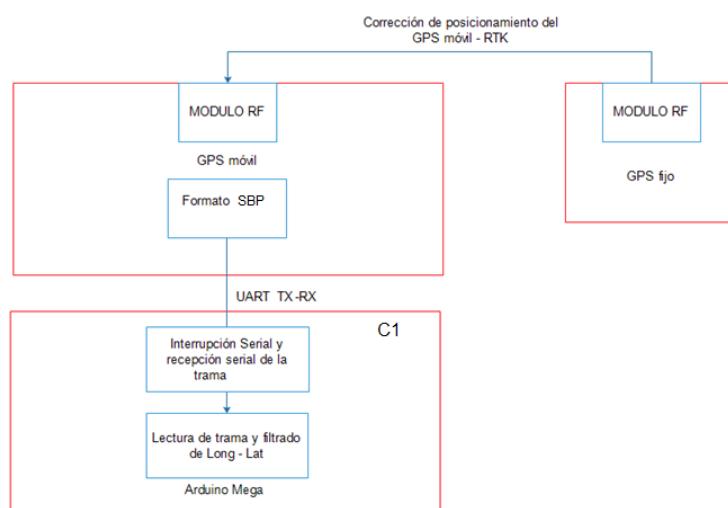


Figura 4. 21: Diagrama de obtención de la posición utilizando el RTK

Sensor de medición inercial – IMU

El sensor IMU elegido es el GY-85 (ver Figura 4.22), que contiene el magnetómetro HMC5883L, acelerómetro ADXL345 y giroscopio ITG3205, cada uno con medición en los 3 ejes.



Figura 4. 22: IMU GY-85

Con el IMU podemos obtener el ángulo y velocidad de giro del robot sobre los 3 ejes, sin embargo sólo interesa el giro que representa la orientación del robot.

El controlador C1 lee mediante el protocolo I2C el IMU, obteniendo la información de cada sensor.

La ubicación del IMU sobre el robot se realizará considerando 3 aspectos principales:

- *La posición central:*

Debido a que los ángulos tienen que medirse respecto al centro de giro del robot.

- *Alejamiento a fuentes de ruido :*

El sensor magnetómetro o compás electrónico es muy sensible a fuentes de campos magnéticos (imanes) y metálicos, es por ello que se colocó lo más separadamente posible de los motores y estructura mecánica.

- *Distancia al controlador C1 no muy grande:*

Debido a que el conductor posee resistencia y capacitancia, a mayor longitud de éste se altera la velocidad de transmisión de datos en el protocolo I2C (se usa para comunicar C1 con el IMU). Es por ello que mediante pruebas se eligió como máxima distancia 1m para un cable de cobre #24.

Encoders

Para medir la velocidad del robot se seleccionó encoders incrementales (ver Figura 4.23). Debido al costo elevado de los encoder comerciales ($>\$ 90$) se utiliza como encoders a motores de bajo costo ($<\$ 10$) que ya poseen encoders instalados.



Figura 4. 23 : Motor DC con eje directo al encoder

El encoder incremental se compone de un disco que posee ranuras además de leds infrarrojos en frente de fototransistores (ver Figura 4.24), gracias a dichas ranuras se envían pulsos de voltaje de 0 a 5V hacia el microcontrolador.

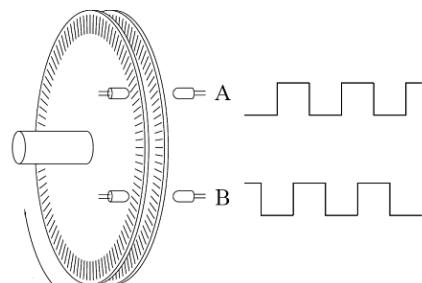


Figura 4. 24: Canales del encoder

El encoder utilizado posee 2 canales. Para la medición de la velocidad sólo se utiliza el canal A del encoder, sin embargo también se ha desarrollado el hardware y software para la medición del canal B para pruebas de control de ángulo de giro de las ruedas.

El controlador lee el encoder por medio de interrupciones externas que son provocadas por los flancos de los pulsos del encoder, éstas se acumulan en contadores.

Sensor de obstáculos

La detección de obstáculos durante el desplazamiento del robot es muy importante ya que de ésta manera se previenen accidentes y se cuida al robot de colisiones. Para ello se diseñó un arreglo de sensores ultrasónicos SRF05.

El principio de funcionamiento de los sensores ultrasónicos (ver Figura 4.25) es por medio de envío(trig-ping) y rebote(echo) de ondas sonoras a alta frecuencia(40KHz).

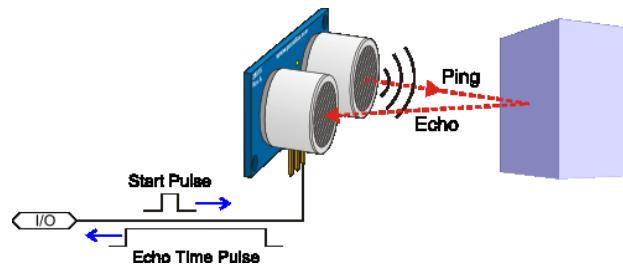


Figura 4. 25: Funcionamiento del sensor ultrasónico

La distribución de los sensores ultrasónicos es de forma circular de tal manera que cubra los 180° en la parte frontal del robot, además el rango de detección de obstáculos es de 10cm a 3m.

4.4.2. Microcontroladores

El robot posee 3 microcontroladores principales los cuales son:

Microcontrolador	Función
C1	<i>Control de la navegación autónoma</i>
	Manejo de las coordenadas geográficas
	Control de la orientación
	Supervisión del consumo energético
	Envío de información para el monitoreo de las variables de navegación del robot
C2	<i>Control de las tareas</i>
	Control del sistema de fumigación
	Adquisición de datos del campo agrícola
	Envío de información hacia el SCADA
C3	<i>Control de velocidad de los motores</i>
	Supervisión del buen funcionamiento de los motores

Tabla 4. 17 : Funcionamiento del sensor ultrasónico

Se eligió el microcontrolador Arduino Mega 2560 R3 (ver Figura 4.26) para la implementación de los controladores debido a las siguientes razones:

Controlador	Microcontrolador	Fundamento
C1	Arduino Mega 2560 R3	Posee los 4 puertos seriales por hardware que se requieren. Además de

		múltiples I/Os y capacidad de RAM para almacenar las geoposiciones de referencia.
C2	Arduino Mega 2560 R3	Posee 16 entradas analógicas para conectar una gran variedad de sensores analógicos, además 54 I/Os que pueden ser configurados como entradas para medición de una gran cantidad de sensores ON/OFF y como salidas cuando se requiera gran cantidad de actuadores.
C3	Arduino Mega 2560 R3	Posee 4 salidas de PWM que se requieren además de 2 puertos seriales por hardware. Además tiene la cantidad suficiente de entradas que generan interrupciones externas para poder leer los pulsos de los 2 encoders.

Tabla 4. 18 : Criterio para la elección de microcontroladores



Figura 4. 26: Arduino Mega 2560 R3

Controlador C1:

El controlador de navegación C1 presenta el siguiente diagrama de conexiones:

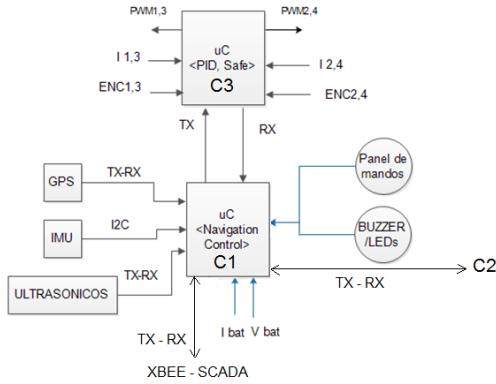


Figura 4. 27: Conexiones del controlador C1

Controlador C3:

El controlador C1 envía las referencias de velocidad de cada motor al controlador C3, de acuerdo a los resultados del control de orientación. El envío se realiza mediante la comunicación serial UART a una velocidad de 115200 Baudios para lograr la mayor rapidez. El controlador C3 calcula las acciones de control basándose en el contador de pulsos de cada encoder y además utiliza información de la corriente como saturador de la acción de control; finalmente envía las acciones de control en forma de modulación de ancho de pulso PWM hacia los drivers.

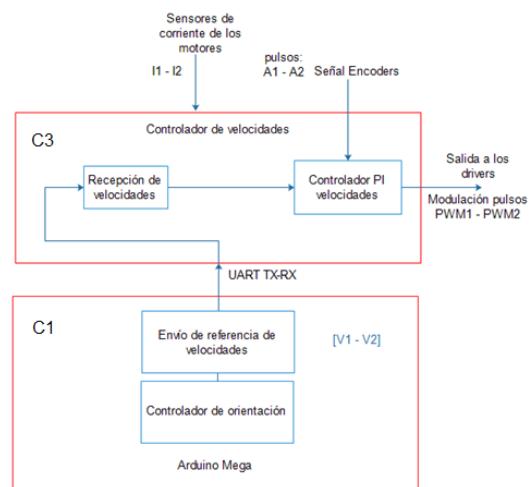


Figura 4. 28: Proceso del controlador C3

Controlador C2:

El controlador de tareas C2 presenta el siguiente diagrama de conexiones:

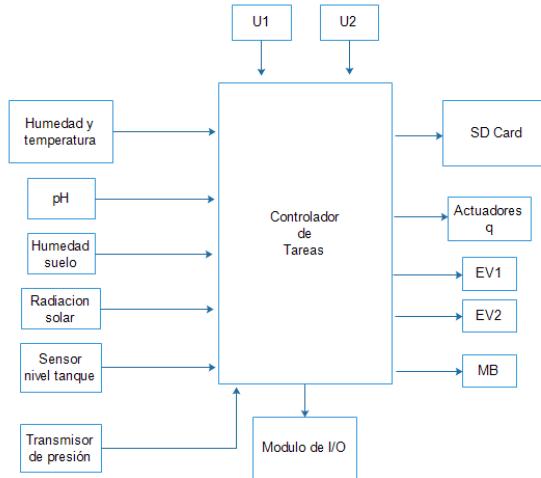


Figura 4. 29: Conexiones con el controlador de tareas

Donde los periféricos principales son:

- Entradas analógicas y digitales para los sensores de campo.
- Entradas digitales para los sensores de ultrasonido que detectan los cultivos laterales (U1, U2).
- Salida tipo PWM para el control de la motobomba DC (MB).
- Salidas digitales para las 2 válvulas solenoides (EV1, EV2).

4.4.3. Potencia

Para el movimiento y ejecución de tareas del robot se requiere de componentes que amplifiquen la potencia de las señales enviadas por el controlador, hacia los actuadores.

En el robot los actuadores son los siguientes:

- 2 Motores DC de alto torque para el movimiento del robot

- 1 Motor DC de alta velocidad para la bomba
- 2 Válvulas solenoide para la fumigación
- Otras actuadores opcionales

Las interfaces de potencia para los actuadores son las siguientes:

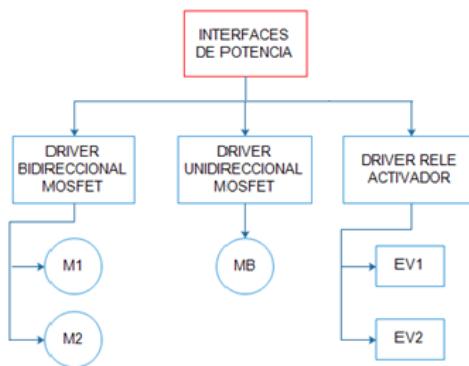


Figura 4. 30: Interfaces de potencia

Siendo:

- M1 y M2: son los motores DC de desplazamiento del robot
- MB: motor DC de la bomba de fumigación
- EV1 y EV2: son las válvulas solenoide izquierda y derecha respectivamente.

Driver de potencia de los motores de desplazamiento:

Los drivers de potencia diseñados son del tipo “Puente H” y utilizan transistores Mosfet de canal N IRF1404 debido al gran requerimiento de corriente por parte de los motores utilizados. Éste driver posee protección gracias a que posee 1 sensor de corriente ACS758 – 100B. El diseño se presenta en el en siguiente esquemático:

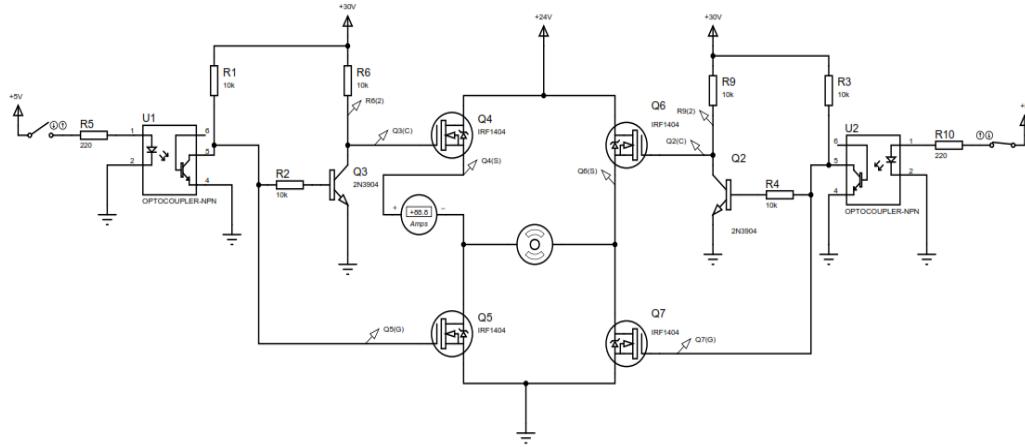


Figura 4. 31 : Esquema de driver tipo puente H utilizando “full channel N Mosfet”

Driver de potencia del motor de la bomba de fumigación

El driver para éste actuador es el mismo que para los motores de desplazamiento con la diferencia que sólo se usa de manera unidireccional.

Driver de potencia de las válvulas solenoídes

Las válvulas solenoide solamente requieren salidas tipo ON/OFF por ello se utiliza como interfaz de potencia un módulo de relés comercial ya que su costo es bajo y además el voltaje de activación es de 5V.

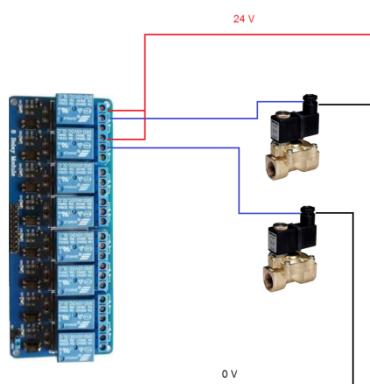


Figura 4. 32: Relés de activación de las válvulas solenoide

Las características eléctricas de los actuadores y sus interfaces de potencia son las siguientes:

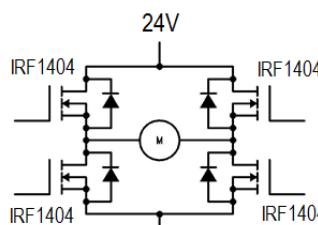
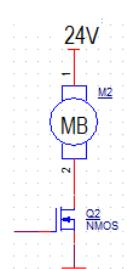
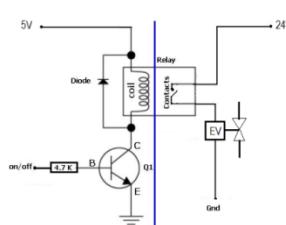
Actuador	Características	Interfaz de potencia
M1, M2	24V – 350W – 200RPM Imin = 4A(vacío) I nominal = 15A Imáx = 50A	Drivers bidireccional del tipo Puente H utilizando Mosfet Canal N IRF1404. 10 Mosfet por cada driver, con disipador. 
MB	24V - 350W 3600 RPM	Drivers unidireccional utilizando Mosfet Canal N IRF1404. 
EV1, EV2	24V – 6W	Driver tipo relé activado con 5V y salida 30VDC -10A máx. 
Otras salidas, opcionales para otras tareas.		Driver tipo relé.

Tabla 4. 19 : Actuadores y sus interfaces de potencia

Elementos de protección

Para las interfaces de potencia que pueden sobrecargarse y tienen más riesgo de falla se colocan elementos de seguridad, como son:

- *Diodo de polaridad DC:*

Éste diodo permite evitar malograr el driver por voltaje inverso debido a malas conexiones de la entrada de alimentación de voltaje.

- *Fusible de protección por sobre corriente:*

Permite evitar que los componentes del driver se malogren por sobre corriente.

- *Sensor de corriente:*

Permite monitorear continuamente la corriente consumida por los motores y poder predecir cortocircuitos o sobrecargas, siendo ésta información enviada al SCADA del robot.

- *Sensor de temperatura del circuito:*

Es un sensor redundante utilizado para aumentar la seguridad de funcionamiento del driver, ya que mide las temperaturas cercanas a los componentes del driver y tiene un límite máximo permisible, que el controlador continuamente va monitoreando. El funcionamiento del sensor de temperatura puede indicar un mal funcionamiento de los disipadores de calor.

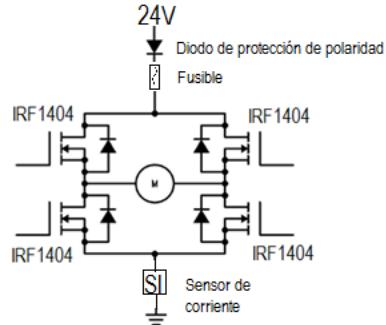


Figura 4. 33: Protecciones en un driver bidireccional Mosfet

Las características técnicas de los sensores de corriente y temperatura son:

Elementos	Características	Implementación
Sensor de corriente de cada driver de potencia para los motores de desplazamiento.	Capacidad de corriente de 50A por cada driver. Señal de salida tipo analógica.	En sensor de corriente de 50A es de la marca Allegro, código ACS758 – 50B.
Sensor de temperatura, es un elemento adicional de protección	El rango de temperaturas a sensar es de 10 – 100°C, ya que cuando el driver está operando en su máxima capacidad, inclusive el dissipador puede llegar a temperaturas altas. Señal de salida tipo analógica.	Para el rango de temperaturas se elige al sensor de temperatura LM35, el cual será instalado muy cercano al dissipador de los drivers.

Tabla 4. 20 : Características técnicas de los sensores de corriente y temperatura

4.4.4. Comunicaciones:

Las comunicaciones se basan básicamente en la interacción entre el SCADA, controlador de navegación y controlador de tareas. Las comunicaciones descritas aquí se refieren a las comunicaciones inalámbricas y comunicaciones alámbricas (entre controladores). El módulo de las comunicaciones inalámbricas se compone de un par Transceivers XBEE PRO S2.



Figura 4. 34: Módulo XBEE PRO S2

Las características del módulo XBEE PRO S2 se muestran en la Tabla 4.21:

Voltaje de operación	3.3V
Consumo de corriente máxima	295mA
Velocidad transmisión	250kbps
Potencia	63mW output (+17dBm)
Alcance de las comunicaciones	1 mile (1600m)
Frecuencia de transmisión	2.4 GHz
Antena	Cable

Tabla 4. 21 : Características del módulo XBEE

Arquitectura de comunicaciones

Se utilizó la siguiente arquitectura de comunicaciones de manera que sólo se usa 1 transceiver para comunicar el robot con el sistema SCADA.

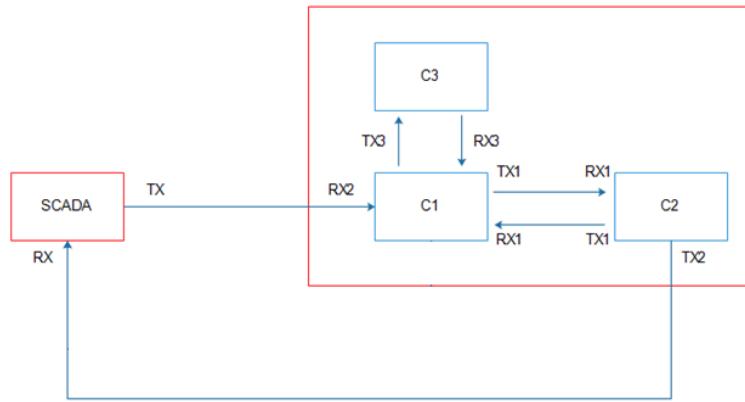


Figura 4. 35: Esquema de las comunicaciones

Siendo:

- **C1, C2 y C3:** los controladores de navegación, de tareas y de velocidades de los motores respectivamente.
- **TX, RX:** Son las vías de transmisión y recepción serial de cada controlador.
- **SCADA:** Es el sistema de supervisión, control y adquisición de datos del robot
- **Comunicaciones inalámbricas (mediante los módulos XBEE):**
 - TX SCADA → RX2, C1
 - TX2, C1 – TX2, C2 → RX SCADA

4.4.5. Sensores para la adquisición de datos

Existe una gran variedad de sensores para agricultura, sin embargo en el proyecto se utilizarán los más comunes y de bajo costo.

Se han considerado los siguientes sensores para la adquisición de datos :

Sensor	Código	Características	Modelo Seleccionado
Humedad del suelo	Soil moisture	Tipo resistivo. Respuesta rápida. Presión media. Salida analógica. Bajo costo. Voltaje de trabajo 3.3-5V.	
pH del suelo	pH Acidity Meter	Respuesta rápida. Presión media. Salida analógica. Bajo costo. Voltaje de trabajo 3.3-5V.	
Temperatura del ambiente	DHT11	0 – 50°C	
	1 muestra por segundo 3 – 5V Protocolo 1Wire	20 – 90 %RH	
Radiación solar	Panel solar	Respuesta rápida. Presión media. Salida analógica. Bajo costo. Voltaje de trabajo 3.3-5V.	

Tabla 4. 22 : Características de los sensores del módulo

4.4.6. Sensores para el sistema de fumigación

Los sensores que intervienen en éste proceso son los sensores detección de los cultivos, nivel tanque y presión del fluido. Para la detección de cultivos y medición del nivel del tanque se utilizan sensores de ultrasonido SRF05 (ver Figura 4.36).



Figura 4. 36: Sensor de ultrasonido SRF05

Para medir la presión del fluido se utiliza un transmisor de presión de la marca DANFOSS (ver Figura 4. 37), éste sensor entrega señal de 4 a 20mA para 0 a 50Bar de presión.



Figura 4. 37: Transmisor de presión DANFOSS MBS 3000

La medición de éstas variables es continua mientras se encuentre en el proceso de fumigación. A continuación se muestra el tipo de medida por cada sensor:

Id	Variable	Tipo de medida
U1	Ultrasónico izquierdo	Digital
U2	Ultrasónico derecho	Digital
N_t	Nivel del tanque	Digital

S_p	Sensor presión	Analógica
-----	----------------	-----------

Tabla 4. 23 : Tipos de señales de los sensores para la fumigación

4.5. Sistema de energía

El sistema de energía utiliza dos baterías de plomo de 12V, un sensor de voltaje en los bornes de las baterías y un sensor de consumo total de corriente del robot.

La configuración de las baterías es en serie para obtener 24V, que es el voltaje que requiere la mayoría de actuadores utilizados, además se dispondrá de salidas de 12V y 5V reguladas por DC-DCs (ver Figura 4.38).

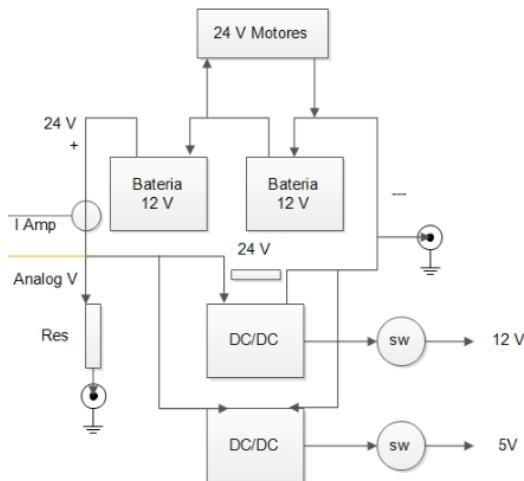


Figura 4. 38: Conexión de las baterías

El manejo inteligente de la energía se realiza mediante un algoritmo que se encuentra integrado en el controlador de navegación y hace uso de la señal de voltaje y corriente instantáneo de los sensores mostrados en la siguiente figura.

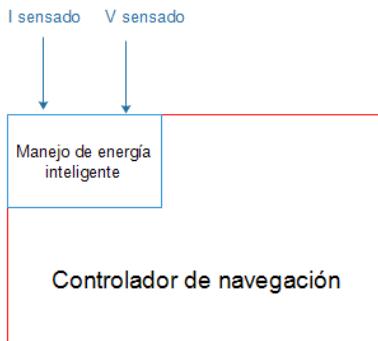


Figura 4. 39: Manejo del consumo instantáneo de energía

Los componentes utilizados poseen las siguientes características técnicas:

Componente	Características	Justificación e Implementación
Batería de plomo	2 Baterías ETNA 12V – 90Ah Plomo – Libre de mantenimiento 13 placas 17 Kg.	Se utiliza baterías de plomo ya que son mucho más fácil y seguras que otros tipos de baterías como las de Litio o Níquel. Además a pesar que su densidad de energía no es tan alta, posee un bajo costo con respecto a las otras baterías. 
Reguladores de voltaje	<i>Reguladores :</i> Regulador entrada 24V, salida 5V Capacidad 3A Regulador entrada 24V, salida 12V Capacidad 2A	Se utilizarán reguladores DC-DC debido a que tienen gran eficiencia de regulación y además que soporta hasta 3A. 2 Reguladores DC-DC LM2596 

Sensor de corriente	<p>Sensor de efecto Hall de 200A. Sensor analógico.</p>	<p>Debido a que no se encontró en el mercado sensor de efecto Hall comercial de 200A se van a utilizar 2 sensores de 100A en paralelo. En sensor de corriente de 100A es de la marca Allegro, código ACS758 – 100B.</p> 
Sensor de voltaje	<p>Sensor basado en un divisor de tensión hasta 5V máximo, que envía una señal análoga de voltaje hacia una entrada análoga del controlador.</p>	<p>Se utiliza un filtro del tipo seguidor de tensión a la salida del divisor de voltaje.</p>

Tabla 4. 24 : Tipos de señales de los sensores para la fumigación

Capítulo 5

Diseño de controladores y algoritmos

5.1. Modelamiento del robot

Para poder desarrollar el sistema de control es necesario realizar el modelamiento del robot, a continuación se desarrolla el modelo del robot móvil.

Modelo del robot móvil:

Se representa al robot móvil como un cuerpo rígido que se desplaza en un plano XY con velocidad lineal “V” y velocidad angular “W” tal como se observa en la Figura 5.1.

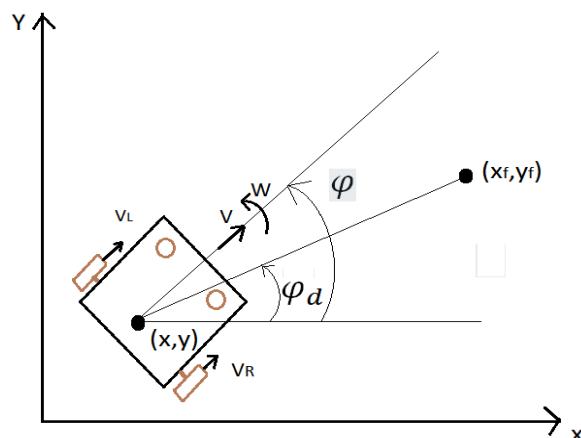


Figura 5. 1 : Representación del robot y sus variables

Siendo:

v : Velocidad lineal del robot

ω : Velocidad rotacional del robot

φ : Ángulo actual del robot respecto al sistema XY

φ_d : Ángulo deseado respecto al sistema XY

V_L : Velocidad lineal de la rueda izquierda del robot

V_R : Velocidad lineal de la rueda derecha del robot

(x, y) : Posición actual del robot

(x_f, y_f) : Posición objetivo del robot

El robot se moverá con velocidad lineal constante en un instante, por lo que:

$$\dot{x} = v \cos(\varphi) \quad (5.1)$$

$$\dot{y} = v \sin(\varphi) \quad (5.2)$$

$$\dot{\varphi} = \omega \quad (5.3)$$

Discretizando:

$$x_{k+1} = x_k + T v_k \cos(\varphi_k) \quad (5.4)$$

$$y_{k+1} = y_k + T v_k \sin(\varphi_k) \quad (5.5)$$

T : Tiempo de discretización

El vector de estados del modelo es:

$$X = \begin{bmatrix} x \\ y \\ \varphi \end{bmatrix} \quad (5.6)$$

La ecuación en espacio de estados del modelo no lineal es:

$$\dot{X} = \begin{bmatrix} -\sin(\varphi) & 0 \\ \cos(\varphi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (5.7)$$

$$v = \frac{R}{2}(u_r + u_l) \quad (5.8)$$

$$\omega = \frac{R}{W}(u_r - u_l) \quad (5.9)$$

Dónde:

u_l : Velocidad angular de la rueda izquierda

u_r : Velocidad angular de la rueda derecha

R: Radio de las ruedas del robot

W: Ancho del robot

5.2. Algoritmo para estimación de la orientación

La orientación del robot es una variable fundamental para el control de la navegación. Para medir la orientación utilizando el IMU se puede medir directamente el magnetómetro sin embargo la medida resulta ruidosa y muy sensible a las perturbaciones externas, es por ello que se necesita utilizar los sensores acelerómetro y giroscopio (en la Tabla 5.1 se presentan las ventajas y desventajas de cada sensor) de tal manera que las medidas se fusionen y resulte la medida de la orientación más exacta. También cabe resaltar que cuando la superficie por donde se desplaza el robot no es plana, el robot se inclina y la medida de la orientación (comúnmente llamado *heading*) del magnetómetro es errónea y es necesario hacer una compensación de la inclinación.

Sensor	Medición	Desventaja	Ventaja
Giroscopio	Mide velocidad angular en cada eje(x, y, z) y a partir de integración se puede calcular los ángulos.	Por la propagación del error en la integración se presenta el problema de la deriva (drift).	-Rápida respuesta. -No es sensible a objetos del entorno.
Acelerómetro	Mide aceleración lineal en cada eje. A partir de las aceleraciones estáticas se puede calcular los ángulos.	-Se tiene que eliminar la componente de gravedad. -Sensible a la vibración (ruido).	-Medición precisa. -No es sensible a objetos del entorno.

Magnetómetro	<p>Mide intensidad de campo magnético en cada eje.</p> <p>A partir de las intensidades en los ejes se puede calcular los ángulos.</p>	<ul style="list-style-type: none"> -Sensible a campos magnéticos y eléctricos externos; y a los objetos que poseen imanes y metales ferrosos. -Sensible al ruido. -La calibración es muy importante. 	<ul style="list-style-type: none"> -Entrega medidas absolutas respecto a las coordenadas geográficas.
--------------	---	---	--

Tabla 5. 1: Ventajas y desventajas de los sensores del IMU

Se elige el método DCM para la Fusión de Sensores ya que de acuerdo al estudio realizado en [19], entrega mejores resultados que el filtro de Kalman y complementario. Por otro lado respecto a método de los Quaterniones no se encontró una comparación sin embargo ya que el DCM dispone de un algoritmo claro y procedimental se eligió para el solucionar el problema de medición con exactitud de la orientación del robot.

5.2.1. Método de los Cosenos Directores (DCM)

El procedimiento que se sigue con el método DCM incluye las siguientes etapas:

- Lectura y calibración de los sensores

El controlador C1 lee mediante el protocolo I2C el IMU, obteniendo la información del acelerómetro, magnetómetro y giroscopio. Sin embargo previamente debe de realizarse un procedimiento de calibración para calcular el offset y escalamiento en para cada sensor y eje.

Para el caso del magnetómetro es como se muestra:

Primero se tienen que obtener los valores máximos y mínimos rotando el magnetómetro alrededor de los ejes X Y Z varias vueltas.

Las ecuaciones para calcular las medidas calibradas en cada eje son las siguientes:

Eje X:

$$MAGN_{X_OFFSET} = \frac{MAGN_{X_MIN} + MAGN_{X_MAX}}{2} \quad (5.10)$$

$$MAG_X_SCALE = \frac{100}{MAGN_X_MAX - MAGN_X_OFFSET} \quad (5.11)$$

$$magnetom[0] = (magnetom[0] - MAGN_X_OFFSET) MAGN_X_SCALE \quad (5.12)$$

Eje Y:

$$MAGN_{Y_OFFSET} = \frac{MAGN_{Y_MIN} + MAGN_{Y_MAX}}{2} \quad (5.13)$$

$$MAG_Y_SCALE = \frac{100}{MAGN_Y_MAX - MAGN_Y_OFFSET} \quad (5.14)$$

$$magnetom[1] = (magnetom[1] - MAGN_Y_OFFSET) MAGN_Y_SCALE \quad (5.15)$$

Eje Z:

$$MAGN_{Z_OFFSET} = \frac{MAGN_{Z_MIN} + MAGN_{Z_MAX}}{2} \quad (5.16)$$

$$MAG_Z_SCALE = \frac{100}{MAGN_Z_MAX - MAGN_Z_OFFSET} \quad (5.17)$$

$$magnetom[2] = (magnetom[2] - MAGN_Z_OFFSET) MAGN_Z_SCALE \quad (5.18)$$

De similar manera se realiza para el acelerómetro (en éste caso se gira al IMU lentamente para no generar aceleraciones adicionales a la gravedad) y giroscopio.

- Compensación de la inclinación (Tilt Compensation)

Los ángulos de Euler RPY son los siguientes:

$$\begin{bmatrix} Roll \\ Pitch \\ Yaw \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (5.19)$$

Cuando el plano sobre el cual se encuentra el IMU se inclina, la intensidad de campo magnético se pierde en los otros 2 ejes y la medición de la orientación (heading) se altera. Para compensar ello se realiza el “Tilt Compensation”

$$mag_x = magnetom[0]\cos(\theta) + magnetom[1]\sin(\phi)\sin(\theta) + magnetom[2]\cos(\phi)\sin(\theta) \quad (5.20)$$

$$mag_y = magnetom[1]\cos(\phi) - magnetom[2]\sin(\phi) \quad (5.21)$$

$$Heading = atan2\left(\frac{-mag_y}{mag_x}\right) \quad (5.22)$$

- Cálculo y actualización de la matriz de rotación (R)

A partir de la matriz de rotación en cada eje X Y Z se obtiene:

$$R = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (5.23)$$

- Normalización de la matriz R

El vector de corrección de tal manera que la componente en X e Y sean ortogonales es el error, en la Figura 5.1 se muestra el error como el vector “d”.

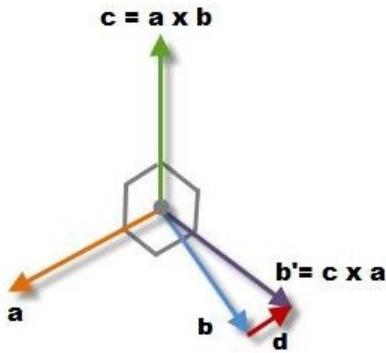


Figura 5. 2 : Vector de corrección

Sean:

$$X = \begin{bmatrix} r_{xx} \\ r_{xy} \\ r_{xz} \end{bmatrix} \quad (5.24)$$

$$Y = \begin{bmatrix} r_{yx} \\ r_{yy} \\ r_{yz} \end{bmatrix} \quad (5.25)$$

El producto punto entre X e Y representa el error:

$$\text{error} = X^T Y \quad (5.26)$$

Las componentes ortogonales son:

$$X_{\text{ortogonal}} = X - \frac{\text{error}}{2} Y \quad (5.27)$$

$$Y_{\text{ortogonal}} = Y - \frac{\text{error}}{2} X \quad (5.28)$$

La componente ortogonal en Z a partir del producto cruz de X e Y:

$$Z_{\text{ortogonal}} = X_{\text{ortogonal}} \times Y_{\text{ortogonal}} \quad (5.29)$$

Luego las componentes normalizadas se calculan:

$$X_{normalizado} = \frac{(3 - X_{ortogonal} \circ X_{ortogonal})}{2} \circ X_{ortogonal} \quad (5.30)$$

$$Y_{normalizado} = \frac{(3 - X_{ortogonal} \circ X_{ortogonal})}{2} \circ X_{ortogonal} \quad (5.31)$$

$$Z_{normalizado} = \frac{(3 - Z_{ortogonal} \circ Z_{ortogonal})}{2} \circ Z_{ortogonal} \quad (5.32)$$

- Cálculo de la corrección de la deriva (Drift Correction)

Sea el vector de error:

$$error' = \begin{bmatrix} r_{zx} \\ r_{zy} \\ r_{zz} \end{bmatrix} g_{referencia} \quad (5.33)$$

La corrección se realiza mediante un controlador PI.

$$wp_{correccion} = Kp \, error' \quad (5.34)$$

$$wi_{correccion} = Ki \, error' \quad (5.35)$$

La medición corregida del giroscopio es:

$$w(t) = w_{gyro}(t) + w_{correccion}(t) \quad (5.36)$$

- Obtención de los ángulos de Euler (RPY)

$$\theta = -\text{asin}(R(2,0)) \quad (5.37)$$

$$\phi = \text{atan2} \left(\frac{R(2,1)}{R(2,2)} \right) \quad (5.38)$$

$$\psi = \text{atan2} \left(\frac{R(1,0)}{R(0,0)} \right) \quad (5.39)$$

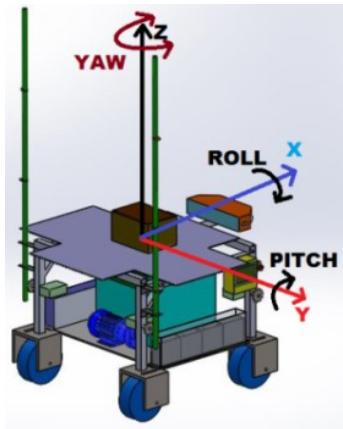


Figura 5. 3 : Ángulos de Attitude del robot

Para la navegación del robot desarrollado en la tesis se utiliza sólo el ángulo Yaw debido a que es el valor de la orientación del robot respecto a las coordenadas geográficas (absolutas).

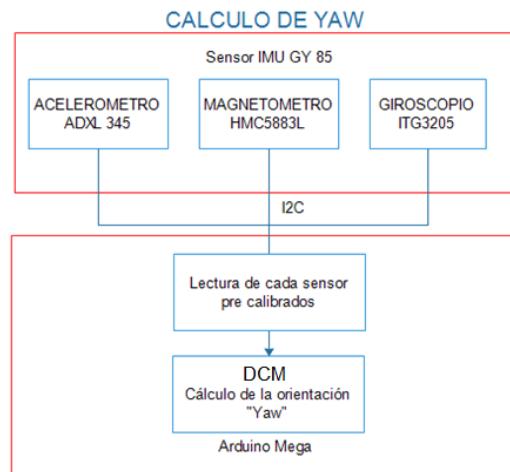


Figura 5. 4 : Diagrama obtención del ángulo Yaw

5.3. Controlador de navegación autónoma

5.3.1. Control basado en puntos

Se desarrolla un control de navegación basado en puntos (geo posiciones) que provienen del arreglo de puntos generado por el *path planning* a partir de la ruta

deseada. A partir de dicho arreglo de puntos, el controlador de navegación calcula la velocidad lineal y orientación que debe de tener el robot en el transcurso de su movimiento.

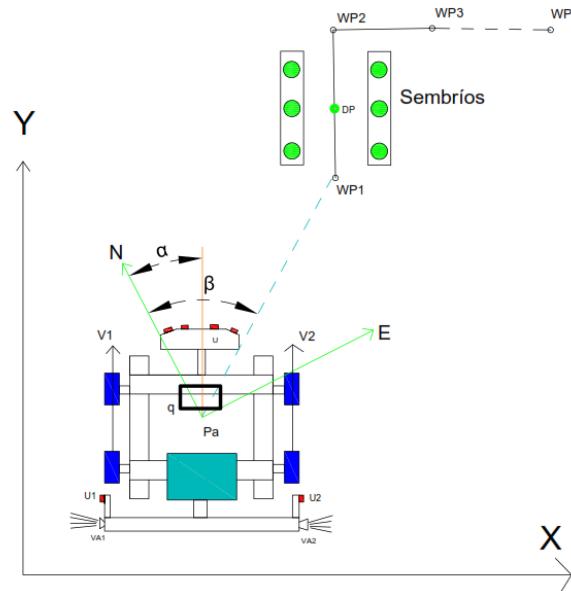


Figura 5. 5: Navegación basada en puntos

Para el control de la navegación utilizamos los controladores anteriormente desarrollados (control de orientación del robot y de velocidades de las ruedas), de tal manera que mediante la posición deseada se calcule la orientación y las velocidades de cada rueda requerida.

El diagrama de bloques para el control de la navegación se presenta en la Figura 5. 6.

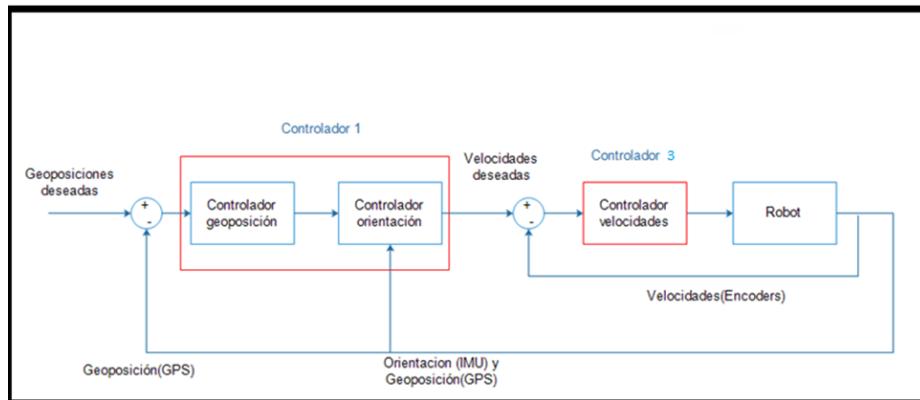


Figura 5. 6 : Navegación basada en puntos

Error de posición

Mientras el robot se va desplazando hacia el punto objetivo, el radio que los une va disminuyendo, por ende la norma cada vez se reduce y tiende a 0. Para visualizar el acercamiento del robot al punto objetivo, utilizamos el error de posición.

Calculamos el error a la posición objetivo como una norma cuadrada, que se puede representar como el radio de cercanía de la siguiente manera:

$$r^2 = (Latitud_f - Latitud_a)^2 + (Longitud_f - Longitud_a)^2 \quad (5. 40)$$

Dónde:

r: radio de cercanía al punto objetivo

5.3.2. Controlador de orientación del robot

Se quiere que el robot tenga su orientación igual a la orientación deseada. Para ello nos basaremos en la velocidad angular como acción de control ya que representa el cambio de orientación en el tiempo.

$$\frac{d\varphi}{dt} = \omega, \quad \frac{\Phi(s)}{W(s)} = \frac{1}{s} \quad (5. 41)$$

$$\varphi_d = \text{atan2}(y_f - y, x_f - x) \quad (5. 42)$$

e: Diferencia entre el ángulo actual y el deseado

$$e = \varphi - \varphi_d \quad (5.43)$$

Discretización de la ecuación de velocidad angular:

$$\varphi_k = \varphi_{k-1} + T * \omega_{k-1} \quad (5.44)$$

$$e_k = \varphi_k - \varphi_{d_k} \quad (5.45)$$

$$e_k = \varphi_k - \text{atan2}(y_f - y_k, x_f - x_k) \quad (5.46)$$

Se considera la velocidad lineal constante por tramos según convenga:

$$v_k = v_0 \quad (5.47)$$

La acción de control se calcularía como una función del error de orientación:

$$\omega_k = f(e_k) \quad (5.48)$$

El tiempo de muestreo se calcula a partir de la función de transferencia del modelo de control de orientación de la ecuación 5.41. La constante de tiempo es igual a 1 y por ende el tiempo de muestreo (T) que es la quinta parte [23] resulta 200ms.

En éste caso la función f será un PI discreto utilizando Tustin [23], de tal manera que se disminuya a 0 el error estacionario, el algoritmo de control es el siguiente:

Acción proporcional	$A_p[k] = K_p (e[k] - e[k - 1])$
Acción integral	$A_i[k] = K_i \frac{T}{2} (e[k] + e[k - 1])$
Acción de control	$w[k] = w[k - 1] + A_p[k] + A_i[k]$

Tabla 5. 2 : Pasos del algoritmo de control PI discreto

Ahora se requiere convertir la velocidad angular en 2 componentes: para la velocidad angular de la rueda derecha y la rueda izquierda.

A partir de las ecuaciones de cinemática del robot móvil tipo diferencial y las ecuaciones que relacionan velocidad lineal con angular mediante el radio de rueda se tiene las siguientes ecuaciones:

$$V_L = R * u_l \quad (5.49)$$

$$V_R = R * u_r \quad (5.50)$$

Se puede despejar la velocidad angular necesaria en cada rueda (izquierda y derecha) de tal manera que se cumpla el control de orientación:

$$u_L = \frac{V}{R} - \frac{1}{2} \frac{wW}{R} \quad (5.51)$$

$$u_R = \frac{V}{R} + \frac{1}{2} \frac{wW}{R} \quad (5.52)$$

Las velocidades angulares calculadas, serán las referencias para los controladores de velocidad, tal como se muestra en el siguiente esquema de bloques:

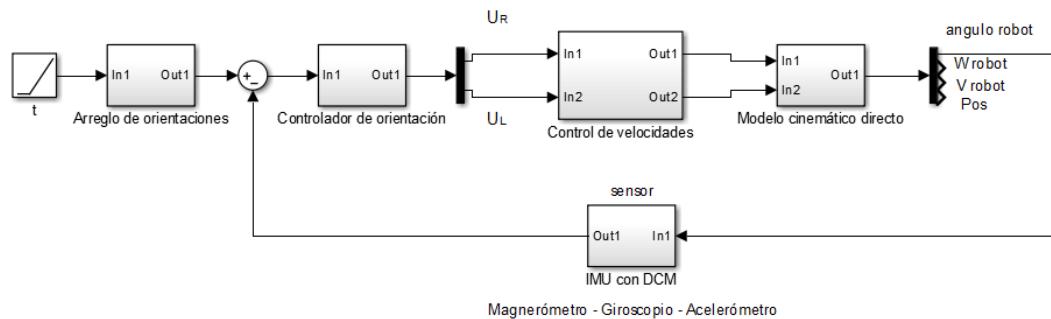


Figura 5. 7 : Diagrama del sistema de control de orientación

5.3.3. Controlador de velocidad de los motores DC

5.3.3.1. Modelo matemático del motor DC + caja reductora

El motor utilizado es un motor DC que funciona normalmente en la dirección de los autos de marca Mitsubishi. Éste motor posee una reducción mediante una caja de engranajes en transmisión tipo tornillo sinfín – corona.



Figura 5.8 : Motor DC y transmisión tornillo sinfín – corona

El control es por corriente de armadura. La representación para el modelamiento se muestra en la Figura 5.9.

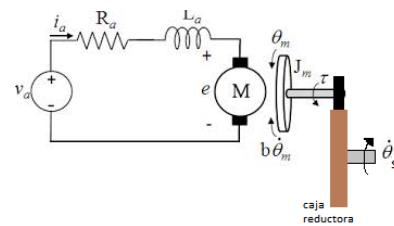


Figura 5.9: Modelamiento del motor con caja reductora

La caja reductora proporciona una reducción de velocidad y aumento de torque.

El ratio de velocidades de la caja reductora es:

$$\frac{\theta_s}{\theta_m} = r_t \quad (5.53)$$

El diagrama de bloques que representa la función de transferencia del motor DC + caja reductora es:

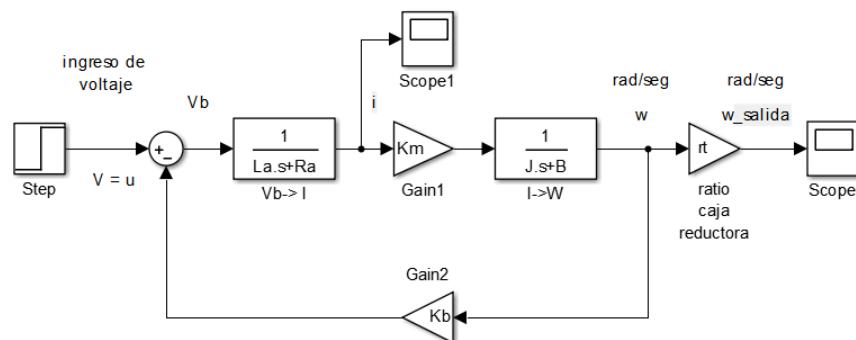


Figura 5.10: Representación en bloques del modelo matemático del motor

Siendo, los parámetros:

Símbolo	Parámetro
R_a	Resistencia de armadura
L_a	Inductancia de armadura
J	Momento de inercia
B	Coeficiente de fricción
K_b	Constante de velocidad / eléctrica
K_m	Constante de torque / mecánica
r_t	Ratio de transmisión de la caja

Tabla 5. 3 : Parámetros del motor con caja reductora

La función de transferencia, con entrada el voltaje al motor (V) y salida la velocidad angular después de la caja reductora (Ws) es:

$$G(s) = \frac{W_s(s)}{V(s)} = \frac{K_m r_t}{L_a J s^2 + (R_a J + L_a B)s + R_a B + K_b K_m} \quad (5.54)$$

Como se puede apreciar son varios los parámetros que se tienen que conocer, es por ello que se puede seguir 2 métodos para obtener la FT:

- *Medición experimental de cada parámetro:* Medición de resistencias, inductancias, constantes, etc., mediante pruebas eléctricas.
- *Identificación de sistemas:* Obtener la FT como un bloque, sin necesidad de conocer el valor de cada parámetro por separado.

Se procederá con el método de identificación de sistemas para poder obtener la FT de la planta. Además cabe recalcar que el modelo general obtenido servirá para relacionarlo con alguna estructura de la identificación.

A partir en adelante se mencionará al sistema “motor DC + caja reductora” como solo motor.

5.3.3.2. Identificación del modelo de los motores

Para diseñar y simular el controlador de velocidad de los motores es necesario conocer su modelo matemático. Debido a que no se disponen de los parámetros para cada motor, se desarrolló la identificación del modelo de los motores.

El método de identificación que se utiliza es el método de “identificación no paramétrica”. Se usa dicho método porque se conoce previamente la función de transferencia del motor en función de los parámetros y es posible relacionarla con las siguientes características: retardos, constantes de tiempo y ganancia estática.

La estructura de modelo no paramétrico seleccionado es:

$$G(s) = \frac{K_p}{(1 + T_{p1}s)(1 + T_{p2}s)} e^{-s\tau} \quad (5.55)$$

Para obtener los datos necesarios para aplicar el método de identificación, se aplicó una entrada tipo escalón en subida y bajada tal como se muestra en la Figura 5.11. Se utilizó para la primera prueba el motor 2, además fue necesario un microcontrolador para enviar la señal de control “ u ” en los tiempos establecidos y además para leer y guardar los datos de entrada y salida. Para el funcionamiento del motor se utilizó un driver de potencia de la capacidad necesaria y además un encoder acoplado al eje de salida del motor.

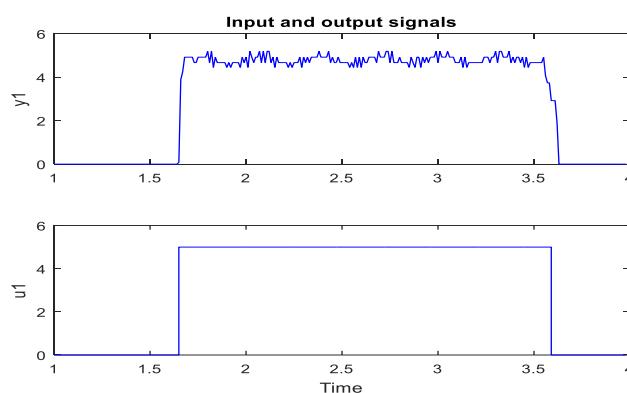


Figura 5. 11: Señales de entrada y salida para la identificación del sistema

Una vez obtenido los datos de entrada y salida (“mydata”), se utilizó el toolbox de identificación de sistemas de Matlab.

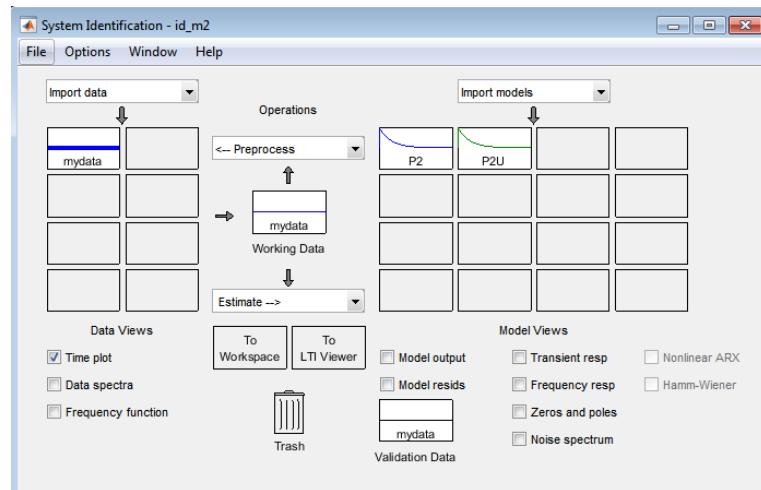


Figura 5. 12: Toolbox “ident” de Matlab

Seleccionando la estimación para el tipo: Process Models y 2 Poles, como se muestra en la Figura 5.13.

Debido a que la planta a identificar no presenta retardo (de acuerdo a las pruebas la salida varía en el mismo tiempo que ingresa la señal de control), no se considera el “Delay”.

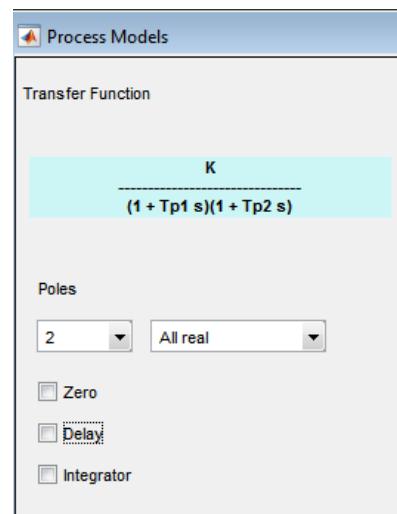


Figura 5. 13: Selección de la estructura de modelo no paramétrico

Los resultados de la identificación entregan los resultados de la Figura 5. 14 con un ajuste del 87.87%.

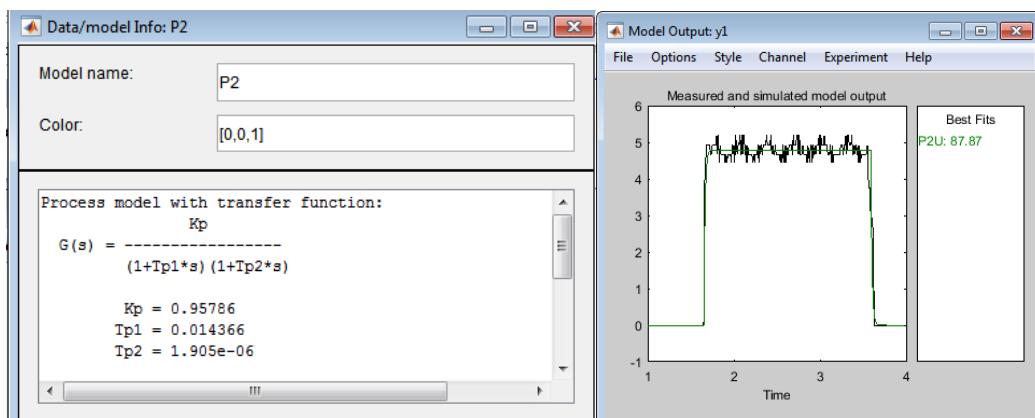


Figura 5. 14 : Resultados de la identificación

Se considera que es un ajuste bueno para el modelo debido a que la presencia de ruido altera los valores de la salida.

$$G(s) = \frac{0.95786}{(1 + 0.014366 s)(1 + 1.905 \cdot 10^{-6} s)} \quad (5.56)$$

El mismo procedimiento desarrollado para el motor 2 se realiza para todos los demás motores. Los resultados de la identificación se presentan en la Tabla 5.4.

Motor	Función de transferencia identificada	FT aproximada
Motor 1	$G(s) = \frac{0.88548}{6.2138 \times 10^{-11} s^2 + 0.0148 s + 1}$	$G(s) = \frac{0.88548}{0.0148 s + 1}$
Motor 2	$G(s) = \frac{0.95786}{2.7367 \times 10^{-8} s^2 + 0.0144 s + 1}$	$G(s) = \frac{0.95786}{0.0144 s + 1}$
Motor 3	$G(s) = \frac{0.89041}{2.2538 \times 10^{-8} s^2 + 0.0153 s + 1}$	$G(s) = \frac{0.89041}{0.0153 s + 1}$
Motor 4	$G(s) = \frac{0.96516}{2.7367 \times 10^{-8} s^2 + 0.0135 s + 1}$	$G(s) = \frac{0.96516}{0.0135 s + 1}$

Tabla 5. 4 : Modelos identificados de los motores

Inicialmente se utilizó 4 motores DC, pero debido a los inconvenientes que hubo en el giro del robot utilizando 4 ruedas motorizadas(skid steer), se cambió a la arquitectura diferencial utilizando 2 ruedas motorizadas y 2 ruedas tipo castor, una en cada extremo de la parte frontal del robot.

Los 2 motores elegidos fueron el motor 2 y 4 debido a que poseen mayor constante K_m y K_b y por ende consumen menos corriente para una misma carga.

Motor	Motor	FT - s
Motor izquierdo (ML)	Motor 2	$G(s) = \frac{0.95786}{0.0144 s + 1}$
Motor derecho (MR)	Motor 4	$G(s) = \frac{0.96516}{0.0135 s + 1}$

Tabla 5. 5 : Modelos de los motores seleccionados: derecho e izquierdo

La respuesta de los modelos de motor obtenidos ante una entrada escalón unitaria, para visualizar su comportamiento, es como se muestra en la Figura 5.15.

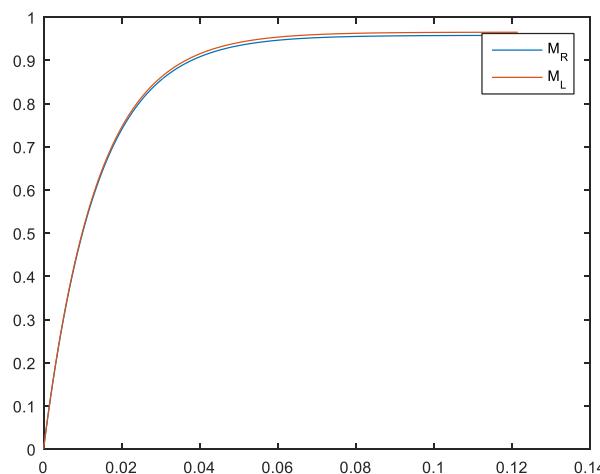


Figura 5. 15 : Respuesta de los modelos identificados ante un escalón unitario

Se puede visualizar que las respuestas obtenidas son muy similares.

5.3.3.3. Diseño del controlador de velocidad del motor DC

Para el diseño del controlador de velocidad del motor se requiere calcular el periodo de muestreo “T”, durante el cual el controlador actualizará la señal de control. También “T” puede ser visto como el periodo de discretización de la planta (motor) o del controlador en rediseño (continuo a discreto).

De manera práctica el periodo de muestreo se elige como una fracción de la constante de tiempo de la planta [23]:

$$T = \frac{\tau}{5} \quad (5.57)$$

Para el caso del motor 2:

$$T = \frac{0.0144}{5} \quad (5.58)$$

$$T \approx 3ms \quad (5.59)$$

Como las constantes de tiempo del modelo matemático de los motores son muy similares, se elige el mismo periodo de muestreo en el diseño del controlador.

El diagrama de bloques para el proceso de control de velocidad del motor es:

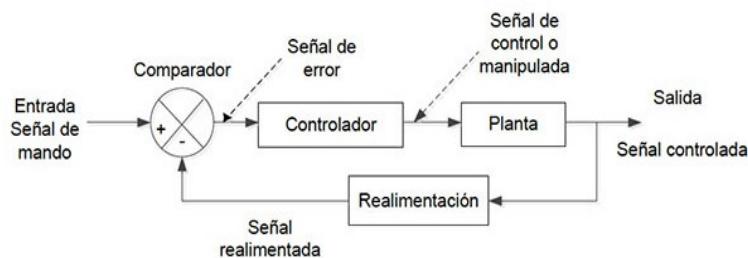


Figura 5. 16 : Proceso de control en lazo cerrado

Siendo la planta el motor, la retroalimentación realizada por un encoder de 100ppv (pulsos por vuelta) y el controlador elegido es un controlador PID discreto debido a las siguientes razones:

- Se tiene un modelo lineal e invariante en el tiempo que representa el motor.
- El controlador PID discreto es más fácil implementar en un microcontrolador y de ajustar en las pruebas.
- De acuerdo a las pruebas en simulación se obtuvo una respuesta del controlador suficientemente buena para la aplicación a realizar.

Para el diseño del controlador PID discreto es posible realizar 2 métodos:

- Diseño en el dominio de continuo “s” y posterior rediseño al dominio discreto “z” utilizando *aproximación bilineal* o también llamado *Tustin*.
- Diseño en el dominio discreto “z”.

Utilizaremos el primer método para el diseño del controlador digital de velocidad de los motores:

Diseño del controlador en tiempo continuo

Se utiliza el siguiente diagrama de bloques:

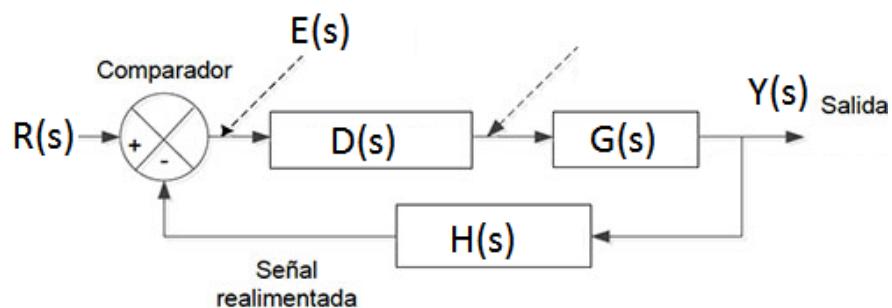


Figura 5. 17 : Diagrama de bloques

La función de transferencia en lazo cerrado es:

$$FTLC(s) = \frac{Y(s)}{R(s)} = \frac{D(s)G(s)}{1 + D(s)G(s)H(s)} \quad (5. 60)$$

Para la estabilidad se analiza el denominador de la FTLC.

$$1 + D(s)G(s)H(s) \quad (5.61)$$

Siendo D(s), el controlador PID continuo de tipo paralelo:

$$D(s) = K_p + \frac{K_i}{s} + K_d s \quad (5.62)$$

Hay muchos métodos para analizar la estabilidad, en éste estudio se parte con el método de Routh Hurwitz [24] (ver Figura 5.18) y la par se eligen valores de las constantes de tal manera que se cumpla los requisitos de diseño.

$$a_2 s^2 + a_1 s + a_0 = 0$$

$$\begin{array}{c|cc} s^2 & a_2 & a_0 \\ s^1 & a_1 & 0 \\ s^0 & b_1 & 0 \end{array}$$

Figura 5. 18: Método de Routh Hurwitz para la estabilidad

Por otro lado de acuerdo a las pruebas realizadas en hardware, se apreció que la señal del encoder es ruidosa y por lo tanto al incluir acción derivativa el ruido se amplifica en la salida ya que la acción derivativa es un filtro “pasa altos”. Es por ello que se tomó como estructura del controlador sólo el PI paralelo, por lo tanto el controlador es:

$$D(s) = K_p + \frac{K_i}{s} \quad (5.63)$$

Además el encoder tiene ganancia unitaria ya que la velocidad es calculada por el mismo microcontrolador.

$$H(s) = 1 \quad (5.64)$$

Para el motor 2 se tiene la siguiente función de transferencia en lazo cerrado:

$$\frac{Y(s)}{R(s)} = \frac{\left(K_p + \frac{K_i}{s}\right)\left(\frac{0.95786}{0.0144 s + 1}\right)}{1 + \left(K_p + \frac{K_i}{s}\right)\left(\frac{0.95786}{0.0144 s + 1}\right)} \quad (5.65)$$

$$\frac{Y(s)}{R(s)} = \frac{0.95786K_p s + 0.95786K_i}{0.0144s^2 + (1 + 0.95786K_p)s + 0.95786K_i} \quad (5.66)$$

Luego de analizar las características de la respuesta:

- Tiempo de establecimiento
- Sobreimpulso máximo

Se obtuvieron las siguientes constantes:

$$K_p = 0.1 \quad (5.67)$$

$$K_i = 9 \quad (5.68)$$

Rediseño del controlador continuo a discreto

Para el rediseño se utilizó la aproximación de Tustin [23]:

$$s = \frac{2}{T} \left(\frac{z - 1}{z + 1} \right) \quad (5.69)$$

Entonces el controlador digital resulta:

$$D(z) = K_p + K_i \frac{T}{2} \left(\frac{z + 1}{z - 1} \right) \quad (5.70)$$

La implementación en el microcontrolador del controlador digital calculado es en tiempo discreto y en forma de ecuación en diferencias, para ello se obtiene las siguientes ecuaciones en diferencias que presentan el algoritmo de control:

Error	$e[k] = SP - ym[k]$
Acción proporcional	$Ap[k] = K_p (e[k] - e[k - 1])$
Acción integral	$Ai[k] = Ki \frac{T}{2} (e[k] + e[k - 1])$
Ley de Control	$u[k] = u[k - 1] + Ap[k] + Ai[k]$

Almacenar valores anteriores	$e[k - 1] = e[k]$ $u[k - 1] = u[k]$
------------------------------	--

Tabla 5. 6 : Pasos del algoritmo de control en tiempo discreto

Prueba para el motor 2, motor izquierdo:

Las pruebas de simulación se realizaron en SIMULINK considerando todos los parámetros calculados y añadiendo el ruido medido a partir de una prueba experimental.

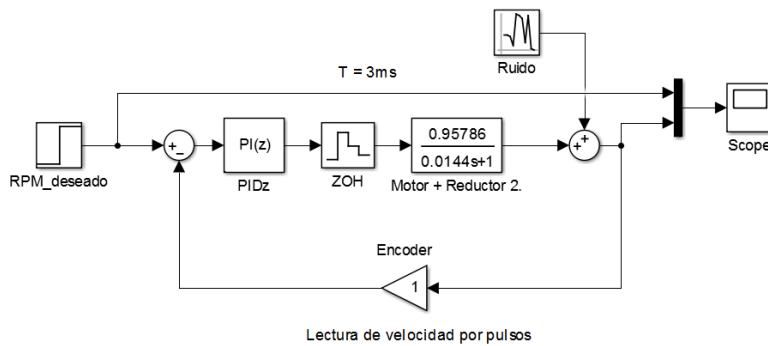


Figura 5. 19: Esquema de control de velocidad del motor 2

La configuración del bloque de PI (z) es:

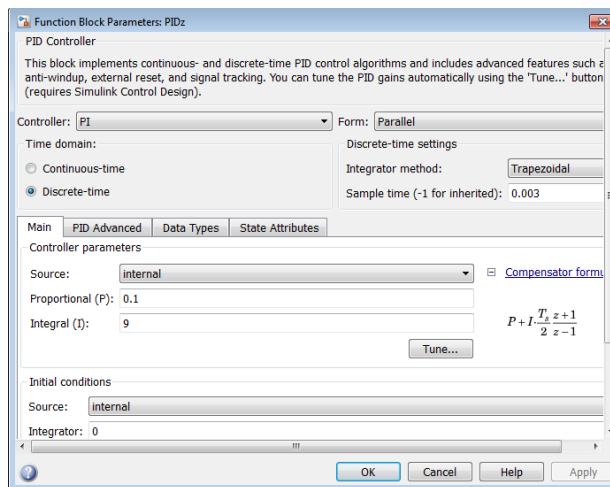


Figura 5. 20: Configuración del bloque PIDz de Matlab para motor 2

Cabe resaltar que el ruido se midió a partir de un conjunto de datos almacenados de la medición del encoder, de los cuales se tomó su media y varianza, y se approximó a un ruido blanco con las siguientes características:

$$\text{media} = 0 \quad (5.71)$$

$$\text{varianza} = 1 \quad (5.72)$$

Finalmente con el controlador para el motor 2 se obtiene la siguiente respuesta:

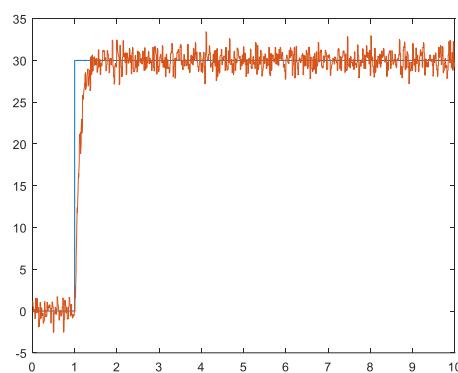


Figura 5. 21 : Respuesta del controlador para referencia de 30RPM

Prueba para el motor 4, motor derecho:

Las pruebas de simulación se realizaron en SIMULINK considerando todos los parámetros calculado y añadiendo el ruido medido a partir de una prueba experimental. Se obtuvo los siguientes parámetros del controlador PIDz:

$$K_p = 0.1 \quad (5.73)$$

$$K_i = 12 \quad (5.74)$$

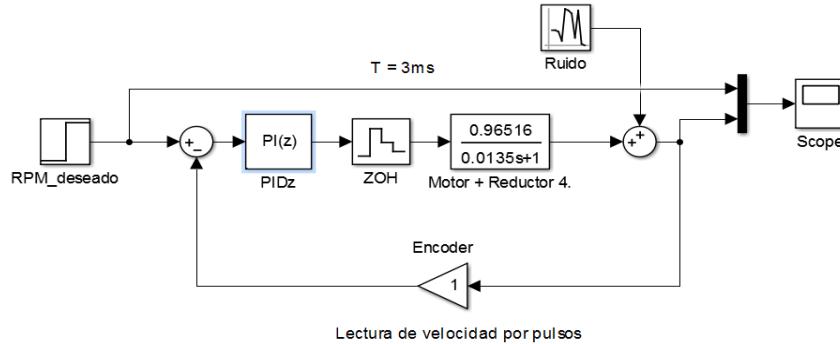


Figura 5. 22: Respuesta del controlador para referencia de 30RPM

La configuración del bloque de PI (z) es:

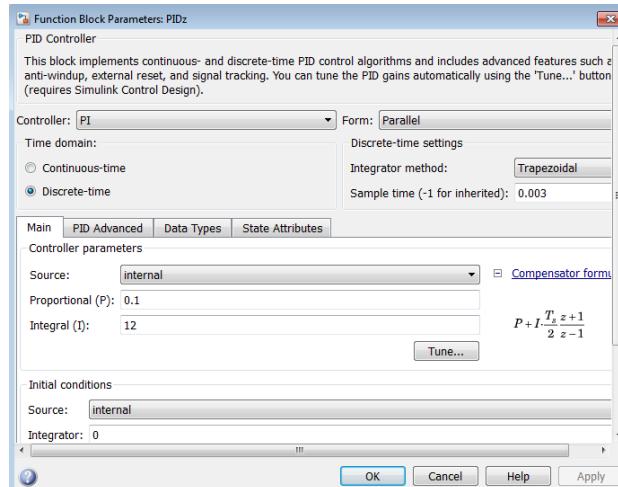


Figura 5. 23 : Configuración del bloque PIDz de Matlab para motor 4

El ruido se midió a partir de un conjunto de datos almacenados de la medición del encoder, de los cuales se tomó su media y varianza, y se approximó a un ruido blanco con las siguientes características:

$$\text{media} = 0 \quad (5.75)$$

$$\text{varianza} = 1 \quad (5.76)$$

Finalmente con el controlador para el motor 4 se obtiene la siguiente respuesta:

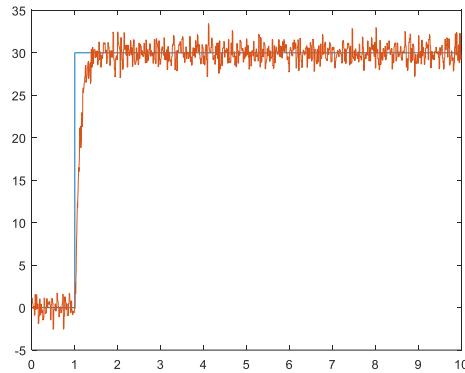


Figura 5. 24 : Respuesta del controlador para referencia de 30RPM

5.3.4. Algoritmos

El método consiste en calcular las velocidades para cada rueda del robot en base a los sensores de navegación para lograr la orientación adecuada. Cuando el robot alcance la orientación deseada seguirá su movimiento en dicha orientación hasta llegar a la vecindad del punto objetivo. En caso el robot detecte obstáculos dinámicos sobre su ruta, tendrá la opción de parar o esquivarlo dependiendo del tipo de obstáculo ya que el espacio disponible sobre el terreno agrícola es limitado.

Se le denomina **vecindad** a los alrededores muy cercanos de la coordenada exacta objetivo, y se define mediante un círculo de radio “ r ”, donde r depende la exactitud requerida.

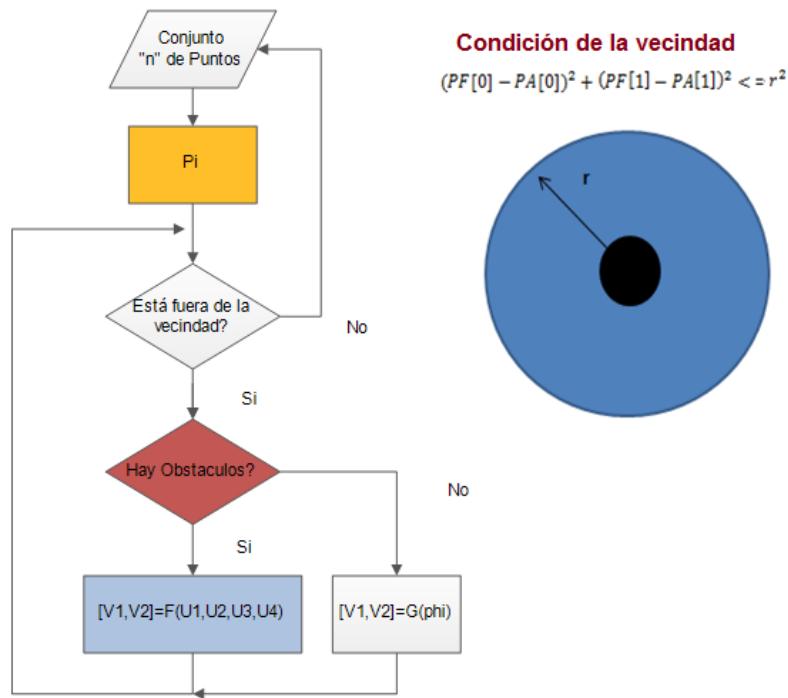


Figura 5. 25 : Algoritmo para la navegación autónoma

Siendo:

- Φ : Orientación del robot
- V_1, V_2 : Velocidades de las ruedas de cada lado del robot.
- U_1, U_2, \dots, U_n : Sensado por ultrasonido de obstáculos
- r : Radio del círculo de vecindad

Por otro lado los controladores que participan en el control de la navegación son C1 y C3. A continuación se muestra el diagrama de flujo de su funcionamiento:

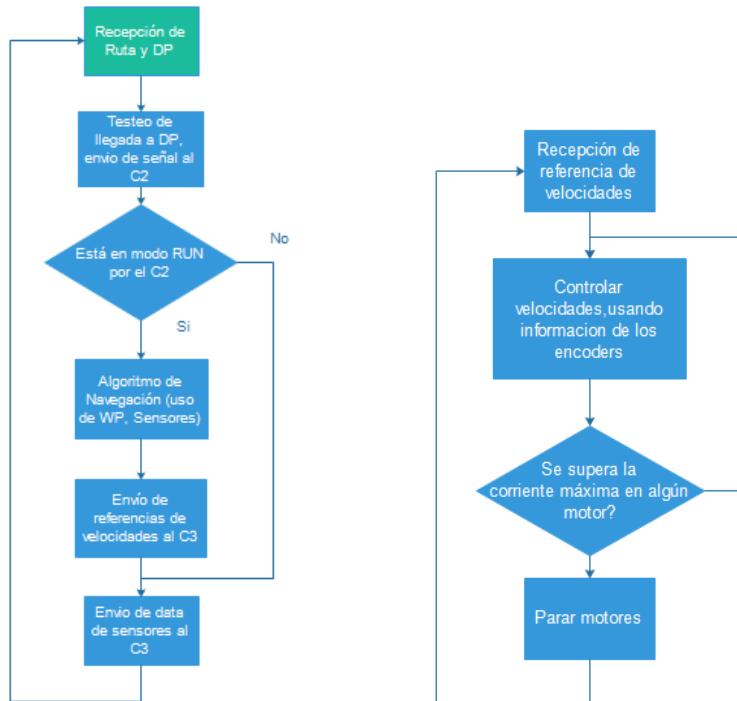


Figura 5. 26 : Diagrama de flujo del controlador C1 y C3 para la navegación

5.4. Controlador de tareas

5.4.1. Controlador de presión de la fumigación

El controlador de la presión deseada se basa en un lazo cerrado de tipo PI, donde la presión deseada se envía desde la interfaz SCADA.

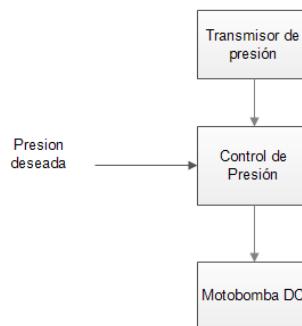


Figura 5. 27 : Control de presión de la línea de presurizado

En el control de la presión se escaló la señal análoga de entrada junto con la salida, se modula por ancho de pulso PWM el voltaje de alimentación del motor DC de la bomba. Se realizó una sintonización utilizando el método de Ziegler Nichols.

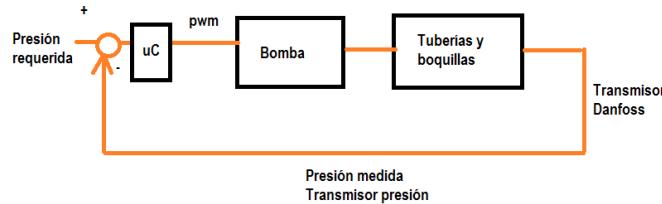


Figura 5. 28 : Lazo de control para la implementación del control de presión

5.4.2. Algoritmo para la fumigación

La activación de las válvulas solenoide se realiza en función a la detección de los cultivos laterales por parte de los sensores de distancia ultrasónicos. Primero se toma lectura de los sensores y en función a la lógica de activación se abren o cierran las válvulas solenoides en cada lado.

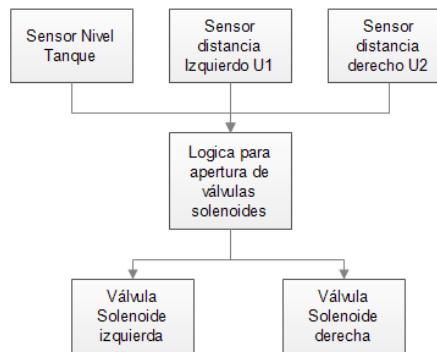


Figura 5. 29 : Diagrama de bloques para la activación de las válvulas solenoides

El diagrama de flujo para el proceso de fumigación integrando los 2 algoritmos anteriores, se muestra a continuación.

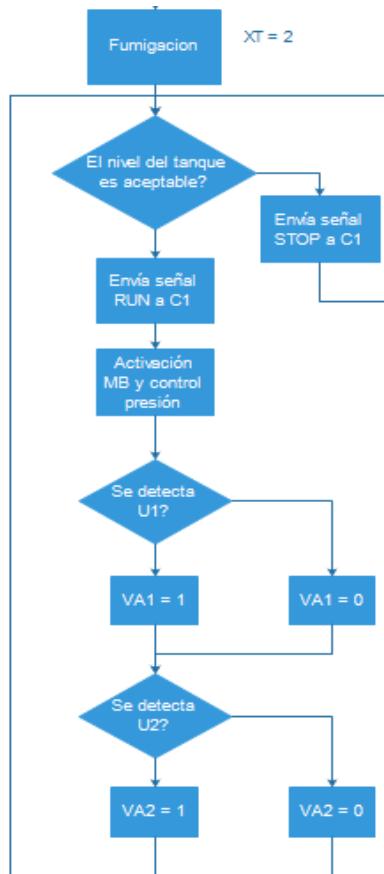


Figura 5. 30 : Diagrama de flujo del proceso de fumigación

5.4.3. Algoritmo para adquisición de datos

El sistema de adquisición de datos, sigue el proceso descrito en la Figura 5.31. Éste proceso lo ejecuta el controlador C2. La temporización se puede asignar desde la interfaz, el controlador para seguir ello utiliza timers. Además la lectura de sensores se realiza leyendo los puertos analógicos y los protocolos digitales correspondientes. Por otro lado el filtrado de ruido se realiza utilizando el método de la “media móvil”.

Para poder enviar hacia el SCADA los datos se ordenan en vectores de bytes con un identificador correspondiente de cada variable.

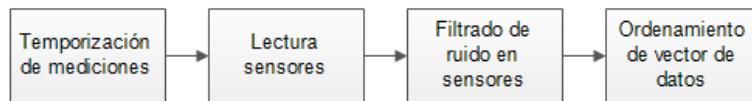


Figura 5. 31 : Proceso de adquisición de datos de los sensores

En la Tabla 5.7 se puede observar los identificadores de cada sensor, el tipo de señal que entrega, y su correspondiente tiempo de muestreo.

Identificadores de trama	Id	Variable	Tipo de señal	Tiempo de muestreo
a	S1	Humedad suelo	Analógica	T1
b	S2	Radiación solar	Analógica	T2
c	S3	pH Suelo	Analógica	T3
d	S4	Temperatura ambiental	Digital	T4
e	S5	Humedad ambiental	Digital	T4

Tabla 5. 7 : Variables medidas y sus características

Capítulo 6

SCADA

6.1. Características del diseño

El SCADA tiene la función de supervisión, control y adquisición de datos, de acuerdo a ello y a las funcionalidades que debe poseer el robot se diseña el software necesario. La supervisión se basa en la visualización de las variables del proceso, el control en poder manejar remotamente al Robot en lazo abierto y cerrado, y la adquisición de datos en almacenar la información importante en base de datos.

Se propone un SCADA con interfaz desarrollada sobre una plataforma de escritorio en Visual Studio, ya que éste software permite utilizar recursos gráficos y varios lenguajes de programación para una misma aplicación, además para el manejo de mapas se utiliza el API de Google Earth, ya que es de uso libre y contiene las funcionalidades suficientes para obtener información de las coordenadas en el espacio de trabajo y graficar rutas sobre él. Por otro lado para las comunicaciones con el Robot se utiliza el puerto de comunicaciones serial UART y finalmente para el almacenamiento de datos se utiliza una base de datos de M-SQL Server.

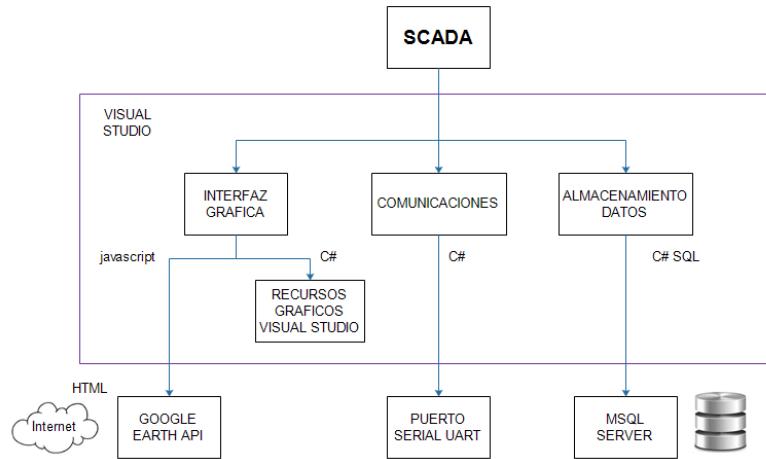


Figura 6. 1: Diagrama de diseño del sistema SCADA

6.2. Planificación de ruta

Para poder asignar una determinada ruta al Robot primero se requiere crear dicha ruta, para ello hay 2 formas:

- Manual: Selección de way points (WP)
- Automático: Selección de la posición inicial y final deseada.

Además es necesario calcular el área restringida para el desplazamiento del robot, ya sea porque sobre ésta se afecta al robot o se afecta a los cultivos.

6.2.1. Cálculo del área restringida

Es necesario realizar el cálculo del lugar geométrico ya que la formación de los sembríos de una columna no forma una línea recta sino que presentan cierta dispersión alrededor de una recta. El lugar geométrico que encierra a una línea de cultivos se muestra a continuación:

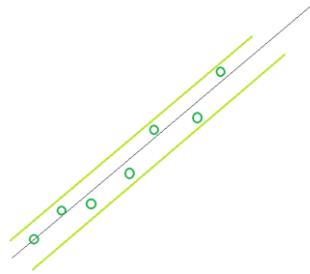


Figura 6.2 : Área restringida limitada por rectas

Para calcular el lugar geométrico se utilizan rectas que encierran completamente los cultivos de una columna, es por ello que cada columna posee su lugar geométrico. Además se podría decir que el conjunto de lugares geométricos de cada columna, representa el lugar geométrico de cultivos total. El procedimiento de cálculo del lugar geométrico de los cultivos es:

- Se calcula la recta que mejor se aproxima a las posiciones de los cultivos de una columna, mediante regresión lineal por mínimos cuadrados.

Se obtiene:

$$y = mr x + br \quad (6.1)$$

Siendo:

- mr: pendiente calculado por regresión a la mejor aproximación
- br: offset de la aproximación

- Se calcula una recta paralela a la recta calculada por regresión de tal manera que encierre al cultivo del más extremo izquierdo de la misma columna.

$$y = mr x + bi \quad (6.2)$$

- Se calcula otra recta paralela a la recta calculada por regresión de tal manera que encierre al cultivo del más extremo derecho de la misma columna.

$$y = mr x + bd \quad (6.3)$$

Finalmente se obtienen las siguientes rectas:

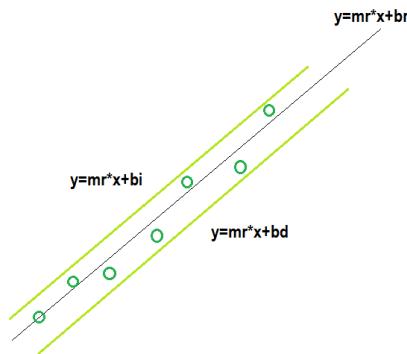


Figura 6.3 : Ecuaciones de las rectas que limitan el área restringida

Cabe recalcar que cada columna tendrá sus propias ecuaciones de rectas con pendientes y offset distintos. El procedimiento para evaluar las posiciones que forman la ruta es el siguiente:

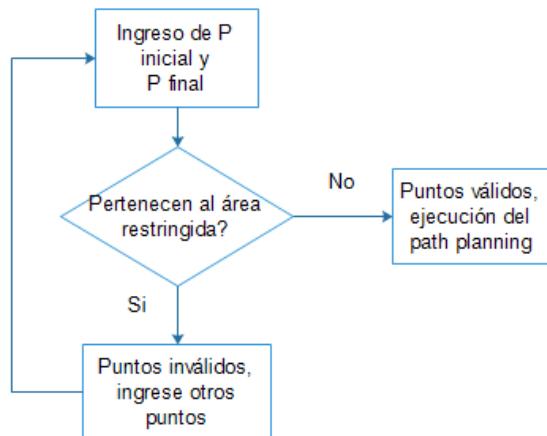


Figura 6.4 : Diagrama de flujo para la formación de ruta

6.2.2. Generación manual de la ruta

Se seleccionan los WP consecutivamente para formar una ruta. La ruta generada se representa como un arreglo o vector de coordenadas de los way points:

$$\text{ruta} = [WP_1, WP_2, WP_3, \dots, WP_n] \quad (6.4)$$

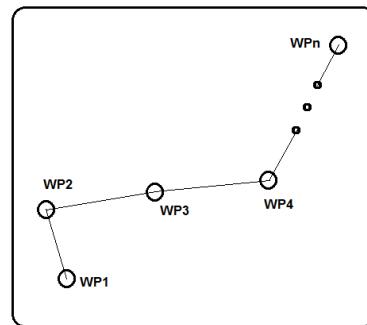


Figura 6. 5 : Ruta generada manualmente

6.2.3. Generación automática de la ruta – path planning

El path planning es la planificación de la trayectoria que tiene que seguir el robot, sobre un mapa estructurado, para poder llegar de un punto inicial a un punto final o seguir una ruta en especial.

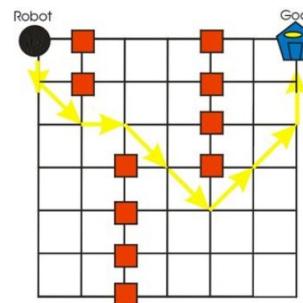


Figura 6. 6 : Inicio y meta en un path planning

En la presente tesis se considera un terreno agrícola con caminos permitidos ya definidos y conocidos.

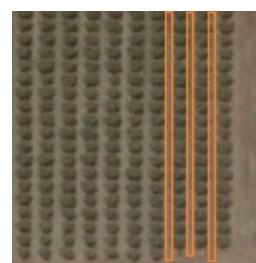


Figura 6. 7 : Caminos permitidos sobre el terreno, se muestran de color naranja

El mapa estructurado ideal es de la siguiente forma:

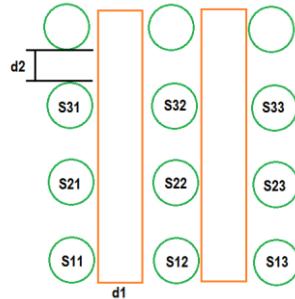


Figura 6. 8 : Mapa estructurado

Dónde:

- $S_{11}, S_{21}, S_{31}, \dots, S_{mn}$: Son los sembríos, pueden ser árboles, plantas, etc.
Poseen coordenadas de longitud y latitud conocidas.
- d_1 : Ancho del camino lineal
- d_2 : Separación entre sembríos.

Lo anterior conlleva a las siguientes consideraciones:

- a. Para la planificación de trayectoria se considera sólo los sembríos como obstáculos estáticos sobre el terreno.
- b. Los obstáculos dinámicos como personas u objetos se considerarán para la evasión o parada por parte del robot en la navegación.
- c. Los “way points” (**WP**) generados por el programa de path planning se basan en las posiciones fijas de los sembríos.
- d. Dependiendo de la distancia de separación entre los sembríos se puede realizar un tipo diferente de path planning ya que pueden haber terrenos cuya separación entre cada sembrío (d_2) sea de magnitud mayor que el

tamaño del robot, y esto ayudaría a que el robot pueda desplazarse libremente sobre 2 dimensiones.

- e. En el caso de que los caminos sean lineales y no exista un espacio considerable entre sembríos, el path planning creará una ruta que sea de extremo a extremo.

Cálculo de los corners:

Los corners son el conjunto de puntos disponibles para ser parte de la ruta planificada, es decir es el conjunto de puntos desde la cual se puede obtener un subconjunto de puntos llamado way points o WP.

$$C = \{C11, C12, C13, \dots, Cnm\} \quad (6.5)$$

$$WP \in C$$

Los corners se representan en el mapa como se presenta:

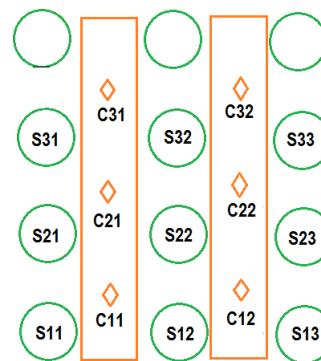


Figura 6. 9 : Corners en el mapa

Los corners se calculan como el promedio de las coordenadas de los 4 sembríos más cercanos, como a continuación se muestra.

$$C11 = \frac{S11 + S21 + S12 + S22}{4} \quad (6.6)$$

Cabe resaltar que las coordenadas de los sembríos son previamente medidas o estimadas con el Google Earth.

En términos generales las coordenadas de latitud y longitud de un corner generado por path planning es:

$$C_{mn_Lat} = \frac{Smn_Lat + S(m+1)n_Lat + Sm(n+1)_Lat + S(m+1)(n+1)_Lat}{4} \quad (6.7)$$

$$C_{mn_Lon} = \frac{Smn_Lon + S(m+1)n_Lon + Sm(n+1)_Lon + S(m+1)(n+1)_Lon}{4} \quad (6.8)$$

Aproximación de los puntos de ingreso a los Corners disponibles:

Una vez que se tienen las ecuaciones del área restringida y los Corners se puede realizar la aproximación de los puntos de ingreso.

Sea C: Conjunto de corners.

$$C = \{C_{11}, C_{12}, C_{13}, \dots, C_{nm}\} \quad (6.9)$$

P in: Punto de inicio

P fin: Punto de fin

D PC: Distancia del punto a un corner

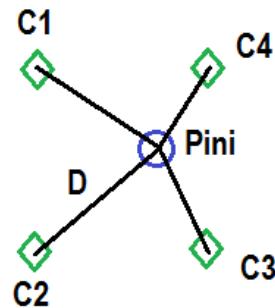


Figura 6. 10 : Búsqueda del corner más cercano al Pin

El corner que minimiza la distancia $D_{PC_{ini}}$ es el más cercano y el elegido como corner de inicio.

$$D_{PC_{ini}} = \sqrt{(Pini_x - C_x)^2 + (Pini_y - C_y)^2} \quad (6.10)$$

De igual manera el corner que minimiza la distancia $D_{PC_{fin}}$ es el más cercano y el elegido como corner de fin.

$$D_{PC_{fin}} = \sqrt{(P_{fin_x} - C_x)^2 + (P_{fin_y} - C_y)^2} \quad (6.11)$$

Cálculo de ruta:

Se calculan los way points necesarios para llegar del punto inicial al punto final.

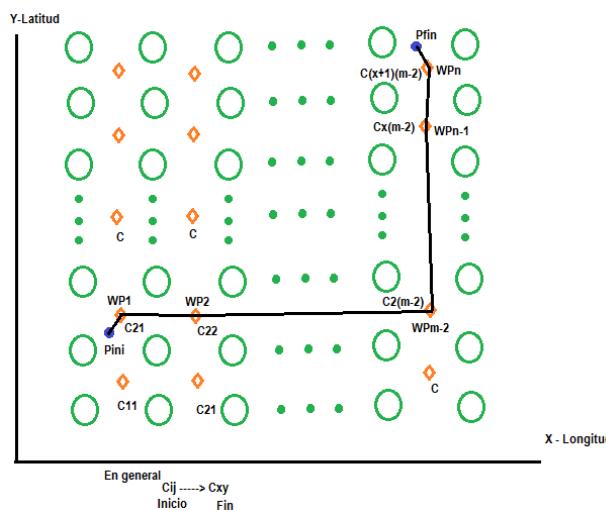


Figura 6. 11 : Ruta calculada

Sea C_{ij} el corner más cercano a P_{ini} y C_{xy} el corner más cercano a P_{fin} . El procedimiento para generar la ruta es el siguiente:

a. **Si $x > i$**

Aumentar i hasta que sea igual a x .

Almacenar en la memoria de way points (WP) los corners:

$$WP = \{C_{ij}, C(i+1)j, \dots, Cxj\}$$

Si $y > j$

Aumentar j hasta que sea igual a y .

$$WP = \{C_{ij}, C(i+1)j, \dots, Cxj, Cx(j+1), \dots, Cxy\}$$

Sino:

Disminuir j hasta que sea igual a y .

$$WP = \{C_{ij}, C(i+1)j, \dots, Cxj, Cx(j-1), \dots, Cxy\}$$

b. Si $x < i$

Aumentar i hasta que sea igual a x .

Almacenar en la memoria de way points (WP) los corners:

$$WP = \{C_{ij}, C(i-1)j, \dots, Cxj\}$$

Si $y > j$

Aumentar j hasta que sea igual a y .

$$WP = \{C_{ij}, C(i-1)j, \dots, Cxj, Cx(j+1), \dots, Cxy\}$$

Sino:

Disminuir j hasta que sea igual a y .

$$WP = \{C_{ij}, C(i-1)j, \dots, Cxj, Cx(j-1), \dots, Cxy\}$$

Pruebas en Visual Studio

Luego de haber sido validado el algoritmo se realizó la codificación del algoritmo en C#. Al crear la interfaz en Visual Studio se agregó el API de Google Maps, gracias a ello se pudo realizar mejores pruebas de los algoritmos y además dar una mejor calidad de representación a las rutas sobre mapas reales.

En la Figura 6. 12 se muestra el funcionamiento del algoritmo de creación del lugar geométrico de los sembríos y de los corners. El lugar geométrico de los sembríos se representa entre rectas de color rojo y los corners con círculos de color blanco.



Figura 6. 12 : Gráfica del área restringida y corners en C#

El cálculo de la ruta que se muestra de color blanco, representa la unión mediante rectas entre los WP y los puntos inicial y final. De tal manera que la ruta es:

$$\text{Ruta} = \{P_{ini}, WP_1, WP_2, \dots, WP_n, P_{fin}\} \quad (6.12)$$

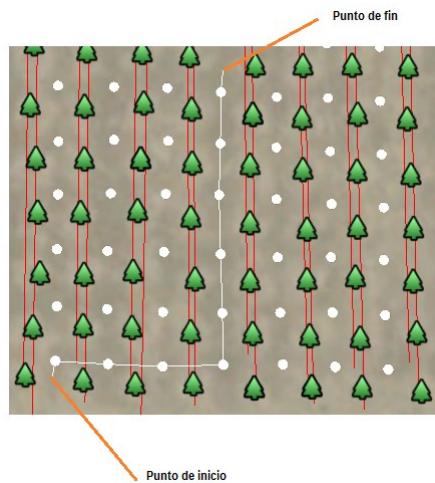


Figura 6. 13 : Ruta calculada por el path planning en C#

6.3. Asignación de tareas

Las tareas establecidas para el Robot son la navegación, fumigación, adquisición de datos y transporte de objetos. Éstas tareas necesitan algunos parámetros (ver Tabla 6.1) para ser asignadas.

Tarea	Parámetro
Navegación	Ruta deseada
Fumigación	Presión de trabajo
	Lado a fumigar (derecho, izquierdo, ambos)
Adquisición de datos	Variables de interés
	Tiempos de adquisición
Transporte de carga	Peso máximo

Tabla 6. 1 : Parámetros para la asignación de tareas al Robot

La interfaz gráfica diseñada para asignar las tareas y sus parámetros se presenta a continuación:

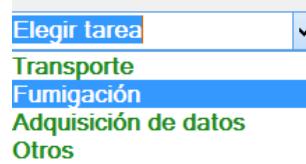


Figura 6. 14 : Selección de la tarea

- Navegación:

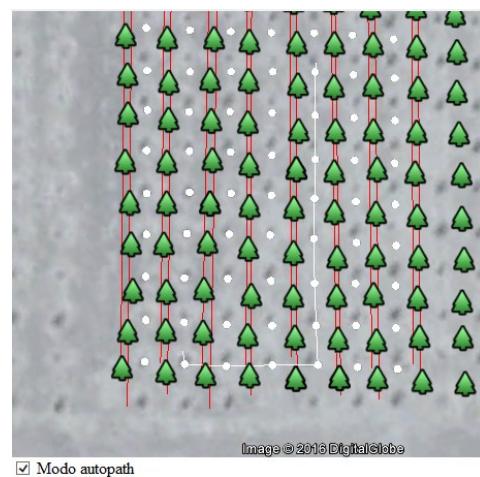


Figura 6. 15 : Asignación de la ruta deseada en modo autopath

- Fumigación:



Figura 6. 16 : Asignación de parámetros para la fumigación

- Adquisición de datos:



Figura 6. 17 : Asignación de parámetros para la adquisición de datos

Finalmente se envía la tarea, cabe resaltar que el transporte de objetos se puede realizar mientras no se realice la fumigación debido al espacio y peso del tanque.

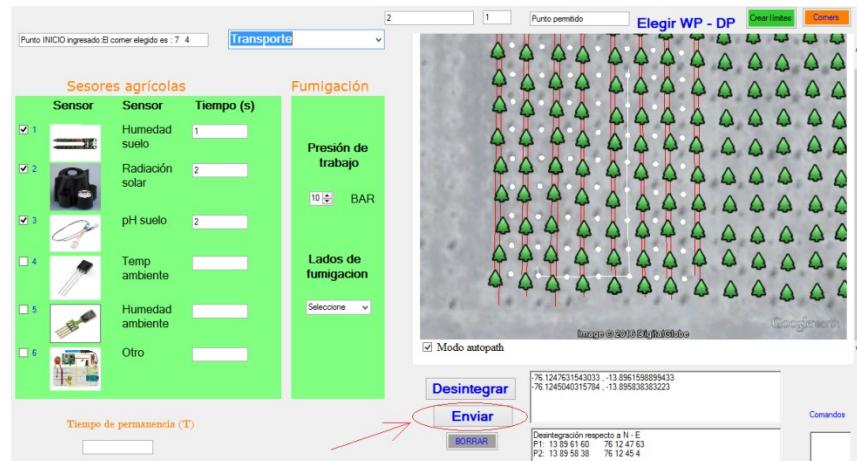


Figura 6. 18 : Asignación de parámetros para la adquisición de datos

6.4. Comandos remotos

Para la etapa de pruebas se desarrolló una ventana de la interfaz SCADA que permite enviar comandos hacia el Robot para controlar su movimiento y además los actuadores del sistema de fumigación y adquisición de datos.

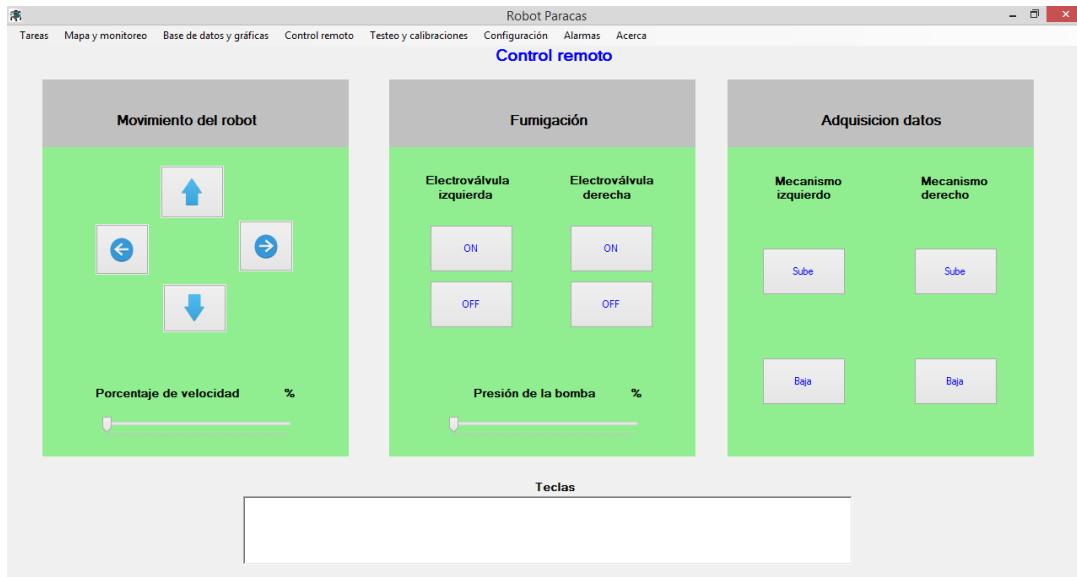


Figura 6. 19 : Ventana para el control remoto del robot

6.5. Monitoreo y supervisión

Para poder conocer el estado del robot y la tarea en ejecución, se requiere los datos de las variables del robot y las tareas. Las variables del robot se visualizan en tiempo real y son volátiles en cambio las variables de las tareas se almacenan en tablas de base de datos.

6.5.1. Monitoreo en Tiempo Real

Debido a que el operador necesita saber si el robot está funcionando correctamente, el SCADA recibe por el puerto Serial todos los datos necesarios que describen su estado como son:

Variables de navegación	Posición: Latitud y longitud
	Orientación: Yaw
	Presencia de obstáculos

	Velocidad lineal
	Voltaje de las baterías
Variables eléctricas	Corriente consumida
	%Energía disponible
	% Nivel del tanque
Variables de las tareas	% Ruta recorrida

Tabla 6. 2 Variables monitoreadas en Tiempo Real

La ventana de la interfaz SCADA desarrollada incluye la visualización de la posición del robot sobre el mapa de Google Earth, tal como se muestra en la Figura 6.20.

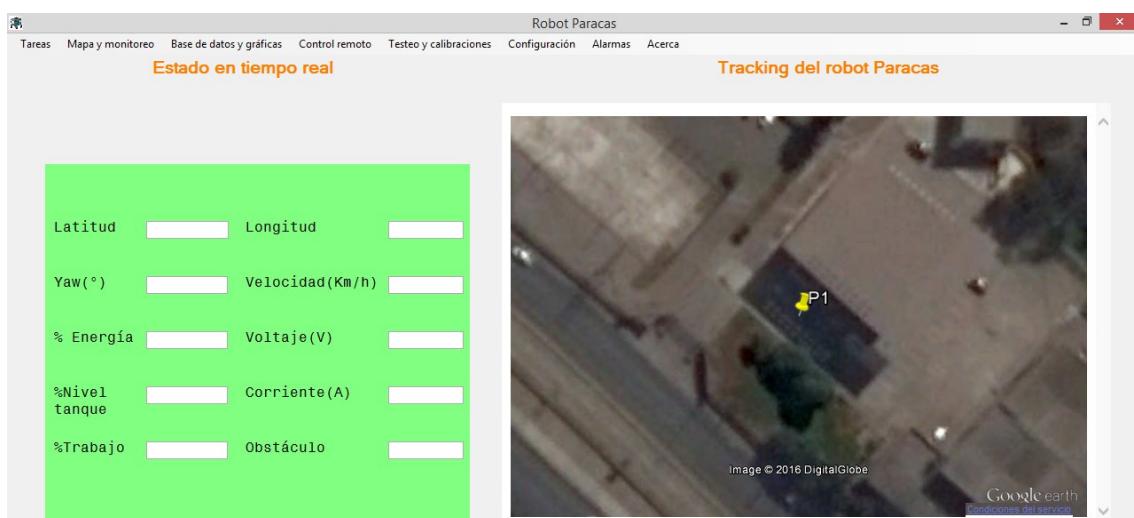


Figura 6. 20 : Ventana para el monitoreo en tiempo real

6.5.2. Almacenamiento en base de datos

Las variables de los sensores agrícolas con la geoposición y estampado de tiempo correspondiente se almacena en tablas dentro de una base de datos creada en MSQl Server. Se utiliza una tabla para cada variable (ver Figura 6.21).

The screenshot shows the Microsoft SQL Server Management Studio (SSMS) interface. On the left, the 'Explorador de objetos' (Object Explorer) tree view is expanded, showing several databases and their tables. A red box highlights the 'robot_paracas' database node, which contains a 'Tablas' (Tables) node with several entries, including 'dbo.humedad_ambiental'. On the right, a query window titled 'SQLQuery1.sql - TOSHIBA\ARTURO.master (TOSHIBA\ARTURO (58)) - Microsoft SQL Server Object Explorer' displays a T-SQL script. The script is a SELECT statement that retrieves top 1000 rows from the 'humedad_ambiental' table, ordered by 'fecha'. Below the script, the 'Resultados' (Results) pane shows the execution output, listing numerous rows of data with columns 'valor' and 'fecha'. At the bottom of the results pane, a green status bar indicates 'Consulta ejecutada correctamente.' (Query executed successfully.)

```

SQLQuery1.sql - TOSHIBA\ARTURO.master (TOSHIBA\ARTURO (58)) - Microsoft SQL Server Object Explorer

SELECT TOP 1000 [valor]
      ,[fecha]
     FROM [robot_paracas].[dbo].[humedad_ambiental]

valor fecha
992 41 31/05/2016 05:10:02
993 44 31/05/2016 05:10:02
994 44 31/05/2016 05:10:03
995 44 31/05/2016 05:10:04
996 44 31/05/2016 05:10:04
997 41 31/05/2016 05:10:05
998 41 31/05/2016 05:10:05
999 41 31/05/2016 05:10:06
1... 41 31/05/2016 05:10:06

```

Figura 6. 21 : Base de datos MSQl y tablas de las variables

La ventana de la interfaz SCADA para visualización temporal y almacenamiento de los datos de las variables en tablas dentro de la base de datos se muestra en la Figura 6.22.

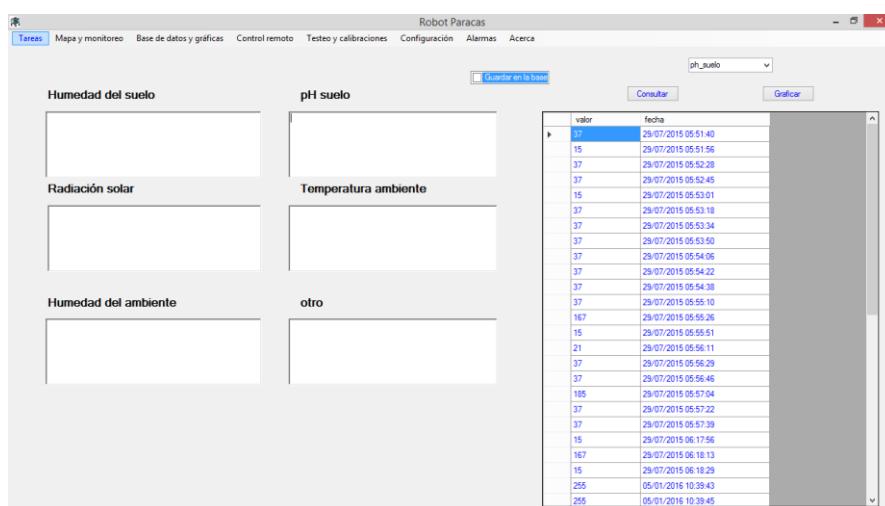


Figura 6. 22 : Interfaz SCADA base de datos y gráficas

Desde la interfaz se puede hacer consultas a la base de datos y graficar su tendencia en el tiempo. También el SCADA posee la ventana de alarmas para recibir alertas de posibles fallos.



Figura 6. 23 : Interfaz SCADA Alarms

Además se desarrolló la ventana de configuración para poder seleccionar el puerto serial del XBEE.

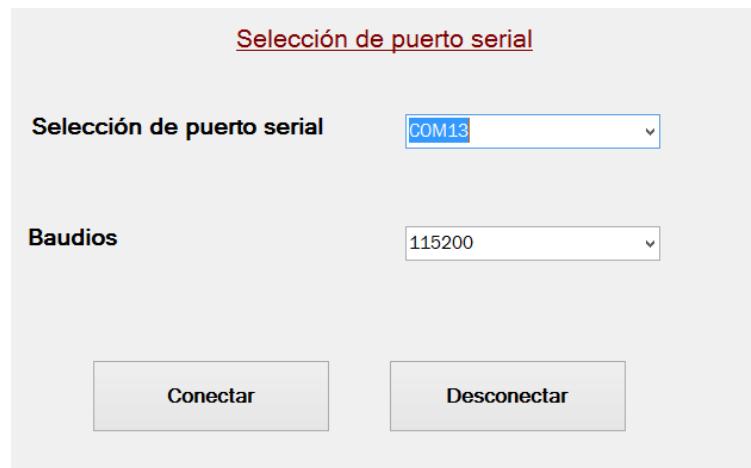


Figura 6. 24 : Selección del puerto serial

Capítulo 7

Implementación

7.1. Implementación mecánica

La estructura mecánica se compone de perfiles de tubo de acero inoxidable y soportes de fierro. El ensamblado se realizó mediante soldadura con electrodos para acero inoxidable 304 como se muestra en la Figura 7. 1.



Figura 7. 1: Ensamblado de la estructura del robot

También se soldó el soporte de los motores sobre los tubos cuadrados verticales.



Figura 7. 2 : Soporte vertical para fijar el motor

Luego se colocó rueda por rueda con su eje cruzando el soporte de fierro.

Posteriormente se asegura su porta rodaje (chumacera).



Figura 7. 3 : Ensamblado de las ruedas

La transmisión de movimiento se realizó mediante piñón y cadena para cada motor, se acopó un piñón de 11 dientes en el eje del motor y otro de 22 dientes en el eje de la rueda (para duplicar la fuerza), éste eje esta embocinado con las ruedas y en los extremos posee rodamientos como se muestra en la Figura 7.4.

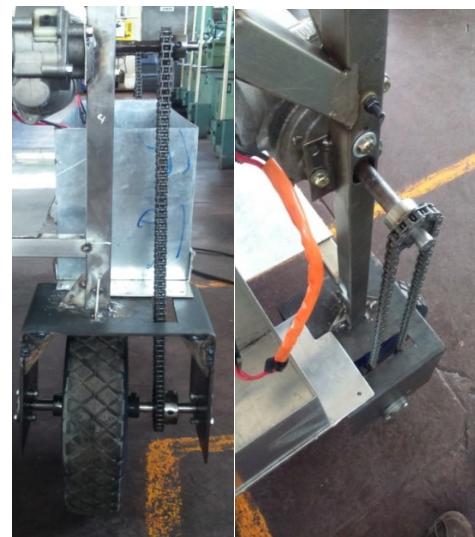


Figura 7. 4: Transmisión de movimiento del motor a la rueda

Luego se colocaron todos los motores (Figura 7.5) e hicieron pruebas de la correcta transmisión de movimiento por medio de las cadenas y piñones para ello se utilizó el circuito electrónico y un programa de test.



Figura 7. 5 : Pruebas de transmisión de movimiento

Posteriormente se colocó la porta baterías en cada lado del robot (ver Figura 7.6).



Figura 7. 6: Porta baterías

Para hacer posible el control del robot en lazo cerrado en velocidades, se colocó el soporte de los encoder a los motores.



Figura 7. 7 : Soporte para fijar el encoder al motor

Finalmente se obtuvo el siguiente sistema mecánico del robot:



Figura 7. 8 : Sistema mecánico del robot

Elementos adicionales para el sistema de fumigación:

Se colocó una plancha de aluminio (por ser inoxidable y de bajo peso) en posición horizontal para que sirva de soporte del tanque de agroquímico, tal como se muestra en la Figura 7.9.

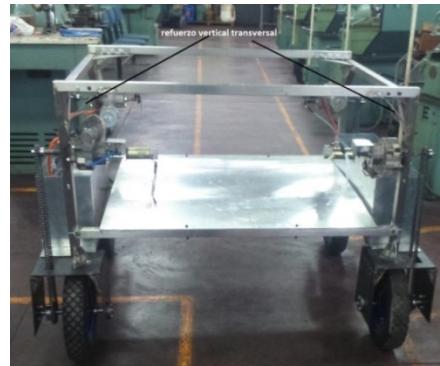


Figura 7.9 : Plancha de aluminio como base del tanque de agroquímico

Además se colocaron 2 barras de pulverización (1 en cada lado) sobre las cuales van instaladas las boquillas con sus mangueras. Son fáciles de montar y desmontar porque poseen una rosca interna y hay un perno soldado en su soporte.

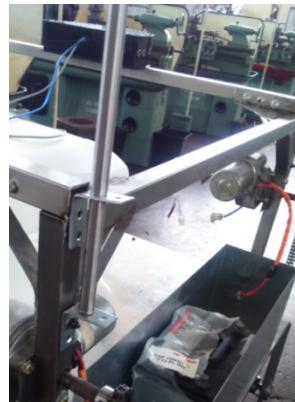


Figura 7. 10 : Barras de pulverización

Además durante esta parte de pruebas surgió la necesidad de agregar refuerzo a la estructura debido a que se producían esfuerzos en los giros del robot debido a su peso, por ello se soldó refuerzos horizontales en las esquinas como se muestra:



Figura 7. 11 : Refuerzo triangular horizontal en las esquinas de la estructura

7.2. Implementación sistema hidráulico

Bomba DC:

Debido a que no existe en el mercado una bomba eléctrica DC de la potencia requerida (de acuerdo a los cálculos anteriores, resultó ser mayor a 176 W), se adaptó un motor DC de 300W y 1800RPM con el impulsor de una bomba monofásica comercial de 1/2HP (ver Figura 7.12).



Figura 7. 12 : Motor DC y motobomba monofásica desarmada

El acoplamiento resultó exitoso, además se maquinó un bloque de PVC para convertirlo en soporte de la bomba como se muestra en la Figura 7.13.



Figura 7. 13 : Acoplamiento del motor DC con el propulsor

Además la bomba se unió con el tanque mediante una tubería con una llave manual (ver Figura 7.14) para mayor maniobrabilidad.



Figura 7. 14 : Motobomba conectada al tanque de agroquímico

Círculo hidráulico:

El circuito se compone de 3 válvulas manuales y 2 solenoides para activar el flujo de líquido, además como parte de la regulación de la presión se encuentra la válvula de alivio y el transmisor de presión. Adicionalmente se utilizó una manguera de

capacidad hasta 50Bar y elementos adicionales como abrazaderas, etc. Las válvulas solenoides y el transmisor de presión fueron cableados hacia el tablero. De forma similar se realizó la conexión de la salida de la bomba al circuito hidráulico y también la conexión de retorno de fluido presurizado al tanque desde la salida de la válvula de alivio. En la Figura 7.15 se muestra la unión física de los componentes mencionados.



Figura 7. 15 : Unión de componentes del circuito hidráulico

Boquillas

Las boquillas se instalaron sobre las barras mediante sólo cintillos (Figura 7.16) ya que se observó que se sujetaban lo suficiente y es posible poder girarlas para darle una dirección de preferencia.



Figura 7. 16 : Boquilla sujetada a la barra de pulverización

A continuación se muestra la boquilla pulverizando el líquido.



Figura 7. 17 : Boquilla pulverizando el fluido

El ángulo de cono de salida depende de la presión que llegue a la boquilla.

El módulo de pulverización se instaló en la parte posterior del Robot (ver Figura 7.18), de tal manera que haya menos probabilidad de fugas que afecten al sistema electrónico y además que se pueden manipular más fácil y rápidamente las válvulas manuales así como también posibles futuros ajustes.



Figura 7. 18 : Módulo de pulverización en la parte posterior del Robot

7.3. Implementación eléctrica – electrónica

7.3.1. Controladores

Los controladores implementados sobre la plataforma de Arduino, están divididos de acuerdo a las funciones específicas que cumple cada uno tal como se muestra en la Tabla 7.1.

C1	Controlador de navegación
C2	Controlador de tareas
C3	Controlador de velocidad de los motores

Tabla 7. 1 : Controladores dentro del robot



Figura 7. 19 : Tarjeta electrónicas de los controladores y borneras de alimentación

7.3.2. Driver de potencia de los motores

Los driver de potencia son del tipo Puente H y utilizan transistores Mosfet de canal N IRF1404 debido al gran requerimiento de corriente por parte de los motores utilizados. Además posee protección ante sobrecargas en el motor, gracias a que incorpora 1 sensor de corriente ACS758 – 100B.

La implementación electrónica sobre una placa se realizó como se muestra en la Figura 7.21, además se agregó disipadores de calor a los transistores para mejorar la eficiencia. También fue necesario incluir un elevador de voltaje XL6009 para saturar a los transistores mosfet .

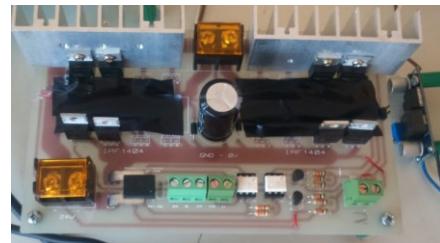


Figura 7. 20 : Driver de potencia implementado

Se realizó pruebas con 12V y 24V además de incluir carga al motor.



Figura 7. 21 : Pruebas del driver de potencia

Como elemento de seguridad redundante ante sobre corrientes se añadió una llave termo magnética y para la actuación rápida ante fallas se colocó 2 Switchs de parada de emergencia que activan a un contactor que abre o cierra el circuito con la alimentación.

Prueba de todos los motores:

Para el conexionado entre las borneras y drivers se utiliza cable #10 y para la conexión de los drivers con los motores se utiliza cable vulcanizado multipolar de 2 polos #10, de ésta manera se obtiene un mayor grado de robustez de las conexiones.

Se colocaron terminales hembras y machos para todos los componentes (motores, encoder, baterías, electroválvulas) (ver Figura 7.23), de manera que se pueda realizar un conexiónado rápido y seguro.

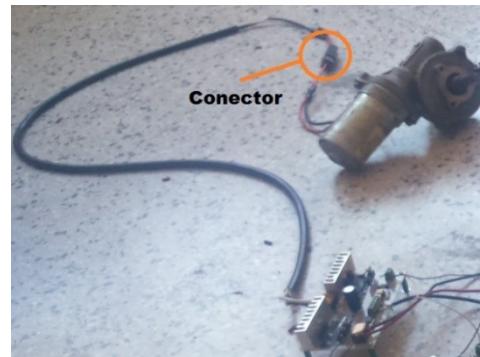


Figura 7. 22 : Conexión entre driver de potencia y motor

En lazo abierto:

Se realizó prueba conjunta de todos los motores con sus drivers en lazo abierto, para ello desde el controlador C3 se le envió una secuencia de PWM con dutty variable.



Figura 7. 23 : Pruebas en lazo abierto de los motores

En lazo cerrado:

Para controlar al motor en lazo cerrado se incorporó un encoder sobre el eje lateral libre de los motores tal como se muestra en la Figura 7.24.



Figura 7. 24 : Motor con encoder incorporado

Pruebas:

Utilizando el controlador Arduino Mega R3 se realizó pruebas en lazo cerrado utilizando el encoder.



Figura 7. 25 : Pruebas del control de velocidad en lazo cerrado del motor

Los datos adquiridos, durante las pruebas, en el monitor serial se muestra a continuación:

30	30.0000	29.0000	20	20.0000	20.0000
30	30.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	29.0000	20	20.0000	20.0000
30	29.0000	31.0000	20	22.0000	20.0000
30	29.0000	31.0000	20	22.0000	20.0000
30	29.0000	31.0000	20	22.0000	20.0000

Figura 7. 26 : Resultados del control en lazo cerrado para 2 motores en simultáneo

Tablero de control:

Se acomodaron las tarjetas de control, drivers de potencia y otros componentes sobre una tabla de madera para poder fijarlas y que estén bien posicionadas.



Figura 7. 27 : Prueba del circuito interconectado sobre la tabla de madera

Para proteger a los circuitos, se colocó dentro de un tablero de control. Además se utilizaron PGs para fijar bien los cables por los agujeros realizados al tablero. Los cables que salen son hacia motores - encoder, baterías, Switchs parada de emergencia, válvulas solenoide y sensores.

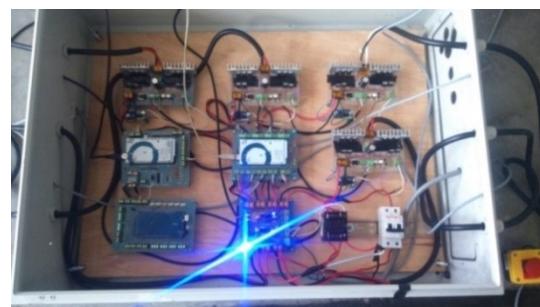


Figura 7. 28 : Tablero eléctrico con todo el circuito incorporado

En la parte frontal del tablero de control se colocó LEDs indicadores, 1 switch de encendido del robot y pulsadores como “reset” de los microcontroladores. El switch de encendido funciona por seguridad con una llave propia.

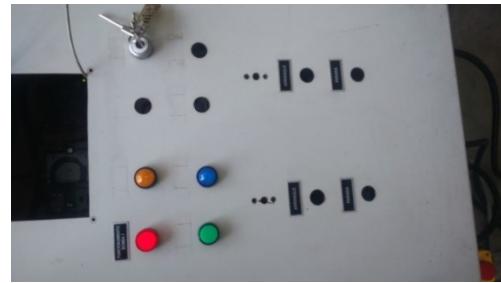


Figura 7. 29 : Parte frontal del tablero eléctrico

Finalmente se realizó las siguientes pruebas para garantizar el correcto funcionamiento del tablero y los controladores:

- a. Comunicación bidireccional con la Laptop por medio del XBEE, la transmisión de datos se realiza desde C2-> Laptop y desde Laptop->C1, se probó con un terminal Serial y también con la interfaz SCADA realizada.
- b. Control de velocidades por medio del C3, lectura de los sensores de corriente. Desde la Laptop se le envió los setpoint de velocidades y se controló de una manera exitosa.
- c. Operatividad del switch de encendido y los de parada de emergencia.

Se puede observar en la Figura 7.31 que la distribución es simétrica y eficiente para una instalación rápida y segura.



Figura 7. 30 : Prueba del control de los motores remotamente desde la laptop

7.3.3. Sensores del sistema de navegación

Instalación del GPS:

Se colocó la antena del GPS en la parte superior de la barra de madera colocada en el parte frontal del robot, de tal manera que se encuentre más alejada de las fuentes de ruido y tenga mejor línea de vista hacia los satélites Figura 7.32. Por otro lado el circuito del GPS se encuentra dentro del tablero ya que éste no requiere estar al aire libre porque solo procesa la información.



Figura 7. 31 : Posición de la antena del GPS

El controlador C3 lee la trama SBP que entrega el GPS por el puerto serial, para mayor velocidad se utilizan 115200 BAUDIOS.

Instalación de los encoders:

Los encoders se colocan en los ejes laterales de los motores tal como se presenta en la Figura 7.33.

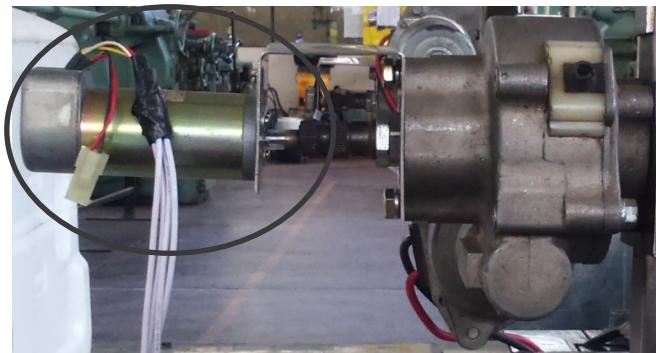


Figura 7.32 : Encoder acoplado al motor

Instalación del IMU:

Luego de analizar los 3 aspectos de para posicionar el IMU, se realizó la instalación en la posición mostrada en la Figura 7.34. Luego se realizó el cableado hacia el tablero de control.



Figura 7.33 : Ubicación del IMU en el robot

7.3.4. Sistema de fumigación

El proceso de fumigación se logra gracias al controlador C2, el cual presuriza la línea que sale del tanque (de 5 a 10bar) activando la motobomba DC (por PWM) y leyendo el valor de la presión mediante el transmisor de presión, además si detecta los sembríos (pej. árboles) mediante los sensores ultrasónicos laterales entonces abre

la válvula solenoide correspondiente utilizando para ello un módulo de relés ya que operan con 24V, llegando así el fluido presurizado que será convertido en pequeñas partículas (pulverización) mediante las boquillas.

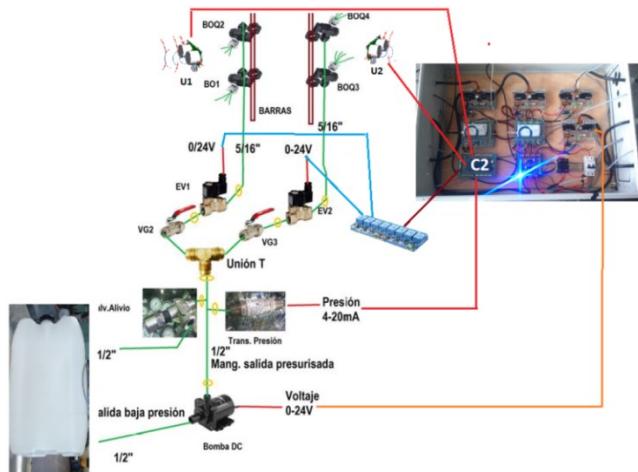


Figura 7. 34 : Implementación del sistema de fumigación

Por otro lado cabe recalcar que para la medición del transmisor de presión, por parte del microcontrolador C2, se requirió convertir la señal de corriente que entrega a voltaje para ello se utilizó un sensor de corriente ACS758 – 05B.

Sensor de detección de cultivos

Para la detección de cultivos a fumigar, se instaló 1 sensor de ultrasonido en cada lado (derecha - izquierda).

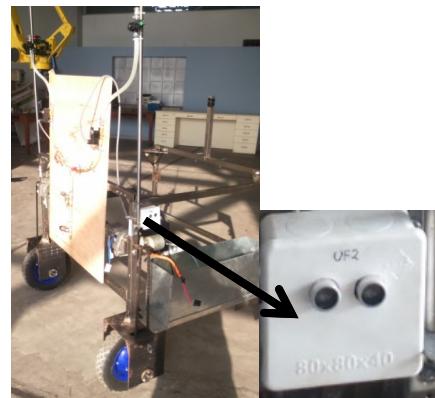


Figura 7. 35 : Sensor ultrasónico para detección lateral de cultivos

En la implementación, los sensores ultrasónicos mencionados se conectan al controlador C2. De acuerdo a la distancia que mida el sensor de ultrasonido, C2 abrirá o cerrará la válvula solenoide del lado correspondiente.

7.3.5. Sistema de adquisición de datos

Se elaboró una placa electrónica (módulo de sensores, ver Figura 7.39) para unir las señales de todos los sensores, de manera que con un solo cable multipolar se comunique al modulo de sensores con el controlador C2. Dicho módulo se coloca en la parte exterior del tablero de control, de tal manera que los sensores ambientales como son el de radiación solar, humedad y temperatura del ambiente tomen buenas mediciones. Además se deja libre la conexión de los sensores de humedad (ver Figura 7. 37) y pH del suelo para luego extender su conexión hacia las puntas resistivas que harán contacto con el suelo del terreno a operar.



Figura 7.36 : Sensor Humedad del suelo, tipo resistivo

El modulo de sensores posee las interconexiones como se muestra en la Figura 7.38 de modo que se logra simplificar el cableado hacia C2.

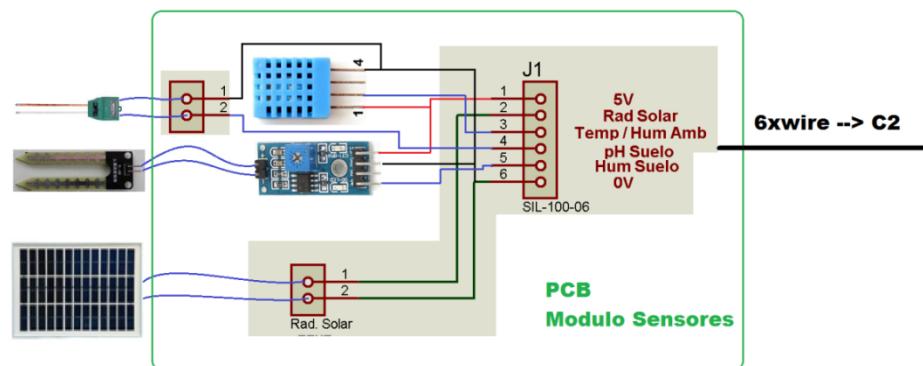


Figura 7.37 : Integración de las conexiones de los sensores

La integración de los sensores en el módulo resultó como se muestra a continuación:



Figura 7.38 : Módulo de sensores

7.3.6. Comunicación del robot con el SCADA

La comunicación entre el Robot y el SCADA es bidireccional, para ello se utilizan módulos transceptores XBEE PRO S2. La posición de los transceptores se muestra en la Figura 7.39.

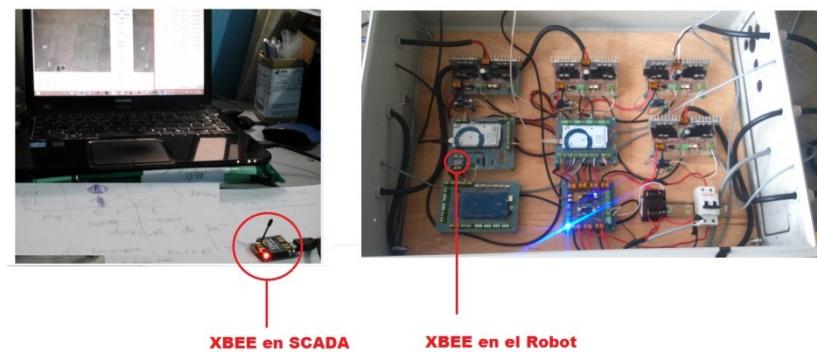


Figura 7. 39 : Disposición de los módulos XBEE PRO S2

7.4. Costo de implementación

El costo de los materiales y servicio de mecanizado para la implementación de la presente tesis se detalla en la Tabla 7.2.

Grupo	Elemento	Modelo	Costo Unit (S/.)	Cantidad	Costo (S.)
Sensores	IMU	GY 85 9dof sensor	75	1	75
	GPS	Piksi	3600	1	3600
	Ultrasónicos	SRF05	20	10	200
	Corriente 50A	ACS714	45	4	180
	Corriente 100A	ACS758LC B	50	2	100
	Sensor temperatura y humedad	DHT11	62	1	62
	Sensor pH	3 In 1 PH Tester Soil	49	1	49

	Sensor humedad del suelo	Moisture sensor module	20	3	60
	Sensor de radiación solar	Solar radiation sensor	149	1	149
Actuadores	Motores DC	24V , 200 RPM 350W	600	4	2400
	Motobomba	1 HP, necesario por el propulsor	250	1	250
	Servomotores	MG995	50	5	250
	Motores con encoder	DC 12V, 100ppv	40	2	80
	Válvulas solenoide		150	2	300
Controladores	Arduino MEGA	Arduino MEGA 2560	75	4	300
	Arduino UNO	Arduino UNO R3	50	1	50
Drivers de potencia	Transistor Mosfet Channel N	IRF1404	300	1	300
	Transistor Mosfet Channel P	irf4905	300	1	300
Comunicaciones	Transceiver RF	XBEE PRO S2	148	2	296
Tarjetas electrónicas	PCBs para control, sensado y potencia.	Fibra vidrio 20cm x 20cm	50	5	250
Complementos electrónicos	Reguladores DC DC	Buck Converter with LM2596	10	3	30
	Buzzer, leds, capacitores, resistores y otros		100	1	100
Baterías	Baterías de Plomo, libre de mantenimiento	Etna 90 Ah	375	2	750
Periféricos	Interruptor parada emergencia		10	3	30
	Módulo de LEDs indicadores		20	1	20
	Salidas tipo Relé	8 canales	25	1	25
Cables y conectores	Cables para los actuadores, sensores, periféricos, baterías.		200	1	200

	Conectores y terminales.				
Tablero eléctrico	Tablero metálico		300	1	300
Estructura	Planchas de aluminio		200	2	400
	Tubos cuadrados de acero	$\frac{3}{4}'' \times 2\text{mm}$	150	2	300
	Tornillos , Remaches , etc.		50	1	50
Transmisión de movimiento	Acoplamientos		50	4	200
	Cadenas y piñones		100	4	400
	Ruedas		65	4	260
Accesorios de fumigación	Barras fumigación, mangueras y tubos de conexión.		600	1	600
	Válvula manual, transmisor de presión, regulador		500	1	500
	Boquillas y portaboquillas	ATR80	45	4	180
	Tanque para el agroquímico		100	1	100
Servicio de mecanizado	Corte, Soldadura, Ensamblado		1000	1	1000

Tabla 7. 2 : Lista de costo de la implementación

El costo total resultante de la implementación es de 14 696 Soles.

Capítulo 8

Pruebas y resultados

8.1. Pruebas de comunicaciones del robot con el SCADA

Se realizó las pruebas de comunicación entre los microcontroladores que posee el robot y el SCADA, para enviar los comandos - tareas al robot y recibir en el SCADA los datos de monitoreo.



Figura 8. 1: Pruebas de comunicaciones entre el robot y SCADA

8.2. Prueba de movimientos del robot

Se realizó pruebas enviando las instrucciones desde la interfaz SCADA en modo de control remoto y envío de referencias para el lazo cerrado. De ésta manera se verificó que el sistema electrónico no tengas fallas en hardware ni software.

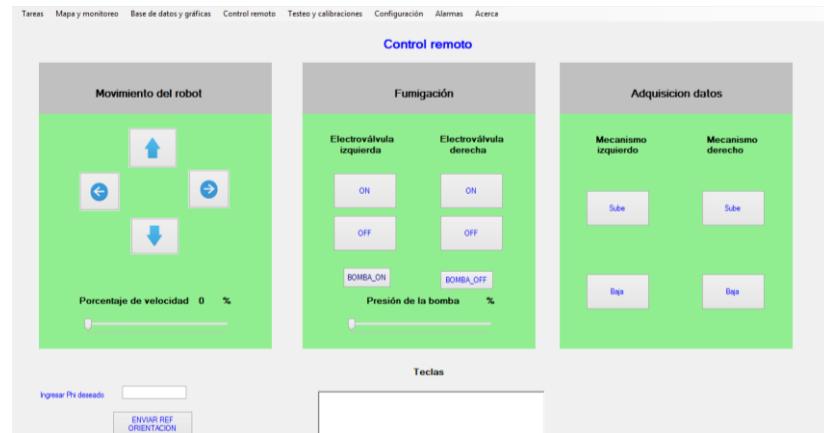


Figura 8. 2 : Ventana del SCADA para pruebas

Una vez recibida el comando-tarea por parte del Robot, éste ejecutó adecuadamente el movimiento.



Figura 8. 3 : Pruebas de movimiento del robot en lazo abierto

8.2.1. Prueba de control de velocidad del motor DC

Se realizó la prueba de control de velocidad del motor asignando la referencia al controlador desde la computadora, obteniendo los siguientes resultados:

Prueba para un escalón de 20 RPM:

Se obtiene un respuesta rápida con error estacionario “0” ya que posee se usa integrador en el controlador, además se puede observar la influencia del ruido sobre la respuesta Figura 8.4.

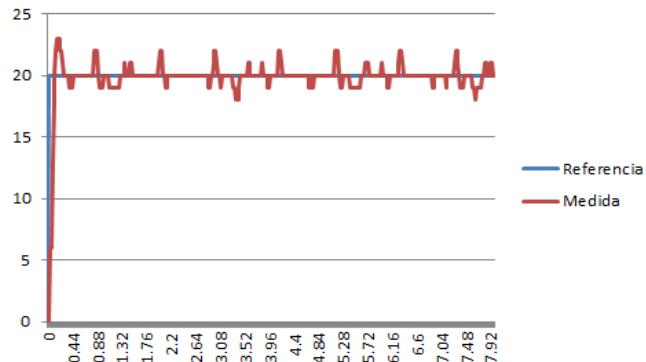


Figura 8. 4 : Respuesta del controlador para referencia de 20RPM

Prueba para para una secuencia de pulsos de amplitud 20 RPM

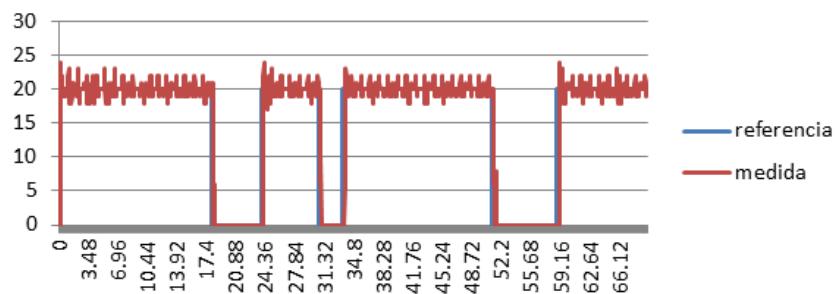


Figura 8. 5 : Respuesta del controlador para una secuencia de pulsos a 20RPM



Figura 8. 6 : Pruebas de movimiento del robot en lazo cerrado

8.3. Prueba de control de orientación del robot

Se le asignó al robot distintas orientaciones desde la interfaz SCADA.

A continuación se presenta las pruebas de control de orientación cada 90° respecto a los ejes dibujados en el suelo:

- **Con velocidad lineal $VI = 0$**

En éste modo el robot solo posee velocidad angular. Cuando llega a la orientación deseada se detiene.



Figura 8. 7 : Pruebas de control de orientación con $VI = 0$

- **Con velocidad lineal $VI = 0.3\text{m/s}$**

En éste modo el robot posee velocidad angular y velocidad lineal. Cuando grá tambiéen se desplaza levemente por la velocidad lineal que posee. Además cuando llega a la orientación deseada, su velocidad angular se hace cero y su velocidad lineal es $VI = 0.3\text{m/s}$ de tal manera que da la impresión de que se mueve en línea recta.





Figura 8. 8 : Pruebas de control de orientación con $Vl=0.3m/s$

- **Gráficas del control de orientación**

Se muestra a continuación el gráfico de la evolución del ángulo de orientación y las velocidades angulares necesarias de cada motor lateral:

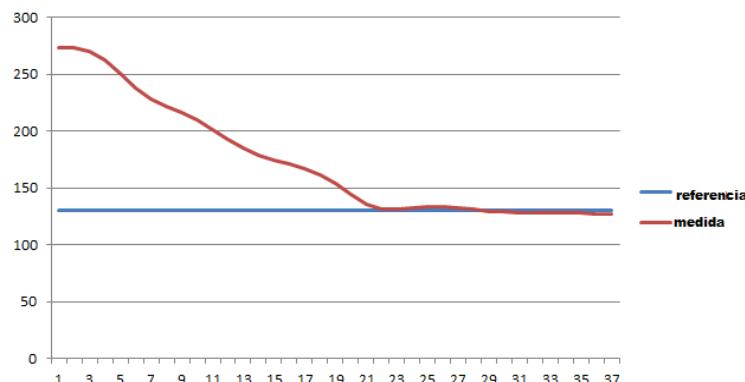


Figura 8. 9 : Control de la orientación para referencia 130° (Respecto al Norte)

El paso de tiempo es de 0.1seg para la tomada de datos:

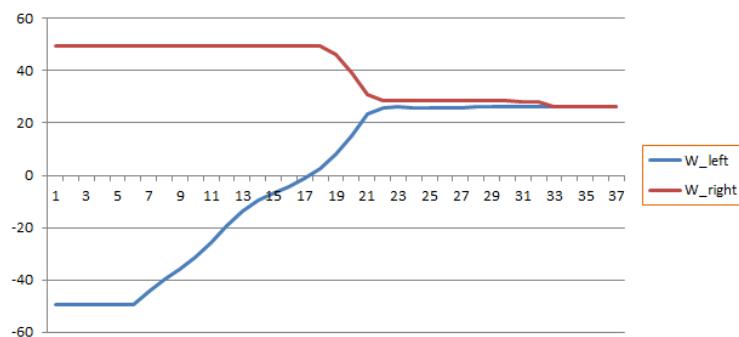


Figura 8. 10 : Velocidades angulares (RPM) de cada motor calculadas por el control de orientación

8.4. Pruebas de navegación autónoma

Debido a que se trabaja con el GPS RTK para mejorar la exactitud de posicionamiento, se requiere colocar uno de los GPS (el que corrige) en la estación fija o base (ver Figura 8.11).



Figura 8. 11: Estación base o fija, desde donde se envía tareas al robot

Las coordenadas de la estación fija se obtienen a partir de una comparación visual de la posición real y el mapa de Google Earth (ver Figura 8.12).



Figura 8. 12 : Visualización de la estación fija en el Mapa de GE



Figura 8. 13 : Coordenada de la estación fija

8.4.1. Pruebas del sistema de control y algoritmo

Se realiza una prueba hacia un punto objetivo:

En ésta prueba como sólo se tiene un punto objetivo para visualizar el control de navegación, se toma el punto objetivo como punto final:

$$P_f = (\text{Longitud}_f, \text{Latitud}_f) \quad (8.1)$$

La posición de partida es la posición actual que entrega el GPS RTK cada cierto tiempo.

$$P_a = (\text{Longitud}_a, \text{Latitud}_a) \quad (8.2)$$

Dónde:

Latitud_f : latitud del punto objetivo

Latitud_a : latitud actual

Longitud_f : longitud del punto objetivo

Longitud_a : longitud actual

Como ya se indicó anteriormente el controlador de navegación se compone internamente de un controlador de orientación y 2 controladores de velocidad (1 para cada motor). El control de orientación posee una referencia de ángulo que se calcula con la posición actual y la posición objetivo.

En la prueba realizada el control de orientación entregó el siguiente resultado.

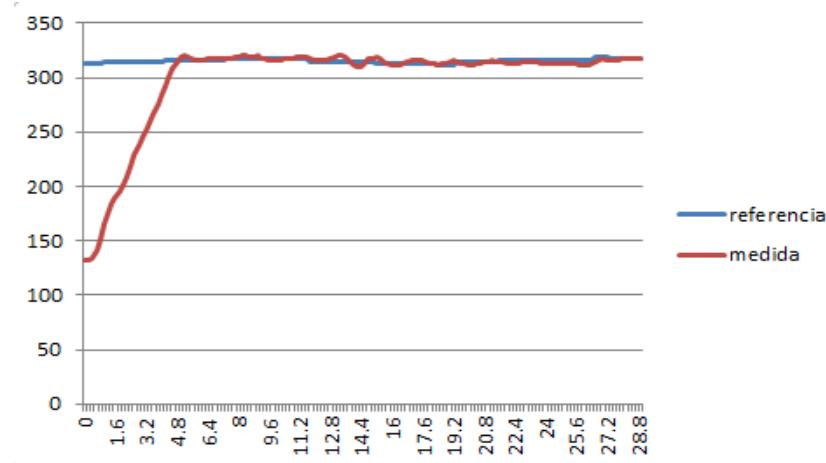


Figura 8.14 : Orientación controlada para llegar al punto objetivo

Además el controlador de orientación entrega las velocidades que debe de tener cada motor y por ende deben de ser controladas por sus respectivos controladores de velocidad:

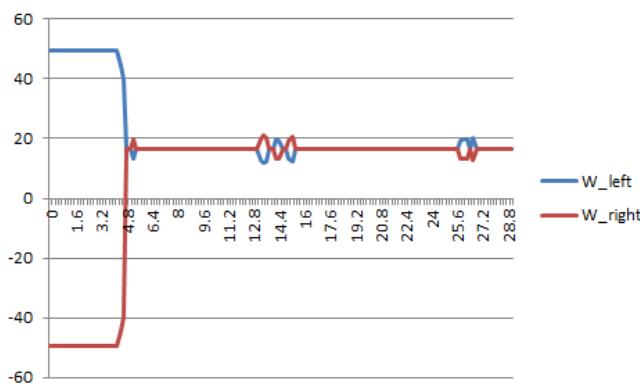


Figura 8.15 : Velocidades angulares de cada motor calculadas para la navegación

Se puede apreciar que las velocidades en cada lado convergen a un mismo valor ya que el robot debe desplazarse linealmente hacia su objetivo con una velocidad que también es calculada por el controlador de navegación. Se puede apreciar pequeñas perturbaciones en las velocidades lineales, ésto se debe al ruido y perturbaciones mientras el robot se desplaza hacia su punto objetivo.

Error de posición

Mientras el robot se va desplazando hacia el punto objetivo, el radio que los une va disminuyendo, tal como se muestra en la Figura 8.13.



Figura 8. 16 : Error de posición durante la navegación

8.4.2. Pruebas utilizando monitoreo remoto

Se hicieron pruebas para llegar hacia 1 solo punto y para seguir una ruta (conjunto de puntos).

- **Prueba hacia 1 punto**

Punto asignado: Ubicado entre el “Laboratorio de Mecatrónica” y el “Instituto de Motores”.

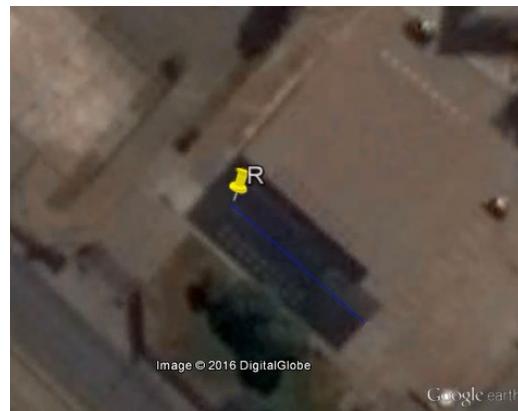


Figura 8. 17 : Asignando el punto objetivo desde la interfaz SCADA

Etapa de orientación del robot hacia su objetivo:



Figura 8. 18 : Orientación y desplazamiento del robot hacia el punto objetivo

El controlador de la navegación(C1) entrega información(ver Figura 8.16) de las variables utilizadas para monitorear el buen funcionamiento del control. A partir de ésta información se puede ver cómo cambian las variables mientras el robot se dirige hacia su objetivo. El orden en que se presenta la información es el siguiente:

[orientación_{referencia}, orientación_{medida}, error_{orientacion}, integral_{error}, W_{angular}, W_{left}, W_{right}, error_{posición}]

Las unidades de la orientación es en grados sexagesimales y las velocidades angulares W en rad/s.

```

130 292.82 -162.82 -10.41 -17.32 -5.20 5.20 0.0000000729
130 287.25 -157.25 -13.55 -17.08 -5.20 5.20 0.0000000729
130 279.18 -149.18 -16.54 -16.57 -5.20 5.20 0.0000000729
130 270.94 -140.94 -19.35 -16.03 -5.20 5.20 0.0000000729
130 252.37 -132.37 -22.00 -15.44 -5.20 5.20 0.0000000729
130 252.70 -122.70 -24.46 -14.72 -5.20 5.20 0.0000000729
129 245.04 -116.04 -26.78 -14.28 -5.20 5.20 0.0000000719
129 237.16 -108.16 -28.94 -13.71 -5.20 5.20 0.0000000719
129 230.28 -101.28 -30.97 -13.22 -5.20 5.20 0.0000000719
129 223.38 -94.38 -32.85 -12.72 -5.20 5.20 0.0000000719
130 216.05 -86.05 -34.57 -12.06 -5.20 5.20 0.0000000682
130 207.10 -77.10 -36.12 -11.32 -5.20 5.20 0.0000000682
130 200.22 -70.22 -37.52 -10.77 -5.20 5.20 0.0000000682
130 191.64 -61.64 -38.75 -10.04 -5.20 5.20 0.0000000682
130 183.53 -53.53 -39.82 -9.34 -5.20 5.20 0.0000000678
130 177.79 -47.79 -40.78 -8.86 -5.20 5.20 0.0000000678
130 169.52 -39.52 -41.57 -8.11 -5.20 5.20 0.0000000678
130 162.06 -32.06 -42.21 -7.43 -5.20 5.20 0.0000000678
130 154.97 -24.97 -42.71 -6.77 -5.08 5.08 0.0000000682
130 147.22 -17.22 -43.05 -6.03 -4.52 4.52 0.0000000682
130 140.90 -10.90 -43.27 -5.42 -4.06 4.06 0.0000000682
130 133.39 -3.39 0.00 -4.67 1.71 1.71 0.0000000682
130 127.43 2.57 0.00 0.26 1.71 1.71 0.0000000682
131 123.50 7.50 0.15 0.77 2.29 1.14 0.0000000652
131 122.76 8.24 0.31 0.86 0.64 -0.64 0.0000000652
131 124.10 6.90 0.45 0.74 2.27 1.16 0.0000000652
131 126.95 4.05 0.53 0.46 2.06 1.37 0.0000000652
131 130.24 0.70 0.00 0.13 1.71 1.71 0.0000000652
131 132.79 -1.79 0.00 -0.18 1.71 1.71 0.0000000652
131 133.17 -2.17 0.00 -0.22 1.71 1.71 0.0000000652
131 131.52 -0.52 0.00 -0.05 1.71 1.71 0.0000000649
131 130.42 0.58 0.00 0.06 1.71 1.71 0.0000000649
131 130.04 0.96 0.00 0.10 1.71 1.71 0.0000000649
131 129.50 1.50 0.00 0.15 1.71 1.71 0.0000000649
130 130.16 -0.16 0.00 -0.02 1.71 1.71 0.0000000636

```

Figura 8. 19 : Evolución de las variables del control en la etapa de orientación

A partir de la información de la etapa de orientación se puede observar que el robot se orienta haciendo una diferencia de velocidades laterales y luego se desplaza con la misma velocidad mientras tenga la orientación deseada, si hubiera alguna perturbación externa el robot recuperará su orientación. Además se puede apreciar que el error de posición va disminuyendo mientras el robot se desplaza.

Etapa de llegada a la posición objetivo:



Figura 8. 20 : El robot llegando a su posición objetivo

El monitoreo de las variables del control se pueden visualizar en la siguiente imagen:

```

151 148.10 2.90 0.00 0.32 1.71 1.71 0.0000000023
151 150.28 0.72 0.00 0.07 1.71 1.71 0.0000000023
149 151.37 -2.37 0.00 -0.24 1.71 1.71 0.0000000020
149 151.45 -2.45 0.00 -0.25 1.71 1.71 0.0000000020
149 151.00 -2.00 0.00 -0.20 1.71 1.71 0.0000000020
149 150.03 -1.03 0.00 -0.10 1.71 1.71 0.0000000020
148 149.41 -1.41 0.00 -0.14 1.71 1.71 0.0000000019
148 149.54 -1.54 0.00 -0.16 1.71 1.71 0.0000000019
148 150.27 -2.27 0.00 -0.23 1.71 1.71 0.0000000019
148 151.48 -3.48 0.00 -0.36 1.71 1.71 0.0000000019
147 152.69 -5.69 -0.11 -0.58 1.28 2.15 0.0000000018
147 153.11 -6.11 -0.24 -0.63 1.24 2.19 0.0000000018
147 152.29 -5.29 -0.34 -0.56 1.29 2.14 0.0000000018
147 149.93 -2.93 0.00 -0.33 1.71 1.71 0.0000000018
145 146.61 -1.61 0.00 -0.16 1.71 1.71 0.0000000016
145 144.25 0.75 0.00 0.08 1.71 1.71 0.0000000016
145 143.85 1.15 0.00 0.12 1.71 1.71 0.0000000016
145 144.92 0.08 0.00 0.01 1.71 1.71 0.0000000016
144 146.94 -2.94 0.00 -0.30 1.71 1.71 0.0000000016
144 148.35 -4.38 -0.09 -0.44 1.38 2.05 0.0000000016
144 149.09 -5.09 -0.19 -0.53 1.32 2.11 0.0000000016
144 148.45 -4.45 -0.28 -0.47 1.36 2.07 0.0000000016
144 146.30 -2.30 0.00 -0.26 1.71 1.71 0.0000000016
143 143.47 -0.47 0.00 -0.05 1.71 1.71 0.0000000015
143 141.20 1.80 0.00 0.18 1.71 1.71 0.0000000015
143 141.12 1.88 0.00 0.19 1.71 1.71 0.0000000015
143 142.31 0.69 0.00 0.07 1.71 1.71 0.0000000015
152 144.31 7.69 0.15 0.78 2.30 1.13 0.0000000011
152 145.97 -6.03 0.27 0.63 2.19 1.24 0.0000000011
152 147.60 4.40 0.36 0.48 2.07 1.36 0.0000000011
152 149.71 2.29 0.00 0.27 1.71 1.71 0.0000000011
-12.025315 -12.025288
Llegue a la vecindad

```

Figura 8. 21 : Evolución de las variables del control cerca del objetivo

Debido a que se consideró en el control de navegación una vecindad con un error de posición cuadrático $1e-9$, se puede apreciar en el evolución de variables del control que al llegar a éste valor de error el robot considera que ya se encuentra en la posición objetivo y se detiene.

Resultado:



Figura 8. 22 : Trayectoria descrita por el Robot para llegar a su posición objetivo

- **Prueba de ruta**

Se realizó sobre la loza deportiva de la Facultad de Ingeniería Mecánica, asignándole la siguiente ruta:



Figura 8. 23 : Ruta asignada al Robot sobre la loza deportiva FIM

A continuación se muestran las imágenes del desplazamiento del robot:



Figura 8. 24 : Desplazamiento del robot siguiendo la ruta

Resultado:

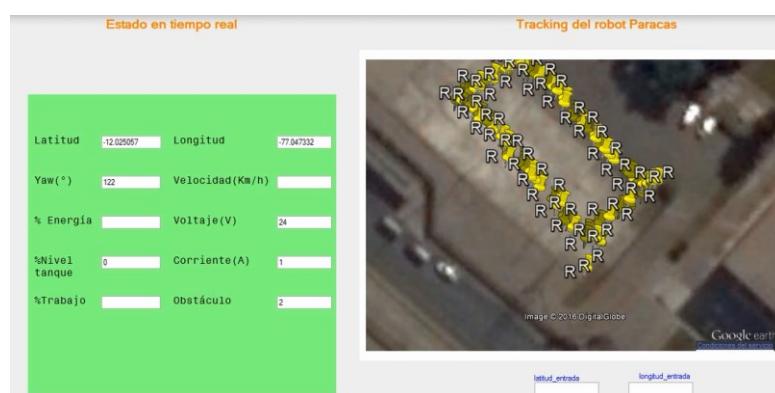


Figura 8. 25 : Ruta seguida por el robot

Cabe resaltar que la exactitud de desplazamiento del robot, se puede ver afectada mientras más rápido se desplaza.

8.5. Prueba del módulo de pulverización

8.5.1. Pruebas de la bomba DC

Determinación de la relación Caudal Q con la velocidad de giro en RPM.



Figura 8. 26 : Pruebas de la motobomba DC

A partir de los datos adquiridos se determinó una relación entre el caudal Q y la velocidad en RPMs de la motobomba DC.

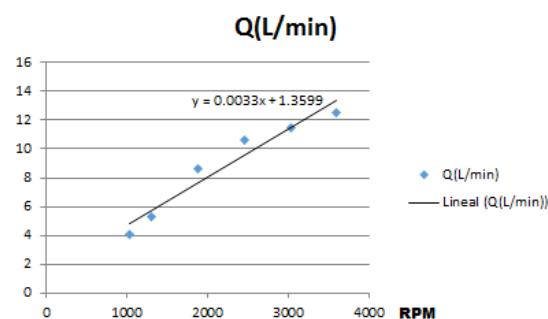


Figura 8. 27 : Determinación experimental del caudal que entrega la motobomba DC



Figura 8. 28 : Pruebas de la motobomba DC

8.5.2. Prueba de activación de válvulas y pulverización



Figura 8. 29 : Activación de las boquillas debido a sensor detección del ultrasónico

8.6. Prueba del módulo de adquisición de datos

Durante las pruebas de navegación también se adquirió datos de los sensores, los cuales se presentan:

8.6.1. Muestra de los datos en tiempo real

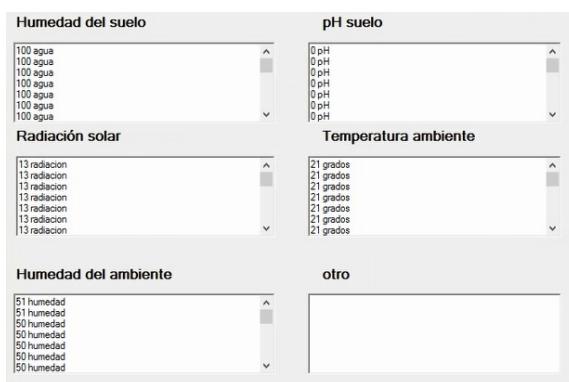


Figura 8. 30 : Datos adquiridos en tiempo real

8.6.2. Consulta a la base de datos

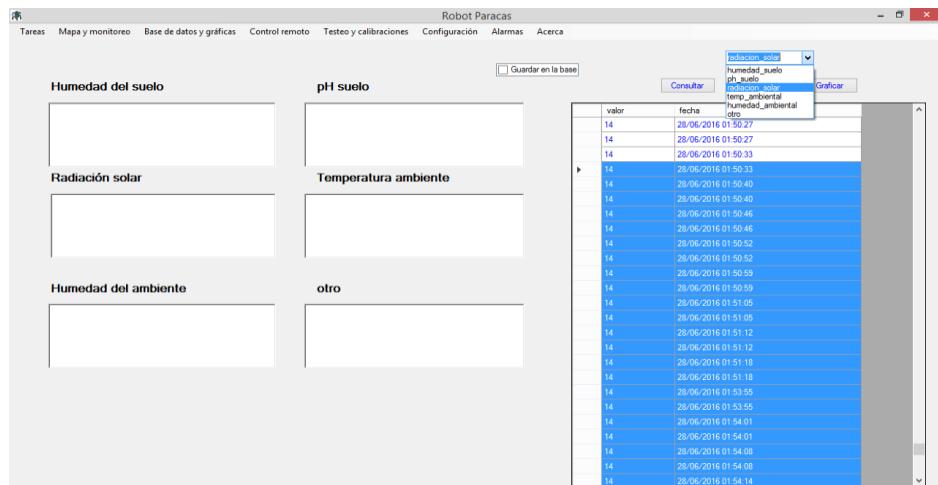


Figura 8. 31 : Consultas a la base de datos de los sensores agrícolas

Es posible graficar los datos almacenados en un rango de fechas preestablecido:

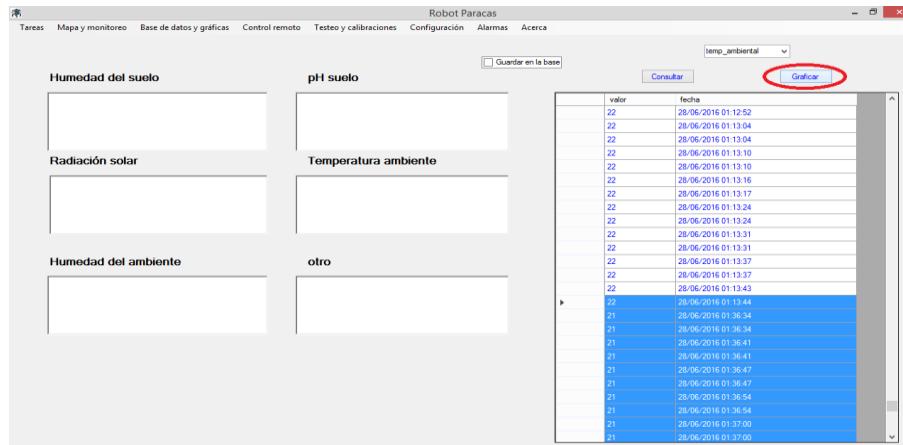


Figura 8. 32 : Seleccionando los datos a graficar

Se obtuvo las siguientes gráficas de los datos de los sensores:

Radiación solar:

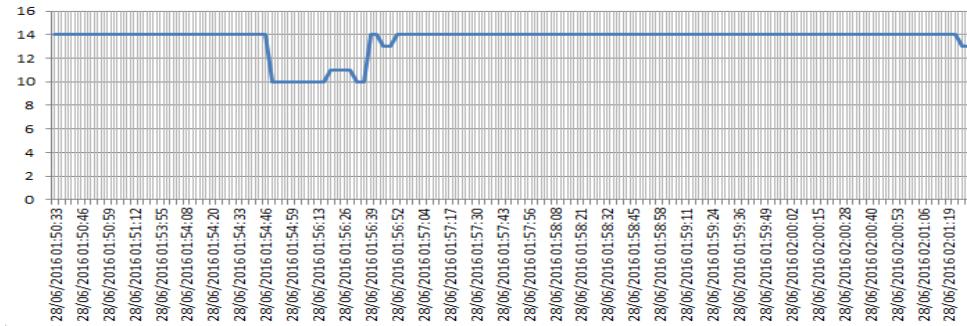


Figura 8. 33 : Mediciones de radiación solar

Humedad ambiental:

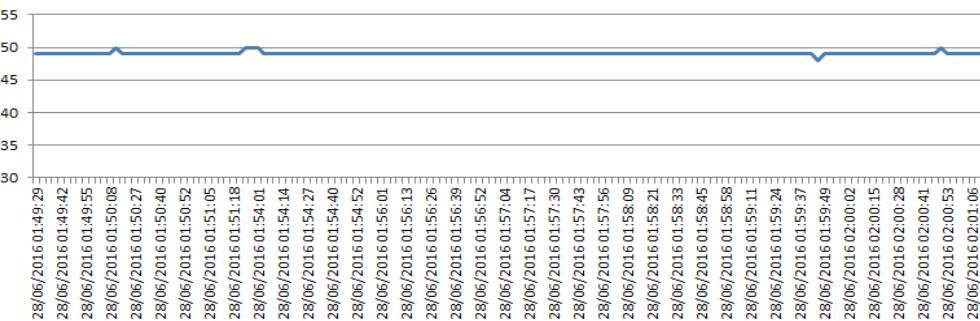


Figura 8. 34 : Mediciones de humedad ambiental

Temperatura ambiental:

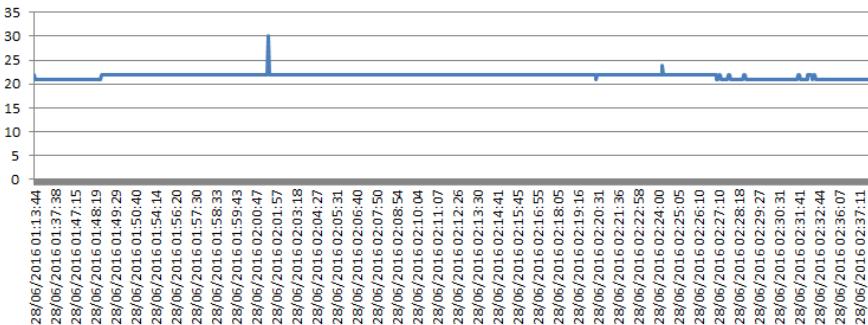


Figura 8. 35 : Mediciones de temperatura ambiental

De igual manera es posible graficar los datos adquiridos de las variables del suelo agrícola (humedad y pH), sin embargo como los datos no son válidos ya que no se tuvo en cuenta el mecanismo para poner en contacto el sensor con el suelo agrícola se dejó el sensor al aire libre para posteriormente acoplarlo a dicho mecanismo.

Conclusiones

1. Se logró controlar la navegación del Robot utilizando un controlador PI digital (discreto) de orientación y 2 controladores PI digitales para la velocidad de los motores DC.
2. Los controladores digitales desarrollados, tal como se muestra anteriormente, fueron discretizados utilizando el tiempo de muestreo según la regla de la quinta parte de la menor constante de tiempo de la función de transferencia en tiempo continuo. De ésta manera se obtuvo buenos resultados, los cuales son mostrados en las gráficas de control de velocidad y orientación.
3. Cuando se realizó pruebas con el GPS en modo “Single” se observó que las mediciones de las geoposiciones no coinciden con los puntos del Mapa de Google Earth y por ello se requiere de una corrección del mapa ya que son imágenes satelitales que presentan deformaciones en ciertas regiones. Sin embargo al utilizar al GPS en modo “RTK” y asignando a la estación fija el valor de su coordenada igual a la que se visualiza en el Mapa, se obtuvo geoposiciones con buena exactitud de los puntos del campo respecto al Mapa.
4. El método de Sensor Fusion DCM estimó la orientación con respuesta rápida y buena exactitud. Además es un método tolerante al ruido.
5. Mientras mayor sea la velocidad de desplazamiento del Robot, se requiere mayores variaciones de la acción de control (voltaje a los motores) y por ende se producen más alteraciones mecánicas que reducen la exactitud del movimiento.

6. La arquitectura de robot móvil Skid Steer aumenta la potencia y torque para el movimiento ya que utiliza 4 motores DC, sin embargo reduce la facilidad de giro y provoca un mayor consumo de corriente. Debido a lo anterior se prefirió trabajar con la arquitectura de robot móvil Diferencial.
7. Se logró recorrer rutas planificadas en la interfaz adquiriendo datos y/o realizando el proceso de fumigación sobre un campo de pruebas (UNI - FIM). De manera similar se ejecutaría el proceso en el terreno agrícola real y con un robot de mayor escala ya que el desarrollado es un robot prototipo.
8. Para las futuras pruebas en uno de los campos agrícolas planteados en la formulación del problema se utilizará el algoritmo de path planning desarrollado. Cabe mencionar que primero se requiere corroborar el mapeo del campo utilizando el mapa de Google Earth y el estado físico actual.
9. La adquisición de datos en el campo de pruebas fue exitoso ya que los datos fueron medidos por el Robot, luego enviado al SCADA y finalmente almacenados en la base de datos de la estación fija. Posteriormente se realizó las consultas a la base de datos para cada variable observando su evolución en el tiempo y geoposición.
10. El proceso de fumigación utilizando el módulo de pulverización y las activaciones de las válvulas solenoides en función a los sensores ultrasónicos laterales fue exitoso ya que no se desperdicia el agroquímico en sectores del campo donde no hay cultivo como sí sucede en los sistemas tradicionales de fumigación. Además de ello es posible regular la presión, del chorro de salida por las boquillas, desde el software lo cual lo hace más versátil.

11. El sistema de fumigación desarrollado (máx. 10Bar, 10L/min) para el Robot prototipo posee todas las características necesarias para un buen funcionamiento, lo cual puede ser escalado para un futuro robot de mayor tamaño y potencia.
12. El robot desarrollado es un prototipo de una futura maquinaria inteligente comercial. Posee características en software y hardware al mismo nivel que las plataformas de investigación de otras empresas a un costo mucho menor.

Recomendaciones

1. El sistema mecánico resultó ser muy influyente en el buen funcionamiento del sistema de control del robot ya que los desalineamiento, falta de ajuste de las piezas, problemas de transmisión de movimiento, etc. alteran considerablemente el funcionamiento.
2. Es necesario que la cadena esté correctamente tensada para no tener problemas en la transmisión de movimiento, para ello será necesario agregar un sistema de tensado de cadena ya que el robot desarrollado no lo tiene y se tuvo que realizar un tensado forzado lo cual provoca un aumento en el consumo de corriente por parte de los motores.
3. Es útil colocar sensores de corriente para proteger a los componentes electrónicos.
4. Para poder obtener mejorares resultados en la navegación se podría cambiar la estrategia de control a un control no lineal multivariable.
5. La antena del GPS RTK debe de poseer línea de vista libre hacia el cielo de tal manera que se pueda comunicar con la mayor cantidad de satélites y trabajar con la mayor exactitud posible,
6. La calibración de los sensores que posee el IMU, como son magnetómetro, acelerómetro y giroscopio es muy importante porque por defecto las medidas entregadas están desfasadas y además necesitan un factor de escalamiento.
7. Es necesario alejar el IMU de los motores y metales ya que el magnetómetro es muy sensible a elementos que producen o alteran campos eléctricos y/o magnéticos.

8. La función de calibración de las mediciones de los sensores de adquisición de datos se puede realizar en el mismo controlador C2 o también en el programa de la interfaz SCADA.
9. Debido a que el magnetómetro posee mediciones discontinuas en los extremos de 0° y 360° , es recomendable aplicar una rotación de coordenadas cuando se tenga una referencia calculada para el controlador de orientación cercana a dichos extremos. En el caso del sistema de control del robot desarrollado se realizó una rotación de 60° para dicho caso.
10. Para el sistema de fumigación, se recomienda instalar una válvula de alivio que posea un rango de presiones no muy grande respecto al rango de operación (0 a 10Bar) ya que se tuvo el problema de no poder variar con precisión la presión de activación de la válvula de alivio porque disminuía y aumentaba muy rápidamente al girar la rosca que lo gradúa.
11. Es importante monitorear continuamente que el tanque no se vacíe, ya que se puede almacenar aire dentro de la bomba y mangueras lo cual es perjudicial.
12. Es necesario proteger a todo el circuito electrónico de los posibles chorros o gotas que provienen de la fumigación, para ello se tiene que cubrir el tablero con un plástico de buena calidad y transparente, o bien disponer de un tablero con protección IP alta.
13. Para el control de la navegación se recomienda que el valor del radio de la vecindad está en función de la exactitud del GPS que se dispone.

Bibliografía

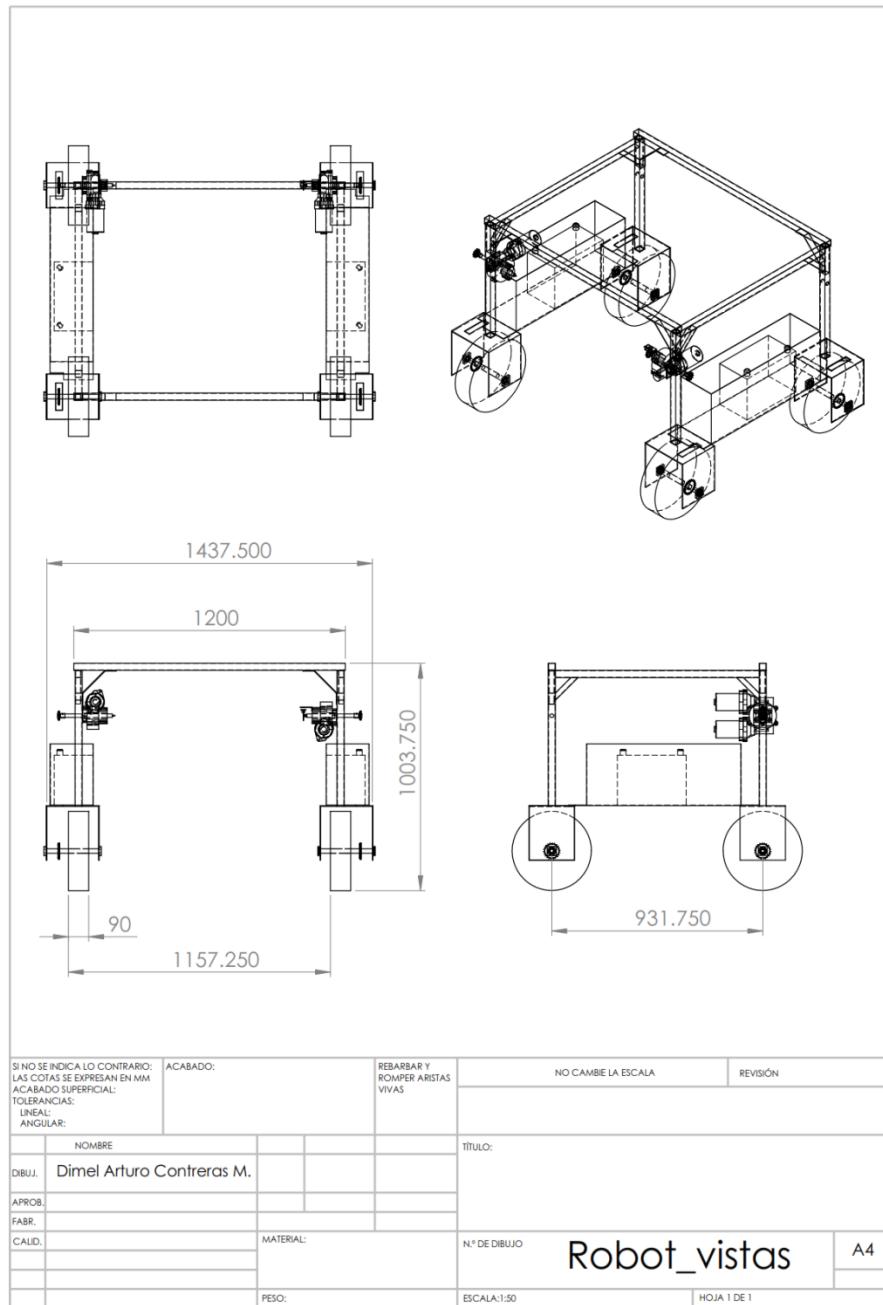
- [1] Emiliano García y Fernando Flego. Agricultura de Precisión. Tecnología Agropecuaria.
- [2] Peru Opportunity Fund. Diagnóstico de la Agricultura en el Perú, informe final.
- [3] “El agro peruano está perdiendo posicionamiento internacional por su falta de tecnificación”. Disponible en: <http://biznews.pe/entrevistas/pedro-alvarez-martinez-ceo-pa-internacional-agro-peruano-esta-perdiendo-posicionamiento>.
- [4] “El mal uso de los agroquímicos”. Disponible en:
<http://www.infobae.com/2013/10/21/1517756-argentina-mal-uso-los-agroquimicos-provoca-problemas-salud>
- [5] Trakür, el robot para fumigar bajo invernadero. Disponible en:
<http://www.redagricola.com/noticias/trakuer-el-robot-para-fumigar-bajo-invernadero>
- [6] Agricultural Robots, Agribot. Disponible en:
<http://www.nepas.eesc.usp.br/roboticaagricola/index.php/robos-agricolas/agribot>
- [7] Project summary, Disponible en: <http://www.rhea-project.eu/>
- [8] Proceedings of the first International Conference on Robotics and associated High-technologies and Equipment for agriculture, 2012 RHEA Project.
- [9] Rover Overview – ArduPilot. Disponible en: <http://rover.ardupilot.com/>
- [10] Products - Grizzly. Disponible en: <http://www.clearpathrobotics.com/>
- [11] Research Robots – Seekur, Disponible en:
http://www.mobilerobots.com/mobile_robots.aspx
- [12] Ing. Agr. Mario P. Bogliani , Ing. Agr. Gerardo Masiá. Pulverizaciones agrícolas. INTA.
- [13] Lía García Pérez. Navegación autónoma de robots en agricultura: un modelo de agentes. Universidad Complutense de Madrid. Madrid, 2005.
- [14] Patricio Abarca .Sistema de control automático.

- [15] Ing. Jorge Cosco Grimaney. Estrategias de control. UNI 2011.
- [16] Oliver J. Woodman .An introduction to inertial navigation. University of Cambridge.
- [17] Santiago Martínez, Rafael Sisto. Posicionamiento y Sensor Data Fusion. Universidad de la República. Uruguay 2009.
- [18]Corrado Santoro . How does a Quadrotor fly? .ARSLAB, Universidad de Catania Italia 2014.
- [19]William Premerlani and Paul Bizard. Direction Cosine Matrix IMU: Theory.
- [20] David Abelardo, García Álvarez. Sistema GNSS. 2008.
- [21] Richard B. Langley. RTK GPS. University of New Brunswick.
- [22] Carlos de Castro Lozano. Introducción SCADA.
- [23] Javier Sotomayor. Implementación de controladores digitales. PUCP 2016.

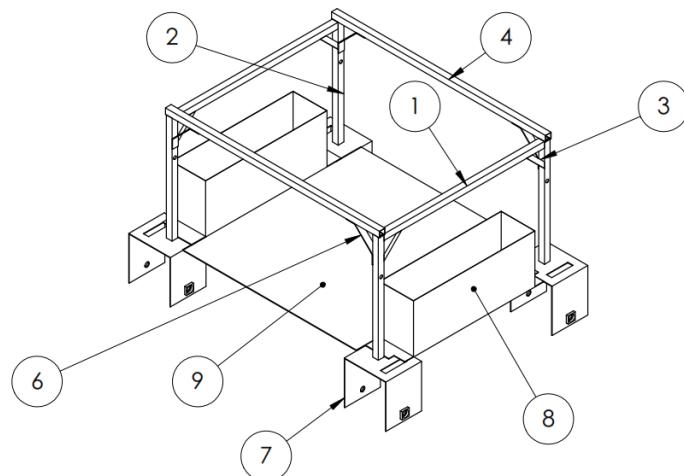
Apéndice A

Planos del sistema mecánico

- Vistas de la estructura mecánica del Robot



- Partes de la estructura mecánica del Robot



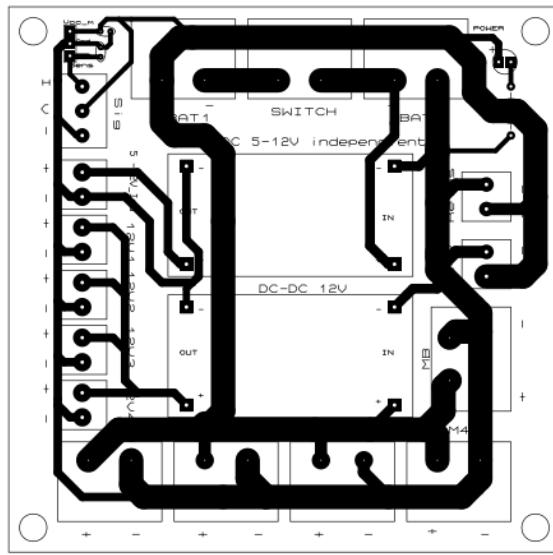
N.º DE ELEMENTO	N.º DE PIEZA	DESCRIPCIÓN	CANTIDAD
1	Lateral_horizontal	tubo cuadrado	2
2	Lateral_vertical	tubo cuadrado	4
3	Triangulo_lateral		4
4	Transversal_horizontal	tubo cuadrado	2
5	Transversal_vertical	tubo cuadrado	4
6	triangulo_transversal		4
7	Soporte	plancha	4
8	caja_batería		2
9	base_inferior	plancha	1

SI NO SE INDICA LO CONTRARIO: LAS COTAS SE EXPRESAN EN MM ACABADO SUPERFICIAL: TOLERANCIAS: LINEAL: ANGULAR:	ACABADO:	REBARBAR Y ROMPER ARISTAS VIVAS	NO CAMBIE LA ESCALA	REVISIÓN
DIBUJ.	TÍTULO:			
DIBUJ.	NOMBRE			
	Dimel Arturo Contreras M.			
APROB.				
FABR.				
CALID.	MATERIAL:	N.º DE DIBUJO	Estructura Mecánica	
			A4	
	PESO:	ESCALA:1:50	HOJA 1 DE 1	

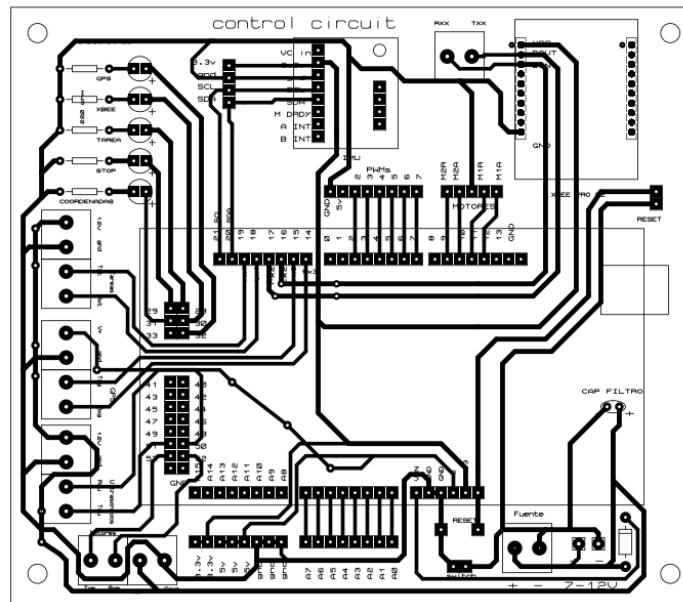
Apéndice B

Esquemas circuitos electrónicos

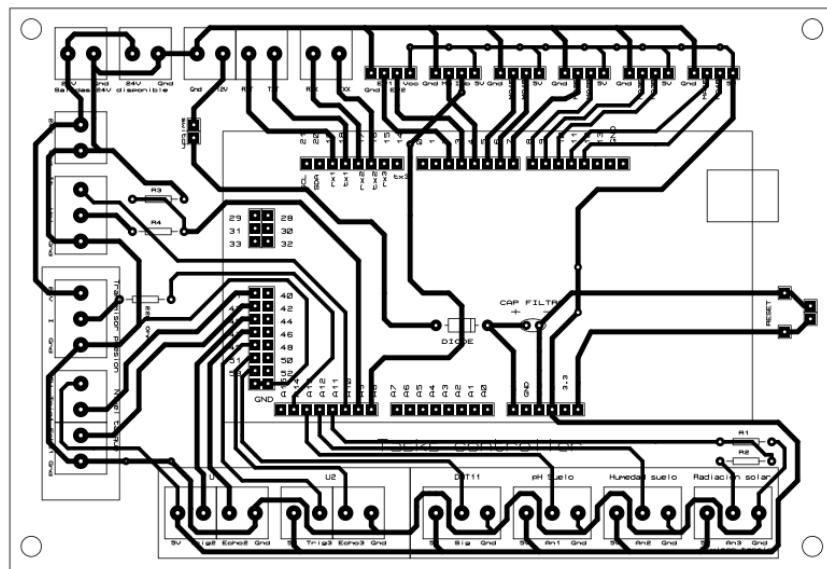
- Diseño de tarjeta electrónica de distribución de energía



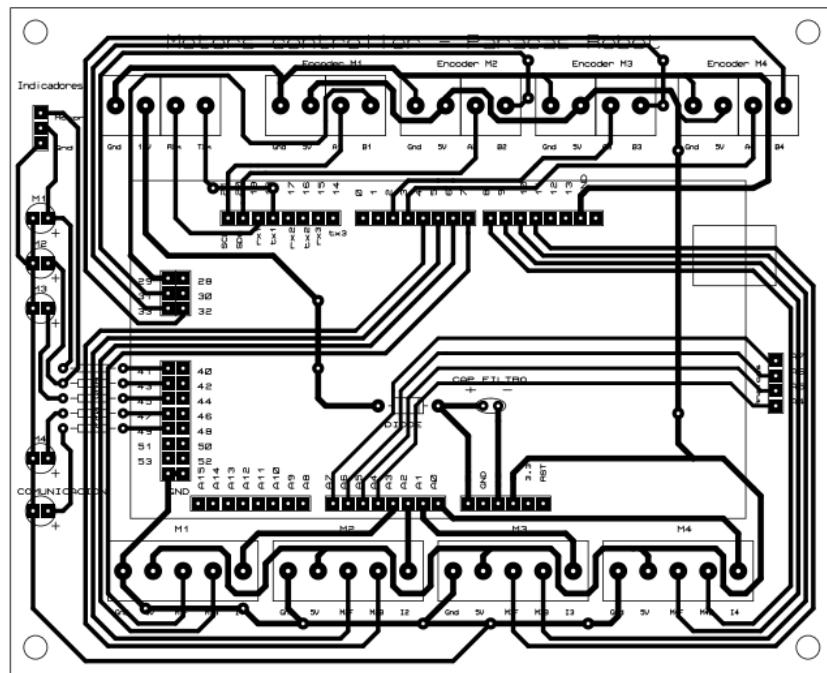
- Diseño de tarjeta electrónica del controlador de navegación C1



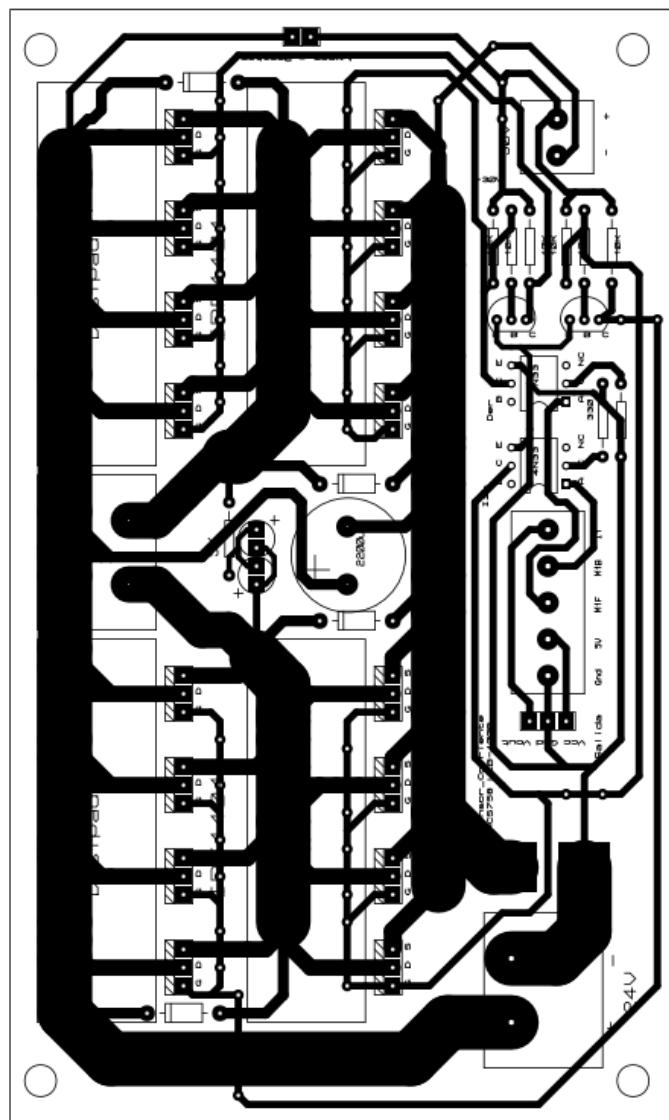
- Diseño de tarjeta electrónica del controlador de tareas C2



- Diseño de tarjeta electrónica del controlador de velocidades C3



- Diseño de tarjeta electrónica del driver de Puente H



Apéndice C

Códigos de los controladores

Controlador C1:

- Método de Sensor Fusion DCM

```

void Compass_Heading()
{
    float mag_x;
    float mag_y;
    float cos_roll;
    float sin_roll;
    float cos_pitch;
    float sin_pitch;
    cos_roll = cos(roll);
    sin_roll = sin(roll);
    cos_pitch = cos(pitch);
    sin_pitch = sin(pitch);
    // Tilt compensated magnetic field X
    mag_x = magnetom[0] * cos_pitch + magnetom[1] * sin_roll * sin_pitch + magnetom[2] * cos_roll
    * sin_pitch;
    // Tilt compensated magnetic field Y
    mag_y = magnetom[1] * cos_roll - magnetom[2] * sin_roll;
    // Magnetic Heading
    MAG_Heading = atan2(-mag_y, mag_x);
}

void Normalize(void)
{
    float error=0;
    float temporary[3][3];
    float renorm=0;
    error= -Vector_Dot_Product(&DCM_Matrix[0][0],&DCM_Matrix[1][0])* .5;
    Vector_Scale(&temporary[0][0], &DCM_Matrix[1][0], error); //eq.19
    Vector_Scale(&temporary[1][0], &DCM_Matrix[0][0], error); //eq.19
    Vector_Add(&temporary[0][0], &temporary[0][0], &DCM_Matrix[0][0]);//eq.19
    Vector_Add(&temporary[1][0], &temporary[1][0], &DCM_Matrix[1][0]);//eq.19
    Vector_Cross_Product(&temporary[2][0],&temporary[0][0],&temporary[1][0]);
    renorm=.5 *(3 - Vector_Dot_Product(&temporary[0][0],&temporary[0][0]));
    Vector_Scale(&DCM_Matrix[0][0], &temporary[0][0], renorm);
    renorm=.5 *(3 - Vector_Dot_Product(&temporary[1][0],&temporary[1][0]));
    Vector_Scale(&DCM_Matrix[1][0], &temporary[1][0], renorm);
    renorm=.5 *(3 - Vector_Dot_Product(&temporary[2][0],&temporary[2][0]));
    Vector_Scale(&DCM_Matrix[2][0], &temporary[2][0], renorm);
}

```

```

void Drift_correction(void)
{
    float mag_heading_x;
    float mag_heading_y;
    float errorCourse;

    //Compensation the Roll, Pitch and Yaw drift.
    static float Scaled_Omega_P[3];
    static float Scaled_Omega_I[3];
    float Accel_magnitude;
    float Accel_weight;

    // Calculate the magnitude of the accelerometer vector
    Accel_magnitude = sqrt(Accel_Vector[0]*Accel_Vector[0] +
                           Accel_Vector[1]*Accel_Vector[1] + Accel_Vector[2]*Accel_Vector[2]);

    Accel_magnitude = Accel_magnitude / GRAVITY; // Scale to gravity.
    Accel_weight = constrain(1 - 2*abs(1 - Accel_magnitude),0,1); //

    Vector_Cross_Product(&errorRollPitch[0],&Accel_Vector[0],&DCM_Matrix[2][0]);
    Vector_Scale(&Omega_P[0],&errorRollPitch[0],Kp_ROLLPITCH*Accel_weight);

    Vector_Scale(&Scaled_Omega_I[0],&errorRollPitch[0],Ki_ROLLPITCH*Accel_weight);
    Vector_Add(Omega_I,Omega_I,Scaled_Omega_I);

    mag_heading_x = cos(MAG_Heading);
    mag_heading_y = sin(MAG_Heading);
    errorCourse=(DCM_Matrix[0][0]*mag_heading_y) - (DCM_Matrix[1][0]*mag_heading_x);

    //Calculating YAW error
    Vector_Scale(errorYaw,&DCM_Matrix[2][0],errorCourse);

    Vector_Scale(&Scaled_Omega_P[0],&errorYaw[0],Kp_YAW);//.01proportional of YAW.
    Vector_Add(Omega_P,Omega_P,Scaled_Omega_P);//Adding Proportional.

    Vector_Scale(&Scaled_Omega_I[0],&errorYaw[0],Ki_YAW);//.00001Integrator
    Vector_Add(Omega_I,Omega_I,Scaled_Omega_I);//adding integrator to the Omega_I
}

void Matrix_update(void)
{
    Gyro_Vector[0]=GYRO_SCALED_RAD(gyro[0]); //gyro x roll
    Gyro_Vector[1]=GYRO_SCALED_RAD(gyro[1]); //gyro y pitch
    Gyro_Vector[2]=GYRO_SCALED_RAD(gyro[2]); //gyro z yaw
}

```

```

Accel_Vector[0]=accel[0];
Accel_Vector[1]=accel[1];
Accel_Vector[2]=accel[2];

Vector_Add(&Omega[0], &Gyro_Vector[0], &Omega_I[0]); //adding proportional term
Vector_Add(&Omega_Vector[0], &Omega[0], &Omega_P[0]); //adding Integrator term

Update_Matrix[0][0]=0;
Update_Matrix[0][1]=-G_Dt*Omega_Vector[2];//-z
Update_Matrix[0][2]=G_Dt*Omega_Vector[1];//y
Update_Matrix[1][0]=G_Dt*Omega_Vector[2];//z
Update_Matrix[1][1]=0;
Update_Matrix[1][2]=-G_Dt*Omega_Vector[0];//-x
Update_Matrix[2][0]=-G_Dt*Omega_Vector[1];//-y
Update_Matrix[2][1]=G_Dt*Omega_Vector[0];//x
Update_Matrix[2][2]=0;

Matrix_Multiply(DCM_Matrix,Update_Matrix,Temporary_Matrix); //a*b=c

for(int x=0; x<3; x++) //Matrix Addition (update)
{
    for(int y=0; y<3; y++)
    {
        DCM_Matrix[x][y]+=Temporary_Matrix[x][y];
    }
}

void Euler_angles(void)
{
    pitch = -asin(DCM_Matrix[2][0]);
    roll = atan2(DCM_Matrix[2][1],DCM_Matrix[2][2]);
    yaw = atan2(DCM_Matrix[1][0],DCM_Matrix[0][0]);
}

void read_sensors() {
    Read_Gyro(); // Read gyroscope
    Read_Accel(); // Read accelerometer
    Read_Magn(); // Read magnetometer
}

void reset_sensor_fusion() {
    float temp1[3];
    float temp2[3];
    float xAxis[] = {1.0f, 0.0f, 0.0f};

    read_sensors();
    timestamp = millis();
}

```

```

// GET PITCH
// Using y-z-plane-component/x-component of gravity vector
pitch = -atan2(accel[0], sqrt(accel[1] * accel[1] + accel[2] * accel[2]));

// GET ROLL
// Compensate pitch of gravity vector
Vector_Cross_Product(temp1, accel, xAxis);
Vector_Cross_Product(temp2, xAxis, temp1);
// Normally using x-z-plane-component/y-component of compensated gravity vector
// roll = atan2(temp2[1], sqrt(temp2[0] * temp2[0] + temp2[2] * temp2[2]));
// Since we compensated for pitch, x-z-plane-component equals z-component:
roll = atan2(temp2[1], temp2[2]);

// GET YAW
Compass_Heading();
yaw = MAG_Heading;

// Init rotation matrix
init_rotation_matrix(DCM_Matrix, yaw, pitch, roll);
}

```

- **Control de la navegación**

```

int Tmuestreo = 200;
boolean flag_control = false; //flag que inicia el controlador
float yaw_v ;
unsigned long t;
unsigned long Tor;
float epsi = 0.5;
float x_a,y_a, x_o, y_o ,vl,wang ,W = 1.15;
int T = 20;
///Variables y parametros para el control
float kp = 0.1,ki = 0.1,int_e = 0, R = 0.175,u1 = 0,u2 = 0,e_ant = 0,wang_ant = 0;
float u1_send = 0,u2_send = 0,v1 = 0,v2 = 0; //Left
int phid;
int vel_lineal;
float phi, phid_p1,phid_p2,beta,e,error;
float umax = 5.2; //Configurar maxima velocidad y aceleración.
float amax = 5.2;
int puntos;
float latitudwp[30]; //Por ahora maximo 30 WP
float longitudwp[30];
int jj;//Contador de puntos dentro del control de navegación

#define C2 Serial1
#define SCADA Serial2
#define RTK Serial3

```

```

int ref = 66; //valor de test para prueba del cambio de coordenadas
long t_ant = 0; //Medicion de tiempo para que ingrese al envio_monitoreo
long t_control = 0; //Medicion de tiempo para que ingrese al control()
long t_gps = 0; //Medicion de tiempo para que ingrese al control()

int puntosgps = 0; //coordenadas gps prueba
float sq_radius=0;
int rot_coord = 0;

void setup()
{
//-----Comunicacion con C2 -----
C2.begin(115200);
//-----IMU-----
cli(); //stop interrupts

Timer3.initialize(20000); // initialize timer1, en microsegundos
// Timer1.pwm(9, 512); // setup pwm on pin 9, 50% duty cycle
Timer3.attachInterrupt(callback); // attaches callback() as a timer overflow interrupt

sei(); //allow interrupts
// Init serial output
Serial.begin(OUTPUT__BAUD_RATE);

//XBEE_repcion*****
SCADA.begin(115200);
for (int j=0;j<400;j=j+1){
    data[j] = 0.000000000000000f;
}
//GPS ****
RTK.begin(115200);
//C3
C3.begin(115200); //XBEE
startMillis = millis();
// Init status LED
pinMode (STATUS_LED_PIN, OUTPUT);
digitalWrite(STATUS_LED_PIN, LOW);

// Init sensors
delay(50); // Give sensors enough time to start
I2C_Init();
Accel_Init();
Magn_Init();
Gyro_Init();

// Read sensors, init DCM algorithm
delay(20); // Give sensors enough time to collect data
reset_sensor_fusion();
G_Dt=0.020;
jj = 0; //Primer WP

}

```

```

void loop(){

    comandos_recepcion(); //RECEPCION COORDENADAS XBEE
    if(millis()-t_gps >200){
        RTK_recepcion();
        t_gps = millis();
    }
    read_sensors();

    yaw_v = TO_DEG(yaw); //Esta medida es respecto al Sur
    //*****Convirtiendo respecto al Norte
    yaw_v = yaw_v-180;

    if(yaw_v < 0)
        yaw_v += 360;

    if(flag_control){
        //Coordenada Objetivo; jj inicia en 0
        beta = calcula_orientacion(latitude,longitude,latitudwp[jj],longitudwp[jj]);

        if(beta >=0 && beta < 20){
            rot_coord = 60;
        }
        else if(beta > 340 && beta <= 360){
            rot_coord = -60;
        }
        else{
            rot_coord = 0;
        }

        phi = yaw_v + rot_coord; //Sensado
        if(phi<0) phi+=360; //Cuando se hace negativo, se aumenta una vuelta
        if(phi>360) phi-=360;

        phid = beta + rot_coord;

        if((millis()-t_control)>=Tmuestreo){ //Tiempo de muestreo del controlador
            Control();
        }

        if(vecindad(jj)){
            jj+=1; //Se pasa al siguiente WP

            if(jj>=puntos){ //Si ya acabó todos los WP, tiene que dejar de control y detener
                flag_control = false;
                detener_robot();
                Serial.println("Detenerse");
            }
        }
    }

    delay(10);
}

```

```

float calcula_orientacion(float lat_a, float lon_a , float lat_f, float lon_f){
    float orid;
    orid=atan2((lat_f-lat_a),(lon_f-lon_a))*57.29582790879777;//en sexagesimales
    orid=90-orid;//segun el sentido de norte y sentido horario

    if(orid<0)
        orid+=360;
    return orid; //orientación en grados sexagesimales
}

boolean vecindad(int jj){
    sq_radius = sq(latitudwp[jj]-latitude)+sq(longitudwp[jj]-longitude);
    if(sq_radius < 0.000000001){
        Serial.print(latitudwp[jj],6);
        Serial.print(" ");
        Serial.println(latitude,6);
        Serial.println("Llegue a la vecindad");
        return true;
    }
    else
        return false;
}

void detener_robot(){
    Serial.println("Robot Termino Ruta");
    // envio serial a C3 para detenerse
    C3.print("x");
    C3.print((char)0);
    C3.print((char)0);
    C3.print((char)0);
    C3.print((char)0);
}

void callback()
{
    compensate_sensor_errors();
    // Run DCM algorithm
    Compass_Heading(); // Calculate magnetic heading
    Matrix_update();
    Normalize();
    Drift_correction();
    Euler_angles();
}

void Control(){
    vl = 0.3;
    maximo_rpm = 40;
    Timer3.detachInterrupt();

    e = (float)phid-phi;
    int_e = int_e + e*T/1000;
}

```

```

if(abs(e)>40){
    maximo_rpm = 20;
}
if(abs(e)<5) {
    kp =0.01;
    ki = 0.005;
}
else if(abs(e)<10){
    kp =0.03;
    ki = 0.01;
}
else{
    kp =0.1;
    ki = 0.01;
}

if(int_e > 500) int_e=500;
//metodo de PI Tustin
float Tm = Tmuestro/1000.0;
wang = wang_ant + Kp*(e-e_ant) + Ki*Tm*(e+e_ant)/2.0;
e_ant = e;
wang_ant = wang;

u1 = (vl/R)+(wang*W/2);
u2 = (vl/R)-(wang*W/2);
Saturadores();

//Recalcular
v1 = R*u1;
v2 = R*u2;
vl = (v1+v2)/2;
wang = (vl-v2)/W;
enviar_c3(u1,u2);

t_control = millis(); //Se actualiza el tiempo anterior
Timer3.attachInterrupt(callback);
}
void Saturadores(){
    if (abs(u1) > umax){
        u1 = sign(u1)*umax;
    }
    if (abs(u2) > umax){
        u2 = sign(u2)*umax;
    }
}

int sign(float x){
    if(x>=0){
        return 1;
    }
    else{
        return -1;
    }
}

```

Controlador C3:

```

//Encoder
#define ppv1 100.0
#define ppv2 100.0
#define ppv3 100.0
#define ppv4 100.0
//Motor pines
#define M1F 6
#define M1B 7
#define M2F 5
#define M2B 4
#define vmax 200 //RPMs
//Encoder pines
#define chA_m1 0
#define chA_m2 0
//Interrupciones externas encoder
#define inte1 2
#define inte2 3
//-----Corrientes -----
#define I_M1 analogRead(A3)
#define I_M2 analogRead(A2)
//Calculo de velocidades por el encoder
unsigned long elapsedTime1,elapsedTime2,oldTime1,oldTime2;
float velocidad1,velocidad2;

//Variables para el controlador
unsigned long lastTime;
double Input1,Input2, Output1,Output2; Setpoint_m1,Setpoint_m2;
double errSum1,errSum2, lastInput1,lastInput2;
double Output1_ant , error1_ant , Output2_ant , error2_ant ;
double kp,ki,kd;
double kp1,kp2,ki1,ki2,kd1,kd2;
int SampleTime = 15;
//Variables adicionales de comando
long referencia=0;
int max_duty=0;
int ref_ant=0;
//Recepcion de las referencias
int val;
char setpoint[2];
//Sensor de corriente
int sensorValue_sin_offset = 0;
int corriente_analog = 0;
float corriente_voltaje = 0;
float corriente = 0;
long t_op = 0;
boolean enable1,enable2,enable3,enable4,enable5;

int V1F=0,V1B=0,V2F=0,V2B=0;
char ibyte;

void setup() {
  Serial.begin(115200);
  Serial1.begin(115200);
}

```

```

pinMode(M1F, OUTPUT);
pinMode(M1B, OUTPUT);
pinMode(M2F, OUTPUT);
pinMode(M2B, OUTPUT);
//configuracion de frecuencia de PWM
TCCR3B = TCCR3B & B11111000 | B00000011; // Pines 2/3/5 set timer 3
TCCR4B = TCCR4B & B11111000 | B00000011; // Pines 6/7/8 set timer 4
TCCR0B = TCCR0B & B11111000 | B00000011; // Pines 4/13 set timer 0
//https://arduino-info.wikispaces.com/Arduno-PWM-Frequency
//pines de interrupcion como entrada
pinMode(chA_m1, INPUT);
pinMode(chA_m2, INPUT);
attachInterrupt(0, blink, CHANGE); //pin2
attachInterrupt(inte1, cal_v1, RISING);
attachInterrupt(inte2, cal_v2, RISING); //pin2
//Seteo de controladores
SetTunings1(0.1, 9, 0); //kp,ki,kd
SetTunings2(0.1, 9, 0);
//Valores iniciales de variables
elapsedTime1=0;
elapsedTime2=0;
//Inicialización de setpoints
setpoint[0] = 100; // [0 100 200] = [-100 0 100]
setpoint[1] = 100;
t_op=millis();
}
void loop() {
    //Recepcion de las referencias de velocidad
    Recepcion_setpoints();
    Compute();
    Output_pwm();
}
void cal_v1(){
    elapsedTime1 = micros() - oldTime1;
    oldTime1 = micros();
    velocidad1 = 60000000.0/(elapsedTime1*ppv1);
    Input1 = map(velocidad1,0,vmax,0,100);
}
void cal_v2(){
    elapsedTime2 = micros() - oldTime2;
    oldTime2 = micros();
    velocidad2 = 60000000.0/(elapsedTime2*ppv2);
    Input2 = map(velocidad2,0,vmax,0,100);
}
void Compute()
{
    unsigned long now = millis();
    int timeChange = (now - lastTime);
    //Serial.println((int)timeChange);
}

```

```

if(Input1 == 0 && Setpoint_m1!=0){ //Para reducir el ruido encoder1
    Input1 = lastInput1;
}
if(Input2 == 0 && Setpoint_m2!=0){ //Para reducir el ruido encoder2
    Input2 = lastInput2; }
if(timeChange>=SampleTime)
{
// Calcula todas las variables de errores.
double error1 = Setpoint_m1 - Input1;
double error2 = Setpoint_m2 - Input2;
errSum1 += error1;
errSum2 += error2;
double dInput1 = (Input1 - lastInput1);
double dInput2 = (Input2 - lastInput2);
// Calculamos la función de salida del PID Tustin///////////
Output1= Output1_ant + kp1 *(error1-error1_ant) + ki1*(error1+error1_ant)/2.0 ;
Output2= Output2_ant + kp2 *(error2-error2_ant) + ki2*(error2+error2_ant)/2.0;
Output1_ant = Output1;
error1_ant = error1;
Output2_ant = Output2;
error2_ant = error2;
//saturacion
if(Output1<0) Output1=0;
if(Output1>100) Output1=100;
Output1=(Output1*255)/100;
if(Output2<0) Output2=0;
if(Output2>100) Output2=100;
Output2=(Output2*255)/100;
lastTime = now;
}
}
void SetTunings1(double Kp, double Ki, double Kd)
{
double SampleTimeInSec = ((double)SampleTime)/1000;
kp1 = Kp;
ki1 = Ki * SampleTimeInSec;
kd1 = Kd / SampleTimeInSec;
}
void SetTunings2(double Kp, double Ki, double Kd)
{
double SampleTimeInSec = ((double)SampleTime)/1000;
kp2 = Kp;
ki2 = Ki * SampleTimeInSec;
kd2 = Kd / SampleTimeInSec;
}
void medir_corrientes(){
    float total_current = calculo_corriente(I_M1) +
    calculo_corriente(I_M2)+calculo_corriente(I_M3)+calculo_corriente(I_M4);
}

```