

HUMANIDADES, CIENCIA, TECNOLOGÍA E INNOVACIÓN EN PUEBLA

ACADEMIA JOURNALS



OPUS PRO SCIENTIA ET STUDIUM

ISSN 2644-0903 online

VOL. 2, NO. 1, 2020

WWW.ACADEMIAJOURNALS.COM

TRABAJO DE INVESTIGACIÓN AUSPICIADO POR EL CONVENIO CONCYTEP-ACADEMIA JOURNALS



CLAUDIA MENA HUERTA

CONTROL DE UN ROBOT ACTIVADO POR CABLES

TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO TECNOLÓGICO DE PUEBLA

DIRECTOR: M.C. VICENTE RAMÍREZ PALACIOS

CO-DIRECTOR: DR. SERGIO JAVIER TORRES MÉNDEZ

VOCAL: DR. JOSÉ RAFAEL MENDOZA VÁZQUEZ

NÚMERO DE SECUENCIA 2-46



**“CONTROL DE UN ROBOT ACTIVADO POR
CABLES”**

PRESENTA:

ING. CLAUDIA MENA HUERTA

DIRECTOR:

M.C. VICENTE RAMÍREZ PALACIOS

CO-DIRECTOR:

DR. SERGIO JAVIER TORRES MÉNDEZ

VOCAL:

DR. JOSÉ RAFAEL MENDOZA VÁZQUEZ

**TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE PUEBLA**

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

PUEBLA, PUE. 8 DE OCTUBRE DE 2019

**QUE PARA OBTENER EL TÍTULO DE:
MAESTRA EN INGENIERÍA ELECTRÓNICA**

Resumen

“CONTROL DE UN ROBOT ACTIVADO POR CABLES”

AUTOR: MENA HUERTA CLAUDIA

En la presente tesis de grado de maestría, se muestra el trabajo realizado referente a un control de un robot paralelo planar activado por cables, con una estructura tipo industrial ubicado en el laboratorio de posgrado en el Tecnológico Nacional de México / Instituto Tecnológico de Puebla.

Las áreas tecnológicas tienen un gran impacto social y económico en nuestros días; los avances científicos han ido creciendo exponencialmente y, sin embargo, en países cuya infraestructura es carente la ciencia y tecnología no logra despuntar como en países de primer mundo. El ámbito de la robótica ha predominado en los últimos años y es necesario estar actualizados en esos temas como conjunto social.

En este proyecto de tesis se estudia, analiza y aplica el conocimiento referente a la robótica y control de un sistema, donde se evalúa un sistema de control de posición de un robot basado en su cinemática.

Las actividades que se desarrollaron en el presente tema de tesis fueron: análisis mecánico y físico del sistema, ecuaciones de cinemática inversa y cinemática directa, cinemática diferencial, análisis de los parámetros configurados en el servo drive, análisis de la programación en el software del PLC/MC/HMI marca Delta, presentación del código de programación para la cinemática del robot paralelo activado por cables, así como los resultados, ajustes y mejoras en el sistema de control de posición.

Finalmente se presentan las conclusiones sobre los beneficios de trabajar con este robot y como complemento, los anexos de los productos generados como evidencias de este tema de tesis para la obtención del grado de maestría.

Abstract

“CABLE-DRIVEN PARALLEL PLANAR ROBOT CONTROL”

AUTHOR: MENA HUERTA CLAUDIA

In the present Master's thesis, it is showed the task done reference to the cable-driven parallel planar robot control, which structure is industrial kind and it is located at postgraduate lab in the Tecnológico Nacional de México / Instituto Tecnológico de Puebla.

The technological field have got a great social and economic impact in our days; scientific advances have been growing exponentially and, nevertheless, in countries whose infrastructure is lacking, science and technology fail to emerge as in first world countries. The field of robotics has dominated in recent years and it is necessary to be updated on these issues as a social group.

In the current thesis project, knowledge about the robotics and control of a system is studied, analyzed and applied, where a position control system of a robot based on its kinematics is evaluated.

The activities that were developed in this thesis topic were: mechanical and physical analysis of the system, equations of inverse kinematics and direct kinematics, differential kinematics, analysis of the parameters configured in the servo drive, analysis of the programming in the PLC / MC / HMI software Delta brand, presentation of the programming code for the kinematics of the parallel robot activated by cables, as well as the results, adjustments and improvements in the position control system.

Finally, the conclusions about the benefits of working with this robot are presented and as a complement, the annexes of the products generated as evidence of this thesis topic for obtaining the master's degree.



"2019, Año del Caudillo del Sur, Emiliano Zapata"

PT-7

DEPI/MIE/024/2019

Puebla, Puebla, **03/OCTUBRE/2019**

ASUNTO: Autorización de impresión de Tesis.

**C. CLAUDIA MENA HUERTA
PRESENTE.**

De acuerdo al Manual de procedimientos para obtener el título de **"Maestra en Ingeniería Electrónica"** en el Tecnológico Nacional de México y habiendo cumplido con todas las indicaciones que el Jurado Revisor le hizo respecto de su trabajo de Tesis titulado:

"CONTROL DE UN ROBOT ACTIVADO POR CABLES"

Se le comunica que se le concede la **autorización para que proceda a la impresión del trabajo presentado.**

ATENTAMENTE
"Excelencia tecnológica con participación humana"

**OMAR FLORES SÁNCHEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS
DE POSGRADO E INVESTIGACIÓN**



SECRETARÍA DE
EDUCACIÓN PÚBLICA
INSTITUTO TECNOLÓGICO
DE PUEBLA
DIV. DE ESTUDIOS DE
POSTGRADO E INVESTIGACIÓN

C.c.p Coordinación de Maestría en Ingeniería Electrónica
OFS/rth



Agradecimientos

- Esta investigación fue realizada gracias al apoyo del Consejo de Ciencia y Tecnología del Estado de Puebla.
- Agradezco a mi director de tesis, el M.C. Vicente Ramírez Palacios por permitirme trabajar bajo su asesoría, al igual que mi co-asesor el Dr. Sergio Javier Torres Méndez y el equipo de trabajo perteneciente a la línea de investigación “Sistemas mecatrónicos interactivos aplicados al control de procesos”, el Dr. José Rafael Mendoza Vázquez y la M.C. Irma Delia Rojas Cuevas, quienes me aportaron sus conocimientos para encaminar el trabajo de tesis correctamente y así poder lograr los objetivos. Fueron pacientes conmigo y me apoyaron en todo momento durante mi permanencia en la maestría de Ingeniería electrónica.
- Agradezco al Tecnológico Nacional de México/Instituto Tecnológico de Puebla por apoyarme en la adquisición del material de trabajo, así como su apoyo para el logro de publicaciones y presentaciones en congresos referentes a los temas de estudio del posgrado.
- Agradezco al Programa Nacional de Posgrados de Calidad perteneciente al Consejo Nacional de Ciencia y Tecnología por el apoyo otorgado durante mi estancia en la maestría.
- Finalmente, agradezco a mis compañeros de generación quienes me aportaron su conocimiento cuando lo necesitaba y fueron una grata compañía.

Muchas gracias a todos.

Dedicatorias

A Dios

Por permitirme seguir adelante con mi proyecto de vida y darme fuerzas en todo momento alentándome a terminar mi tema de tesis.

A mis padres José Abraham Mena y Salomé Huerta

Quienes me apoyaron desde el momento en que decidí hacer un posgrado; son mi mayor motivación para ser profesionalmente buena y, sobre todo, para mejorar mi persona.

A mis hermanas Sulma Erika Mena y Ariane Mena

Quienes me motivaron y me alegraron muchos momentos con su compañía y apoyo.

A Pedro Armando Rosas

Quien me apoyó en todo momento y me enseñó que no debo rendirme ante situaciones complicadas y ser constante en el camino.

Y a mi sobrina Andrea Sofía

Quien me dio su cariño incondicional y su alegría en este proceso.

Contenido

CAPÍTULO I INTRODUCCIÓN.....	1
1.1 Antecedentes (estado del arte).....	2
1.2 Planteamiento del problema	6
1.3 Justificación.....	7
1.4 Objetivos.....	8
1.5 Metodología	9
1.6 Estructura de la Tesis.....	10
CAPÍTULO II FUNDAMENTO TEÓRICO	11
2.1 Robótica.....	11
2.1.1 Robótica industrial	14
2.2 Teoría de robots.....	18
2.2.1 Arquitectura típica de robots	19
2.2.1.1 Estructura mecánica y elemento terminal de un robot	20
2.2.1.2 Sistemas de transmisión, actuación y sensorial	23
2.2.2 Grados de libertad y espacio de trabajo de un robot	26
2.2.3 Tipos de robots.....	29
2.2.3.1 Robots seriales	30
2.2.3.2 Robots paralelos.....	31
2.2.3.3 Robots paralelos activados por cables	32
2.3 Cinemática	34
2.3.1 Cinemática directa.....	35
2.3.2 Cinemática inversa	36
2.3.3 Cinemática diferencial.....	37
2.3.4 Singularidades.....	39
2.3.5 Cinemática de aceleración.....	40
2.4 Algoritmos de control de movimiento.....	41
2.4.1 Algoritmos conforme al modelo cinemático	41
2.4.2 Algoritmos conforme al modelo dinámico.....	44
2.5 Control	45
2.5.1 Control Proporcional (P)	46
2.5.2 Control Proporcional Integral (PI).....	47
2.5.3 Control Proporcional Derivativo (PD)	51
2.5.4 Control Proporcional Integral Derivativo (PID).....	54

2.6	automatización	57
2.6.1	Arquitectura del proceso de automatización.....	57
2.6.2	Dispositivos de automatización	59
2.6.2.1	Controlador Lógico Programable (PLC)	59
2.6.2.2	Servo drive y servomotor	61
2.6.2.3	Otros dispositivos de automatización	62
2.6.3	Redes de comunicación industriales	63
2.6.4	Protocolo de comunicación	64
2.6.4.1	Protocolo CANopen	64
CAPÍTULO III DESARROLLO		69
3.1	Análisis mecánico del robot	69
3.1.1	Clasificación y descripción del sistema mecánico	69
3.1.2	Descripción de los grados de libertad	72
3.2	Arquitectura del sistema	74
3.2.1	Arquitectura de control.....	75
3.2.2	Arquitectura eléctrica del sistema	79
3.3	Sistema de control.....	81
3.4	Análisis cinemático del robot cableado	83
3.4.1	Cinemática inversa	83
3.4.2	Cinemática directa.....	89
3.4.3	Cinemática diferencial y singularidades	92
3.5	Programación en software Delta.....	96
3.5.1	Programación del Controlador Lógico Programable Delta	97
3.5.1.1	Lenguaje de Programación en ISPSOft	98
3.5.1.2	Ayuda en pantalla	99
3.5.1.3	Generar un programa en ISPSOft v3.04	100
3.5.1.4	Configuración de la red en ISPSOft v3.04.....	107
3.5.2	Programación del Control de Movimiento Delta	110
3.5.2.1	Configuración de la red en CANopen Builder v6.02.....	111
3.5.2.2	Generar un programa en CANopen Builder v6.02	117
3.5.2.3	Compilación, carga y monitoreo en CANopen Builder	120
3.5.3	Programación de la Interfaz Usuario – Máquina Delta.....	122
3.5.3.1	Generar un programa en DOPSOft v2.00.07	122
3.5.3.2	Compilación, carga y monitoreo en DOPSOft	130
3.6	Parametrización del servo drive.....	132

3.6.1	Análisis del Control de posición del Servo drive ASDA A2-M	136
CAPÍTULO IV RESULTADOS.....		141
4.1	Cálculo de posición inicial.....	141
4.2	Programa inicial para simular en Matlab-Simulink.....	143
4.2.1	Modelo del robot paralelo activado por cables	143
4.2.2	Modelo de la cinemática inversa del CDPR	146
4.3	Simulaciones.....	149
4.4	Programa en PLC/MC/HMI.....	153
4.5	Pruebas iniciales	155
4.5.1	Análisis y revisión de primeros resultados.....	162
4.6	Ajustes finales	170
4.7	Programa final para simular.....	172
4.7.1	Simulaciones finales	172
4.8	Resultados experimentales finales	174
4.9	Tablas de resultados finales	176
4.9.1	Análisis y discusión de resultados finales.....	179
Capítulo V Conclusiones.....		185
5.1	Conclusiones.....	185
5.2	Problemas encontrados.....	188
5.3	Trabajo futuro.....	189
5.4	Publicaciones	190
Referencias.....		191
ANEXO A Especificaciones del Servomotor		195
ANEXO B Especificaciones del Servo drive.....		197
ANEXO C Arquitectura eléctrica del sistema de control		198
ANEXO D Código del programa cinemático en ISPSOft.....		200
ANEXO E Código del programa de movimiento en CANopen Builder.....		244
ANEXO F HMI en DOPSoft		254
ANEXO G Código de programa para la obtención de las coordenadas de referencia contra las reales en Matlab		255
ANEXO H Código de programa para la obtención DE gráficas de error de posición en Matlab		258
ANEXO I Artículo publicado N° 1		264
ANEXO J Artículo publicado N° 2		270
ANEXO K Participación en Congreso CICOS		283

Índice de tablas

<i>Tabla 2.1 Clasificación de los robots por la AFRI.....</i>	<i>12</i>
<i>Tabla 2.2 Clasificación de los robots según T. M. Knasel.....</i>	<i>13</i>
<i>Tabla 2.3 Clasificación de las articulaciones de un robot.....</i>	<i>21</i>
<i>Tabla 2.4 Sistema de transmisión de un robot.....</i>	<i>23</i>
<i>Tabla 2.5 Tipos de actuadores para robots</i>	<i>24</i>
<i>Tabla 2.6 Tipos de sensores para robots</i>	<i>25</i>
<i>Tabla 2.7 Otras clasificaciones del robot conforme sus características</i>	<i>29</i>
<i>Tabla 3.1 Configuración Hardware PLC Delta</i>	<i>76</i>
<i>Tabla 3.2 Configuración Hardware PLC Delta</i>	<i>77</i>
<i>Tabla 3.3 Requerimientos para configuración de red.....</i>	<i>96</i>
<i>Tabla 3.4 Proceso de ciclo de trabajo del PLC Delta</i>	<i>97</i>
<i>Tabla 3.5 Botones y sus características para la HMI</i>	<i>127</i>
<i>Tabla 3.6 Parámetros modificados para el proyecto “Control de un CDPR”</i>	<i>134</i>
<i>Tabla 3. 7 Comparación de control feedback contra feedforward</i>	<i>137</i>
<i>Tabla 4.1 Primeras pruebas: Mediciones de puntos en el plano (Px, Py) y ángulo de inclinación de la plataforma.....</i>	<i>155</i>
<i>Tabla 4.2 Primeras pruebas: Mediciones de longitudes de cables l1 y l2 en cada punto.....</i>	<i>156</i>
<i>Tabla 4.3 Primeras pruebas: Mediciones de pulsos que debe girar cada servomotor en cada punto</i>	<i>157</i>
<i>Tabla 4.4 Resultados finales: Mediciones de puntos en el plano (Px, Py) y ángulo de inclinación de la plataforma.....</i>	<i>176</i>
<i>Tabla 4.5 Resultados finales: Mediciones de longitudes de cables l1 y l2 en cada punto.....</i>	<i>177</i>
<i>Tabla 4.6 Resultados finales: Mediciones de pulsos que debe girar cada servomotor en cada punto</i>	<i>178</i>
<i>Tabla 4.7 Comparación de promedio de error de las pruebas iniciales contra las pruebas finales</i>	<i>184</i>

Índice de figuras

<i>Figura 1.1 Simulación para mostrar la evasión de obstáculos del cable del robot CALOWI</i>	3
<i>Figura 1.2 Metodología para el proyecto Control de un Robot Cableado</i>	9
<i>Figura 2.1 Robot hidráulico y su control, perteneciente al tipo A</i>	13
<i>Figura 2.2 Primer robot humanoide en el espacio, tipo 5ª generación</i>	14
<i>Figura 2.3 Robot transportador de costado de coche</i>	16
<i>Figura 2.4 Instalaciones anuales de robots industriales por región</i>	18
<i>Figura 2.5 Arquitectura típica de un robot</i>	19
<i>Figura 2.6 Esquema de la arquitectura del robot Mitsubishi RV-M1</i>	20
<i>Figura 2.7 Clasificación de los eslabones de un robot</i>	21
<i>Figura 2.8 Simplificación de tipos de articulación de un robot</i>	22
<i>Figura 2.9 a) Robot hidráulico HYD-2800 Degen Systems y su b) Diagrama esquemático, c) Tabla de alcances mínimos y máximos y, d) Obtención de grados de libertad</i>	27
<i>Figura 2.10 a) Tipos de espacio de trabajo de un robot y b) Diseño del espacio de trabajo (abajo) del robot Mitsubishi (arriba)</i>	28
<i>Figura 2.11 Manipulador serial o brazo robótico “CRYA”</i>	30
<i>Figura 2.12 Esquema de un robot paralelo</i>	31
<i>Figura 2.13 Ejemplo de un Robot paralelo activado por cables (CDPR)</i>	32
<i>Figura 2.14 Modelos cinemática directa y cinemática inversa de un robot</i>	36
<i>Figura 2.15 Funcionamiento del control cinemático (sombreado)</i>	41
<i>Figura 2.16 Diferentes trayectorias articulares para un robot SCARA a) movimiento eje a eje, b) movimiento simultáneo de ejes, c) trayectoria coordinada, d) trayectoria continua rectilínea</i>	43
<i>Figura 2.17 Sistema de control de un robot</i>	45
<i>Figura 2.18 Configuración del control proporcional</i>	47
<i>Figura 2.19 Configuración del control integral</i>	48
<i>Figura 2.20 a) Configuración del control PI y b) Diagrama a bloques del control PI en Matlab-Simulink</i>	49
<i>Figura 2.21 a) Programación del lugar de las raíces del PI y b) Gráfica del lugar de las raíces del control PI en Matlab</i>	50
<i>Figura 2.22 Configuración del control derivativo</i>	51
<i>Figura 2.23 a) Configuración del control PD y b) Diagrama a bloques del control PD en Matlab-Simulink</i>	52
<i>Figura 2.24 a) Programación del lugar de las raíces del PD y b) Gráfica del lugar de las raíces del control PD en Matlab</i>	53

<i>Figura 2.25 Diagrama a bloques del control PID en Matlab-Simulink</i>	<i>54</i>
<i>Figura 2.26 a) Programación del lugar de las raíces del PID y b) Gráfica del lugar de las raíces del control PID en Matlab.....</i>	<i>55</i>
<i>Figura 2.27 Control PID forma interactuante.....</i>	<i>56</i>
<i>Figura 2.28 Arquitectura del proceso de automatización</i>	<i>58</i>
<i>Figura 2.29 Controlador Lógico Programable marca Delta DVP20SX2</i>	<i>60</i>
<i>Figura 2.30 Servomotor y servo drive marca Delta</i>	<i>62</i>
<i>Figura 2.31 Niveles de jerarquía de redes de comunicación.....</i>	<i>63</i>
<i>Figura 2.32 Aplicación de CAN-bus para reducir la cantidad de cableado.....</i>	<i>65</i>
<i>Figura 2.33 Niveles del protocolo CANopen</i>	<i>67</i>
<i>Figura 2.34 Comparación entre protocolos de comunicación a) CANopen contra Profibus DP y b) CANopen contra DeviceNet</i>	<i>67</i>
<i>Figura 2.35 CANopen usando dispositivos Delta</i>	<i>68</i>
<i>Figura 3.1 Medidas del robot planar paralelo</i>	<i>70</i>
<i>Figura 3.2 Tambor acoplado a la flecha del servomotor</i>	<i>71</i>
<i>Figura 3.3 Robot activado por cables estudiado</i>	<i>71</i>
<i>Figura 3.4 Diagrama esquemático inicial del robot planar paralelo.....</i>	<i>72</i>
<i>Figura 3.5 Diagrama esquemático final del robot planar paralelo</i>	<i>72</i>
<i>Figura 3.6 Arquitectura del sistema de control usando Hardware Delta.....</i>	<i>74</i>
<i>Figura 3.7 Parte frontal del PLC / MC Delta DVP10MC11T</i>	<i>76</i>
<i>Figura 3.8 a) Parte lateral derecha, b) Parte superior y c) Parte lateral izquierda del DVP10MC11T</i>	<i>77</i>
<i>Figura 3.9 Distribución de dispositivos de potencia y control en tablero</i>	<i>79</i>
<i>Figura 3.10 Arquitectura eléctrica del sistema de potencia y control.....</i>	<i>80</i>
<i>Figura 3.11 Esquema a bloques del sistema de control del CDPR.....</i>	<i>82</i>
<i>Figura 3.12 Diagrama para obtención de la cinemática inversa del Robot paralelo cableado planar</i>	<i>83</i>
<i>Figura 3.13 Ciclo de trabajo del PLC Delta</i>	<i>97</i>
<i>Figura 3.14 Lenguaje de Programación "Ladder Diagram" para el PLC</i>	<i>98</i>
<i>Figura 3.15 Ayuda del ISPSOft</i>	<i>99</i>
<i>Figura 3.16 Creación de un nuevo programa en ISPSOft.....</i>	<i>100</i>
<i>Figura 3.17 Ventana principal de ISPSOft</i>	<i>101</i>
<i>Figura 3.18 Configuración del nuevo programa</i>	<i>102</i>
<i>Figura 3.19 a) Edición de símbolos y b) Tabla de símbolos creada.....</i>	<i>103</i>
<i>Figura 3.20 Colocación de contactos y bobinas.....</i>	<i>104</i>
<i>Figura 3.21 Ejemplo de una línea de programación.....</i>	<i>105</i>

<i>Figura 3.22</i>	<i>Uso de Bloques/Instrucciones de Programación.....</i>	<i>105</i>
<i>Figura 3.23</i>	<i>Configurar datos de entrada y de destino para una API.....</i>	<i>106</i>
<i>Figura 3.24</i>	<i>Registros especiales de lectura y escritura entre PLC – MC</i>	<i>106</i>
<i>Figura 3.25</i>	<i>Configuración del driver 1 Conexión RJ45.....</i>	<i>107</i>
<i>Figura 3.26</i>	<i>Configuración de comunicación por RJ45.....</i>	<i>108</i>
<i>Figura 3.27</i>	<i>Proceso de transferencia del programa al PLC.....</i>	<i>109</i>
<i>Figura 3.28</i>	<i>Modo online.....</i>	<i>109</i>
<i>Figura 3.29</i>	<i>Ventana principal de CANopen Builder</i>	<i>110</i>
<i>Figura 3.30</i>	<i>Configuración de IP Address.....</i>	<i>112</i>
<i>Figura 3.31</i>	<i>Configuración segundo drive “Ethernet”</i>	<i>112</i>
<i>Figura 3.32</i>	<i>Configuración de la red Ethernet en CANopen Builder</i>	<i>113</i>
<i>Figura 3.33</i>	<i>Ajuste de la comunicación por ethernet</i>	<i>113</i>
<i>Figura 3.34</i>	<i>Configuración de ethernet lograda</i>	<i>114</i>
<i>Figura 3.35</i>	<i>Detección del servo drive en la red CANopen.....</i>	<i>114</i>
<i>Figura 3.36</i>	<i>Configuración de “Node List” en el Controlador DVP10MC11T</i>	<i>115</i>
<i>Figura 3.37</i>	<i>Configuración del servo drive con dirección 1</i>	<i>116</i>
<i>Figura 3.38</i>	<i>Generar nuevo programa en CANopen Builder v6.02.....</i>	<i>117</i>
<i>Figura 3.39</i>	<i>Configuración del nuevo Programa en MC</i>	<i>117</i>
<i>Figura 3.40</i>	<i>Instrucciones correspondientes a las librerías en Motion Control</i>	<i>118</i>
<i>Figura 3.41</i>	<i>Entradas, salidas y comentarios en MC.....</i>	<i>119</i>
<i>Figura 3.42</i>	<i>Tipos de datos para cada bloque en sección “Ayuda”.....</i>	<i>119</i>
<i>Figura 3.43</i>	<i>Compilación del programa en CANopen Builder.....</i>	<i>120</i>
<i>Figura 3.44</i>	<i>Descarga del programa al MC</i>	<i>120</i>
<i>Figura 3.45</i>	<i>Monitoreo del programa y su ejecución en MC</i>	<i>121</i>
<i>Figura 3.46</i>	<i>Creación y configuración de la pantalla HMI</i>	<i>122</i>
<i>Figura 3.47</i>	<i>Comunicación por RJ45.....</i>	<i>123</i>
<i>Figura 3.48</i>	<i>Opciones de “Element Toolbar”</i>	<i>124</i>
<i>Figura 3.49</i>	<i>Configuración del botón en HMI.....</i>	<i>125</i>
<i>Figura 3.50</i>	<i>Finalizar la apariencia del botón.....</i>	<i>125</i>
<i>Figura 3.51</i>	<i>Dirección de escritura para el botón (marca M0)</i>	<i>126</i>
<i>Figura 3.52</i>	<i>Coordenadas y tamaño del botón</i>	<i>126</i>
<i>Figura 3.53</i>	<i>Agregar texto en la pantalla HMI.....</i>	<i>128</i>
<i>Figura 3.54</i>	<i>Elemento para introducir datos</i>	<i>129</i>
<i>Figura 3.55</i>	<i>Configuración de Numeric Entry.....</i>	<i>129</i>
<i>Figura 3.56</i>	<i>Compilación de la HMI terminada</i>	<i>130</i>
<i>Figura 3.57</i>	<i>HMI on-line Simulation</i>	<i>131</i>
<i>Figura 3.58</i>	<i>HMI para completar el proyecto de MC y PLC terminado</i>	<i>131</i>

<i>Figura 3.59 Tipos de parámetros en el servo drive</i>	<i>132</i>
<i>Figura 3.60 Conexión de servo drive al software</i>	<i>133</i>
<i>Figura 3.61 a) Entorno de ASDA Soft y b) Conexión del servo drive al software</i>	<i>133</i>
<i>Figura 3.62 Columnas de parámetros de ASDA Soft</i>	<i>134</i>
<i>Figura 3. 63 Diagrama a bloques del Control de posición Feedforward y Feedback del Servo drive</i>	<i>136</i>
<i>Figura 3.64 Position Feed Forward Gain y P2-03 Smooth Constant of Position Gain.....</i>	<i>137</i>
<i>Figura 3.65 Efectos de KPP (ganancia proporcional).....</i>	<i>139</i>
<i>Figura 4.1 Posición Home del CDPR</i>	<i>142</i>
<i>Figura 4. 2 Modelo del CDPR sección izquierda.....</i>	<i>143</i>
<i>Figura 4. 3 Configuración de dimensiones de la plataforma móvil.....</i>	<i>144</i>
<i>Figura 4. 4 Modelo completo del Robot Paralelo activado por cables planar en Matlab-Simulink .</i>	<i>145</i>
<i>Figura 4. 5 Modelación virtual de la cinemática inversa con $\theta = 0$.....</i>	<i>146</i>
<i>Figura 4. 6 Primer bloque de programación en Matlab: Cinemática Inversa.....</i>	<i>147</i>
<i>Figura 4. 7 Segundo bloque de programación en Matlab: Conversión a grados.....</i>	<i>148</i>
<i>Figura 4. 8 Simulación de tercera coordenada.....</i>	<i>149</i>
<i>Figura 4.9 Simulación de posición 'Home'</i>	<i>150</i>
<i>Figura 4.10 Simulación de la primera coordenada (0, 320).....</i>	<i>150</i>
<i>Figura 4.11 Simulación de la segunda coordenada (0, -450).....</i>	<i>151</i>
<i>Figura 4.12 Simulación de la tercera coordenada (640, 0).....</i>	<i>151</i>
<i>Figura 4.13 Simulación de la sexta coordenada (-600, 300).....</i>	<i>152</i>
<i>Figura 4.14 Simulación de la octava coordenada (-640, -450).....</i>	<i>152</i>
<i>Figura 4.15 Diagrama de Programa en PLC/HMI.....</i>	<i>153</i>
<i>Figura 4.16 Prueba inicial: a) Posición Home (0, 0). b) Plataforma móvil con 0°</i>	<i>158</i>
<i>Figura 4.17 Prueba inicial: a) Segunda coordenada (0, -450). b) Plataforma móvil con inclinación 0</i>	<i>159</i>
<i>Figura 4.18 Prueba inicial: a) Sexta coordenada (-600, 300) b) Plataforma móvil con inclinación de 21°.....</i>	<i>160</i>
<i>Figura 4.19 Prueba inicial: a) Séptima coordenada (640, -450) b) Plataforma móvil con inclinación de 47°</i>	<i>161</i>
<i>Figura 4.20 Programa para graficar posición de la plataforma móvil en Matlab.....</i>	<i>162</i>
<i>Figura 4.21 Gráfica de segundo par de coordenadas</i>	<i>163</i>
<i>Figura 4.22 Gráfica de sexto par de coordenadas</i>	<i>164</i>
<i>Figura 4.23 Gráfica de séptimo par de coordenadas</i>	<i>164</i>
<i>Figura 4.24 Programa para graficar la norma Euclídea y el error de posición de cada coordenada en Matlab</i>	<i>165</i>

<i>Figura 4.25 Gráfico del módulo de los vectores de posición deseados y reales</i>	<i>166</i>
<i>Figura 4.26 Gráfico del error de posición de la plataforma móvil</i>	<i>167</i>
<i>Figura 4.27 Error de posición en primera coordenada</i>	<i>168</i>
<i>Figura 4.28 Gráfica porcentaje de error de posición en eje x.....</i>	<i>168</i>
<i>Figura 4.29 Gráfica porcentaje de error de posición en eje 'y'</i>	<i>169</i>
<i>Figura 4.30 Bloque cinemática inversa con grado de inclinación</i>	<i>172</i>
<i>Figura 4.31 Simulación de coordenada (627.4, -14.57) con $\theta = 44.51^\circ$.....</i>	<i>173</i>
<i>Figura 4.32 Nueva prueba en (-627, -15) y un ángulo de inclinación de -42°</i>	<i>174</i>
<i>Figura 4.33 Nueva prueba en (603, 295) y un ángulo de inclinación de 21°</i>	<i>175</i>
<i>Figura 4.34 Nueva prueba en (-603, 295) y un ángulo de inclinación de -21°.....</i>	<i>175</i>
<i>Figura 4.35 Gráfica de tercer par de coordenadas en prueba final</i>	<i>179</i>
<i>Figura 4.36 Gráfica de sexto par de coordenadas en prueba final.....</i>	<i>180</i>
<i>Figura 4.37 Gráfica de séptimo par de coordenadas en prueba final.....</i>	<i>180</i>
<i>Figura 4.38 Gráfica prueba final de módulo de cada coordenada.....</i>	<i>181</i>
<i>Figura 4.39 Gráfica prueba final porcentaje de error de posición según su módulo.....</i>	<i>182</i>
<i>Figura 4.40 Gráfica prueba final porcentaje de error de posición en eje 'x'</i>	<i>183</i>
<i>Figura 4.41 Gráfica prueba final porcentaje de error de posición en eje 'y'</i>	<i>183</i>

CAPÍTULO I INTRODUCCIÓN

A lo largo de los años, el ser humano ha buscado facilitar y optimizar las operaciones de un proceso cuyo fin es generar un producto. A partir del año 1760 se crearon máquinas para automatizar los procesos, originando un incremento en la producción, así como una reducción de costos y de tiempos de fabricación; a esa época se le conoce como “Revolución Industrial”. A pesar de que las primeras máquinas industriales poseían un diseño complejo, tenían el inconveniente de hacer tareas muy sencillas y ejecutarlas una a la vez, obligando al hombre a pensar en la creación de máquinas con la característica de ejecutar movimientos parecidos a los del humano y así poder generar tareas más complicadas. Para esa misión se tomó como referencia el concepto de los autómatas, los cuales ya existían desde el año 10 D.C. y fueron denominados así por la cultura griega. Estos últimos son máquinas que imitan la figura y movimientos de un ser animado (Barrientos, Peñín, Balaguer, & Aracil, 2007); muchos de esos ingenios mecánicos fueron creados por artesanos dedicados a la relojería con el fin de entretener a la gente de la corte y ser atractivo para las ferias.

La unión de esos dos conceptos, una máquina industrial en conjunto con un autómata, dieron como resultado los inicios de una revolución tecnológica que han dado paso a la creación del robot. Fue en el año de 1921 cuando el escritor checo Karel Capek en su obra *Rossum's Universal Robot (R.U.R)* usó por primera vez la palabra robot; la popularidad del robot se debe a los matices dados por las novelas y películas de ciencia ficción, sin embargo, no hay semejanza alguna entre estos robots con los reales de tipo industrial.

En los últimos años hubo una creciente demanda de robots en distintas áreas como la industria, la medicina y el área académica. Gracias a eso, se ha dado un importante avance en la investigación y desarrollo de los robots; en el caso particular, se hablará de los robots paralelos activados por cables y se abarcarán algunos trabajos con ellos durante la última década.

1.1 ANTECEDENTES (ESTADO DEL ARTE)

Bruckmann y Pott (2013) mencionan que el origen de los robots paralelos activados por cables se remonta en EE. UU. y Japón a finales de la década de 1980, al combinar los primeros conceptos de grúas, manipuladores cinemáticos paralelos y robots activados por cables. Su gran aceptación en el campo de la robótica se debe a ciertas cualidades que, por el contrario, no poseen los robots paralelos convencionales y los manipuladores seriales. Algunas mejoras relevantes son:

- Mayor tamaño del espacio de trabajo.
- Carga útil máxima y las capacidades dinámicas causadas por la eficiencia en la transmisión de fuerzas a través de los cables.
- Tiene estructuras ligeras (baja masa del robot).
- Transportabilidad en la mayoría de los casos.
- Bajo costo de construcción.

Para ilustrar una de las mejoras anteriores, en el estudio “On the workspace of suspended cable-driven parallel robots” (Merlet, 2016) se hace referencia a la importancia del espacio de trabajo como criterio de rendimiento de un robot. En comparación con los manipuladores seriales, el cálculo del espacio de trabajo de un robot paralelo es más complejo debido al acoplamiento entre las capacidades traslacionales y de orientación. Sin embargo, comparando el espacio de trabajo de un robot paralelo convencional con el de un robot paralelo activado por cables, el primero al cual también llaman Robot Paralelo con Patas Rígidas (por sus siglas en inglés PRRL) presenta limitaciones como la carrera limitada de los actuadores (debido a sus eslabones rígidos), las rotaciones limitadas de las uniones pasivas, las singularidades y la interferencia de pierna.

Diversos estudios de los robots paralelos activados por cables se han presentado a lo largo de los últimos 10 años; ejemplo de ello se muestra en la investigación de Lahouar, Ottaviano, Zeghoul, Romdhane y Ceccarelli (2009) donde se presenta un método de planificación de rutas que simultáneamente tiene la característica de

evadir obstáculos; gracias a este método se evitan colisiones del efector final del robot llamado CALOWI cuando éste ejecuta una trayectoria tal como se muestra en la figura 1.1; estas trayectorias generalmente se realizan usando robots manipuladores seriales, sin embargo, en este estudio se adaptan esas trayectorias en el tipo de robot paralelo activado por cables debido a las mejoras que se plantean en párrafos anteriores.

El método propuesto en su estudio se compone de dos modos: el primero está activo cuando no existen obstáculos en la trayectoria del robot; por lo tanto, en este modo el robot genera una trayectoria recta hacia la meta. El segundo modo se presenta cuando sí existe un obstáculo presente en la trayectoria; durante este modo el robot busca la forma de evadir ese obstáculo, además este algoritmo tiene la capacidad de evitar colisiones de objetos con los cables del robot. Estos algoritmos no garantizan que el robot tome la ruta más corta cuando existen obstáculos, sin embargo, sí cerciora llegar a la meta sin importar cuántos objetos se encuentre el robot durante su camino.

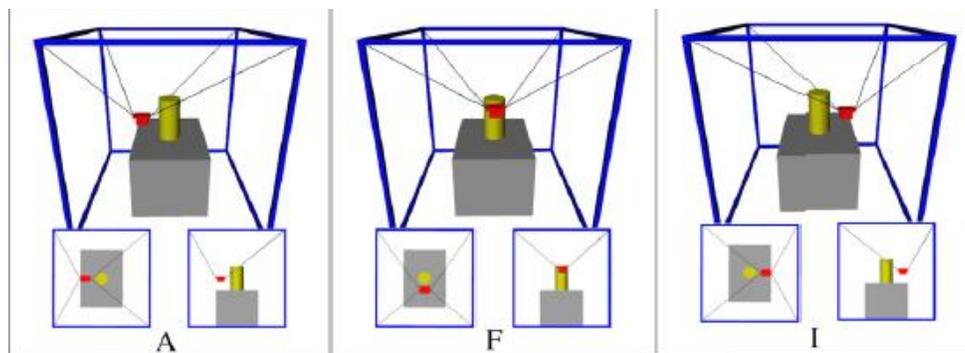


Figura 1.1 Simulación para mostrar la evasión de obstáculos del cable del robot CALOWI (Lahouar et al., 2009)

Actualmente, se ha desarrollado en gran cantidad este tipo de robots debido a sus aplicaciones en los campos de intralogística, fisioterapia y manejo industrial de materiales a gran escala. Un ejemplo de ello son los robots encaminados a la terapia y rehabilitación para personas que padecen apoplejía, pacientes con parálisis o paresia periférica y para aquellas personas que sufrieron un accidente con

consecuencias graves, tal como el deterioro de sus extremidades superiores; por consiguiente, esas personas requieren terapias como las que efectúa el robot llamado “NeReBot” creado por Massiero y su equipo de investigación (2009), el cual genera movimientos a las partes afectadas de la persona que, de acuerdo con la fisioterapia, ayudan a contrarrestar esos problemas. Este tipo de robots realiza movimientos repetitivos en las extremidades o áreas afectadas del paciente después del sufrimiento de una lesión neurológica, ayudando a mejorar significativamente la recuperación de fuerza y movimiento del paciente. Sin embargo, no se ha dedicado mucho tiempo en el estudio y aplicación de este tipo de robots que aportan grandes beneficios en el área médica.

Por otra parte, en el año 2014, se realizó un Robot Paralelo Activado por Cables Reconfigurable (llamado RCDPR por sus siglas en inglés); a diferencia de los CDPRs (Robots paralelos activados por cables) cuya arquitectura es fija, estos robots tienen la capacidad de modificar sus parámetros geométricos para adaptarse de acuerdo con sus propias características (Gagliardini, Caro, Gouttefarde, Wenger, & Girin). Al modificar las dimensiones de su propia base, este robot tiene la ventaja de poder emplearse en operaciones industriales donde se requiera trabajar con medianas o grandes estructuras. El robot de estudio de este artículo se destinó para el pintado y chorreado de una estructura tubular grande.

En el caso de los robots paralelos activados por cables, es más complicado generar un controlador con retroalimentación en comparación con los robots paralelos convencionales, porque los cables deben permanecer en tensión para todas las maniobras en su espacio de trabajo. Khosravi y Taghirad (2014) proponen un controlador PID robusto para el control del robot cableado; se presenta el método directo de Lyapunov para analizar la estabilidad robusta del sistema de circuito cerrado con el algoritmo de control propuesto. Sus pruebas las generaron con un robot paralelo activado por cables planar con tres grados de libertad, asegurando la efectividad de su algoritmo propuesto a través de diferentes trayectorias deseadas.

En la publicación de Rodríguez y su grupo de investigación (2015) se menciona la dificultad de manipular equipamientos o cargas pesadas a través de una máquina de elevación conocida como puente grúa, cuyo funcionamiento se logra mediante un operador, sin embargo, esto no asegura exactitud en la posición ni en la orientación del elemento terminal. Si se busca manipular una pieza compleja con este sistema, se emplean utillajes cuyo costo es tres veces superior al del propio puente grúa, además, los tiempos de manipulación y ensamblaje son elevados y existe un riesgo elevado de accidentes laborales. Por esta razón, en Europa se está desarrollando un robot llamado CoGiRo con la capacidad de ser controlado y actuado a través de cables, posicionando y orientando objetos pesados en una ubicación deseada sin la manipulación directa de un operario.

Otra aplicación que se le ha dado a los robots paralelos cableados se observa en el trabajo de Barry, Fisher y Vaughan (2016) donde se presenta un robot paralelo cableado para inspeccionar estructuras grandes de paredes verticales, tales como tanques de almacenamiento de químicos, grandes cascos de barcos y edificios de gran altura; sin embargo, generalmente no se usa este tipo de robots para estas tareas porque se presentan vibraciones en el sistema, provocando una disminución en la precisión de la inspección. Estas vibraciones son generadas por la naturaleza flexible de los cables y para la corrección de ese error, los autores implementan un control sobre este sistema robótico a través de la dinámica del robot.

Finalmente, en el estudio de Rushton y Khajepour (2018) se presentan algoritmos cuya finalidad es implementar aplicaciones con ausencia de vibraciones en el sistema robótico. En este caso, la eliminación de vibraciones se genera mecánicamente; posteriormente se estabiliza el sistema robótico planar utilizando solo cables y así atenuar las vibraciones de la plataforma móvil. Todas las investigaciones presentadas en este apartado se han validado científicamente, presentando opciones para crecer en este campo de la robótica.

1.2 PLANTEAMIENTO DEL PROBLEMA

Un robot cableado de tipo paralelo es un dispositivo que genera movimiento a través de sus cables. Uno de los extremos de cada cable está conectado a un actuador (2 servomotores en total con un tambor acoplado), mientras que, en el otro extremo de cada cable, se sujeta una plataforma móvil la cual es trasladada y/o rotada en un espacio limitado por sus características mecánicas. Generalmente estos sistemas son empleados en actividades industriales, sin embargo, también se les ha dado aplicaciones médicas y militares.

Los robots tienen la cualidad de desarrollar procesos sin la intervención de un usuario o con la mínima intervención de éste. Para esto es necesario un sistema de control el cual, en términos generales, recibe información del exterior (entrada o input), la procesa a través de ciertos algoritmos de control y finalmente emite esa señal procesada como una respuesta (salida u output) que nos generará diferentes comportamientos del sistema según las circunstancias.

En este momento se dispone del robot planar paralelo activado por cables tipo grúa, sin embargo, no se dispone de un sistema de control de posición para la plataforma móvil del robot. Se desconoce si es posible ordenarle al robot movimientos traslacionales de la plataforma con la siguiente especificación: que el ángulo de la plataforma móvil en todo momento sea cero, es decir, que no exista rotación alguna.

No se dispone de algoritmos para generar movimiento y que puedan ser retroalimentados por los encoders provistos por los servomotores del robot; se desconoce la viabilidad de usar solamente estos dispositivos como retroalimentación en el sistema de control. Las bondades de disponer de un sistema de control implementado en el robot activado por cables son generar movimientos deseados en la plataforma, además de tener un mayor conocimiento del sistema en general y el tipo de control adecuado para el robot.

1.3 JUSTIFICACIÓN

Se requiere implementar un sistema de control retroalimentado por encoder en un robot paralelo activado por cables en el Tecnológico Nacional de México / Instituto Tecnológico de Puebla, para evaluar la viabilidad del funcionamiento del algoritmo de posición a través de la cinemática; determinar si se logra mover la plataforma móvil según lo requiera el usuario, así como evaluar el espacio de trabajo donde la plataforma tenga traslaciones y/o rotaciones con una rotación nula.

Se determina dar solución al problema implementando un algoritmo de posición de la plataforma del robot, usando el control del servo drive, con la finalidad de observar qué beneficios aporta al sistema de control. También, se espera comprender en gran parte el funcionamiento del sistema y como consecuencia, se busca ubicar la plataforma en una posición deseada dentro del espacio de trabajo.

Es primordial mencionar que en México no existe un campo de investigación amplio referido a este tipo de robots y por consecuencia, su aplicación en la industria es escaso. El impacto de los robots paralelos activados por cables no se ha visto reflejado en este país como en otros países europeos, por consiguiente, se busca tener un sistema de control propio en un robot planar paralelo activado por cables. Así mismo, los estudios que se han realizado sobre este tipo de robots se han enfocado más a los robots espaciales o planares con cuatro cables y cuatro servomotores como articulaciones.

En general, el trabajo aportará grandes beneficios en la investigación de los robots paralelos activados por cables y también en la industria.

Derivado de ello, se propone generar el siguiente proyecto “Control de un robot activado por cables” con la determinación de dar una solución al problema que se plantea en este apartado, empleando un robot paralelo activado por cables tipo planar y usando solamente dos cables.

1.4 OBJETIVOS

General:

Diseñar e implementar un sistema de control retroalimentado por encoder, en un robot paralelo actuado por cables con el fin de controlar la posición del efector final.

Específicos:

- Investigar sobre robots activados por cables, sus características y algoritmos de control, sus aplicaciones en diversas áreas, así como los sistemas robóticos cableados existentes.
- Categorizar el tipo de robot estudiado.
- Analizar el sistema robótico cableado y obtener su modelo cinemático.
- Diseñar un algoritmo de posición retroalimentado por encoder y ubicar el efector final en su posición correcta.
- Aplicar el sistema de control al robot de estudio y realizar pruebas experimentales, así como interpretar los resultados obtenidos de la experimentación.
- Evaluar el sistema de control retroalimentado por encoder en un robot cableado y observar sus ventajas y desventajas.

1.5 METODOLOGÍA

El proyecto Control de un Robot Activado por Cables requiere de un proceso metodológico para su elaboración. Esta metodología consta de 6 etapas (ver figura 1.2); se comienza con el conocimiento del sistema robótico paralelo cableado, clasificando el robot por su tipo de movimiento, número de grados de libertad, tipo de estructura cinemática, tipo de actuación, tipo de transmisión y, geometría del espacio de trabajo.



Figura 1.2 Metodología para el proyecto Control de un Robot Cableado

La segunda etapa es el análisis y obtención de las ecuaciones matemáticas para determinar la posición deseada de la plataforma de acuerdo con la longitud de los cables del robot y la velocidad de giro de cada servomotor; esto se logrará a través del análisis de la cinemática inversa del robot. Posteriormente se procede a la simulación del posicionamiento de la plataforma móvil con respecto a la cinemática del robot en el software Simulink de Matlab; por medio de la simulación se comprobarán los resultados de las ecuaciones de la cinemática del robot. En la cuarta etapa se integrará la arquitectura de comunicación al sistema robótico, se interconectarán eléctricamente los dispositivos que conforman la etapa de potencia y la etapa de control del robot paralelo cableado de acuerdo con la Norma Oficial Mexicana NOM-001-SEDE-2012; ambas etapas conforman el tablero eléctrico del sistema. Se prosigue con el análisis de los algoritmos de movimiento usando el control del sistema. Por consiguiente, la última etapa refiere a la programación del sistema de control a través del software del controlador lógico programable marca Delta modelo DVP10MC11T y una interfaz de usuario. Se eligió ese PLC debido a su bajo costo y aplicaciones a nivel industrial. En esta etapa también se generarán las pruebas y resultados del sistema de control.

1.6 ESTRUCTURA DE LA TESIS

En el capítulo uno se presenta una explicación acerca de los antecedentes de los robots paralelos activados por cables y el control en estos sistemas; también se aborda el problema principal y se justifica el tema de tesis respecto a esa cuestión; por consiguiente, se exponen el objetivo general y los específicos, imprescindibles para erradicar el problema suscitado. Finalmente se presenta la metodología, donde se explican los pasos y métodos necesarios para atacar cada punto del tema de investigación.

La teoría crucial, que se asume como base importante para desempeñar el tema de tesis eficazmente, es presentada en el segundo capítulo. Nociones de robótica, cinemática y automatización se abarcan en esta sección; se presentan conceptos desde tipos de robots y elementos del robot, hasta la robótica industrial y la automatización.

Seguidamente, en el tercer capítulo se presenta el desarrollo de las actividades necesarias para el cumplimiento de los objetivos. En primera estancia, se analizó la estructura mecánica del robot paralelo activado por cables; se presentan los componentes del robot, así como su clasificación. De modo semejante, se analizó la arquitectura de control. Otro aspecto trascendental para alcanzar los objetivos es el estudio de la cinemática del robot. Finalmente se analiza el sistema de control y el modo de programación en los distintos dispositivos industriales que se emplearon.

En el capítulo cuatro se exponen los resultados obtenidos durante el proceso del proyecto de tesis: cálculos obtenidos de las distintas ecuaciones del modelo cinemático del robot, simulaciones del robot en el software Matlab-Simulink, programación de las ecuaciones en un controlador industrial y realización de pruebas con el robot estudiado. Finalmente, en el quinto capítulo se infieren los distintos resultados y se comenta si se lograron los objetivos iniciales. Además, se presentan propuestas para mejorar el estudio respecto al robot paralelo activado por cables estudiado.

CAPÍTULO II FUNDAMENTO TEÓRICO

En este capítulo se presentan los conceptos teóricos más importantes sobre robótica general y robots paralelos activados por cables, cuyo estudio es necesario para el desarrollo del proyecto de tesis.

2.1 ROBÓTICA

La robótica se describe como el área dedicada al estudio de la teoría, el diseño, la elaboración y la aplicación de robots. En materia académica, la robótica es un campo relativamente joven cuyos objetivos tienen un alto grado de dificultad, motivo por el cual la robótica está compuesta de muchas disciplinas, tales como la mecánica, el control, la electrónica y las TICs. La implementación de robots en diversas áreas profesionales se ha incrementado en los últimos años, principalmente en el ámbito industrial.

En *Parallel Robots* de Taghirad (2013), se define al robot como un agente artificial mecánico o virtual, generalmente un sistema electromecánico el cual, por su apariencia o movimientos, genera la sensación de tener su propio organismo. Por otra parte, la Asociación Francesa de Normalización (AFNOR) tiene un concepto más complejo del robot:

- Manipulador automático servocontrolado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectorias variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material.

En el mundo de la robótica, generalmente se usan la inteligencia artificial y la disciplina informática, pero existe una característica de mayor importancia en la robótica: el movimiento del robot en su entorno físico.

Un robot típico puede tener algunas de las siguientes propiedades:

- No es natural y ha sido creado artificialmente.
- Puede percibir su entorno.
- Puede manipular cosas en su entorno.
- Tiene cierto grado de inteligencia.
- Es programable.
- Puede moverse con uno o más ejes de movimiento.
- Parece tener intención o agencia

Actualmente existe una gran variedad de robots los cuales se han clasificado por sus diferentes propiedades. Ejemplo de ello es la clasificación establecida por la Asociación Francesa de Robótica Industrial (AFRI), la cual se muestra en la tabla 2.1. Desde otra perspectiva, T. M. Knasel propuso otra clasificación con respecto a la generación del robot, como se observa en la tabla 2.2.

Tabla 2.1 Clasificación de los robots por la AFRI.
(Barrientos et al., 2007)

<i>Tipo A</i>	Manipulador con control manual o telemando (ver figura 2.1).
<i>Tipo B</i>	Manipulador automático con ciclos preajustados; regulación mediante fines de carrera o topes; control por PLC; accionamiento neumático, eléctrico o hidráulico.
<i>Tipo C</i>	Robot programable con trayectoria continua o punto a punto. Carece de conocimientos sobre su entorno.
<i>Tipo D</i>	Robot capaz de adquirir datos de su entorno, readaptando su tarea en función de éstos.

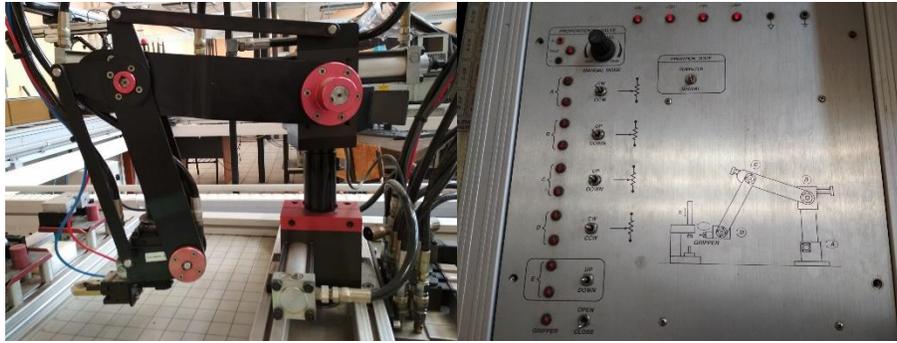


Figura 2.1 Robot hidráulico y su control, perteneciente al tipo A
(Cortesía del CIM del Instituto Tecnológico de Puebla)

Tabla 2.2 Clasificación de los robots según T. M. Knasel.
(Barrientos et al., 2007)

<i>Generación</i>	<i>Nombre</i>	<i>Tipo de control</i>	<i>Grado de movilidad</i>	<i>Usos más frecuentes</i>
1ª (1982)	Pick & place	Fines de carrera, aprendizaje	Ninguno	Manipulación, servicio de máquinas
2ª (1984)	Servo	Servocontrol, trayectoria continua, prog. condicional	Desplazamiento por vía	Soldadura, pintura
3ª (1989)	Ensamblado	Servos de precisión, visión, tacto, prog. off-line	AGV Guiado por vía	Ensamblado Desbarbado
4ª (2000)	Móvil	Sensores inteligentes	Patas Ruedas	Construcción Mantenimiento
5ª (2010)	Especiales	Controlados con técnicas de IA	Andante Saltarín	Uso militar Uso espacial (ver figura 2.2)

Gran parte del estudio generado sobre robots sirve para su posterior implementación en la industria; el área industrial tiene un fuerte impacto en la sociedad y la economía, de modo que se creó una rama importante de la robótica: la robótica industrial.



Figura 2.2 Primer robot humanoide en el espacio, tipo 5ª generación
(Robonaut 2 de National Aeronautics and Space Administration)

2.1.1 Robótica industrial

La robótica industrial es la disciplina relacionada con el diseño, el control y sus aplicaciones en la industria, cuyos productos ya han alcanzado un nivel tecnológico elevado (Siciliano, Sciavicco, Villani, & Oriolo, 2009).

Con respecto al robot industrial y su definición, distintas organizaciones lo han delimitado como se presenta a continuación:

- *Asociación de Industrias Robóticas (RIA por sus siglas en inglés):* Un robot industrial es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas.
- *Organización Internacional de Estándares (ISO por sus siglas en inglés):* Manipulador multifuncional reprogramable con varios grados de libertad,

capaz de mover materias, piezas, herramientas o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas.

- *Federación Internacional de Robótica (IFR por sus siglas en inglés)*: Por robot industrial de manipulación se entiende a una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento.

De acuerdo a las definiciones anteriores, se puede entender que un robot industrial es un dispositivo mecánico con capacidad de manipulación y que incorpora un control más o menos complejo. En cambio, un sistema robotizado es un concepto más desarrollado, el cual abarca todos aquellos dispositivos que realizan tareas de forma automática en sustitución de un ser humano y a su vez, pueden incorporar uno o varios robots (Barrientos et al., 2007).

Un robot industrial opera en un entorno estructurado cuyas características geométricas o físicas se conocen principalmente a priori, por lo tanto, la autonomía del robot tiene ciertos límites; a ese aspecto se le conoce como “espacio de trabajo de un robot”.

En la década de los 60's se crearon los primeros robots industriales, así como las máquinas de control numérico y los teleoperadores. Sus principales características son: Adaptabilidad a situaciones desconocidas anticipadas gracias al uso de sensores, precisión de posicionamiento debido al uso de técnicas de control de retroalimentación y repetibilidad en su ejecución en vista de la programabilidad de varias operaciones.

La tecnología de los robots industriales ha perfeccionado otro término fabril: la automatización. Ésta denota la sustitución del ser humano por máquinas en un proceso de fabricación y no solamente al ejecutar operaciones físicas, sino también en el procesamiento de información sobre el estado del proceso.

En un proceso de fabricación, el objeto a confeccionar debe transferirse de un punto de operación a otro sin sufrir alteración alguna; los procedimientos generales por donde pasa un producto son: almacenado, fabricado, ensamblado, terminado y embalado. El producto debe permanecer en las mejores condiciones desde su ingreso al proceso hasta su término, debido a los estándares de calidad establecidos con antelación. En la mayoría de esas operaciones se aprovecha la capacidad de los robots para levantar objetos, moverlos en el espacio y liberarlos en su destino. Algunas de las aplicaciones típicas de los robots en la industria son:

- Paletizar (colocar objetos en un palet ordenadamente).
- Cargar y descargar piezas (ver figura 2.3).
- Atender las herramientas de la maquinaria.
- Clasificar piezas.
- Embalajar.

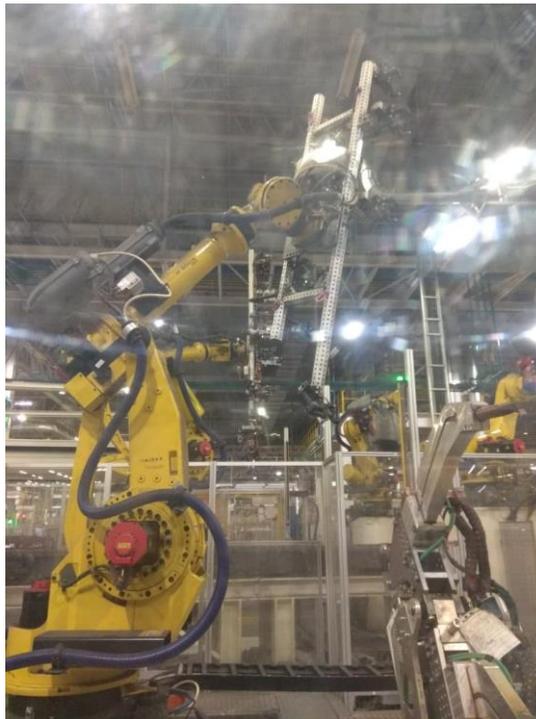


Figura 2.3 Robot transportador de costado de coche
(Cortesía de VW de México)

Por otra parte, los robots en la industria también son requeridos para generar otras tareas, igual de importantes, como son:

- Soldadura por arco y por puntos.
- Pintura y revestimiento.
- Pegado y sellado.
- Corte por láser y chorro de agua.
- Fresado y taladrado.
- Fundición y pulverización a presión.
- Desbarbado y rectificado.
- Atornillado, cableado y fijación
- Montaje de grupos mecánicos y eléctricos.
- Montaje de tableros electrónicos.
- Inspección de objetos.
- Búsqueda de contorno.
- Detección de imperfecciones de fabricación, entre otros.

Durante el pasado año 2018, las instalaciones de robots industriales en el mundo aumentaron un 6% con respecto al año anterior, el equivalente a 422,271 unidades (ver figura 2.4), por un valor de 16.5 mil millones de dólares. Eso significa una densidad promedio en la industria manufacturera de 99 robots por cada 10,000 empleados. Se debe tomar en cuenta que este promedio global incluye a los países con mayor stock de robots; Europa es la región con la mayor densidad de robots, con un valor promedio de 114 unidades. En América, el valor es de 99 unidades y en Asia/Australia es de 91 unidades. En los últimos años, debido al gran volumen de instalaciones de robots, Asia tiene la tasa de crecimiento de densidad de robots más alta y está a punto de alcanzar a las Américas. Entre 2013 y 2018, la tasa de crecimiento promedio anual de la densidad de robots en Asia fue del 16%, en América del 9% y en Europa del 6% (IFR International Federation of Robotics, 2019).

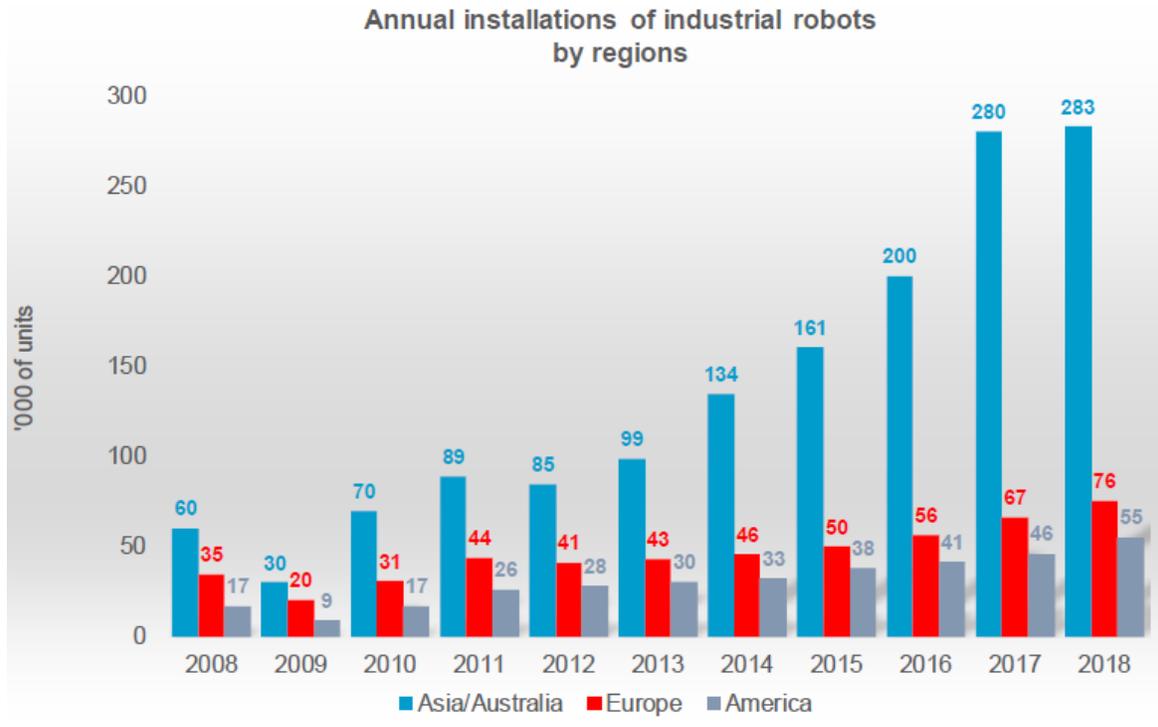


Figura 2.4 Instalaciones anuales de robots industriales por región
(IFR International Federation of Robotics, 2019)

2.2 TEORÍA DE ROBOTS

Al analizar un sistema robótico se subraya la característica de ser muy complejo, por esta razón, la funcionalidad del robot se representa por varios subsistemas. Los robots manipuladores o robots seriales, los cuales más adelante se detallarán, son el tipo de robot más común dentro del ámbito industrial; sin embargo, al mencionar la arquitectura robótica se engloba a todos los tipos de robots.

Otros conceptos teóricos de la robótica son la cadena cinemática, grados de libertad (movilidad del mecanismo robótico) y espacio de trabajo del robot.

2.2.1 Arquitectura típica de robots

Los elementos que conforman la arquitectura típica del robot son: estructura mecánica, transmisiones, sistema de accionamiento, sistema sensorial, sistema de control y elementos terminales.

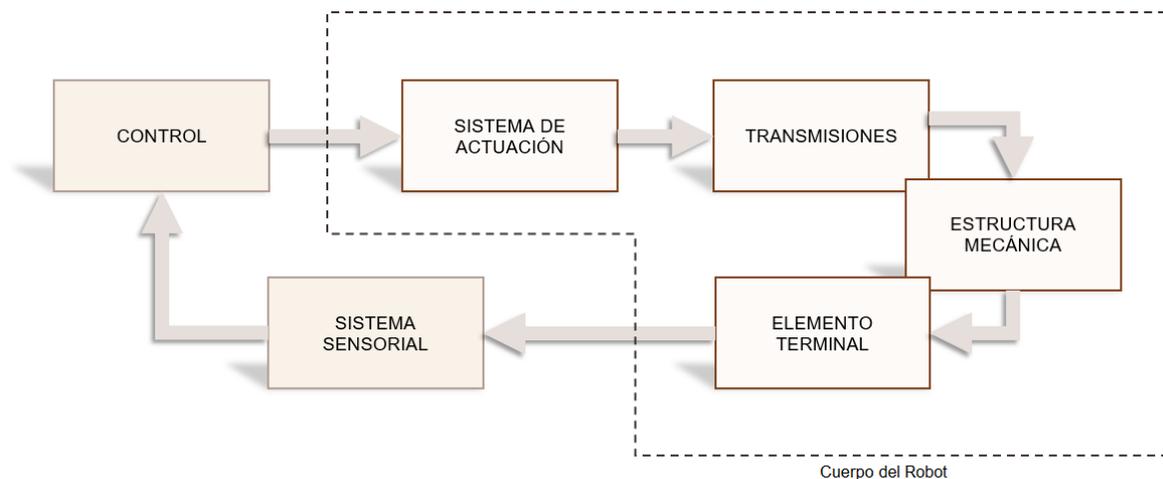


Figura 2.5 Arquitectura típica de un robot

Como se observa en la figura 2.5, el sistema robótico consta de distintas etapas; la estructura mecánica hace referencia a los tipos de articulaciones que posee un robot; el objetivo de las articulaciones es conectar los eslabones entre ellos, generando cadenas cinemáticas en el robot. Por otra parte, los sistemas de transmisión y reducción, como lo dice su nombre, transmiten a las articulaciones los movimientos generados por el sistema de actuación. El sistema sensorial es un conjunto de sensores internos que transmiten señales hacia la etapa de control; éste último se encarga de controlar los movimientos del robot a través de algoritmos programados de control. En última estancia se presenta el elemento terminal o efector final el cual se ubica generalmente en el extremo del robot y puede ser un dispositivo de sujeción, pinzas, dispositivos de corte, de soldadura, entre otras herramientas; la elección del efector final depende de la función asignada al robot. En la figura 2.6 se observa un ejemplo de Arquitectura del robot Mitsubishi.

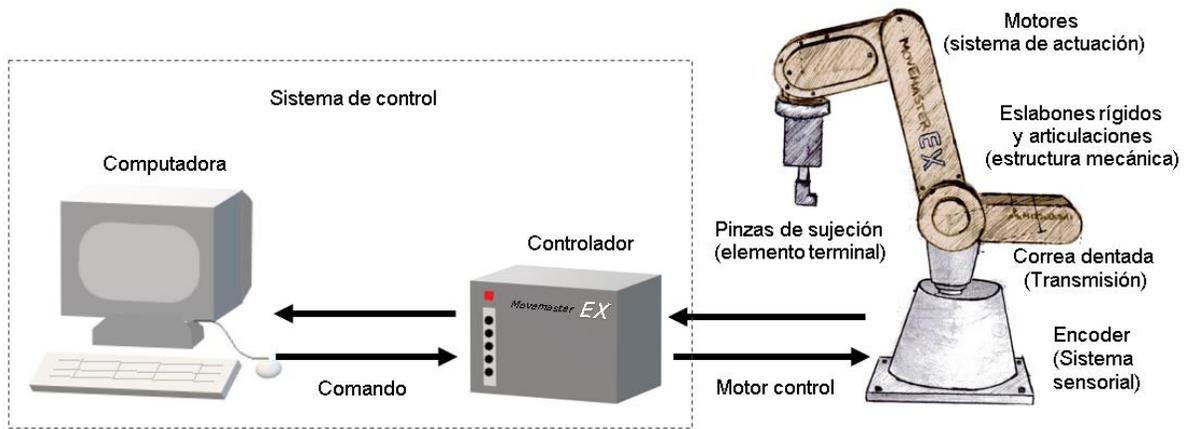


Figura 2.6 Esquema de la arquitectura del robot Mitsubishi RV-M1

2.2.1.1 Estructura mecánica y elemento terminal de un robot

La estructura mecánica del robot consiste en la conexión secuencial de componentes rígidos, también llamados eslabones, a través de uniones móviles conocidas como articulaciones; a ese conjunto de eslabones con articulaciones se le conoce como cadena cinemática del robot. Desde el punto de vista de la cinemática, si entre dos o más eslabones conectados no se produce un movimiento relativo, se considera un enlace único (Jazar, 2010).

Las características de los eslabones como la rigidez y el diseño, tienen una aportación significativa en la precisión y el rendimiento de un robot. Sin embargo, en los robots activados por cables se utilizan enlaces flexibles debido al tipo de aplicaciones y limitaciones del robot, como se verá en el capítulo siguiente; en el caso de este tipo de robots se permite suponer que los cables son cuerpos rígidos para su análisis dinámico solo si el robot se opera con alta rigidez y las tensiones internas de los cables son relativamente altas.

Las conexiones entre los eslabones del robot se llaman uniones o nodos (Taghirad, 2013). De acuerdo al número de uniones que contenga un eslabón se puede clasificar en binario, ternario o cuaternario, es decir, hay eslabones que tienen dos, tres y cuatro uniones respectivamente, como se muestra en la figura 2.7.

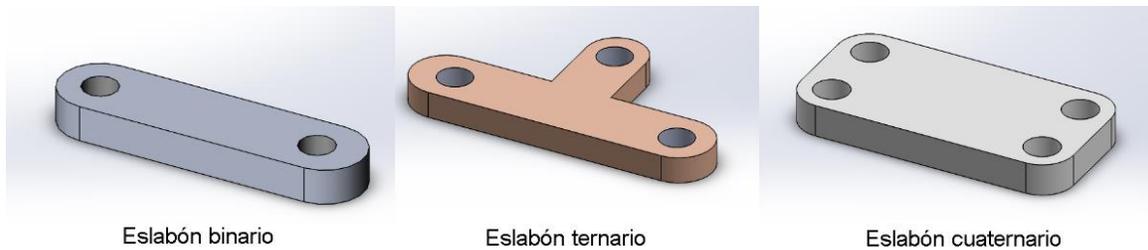
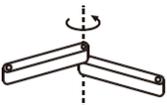
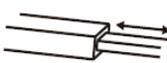
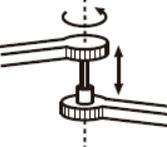


Figura 2.7 Clasificación de los eslabones de un robot

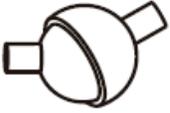
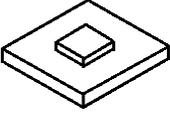
El cuerpo robótico también posee un eslabón rígido, el cual carece de movimiento y es donde se ubica el sistema de referencia, útil para el análisis cinemático del robot. A este eslabón se le denomina tierra o base.

Por otra parte, una articulación proporciona algunas restricciones físicas sobre el movimiento relativo entre los dos miembros de conexión. Debido al movimiento relativo requerido en un par cinemático, se pueden distinguir diferentes tipos de articulaciones, mostrados en la tabla 2.3.

Tabla 2.3 Clasificación de las articulaciones de un robot

Tipo de articulación	Esquema (Lynch & Park, 2017)	Símbolo	Características	Grados de libertad
Revolución		R	Articulación que solo permite la rotación alrededor de un eje entre dos eslabones.	1
Prismática		P	Articulación que permite el traslado sobre un eje entre dos eslabones.	1
Cilíndrica		C	Articulación que posibilita la rotación sobre un eje, además de la traslación independiente a lo largo de otro eje.	2

Continuación de Tabla 2.3 Clasificación de las articulaciones de un robot

Tipo de articulación	Esquema (Lynch & Park, 2017)	Símbolo	Características	Grados de libertad
Universal		T	Articulación que posibilita la rotación de dos ejes independientes. Se puede generar a través de dos articulaciones de revolución consecutivas.	2
Esférica		S	Permite la rotación libre sobre los tres ejes de un eslabón con respecto a otro, por medio de una esfera. En este caso no existe traslación de ningún tipo.	3
Planar		E	Articulación que permite dos grados de libertad traslacionales a lo largo de un plano de unión y un grado de libertad rotacional alrededor de un eje normal al plano de unión.	3

En vista de que solo existen dos tipos de movimiento en una articulación, rotacional y/o traslacional, el tipo de articulación presente en un robot se puede reducir en dos tipos (véase figura 2.8); de esta forma se simplifica el análisis del robot.

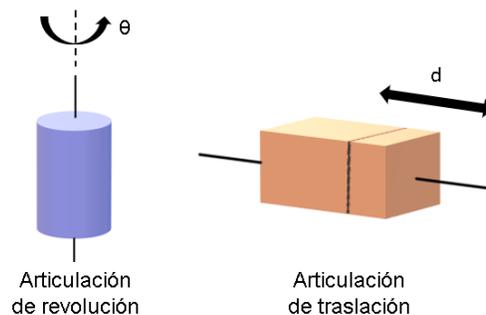


Figura 2.8 Simplificación de tipos de articulación de un robot

El efector final se encuentra conectado en el último enlace del robot, con la finalidad de generar la tarea requerida. En el caso de los robots manipuladores o seriales, el efector final se compone de una muñeca y una herramienta, que funge la acción de una mano. Las herramientas varían dependiendo de la aplicación a ejecutar, por ejemplo, se implementan pinzas, dispositivos de soldadura, de pintura, de corte, entre otros.

2.2.1.2 Sistemas de transmisión, actuación y sensorial

El sistema de transmisión es el encargado de transmitir el movimiento que genera el sistema de actuación hacia las articulaciones. Generalmente, los movimientos que provocan los dispositivos de actuación tienen grandes aceleraciones, por lo tanto, la finalidad de la transmisión es reducir al máximo el momento de inercia; otro fin de este sistema es convertir movimiento circular en lineal o viceversa, dependiendo de la aplicación del robot (ver tabla 2.4).

Tabla 2.4 Sistema de transmisión de un robot (Barrientos et al., 2007)

<i>Entrada-Salida</i>	<i>Denominación</i>	<i>Ventajas</i>	<i>Inconvenientes</i>
<i>Circular-Circular</i>	Engranaje	Pares altos	Holguras
	Correa dentada	Distancia grande	
	Cadena		Ruido
	Paralelogramo		Giro limitado
	Cable		Deformabilidad
<i>Circular-Lineal</i>	Tornillo sin fin	Poca holgura	Rozamiento
	Cremallera	Holgura media	
<i>Lineal-Circular</i>	Parale. articulado	Holgura media	Control difícil
	Cremallera		Rozamiento

Los actuadores, por otra parte, son dispositivos que generan movimiento para cambiar la configuración del robot; es decir, son los encargados de modificar la ubicación geométrica del efector final del robot a través de su potencia, actuando contra fuerzas externas como la gravedad, la inercia, entre otras. Estos dispositivos deben ser controlables y se clasifican en: eléctricos, hidráulicos o neumáticos. La elección del tipo de actuador para el diseño y la construcción de un robot se determina a partir de la aplicación industrial requerida, así como las características convenientes e inconvenientes que se presentan en la tabla 2.5.

Tabla 2.5 Tipos de actuadores para robots (Barrientos et al., 2007)

	<i>Neumático</i>	<i>Hidráulico</i>	<i>Eléctrico</i>
<i>Energía</i>	Aire a presión (5-10 bar)	Aceite mineral (50-100 bar)	Corriente eléctrica
<i>Opciones</i>	Cilindros Motor de paletas Motor de pistón	Cilindros Motor de paletas Motor de pistones axiales	Motor C.D. Motor C.A. Motor paso a paso
<i>Ventajas</i>	Baratos Rápidos Sencillos Robustos	Rápidos Alta relación potencia- peso Autolubricantes Alta capacidad de carga Estabilidad frente a cargas estáticas	Precisos Fiables Fácil control Sencilla instalación Silenciosos
<i>Desventajas</i>	Dificultad de control continuo Instalación especial (compresor, filtros) Ruidoso	Difícil mantenimiento Instalación especial (filtros, eliminación aire) Frecuentes fugas Caros	Potencia limitada

El sistema sensorial se dedica a detectar variables del mecanismo robótico para su posterior configuración. Entonces, un sensor es un dispositivo que se ajusta en el robot, el cual toma información de una o más variables que se requieren medir y, seguido las interpreta el sistema de control, el cual se definirá más adelante; así el usuario podrá manipular el robot de acuerdo a los requerimientos predefinidos. Los sensores más comunes se presentan en la tabla 2.6.

Tabla 2.6 Tipos de sensores para robots

<i>Sensores</i>	<i>Descripción</i>	<i>Variables medidas</i>	<i>Tipos</i>
<i>Internos</i>	Captan la información relacionada con el estado del robot	Posición	Encoder incremental y absoluto Potenciómetro LVDT (transformador diferencial de variación lineal) Resolver
		Velocidad	Sensores de posición Tacómetro Efecto Hall
		Aceleración y fuerza	Galgas extensiométricas Interruptores de efecto Hall Interruptores piezoeléctricos
<i>Externos</i>	Captan la información relacionada con el entorno del robot	De contacto	Interruptores de límite Interruptores neumáticos Sensores piezoeléctricos Transductores de presión
		Sin contacto	De proximidad (inductivos, capacitivos, ópticos) Efecto Hall De microondas, ultrasónicos Láser y visión

2.2.2 Grados de libertad y espacio de trabajo de un robot

Todo robot existente tiene ciertas restricciones mecánicas que limitan el movimiento relativo entre dos eslabones. Los grados de libertad de un mecanismo (*g.d.l.* o *d.o.f.* por sus siglas en inglés) son el número total de variables independientes necesarias para definir la configuración de un robot manipulador.

El análisis realizado para determinar el número de grados de libertad de un robot es a través del número de enlaces y, del número y tipo de articulaciones presentes en el robot. Debe tomarse en cuenta que una articulación con n grados de libertad se puede interpretar como la unión de n articulaciones binarias con un grado de libertad; por ejemplo, en el caso de la articulación esférica s , puede reescribirse como tres articulaciones binarias independientes rotativas, cuyos ejes se cortan.

Se define una variable λ , representando los grados de libertad de los movimientos que ocurren en un espacio, es decir para un mecanismo planar $\lambda = 3$ y para un mecanismo espacial $\lambda = 6$. Entonces, los grados de libertad de un robot se encuentran a partir de la siguiente relación:

$$n = \lambda L - \sum_{i=1}^J (\lambda - f_i) \quad \text{ecuación 2.1}$$

Donde:

n = Número de grados de libertad

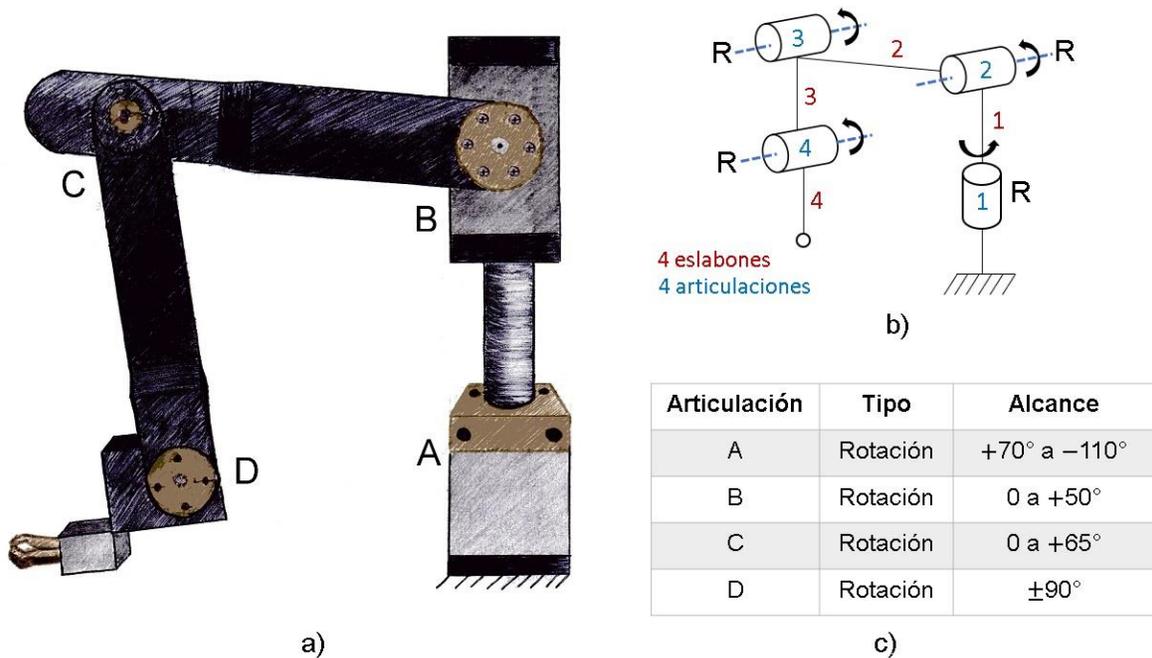
λ = Número de grados de libertad máximos: 3 planar y 6 espacial

L = Número de eslabones móviles

J = Número de articulaciones

f_i = Número de grados de libertad de la articulación i

Para la obtención del número de grados de libertad se puede utilizar un gráfico ilustrativo, tal como se observa en la figura 2.9.



Datos:

$$\lambda = 6$$

$$L = 4$$

$$J = 4$$

$$f_1 = f_2 = f_3 = f_4 = 1$$

Cálculo de grados de libertad:

$$n = \lambda L - \sum_{i=1}^J (\lambda - f_i)$$

$$n = ((6)(4) - 4(6 - 1))g. d. l.$$

$$n = (24 - 20)g. d. l.$$

$$\boxed{n = 4 g. d. l.}$$

d)

Figura 2.9 a) Robot hidráulico HYD-2800 Degen Systems y su b) Diagrama esquemático, c) Tabla de alcances mínimos y máximos y, d) Obtención de grados de libertad.

Como se observó en la figura anterior, el robot HYD-2800 de Degen Systems tiene 4 g.d.l. de acuerdo a la ecuación 2.1, porque sus 4 articulaciones (A-D) son del tipo rotacional, es decir, el número de grados de libertad que tienen las articulaciones de forma independiente es: $f_{1-4} = 1$. En el caso de existir una articulación con más de un grado de libertad, el esquema de esa articulación se representará como la secuencia de varias articulaciones rotacionales y/o traslacionales.

El espacio de trabajo de un robot es un aspecto importante para el análisis robótico; éste es el conjunto de todas las posiciones y orientaciones del elemento terminal. El espacio de trabajo se determina conforme a los rangos físicos admisibles de los eslabones y articulaciones del robot, es decir, está limitado por la geometría del robot y las restricciones mecánicas de sus elementos.

Existen dos tipos de espacio de trabajo de un robot: el diestro y el accesible. El espacio diestro es un subconjunto del tipo accesible, como se muestra en la figura 2.10. El espacio de trabajo accesible es el volumen donde el efector final accede a cada punto y mínimo una orientación. Mientras que el espacio de trabajo diestro es el volumen dentro del cual el elemento terminal puede alcanzar cada punto en todas las orientaciones posibles.

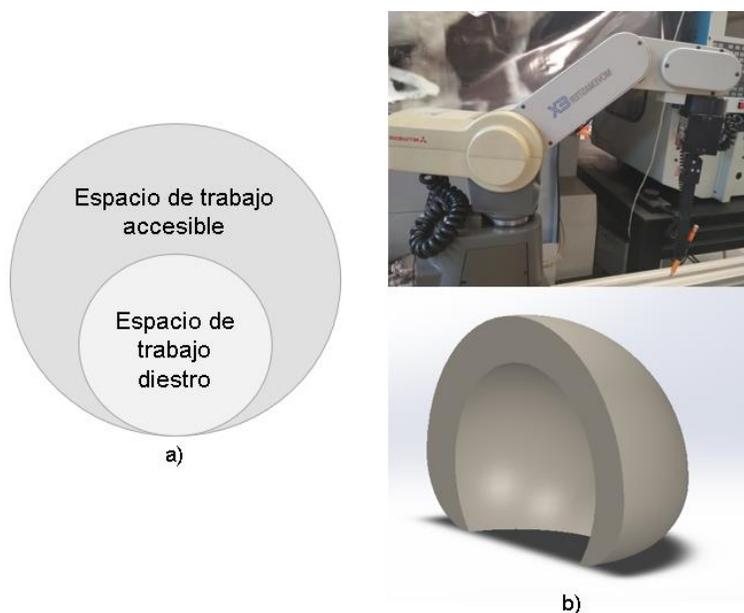


Figura 2.10 a) Tipos de espacio de trabajo de un robot y
b) Diseño del espacio de trabajo (abajo) del robot Mitsubishi (arriba)

El cálculo del espacio de trabajo puede realizarse a través de dos métodos: el gráfico y el analítico. El primero se genera dibujando los rangos máximos y mínimos de las articulaciones, mientras que el segundo método se da a través de ecuaciones que representen el espacio de trabajo, tal como las integrales (área bajo la curva).

2.2.3 Tipos de robots

En apartados anteriores, se muestran algunas clasificaciones oficiales de robots industriales de acuerdo al concepto de diversas instituciones o autores dedicados a la investigación de la robótica. Sin embargo, los robots también pueden clasificarse conforme a sus diversos parámetros, como se presentan en la tabla 2.7.

Tabla 2.7 Otras clasificaciones del robot conforme sus características

<i>Clasificaciones</i>	<i>Tipos de robots</i>
<i>Tipo de movimiento</i>	Planar, esférico y espacial.
<i>Número de grados de libertad</i>	De propósito general ($n = \lambda$), redundante ($n > \lambda$) y deficiente ($n < \lambda$).
<i>Estructura cinemática</i>	Serial, paralelo e híbrido.
<i>Tipo de actuación</i>	Eléctrico, hidráulico y neumático.
<i>Tipo de transmisión</i>	Directa y convencional.
<i>Geometría del espacio de trabajo</i>	Cartesiano, cilíndrico, esférico e irregular.

Una de las clasificaciones más usuales es por el “número de grados de libertad del robot”. Por ejemplo, en la industria se utilizan robots que manipulan objetos en el espacio libremente, por lo tanto, se requiere tengan seis grados de libertad. Por otra parte, es suficiente que un robot para aplicaciones de ensamblado tenga cuatro grados de libertad. Un robot redundante es un robot con más actuadores que los grados de libertad requeridos y un robot deficiente es uno con menos actuadores que los grados de libertad requeridos. Por ende, un robot redundante proporciona más libertad para mover un objeto mientras evita los obstáculos o singularidades que pueden ocurrir dentro del espacio de trabajo del robot (Taghirad, 2013).

Otra forma de clasificación común del robot es por su cadena o estructura cinemática, del cual derivan los tipos de robot serial, paralelo e híbrido.

2.2.3.1 Robots seriales

Acorde a su estructura cinemática, un robot serial o también denominado un manipulador tiene una cadena cinemática de lazo abierto, puesto que uno de sus eslabones se encuentra conectado a tierra, pero nunca cierra su cadena, sino que finaliza su cadena con el efector final el cual se encuentra suspendido. Son conocidos comúnmente como brazos robóticos debido a su parecido con el brazo humano, pues sus articulaciones conforman un hombro, un codo y una muñeca generando esas mismas funciones (ver figura 2.11). Este tipo de robots es el más común en la industria; la aplicación más popular en los manipuladores con cuatro grados de libertad es el ensamblaje; a este robot especial con 4 g.d.l. que cumple este tipo de tareas se le denomina robot SCARA.

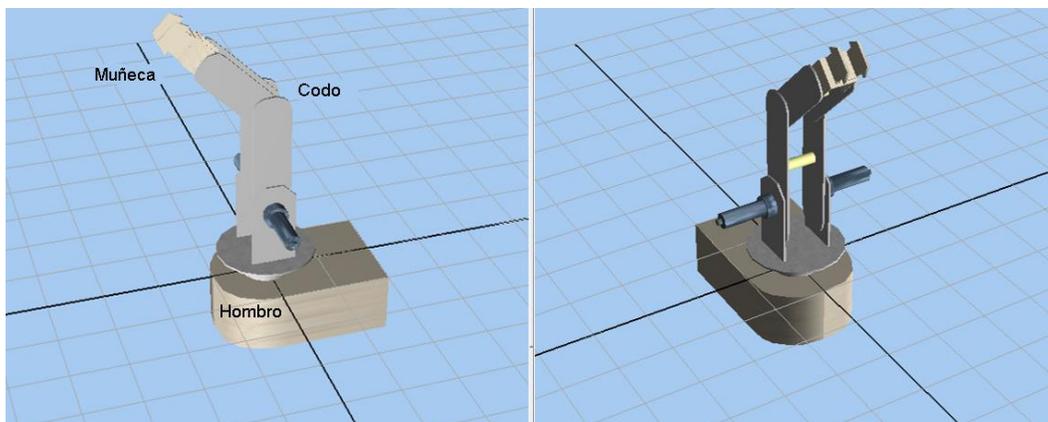


Figura 2.11 Manipulador serial o brazo robótico “CRYA”

La ventaja más significativa del robot serial es tener un espacio de trabajo grande, respecto al volumen del robot y espacio ocupado. Sin embargo, presenta ciertas desventajas, principalmente su baja rigidez causada inherentemente por su estructura cinemática abierta (Taghirad, 2013). Entre otras desventajas se encuentran los errores de posicionamiento acumulativos de enlace a enlace; tienen una relación masa-carga relativamente alta además de una capacidad de carga útil efectiva relativamente baja, teniendo como consecuencia robots con un mínimo de seis grados de libertad para colocar un objeto en el espacio de trabajo del robot.

2.2.3.2 Robots paralelos

Por el contrario, un robot paralelo (ver figura 2.12) es un mecanismo de cadena cinemática de lazo cerrado, cuyo efector final es una plataforma unida por varias cadenas cinemáticas independientes (eslabones). En el caso de una cadena abierta todas sus articulaciones son actuadas, sin embargo, en el caso del tipo de robots paralelos cuya cadena cinemática es cerrada, solo un subconjunto de sus articulaciones puede ser actuadas.

Generalmente sus extremos consisten en un tipo de articulación prismática activada conectada a la plataforma móvil por medio de articulaciones pasivas esféricas y/o universales, es decir que no son actuadas. Gracias a esa configuración, los enlaces sienten solo tracción o compresión, no flexión, aumentando su precisión de posición y permitiendo una construcción más ligera. Los robots paralelos tienen además una gran rigidez estructural, porque la plataforma móvil está soportada por varias extremidades al mismo tiempo. Debido a estas características se obtiene una amplia gama de capacidades de movimiento. Su mayor inconveniente es su espacio de trabajo limitado porque las extremidades pueden colisionar y, además, cada extremidad tiene una serie de articulaciones pasivas que tienen sus propios límites mecánicos. Otra desventaja de los robots paralelos es que pueden perder por completo su rigidez en posiciones singulares (configuraciones que limitan el movimiento del robot), por lo tanto, el robot obtiene grados adicionales de libertad, que son incontrolables, volviéndose inestable o móvil en estas posiciones.

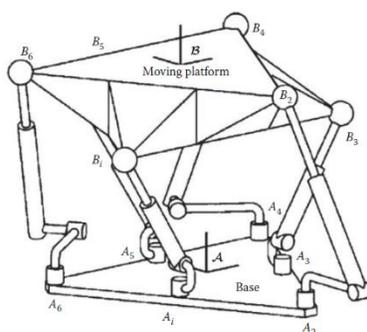


Figura 2.12 Esquema de un robot paralelo (Taghirad, 2013)

2.2.3.3 Robots paralelos activados por cables

Los robots paralelos activados por cables son una clase especial de los robots paralelos, en donde los eslabones extensibles rígidos son reemplazados por cables accionados (Khosravi & Taghirad, 2014). Como mencionan Gagliardini y Caro (2014), se les nombra Robots Paralelos Accionados por Cables (por sus siglas en inglés CDPRs) a aquellos que están conformados por una plataforma móvil conectada a una base fija por medio de cables, dicho en otras palabras, son robots cuyo efector final está conectado a la base (plataforma fija) mediante múltiples cables que modifican su longitud gracias al accionamiento de motores, como se muestra en la figura 2.13. En efecto, la plataforma móvil se mueve hacia la posición y orientación deseadas por actuadores, extendiendo o retrayendo los cables.

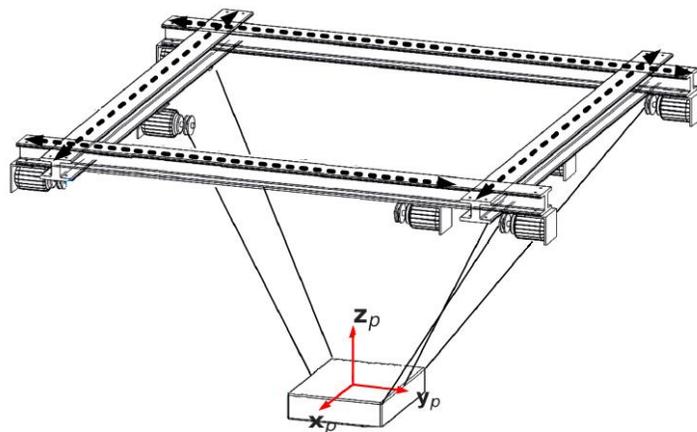


Figura 2.13 Ejemplo de un Robot paralelo activado por cables (CDPR)
(Gagliardini et al., 2014)

Los robots paralelos basados en cables son un tipo de manipulador robótico que recientemente ha atraído el interés por tareas de manipulación de grandes espacios de trabajo (Liu, Qiu, Su, & Chang, 2014). Muestran varias ventajas prometedoras sobre sus contrapartes de enlace rígido, como una estructura mecánica simple y ligera, inercia de momento bajo, gran espacio de trabajo accesible y movimiento a alta velocidad. Por lo tanto, una amplia variedad de robots basados en cable se ha utilizado en rehabilitación médica, transporte de material, experimentos en túneles

de viento, observación astronómica y otros campos. Sin embargo, los robots paralelos basados en cables sacrifican cierta precisión y estabilidad debido al uso de cables en lugar de enlaces rígidos.

Basado en el número de cables (m) y el número de grados de libertad (n), los CDPR se pueden clasificar en tres tipos (movilidad y estática): incompletamente restringidos cuando $n + 1 > m$, completamente restringidos cuando $n + 1 = m$ y redundantemente restringido cuando $n + 1 < m$.

En el tipo totalmente restringido, los cables pueden crear cualquier llave tirando del efector final o de manera equivalente, para un conjunto dado de longitudes de cable, el efector final no se puede mover en la posición y orientación. Para restringir completamente un CDPR, el número de cables que impulsan el efector final debe ser al menos uno más que el número de grados de libertad del robot. Un CDPR no tiene restricciones si la posición y orientación del efector final en el robot está determinada solo por su gravedad. Mientras que, si la posición y orientación del efector final está completamente determinada por las longitudes de los cables, el CDPR está limitado total o redundantemente.

Como se presentó anteriormente, los mecanismos paralelos con cadenas cerradas consisten en fijar y mover plataformas móviles conectadas por un conjunto de patas. Las patas en sí mismas son típicamente cadenas abiertas, pero a veces también pueden ser otras cadenas cerradas (como el robot Delta). Entonces, al analizar los CDPRs deben ser estudiados como robots paralelos, puesto que sus cables actúan como eslabones y entonces, se determina que la cadena cinemática de los robots paralelos accionados por cables es cerrada.

A diferencia de los robots paralelos convencionales, los CDPRs tienen fuerzas de tensión o flexión, mientras que no poseen fuerzas de tracción ni compresión; los cables solo pueden ser conducidos por tensión positiva para mantener la forma de la línea recta en lugar de la compresión negativa.

2.3 CINEMÁTICA

Como menciona Jazar (2010), la cinemática se describe como la ciencia de la geometría en movimiento, limitándose a una descripción geométrica pura del movimiento por medio de la posición, la orientación y sus derivadas de tiempo (velocidades y aceleraciones), sin considerar las fuerzas que originan dicho movimiento.

Las cadenas cerradas admiten una variedad mucho mayor de diseños que las cadenas abiertas y su análisis cinemático y estático es, por consiguiente, más complicado. Esta complejidad puede atribuirse a dos características definitorias de las cadenas cerradas: primero, no todas las uniones se accionan y, segundo, las variables conjuntas deben satisfacer una serie de ecuaciones de restricción de cierre de lazo, que pueden ser independientes o no dependiendo de la configuración del mecanismo.

La presencia de uniones no activadas (o pasivas), aunado al hecho de que el número de uniones actuadas puede diseñarse deliberadamente para exceder los grados de libertad cinemáticos del mecanismo (se dice que dichos mecanismos se activan de manera redundante) no solo hace que el análisis cinemático sea más desafiante puesto que también introduce nuevos tipos de singularidades que no están presentes en cadenas.

Una observación general que se puede hacer para los mecanismos en serie frente a los mecanismos paralelos es la siguiente: para las cadenas en serie, la cinemática directa es generalmente sencilla, mientras que la cinemática inversa puede ser compleja (por ejemplo, puede haber múltiples soluciones o ninguna solución). Para los mecanismos paralelos, la cinemática inversa es a menudo relativamente sencilla, mientras que la cinemática directa puede ser bastante compleja: un conjunto de valores elegidos arbitrariamente puede no ser factible o puede corresponder a múltiples configuraciones posibles de la plataforma (Lynch & Park, 2017).

2.3.1 Cinemática directa

Para representar y describir la ubicación de un objeto en tres dimensiones, tomando como base un sistema de referencia fijo, se requiere emplear el álgebra vectorial y matricial. En los robots seriales, generalmente, el sistema de referencia se ubica en el primer eslabón rígido, es decir, su base y posteriormente se describe la localización de cada eslabón con respecto al sistema de referencia. En contraparte, el sistema de referencia en los robots paralelos se ubica en el efector final.

A partir de un conjunto de parámetros físicos que definen la geometría de un robot dado, así como de los ángulos (rotacional) y/o desplazamientos (traslacional) articulares se halla la posición y orientación del efector final en el espacio tridimensional. Para su obtención, se requiere tener las variables de articulación de un robot y determinar la posición y orientación de cada eslabón o enlace del robot. Finalmente, a través de un método de movimiento rígido en el cual se adjunta un marco de coordenadas a cada enlace y se determina su configuración en los sistemas vecinos, se puede encontrar la posición y orientación del efector final del robot; existe una solución única. A tal análisis de le denomina cinemática directa.

La resolución para el problema de la cinemática directa puede darse a través de matrices de transformación homogénea, por el algoritmo de Denavit-Hartenberg o mediante el uso de cuaternios. Por lo tanto, se puede representar a la cinemática directa como se muestra en las ecuaciones 2.2 a – f:

$$\begin{aligned} Px &= f_x(q_1, \dots, q_i) & \alpha &= f_\alpha(q_1, \dots, q_i) \\ Py &= f_y(q_1, \dots, q_i) & \beta &= f_\beta(q_1, \dots, q_i) \\ Pz &= f_z(q_1, \dots, q_i) & \gamma &= f_\gamma(q_1, \dots, q_i) \end{aligned} \quad \text{ecuación 2.2 a) – f)}$$

Donde: i es el número de grados de libertad del robot.

$f_x, f_y, f_z, f_\alpha, f_\beta$ y f_γ son las funciones del modelo C.D. del robot.

2.3.2 Cinemática inversa

La cinemática inversa se presenta a partir de un conjunto de parámetros físicos que definen la geometría de un manipulador dado, así como de una posición y orientación específica del efector final se halla el conjunto de ángulos y/o desplazamientos articulares que dan como resultado dicha posición y orientación del efector final.

Para los robots en serie, el mapeo desde las posiciones conjuntas hasta la posición y orientación del efector final es fácil, mientras que el mapeo inverso es más difícil. Por el contrario, para un robot paralelo el mapeo inverso es relativamente sencillo de analizar, sin embargo, el mapeo directo se complica si se inicia con su análisis. Por ende, en este caso es más usual el modelo cinemático inverso, debido a que las tareas más frecuentes en los manipuladores implican posicionamiento de su efector final o de alguna otra de sus partes. La cinemática inversa se puede representar como la ecuación 2.3 (Barrientos et al., 2007):

$$q_k = f_k(x, y, z, \alpha, \beta, \gamma) \quad \forall \quad k = 1 \dots n \text{ (g. d. l.)} \quad \text{ecuación 2.3}$$

Las características cinemáticas de un CDPR se analizan de forma diferente a los manipuladores paralelos convencionales, por las características intrínsecas de los cables que deben estar en tensión y con valores de fuerzas limitados. En resumen, en la figura 2.14 se representan los modelos de cinemática directa y cinemática inversa para un robot.



Figura 2.14 Modelos cinemática directa y cinemática inversa de un robot

2.3.3 Cinemática diferencial

La cinemática diferencial corresponde, en términos matemáticos, a la derivada de la variable principal que se analiza en la cinemática directa e inversa: la posición. Por consiguiente, la variable examinada en la cinemática diferencial es la velocidad. La cinemática diferencial directa, de forma semejante a la cinemática directa de posición, proporciona las velocidades del efector final del robot a partir de las velocidades instantáneas de las articulaciones. En contraparte, la cinemática diferencial inversa se utiliza para conocer las velocidades de las articulaciones a partir de la velocidad del elemento terminal.

En el caso de los robots de cadena cinemática abierta, el objetivo de la C.D.D. es relacionar las velocidades de todas las articulaciones con la velocidad del efector final, sin embargo, el análisis de la C.D.D. en los robots paralelos convencionales y cableados se complica por el hecho de que no todas las uniones se accionan. Por ello, solo las articulaciones accionadas pueden prescribir velocidades de entrada, mientras que las velocidades de las uniones pasivas se determinan a partir de las ecuaciones de restricción cinemática y se requieren para determinar finalmente la torsión del marco del efector final del robot paralelo.

La relación entre el vector de velocidades de articulación y las velocidades lineales y angulares del efector final de los robots paralelos es muy útil y está definido por las ecuaciones de restricción cinemática; éste se llama Jacobiano y desempeña un papel importante en la velocidad y el análisis estático del robot paralelo. El Jacobiano de la cinemática inversa puede obtenerse directamente a través del análisis estático. Véase en la ecuación 2.4 la definición de la matriz jacobiana.

La función $f: \mathbf{R}^n \rightarrow \mathbf{R}^m$ se define como:

$$\mathbf{x} = f(\mathbf{q}) \leftrightarrow \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} f_1(q_1, q_2, \dots, q_n) \\ f_2(q_1, q_2, \dots, q_n) \\ \vdots \\ f_m(q_1, q_2, \dots, q_n) \end{bmatrix}$$

Basado en las reglas de diferenciación:

$$dx = \sum_{i=1}^n \frac{\partial f}{\partial q_i} dq_i \leftrightarrow \begin{bmatrix} dx_1 \\ dx_2 \\ \vdots \\ dx_m \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \frac{\partial f_1}{\partial q_2} & \dots & \frac{\partial f_1}{\partial q_n} \\ \frac{\partial f_2}{\partial q_1} & \frac{\partial f_2}{\partial q_2} & \dots & \frac{\partial f_2}{\partial q_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial q_1} & \frac{\partial f_m}{\partial q_2} & \dots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} = [J]_{m \times n} \begin{bmatrix} dq_1 \\ dq_2 \\ \vdots \\ dq_n \end{bmatrix} \quad \text{ecuación 2.4}$$

La matriz J es llamada matriz Jacobiana o Jacobiano. Esta matriz representa una transformación lineal para un robot con n g.d.l. la cual analiza las velocidades espaciales articulares y a través de la ecuación 2.5 de la cinemática diferencial directa, se obtienen las velocidades espaciales de tareas del efector final.

$$\dot{P} = J(q)\dot{q} \quad \text{ecuación 2.5}$$

Donde:

- \dot{P} : Es la matriz de velocidades del efector final con respecto al sistema de referencia (velocidades lineares y velocidades angulares).
- $J(q)$: Es la matriz jacobiana \rightarrow El método directo para obtener el Jacobiano es a través de las derivadas de las ecuaciones de cinemática directa.
- \dot{q} : Es la matriz de velocidades de articulaciones.

Por otra parte, para el cálculo de la cinemática diferencial inversa (C.D.I.) se dan diversos casos para su solución de acuerdo a la clasificación por número de grados de libertad del robot, ya vista con antelación.

- $m = n$, hay un número finito de solución a la ecuación 2.6 de C.D.I:

$$\dot{q} = (J(q))^{-1} \dot{P} \quad \text{ecuación 2.6}$$

- $m > n$, significa que no hay suficientes grados de libertad en el robot, por lo tanto, podría no existir solución al problema.
- $m > n$, hay más grados de libertad en el robot, por consiguiente, hay una infinidad de soluciones al problema y significa que hay redundancia en el sistema robótico.

2.3.4 Singularidades

La C.D.I. de un robot se obtiene a través de la ecuación 2.6, con la condición de que no existan configuraciones singulares. Matemáticamente, una configuración singular es aquella donde el determinante del Jacobiano J es nulo, como se muestra en la ecuación 2.7.

$$\det[J(\mathbf{q})] = 0 \rightarrow \text{existe singularidad} \quad \text{ecuación 2.7}$$

Físicamente, las singularidades se presentan como configuraciones donde se reduce la movilidad del sistema robótico, por ende, no es posible obligar al robot a que su efector final ejecute cualquier movimiento; otra forma de determinar una configuración singular es cuando existen infinitas soluciones al problema de la cinemática inversa; finalmente, en la vecindad de una singularidad, pequeñas velocidades en el espacio operativo pueden causar grandes velocidades en el espacio articular (Siciliano et al., 2009).

Existen dos tipos de singularidades: las singularidades de límite y las singularidades internas. El primer tipo se presenta cuando el robot se encuentra extendido o retraído; este tipo de singularidades se evita limitando al robot para que no llegue más allá de su espacio de trabajo accesible. Por otra parte, están las singularidades internas que ocurren dentro del espacio de trabajo accesible y generalmente son causadas por la alineación de dos o más ejes de movimiento; también deben su causa a las configuraciones particulares del efector final. En este caso sí se

presenta un problema grave porque las singularidades se pueden encontrar en cualquier parte del espacio de trabajo accesible para una ruta planificada en el espacio operativo.

2.3.5 Cinemática de aceleración

La cinemática respecto a la aceleración del sistema robótico, también se divide en cinemática de aceleración directa e inversa. En el caso de un robot con n grados de libertad, su cinemática de aceleración directa (C.A.D.) se expresa a través de la derivada del Jacobiano, la cual mapea las aceleraciones espaciales de las articulaciones y así obtener las aceleraciones espaciales de las posiciones y orientaciones del efector final. A través de derivar la ecuación 2.5 se obtiene la C.A.D., mostrada en la ecuación 2.8.

$$\begin{aligned}\dot{\mathbf{P}} &= \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \\ \ddot{\mathbf{P}} &= \frac{d\dot{\mathbf{P}}}{dt} = \frac{d}{dt}(\mathbf{J}(\mathbf{q})\dot{\mathbf{q}}) \\ \ddot{\mathbf{P}} &= \dot{\mathbf{J}}(\mathbf{q})\dot{\mathbf{q}} + \mathbf{J}(\mathbf{q})\ddot{\mathbf{q}}\end{aligned}\tag{ecuación 2.8}$$

Para la obtención de la cinemática de aceleración inversa (C.A.I.) se toma como base la ecuación 2 y se obtiene la nueva ecuación 2.9, la cual muestra que, dadas las aceleraciones del efector final, se pueden obtener las aceleraciones de las articulaciones, si es que éstas existen.

$$\begin{aligned}\dot{\mathbf{q}} &= (\mathbf{J}(\mathbf{q}))^{-1}\dot{\mathbf{P}} \\ \ddot{\mathbf{q}} &= \frac{d\dot{\mathbf{q}}}{dt} = \frac{d}{dt}[(\mathbf{J}(\mathbf{q}))^{-1}\dot{\mathbf{P}}] \rightarrow \mathbf{J}(\mathbf{q})^{-1} = \mathbf{J}^{-1} \\ \ddot{\mathbf{q}} &= \frac{d}{dt}(\mathbf{J}^{-1})\dot{\mathbf{P}} + \mathbf{J}^{-1}\ddot{\mathbf{P}}\end{aligned}\tag{ecuación 2.9}$$

2.4 ALGORITMOS DE CONTROL DE MOVIMIENTO

La planificación de movimiento es el problema de encontrar un movimiento de robot desde un estado inicial hasta un estado objetivo que evite obstáculos en el entorno y satisfaga otras restricciones, como límites de unión o límites de torque (Lynch & Park, 2017). Frecuentemente, los algoritmos de planificación de movimiento son considerados para poder ejecutar una tarea específica del robot. Su objetivo es generar las entradas de referencia para el sistema de control de movimiento, garantizando las trayectorias en el sistema robótico. Se debe especificar una serie de parámetros para describir la trayectoria deseada. La planificación consiste en generar una secuencia de tiempo de los valores alcanzados por una función de interpolación (típicamente un polinomio) de la trayectoria deseada.

2.4.1 Algoritmos conforme al modelo cinemático

Este algoritmo (véase figura 2.15) permite establecer las trayectorias que debe seguir cada articulación del robot a lo largo del tiempo, logrando así los distintos requerimientos del usuario, como la trayectoria cartesiana del elemento terminal.

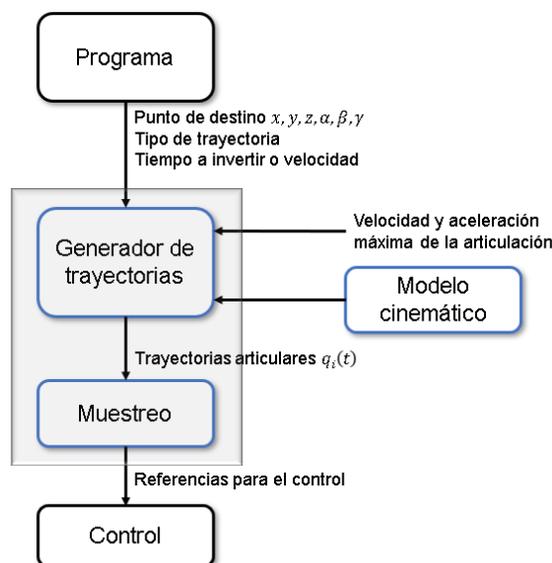


Figura 2.15 Funcionamiento del control cinemático (sombreado) (Barrientos et al., 2007)

En primera estancia, se debe convertir la especificación del movimiento en una trayectoria en espacio cartesiano. Luego se muestrea la trayectoria cartesiana y se obtiene un número finito de puntos de la trayectoria, dados en puntos x, y, z y ángulos α, β, γ .

Usando la transformación homogénea inversa, se convierten esos puntos en coordenadas articulares q_i . Se toma en cuenta la posibilidad de ausencia de solución y si existen puntos singulares, para asegurar la continuidad de la trayectoria. Posteriormente se interpolan los puntos articulares obtenidos y finalmente se muestrea la trayectoria articular para generar referencias al control.

Debe considerarse alguno de los siguientes tipos de trayectoria:

- Trayectoria punto a punto: En este modelo cada articulación cambia desde su posición inicial a la final sin tomar en cuenta el estado de las demás articulaciones. Cada actuador trata de llegar a su punto de destino en el menor tiempo posible, existiendo el movimiento eje a eje y el movimiento simultáneo de ejes. En el primer caso, los ejes se mueven uno a la vez, es decir, primero se mueve una articulación y, cuando ésta llegue a su destino, comienza su movimiento la segunda articulación y así sucesivamente. En el segundo caso, los actuadores se mueven simultáneamente con su propia velocidad, generando distintos tiempos de finalización de movimiento para cada eje. La trayectoria punto a punto no es muy implementada, salvo en robots con unidades de control muy limitadas.
- Trayectorias coordinadas o isócronas: En este tipo de trayectoria, se toma en cuenta la velocidad de la articulación más lenta, puesto que no se debe forzar a los actuadores a trabajar a velocidades y aceleraciones que no pueden alcanzar. Por lo tanto, se ralentiza el movimiento de los demás actuadores, implementando este modelo para sincronizar los tiempos de los ejes y que acaben su movimiento simultáneamente. Esto representa una descripción del extremo del robot no significativa.

- Trayectorias continuas: En este último modelo se pretende que la trayectoria que sigue el efector final del robot sea conocida por el usuario, por lo tanto, se necesita calcular de manera continua las trayectorias articulares; de esta forma se logra que el extremo del robot tenga una trayectoria deseada.

Véase un ejemplo del tipo de trayectorias efectuadas por el robot SCARA de 2 g.d.l. en la figura 2.16. En efecto, se puede definir el control cinemático como aquel que opta por trayectorias ideales que ejecutará el robot, sin embargo, en la realidad no es posible ajustar el movimiento a todas las especificaciones dadas por el usuario, porque las propiedades dinámicas de un sistema robótico actúan como factor limitante para lograr una sincronía entre la trayectoria deseada $q_d(t)$ y la trayectoria real $q(t)$.

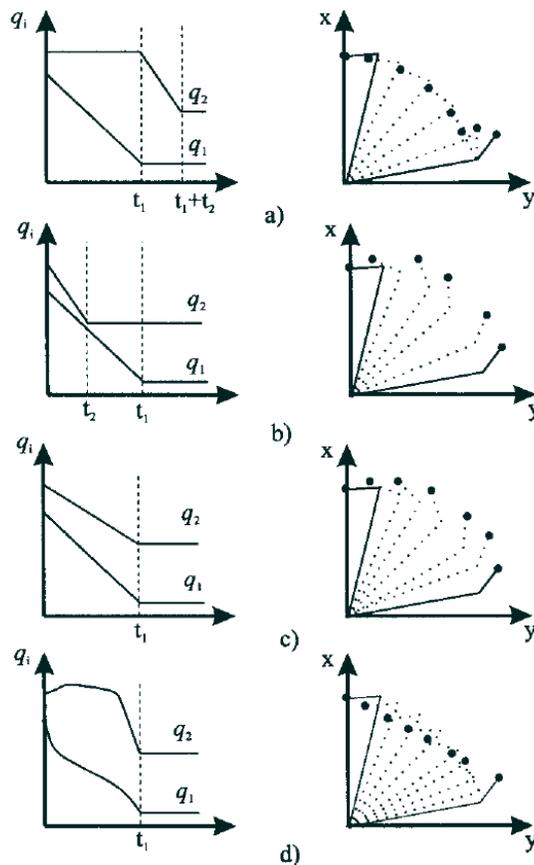


Figura 2.16 Diferentes trayectorias articulares para un robot SCARA a) movimiento eje a eje, b) movimiento simultáneo de ejes, c) trayectoria coordinada, d) trayectoria continua rectilínea (Barrientos et al., 2007)

2.4.2 Algoritmos conforme al modelo dinámico

El control basado en el modelo dinámico toma como herramientas la teoría del servocontrol, tal como representación interna, estado, estabilidad de Lyapunov, control PID, control adaptativo, entre otros conceptos. Su objetivo principal pretende que las trayectorias reales del robot sean más parecidas a las trayectorias propuestas por el usuario. Si bien esa es una de las ventajas del control dinámico, en la práctica, la complejidad de este tipo de control genera un mayor coste económico en la parte tecnológica, porque se conoce que el modelo dinámico de un sistema robótico es firmemente no lineal, multivariable y de parámetros variantes.

Este control es usado mayormente en el espacio articular $q(t)$, sin embargo, también puede optarse por ubicarlo en el espacio cartesiano, controlando así la trayectoria del efector final $j(t)$. Como resultado, existen distintas técnicas de control dinámico, como el control monoarticular y el multiarticular. Dentro del control monoarticular y multiarticular se ubican el control PID, el control PID con prealimentación y el control PD con compensación de gravedad.

- Control monoarticular: En este modelo se consideran características del robot como sus dimensiones, peso, tipo de actuadores, entre otras. Se introduce un modelo de un reductor de elevada relación, donde el modelo dinámico del robot se simplifica al modelo dinámico de cada articulación, sin embargo, este tipo de control tiene inconvenientes, como el considerable aumento del rozamiento viscoso y la introducción de holguras y elasticidad en la transmisión.
- Control multiarticular: No es muy usual considerar a cada articulación por separado (como en el control anterior) y suponer que cada una no se ve afectada más que por sus propias características dinámicas. Divergente a las técnicas basadas en el desacoplamiento de las articulaciones, las técnicas de control acoplado (control multiarticular) contemplan el modelo dinámico real del robot, ocasionando un mejor comportamiento del sistema robótico.

2.5 CONTROL

Como menciona Hernández Gaviño (2010), los controladores son elementos que se adicionan a la configuración original con la finalidad de mejorar las características de respuesta de los sistemas y así satisfacer las especificaciones de funcionamiento tanto en el estado transitorio como en el estacionario.

Se requiere elegir un objetivo de control que se relacione con la tarea y el entorno del sistema; para alcanzar este fin se puede utilizar un control de retroalimentación. El control de retroalimentación utiliza sensores de posición, velocidad y fuerza para medir el comportamiento real del robot, lo compara con el comportamiento deseado y modula las señales de control enviadas a los actuadores, tal como se presenta en la figura 2.17. La retroalimentación se utiliza en casi todos los sistemas de robot.

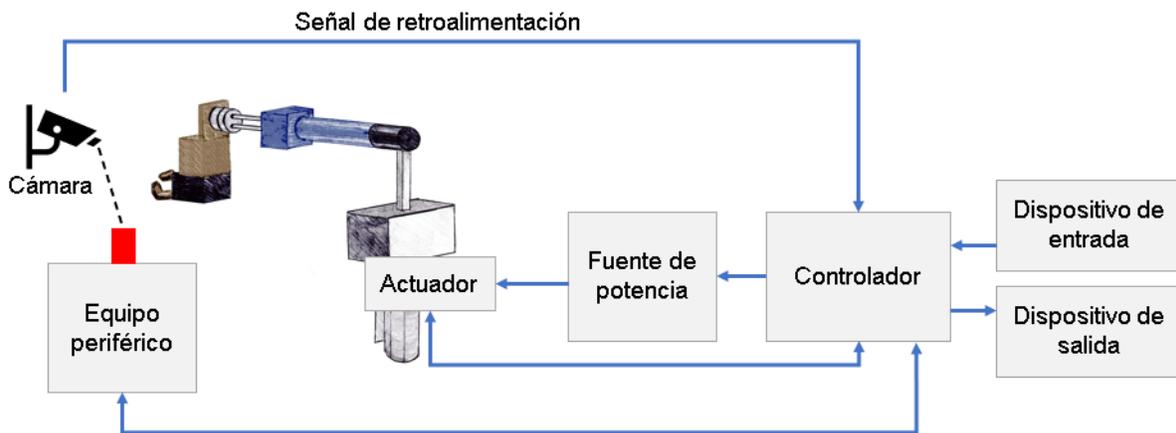


Figura 2.17 Sistema de control de un robot

Generalmente, se utiliza el ajuste de ganancia como primera estancia para poder manipular y modificar las características de respuesta de un sistema. Éste incremento en ganancia (el cual se conoce como control proporcional) tiene como resultado principal mejorar el funcionamiento en estado estable, sin embargo, en el estado transitorio se presenta una respuesta indeseada. Debido a ese

comportamiento, se generaron otros elementos de control que dan lugar a diversos tipos de controladores, como:

- Control proporcional (P).
- Control integral (I).
- Control derivativo (D).

Pero se observó que, para tareas con un grado mayor de complejidad, se requería de la interacción de esos elementos, dando lugar a las siguientes configuraciones:

- Control proporcional-integral (PI).
- Control proporcional-derivativo (PD).
- Control proporcional-integral-derivativo (PID).

Estos controladores agregan polos y/o ceros al sistema controlado. El efecto de agregar polos en un sistema genera que el lugar geométrico de las raíces tenga un desplazamiento hacia la derecha del eje $j\omega$, haciendo inestable al sistema pues reduce su estabilidad relativa. Por otra parte, al adicionar ceros en un sistema, se provoca un desplazamiento del lugar geométrico de las raíces hacia la izquierda, haciendo más estable al sistema controlado.

2.5.1 Control Proporcional (P)

Típicamente es un amplificador simple con una ganancia constante K , puesto que la señal de control a la salida del controlador está relacionada con la entrada del controlador mediante esa constante proporcional, en otras palabras, la salida del controlador $v(t)$ es proporcional al error $e(t)$, tal como se muestra en la ecuación 2.10 y su equivalente en el dominio s es presentada en la ecuación 2.11.

$$v(t) = K_p e(t) \quad \text{ecuación 2.10}$$

$$V(s) = K_p E(s) \quad \therefore \quad G_c(s) = \frac{V(s)}{E(s)} = K_p \quad \text{ecuación 2.11}$$

Prácticamente, un controlador con ganancia K_p (proporcional) es un amplificador del tipo inversor/no inversor como se observa en la figura 2.18 y a eso se debe su comportamiento. Cuando es aplicado este controlador en un sistema mecánico, con una ganancia mínima, se presenta un buen comportamiento en estado transitorio, sin embargo, en el régimen estable se manifiesta un error elevado.

En cambio, si se quiere corregir el error de estado estable se debe aumentar la ganancia, pero ahora, el nuevo comportamiento en el régimen transitorio, adquirirá inestabilidad del sistema y habrá un error significativo. Por consiguiente, no es conveniente usar el controlador P en un sistema, sino que se debe acompañar de algún otro elemento que modifique esos errores en estado transitorio.

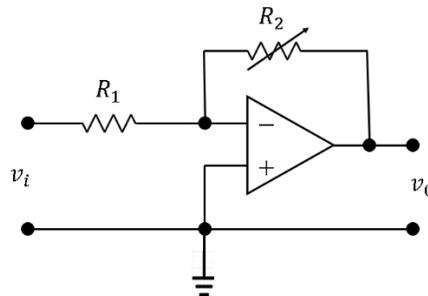


Figura 2.18 Configuración del control proporcional

2.5.2 Control Proporcional Integral (PI)

Un control integral (véase figura 2.19) es aquel donde la salida del controlador $v(t)$ es proporcional a la integral del error $e(t)$, tal como se muestra en la ecuación 2.12; su equivalencia en el dominio s se presenta en la ecuación 2.13.

$$v(t) = K_i \int e(t) dt \quad \text{ecuación 2.12}$$

$$V(s) = \frac{K_i}{s} E(s) \quad \therefore \quad G_c(s) = \frac{V(s)}{E(s)} = \frac{K_i}{s} \quad \text{ecuación 2.13}$$

K_i es la ganancia del control integral y ésta puede escribirse en términos de K_p y T_i (véase ecuación 2.14). T_i es un factor de proporcionalidad que indica el tiempo de integración.

$$K_i = \frac{K_p}{T_i} \quad \text{ecuación 2.14}$$

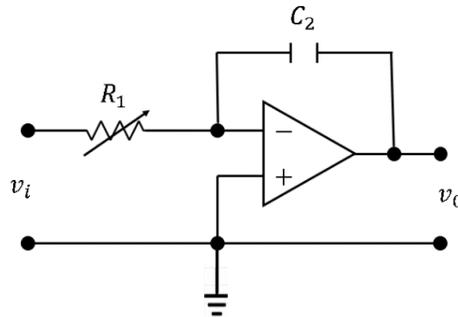


Figura 2.19 Configuración del control integral

Sin embargo, no existen controladores que actúen únicamente con acción integral; siempre actúan en combinación con reguladores de una acción proporcional, complementándose los dos tipos de reguladores, primero entra en acción el regulador proporcional (instantáneamente) mientras que el integral actúa durante un intervalo de tiempo T_i . Por lo tanto, existe el controlador proporcional-integral (PI).

La salida $v(t)$ del control PI, es proporcional al error $e(t)$, sumado a una cantidad proporcional a la integral del error $e(t)$, como se presenta en la ecuación 2.15 y su expresión en el dominio s se muestra en la ecuación 2.16.

$$v(t) = K_p e(t) + K_i \int e(t) dt \quad \text{ecuación 2.15}$$

$$V(s) = K_p E(s) + \frac{K_i}{s} E(s) \quad \text{si: } K_i = \frac{K_p}{T_i}$$

$$G_c(s) = \frac{V(s)}{E(s)} = K_p \left[1 + \frac{1}{T_i s} \right]$$

$$G_c(s) = K_p \left[\frac{s + 1/T_i}{s} \right] = K_p \left[\frac{s + (K_i/K_p)}{s} \right] \quad \text{ecuación 2.16}$$

El control proporcional-integral (figura 2.20) tiene las siguientes características:

- Reduce el amortiguamiento.
- Incrementa el máximo pico de sobreimpulso.
- Decrece el tiempo de elevación.
- Se mejoran los márgenes de ganancia y fase.
- El error de estado estable mejora por el incremento del tipo de sistema.

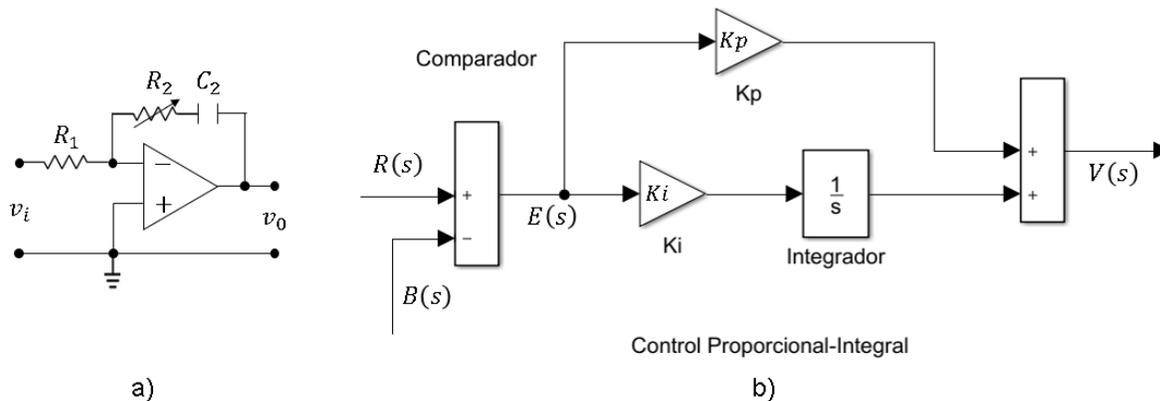


Figura 2.20 a) Configuración del control PI y
b) Diagrama a bloques del control PI en Matlab-Simulink

En esta configuración se reduce el error de estado estable y se debe a la adición de un polo en el origen por la configuración integral, como se muestra en la figura 2.21, aumentando el tipo de sistema y sobrecorrigiendo el error; sin embargo, esta característica genera una respuesta muy oscilatoria e inclusive inestable en un sistema.

```

1  % POLOS Y CEROS DE CONTROLADOR PI
2  % Claudia Mena Huerta
3  % Instituto Tecnológico de Puebla
4
5  clear all;
6  clc;
7
8  Kp = 2;
9  Ki = 5;
10 num = [Kp Ki];
11 den = [1 0];
12 % Se crea la función de transferencia del control PI con Kp=2 y Ki=5
13 PI = tf(num,den)
14 % Se obtiene el lugar de las raíces del control PI
15 figure(1)
16 rlocus(PI);

```

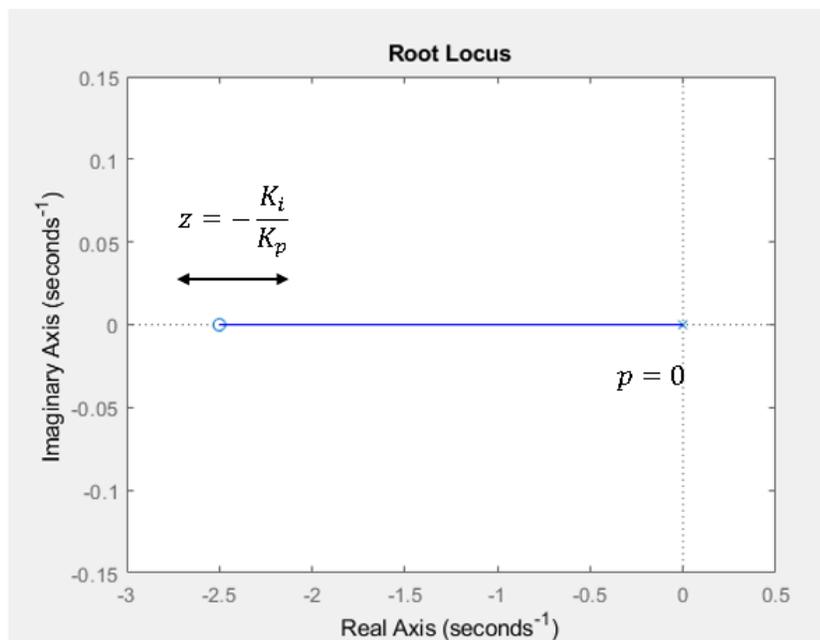
Command Window

```

PI =
      2 s + 5
  -----
         s

```

a)



b)

Figura 2.21 a) Programación del lugar de las raíces del PI y
b) Gráfica del lugar de las raíces del control PI en Matlab

2.5.3 Control Proporcional Derivativo (PD)

Por su parte, un control derivativo (véase figura 2.22) tiene una salida de controlador $v(t)$ proporcional a la derivada del error $e(t)$, tal como se muestra en la ecuación 2.17; su equivalencia en el dominio s se presenta en la ecuación 2.18.

$$v(t) = K_d \frac{de(t)}{dt} \quad \text{ecuación 2.17}$$

$$V(s) = K_d s E(s) \quad \therefore \quad G_c(s) = \frac{V(s)}{E(s)} = K_d s \quad \text{ecuación 2.18}$$

K_d es la ganancia del control derivativo y ésta puede escribirse en términos de K_p y T_d (véase ecuación 2.19). T_d es un factor de proporcionalidad que indica el tiempo de derivación.

$$K_d = K_p T_d \quad \text{ecuación 2.19}$$

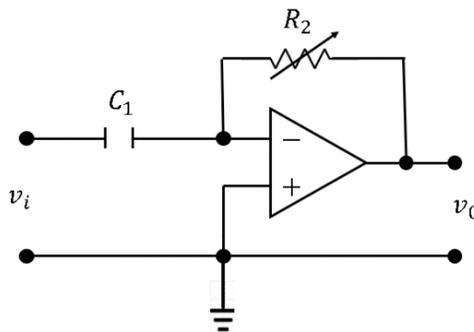


Figura 2.22 Configuración del control derivativo

El control derivativo no es de utilidad si no actúa junto con otro tipo de acción de control, ya que, aislado, el control derivativo no responde a errores de estado estable; por consiguiente, existe el controlador proporcional-derivativo (PD). Este tipo de controlador se requiere analizar antes de implementar como tal un controlador del tipo PID.

El controlador PD contiene en su modelo analítico el comportamiento de K_p y K_d que son las constantes proporcional y derivativa respectivamente. Las funciones que la representan se muestran en las ecuaciones 2.20 y 2.21.

$$v(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad \text{ecuación 2.20}$$

$$V(s) = K_p E(s) + K_d s E(s) \quad \text{si: } K_d = K_p T_d$$

$$G_c(s) = \frac{V(s)}{E(s)} = K_p [1 + T_d s]$$

$$G_c(s) = K_p T_d \left[s + \frac{1}{T_d} \right] = K_p T_d \left[s + \frac{K_p}{K_d} \right] \quad \text{ecuación 2.21}$$

El control proporcional-derivativo (figura 2.23) tiene las siguientes características:

- Incrementa el amortiguamiento y reduce el máximo pico de sobreimpulso.
- El tiempo de elevación experimenta pequeños cambios.
- Se mejoran los márgenes de ganancia y fase.
- El error de estado estable presenta pequeños cambios.
- El tipo de sistema permanece igual.

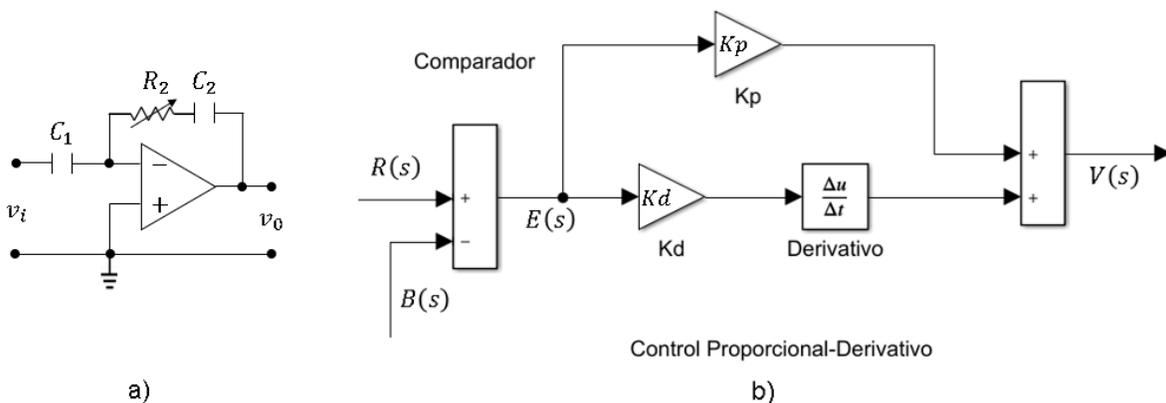


Figura 2.23 a) Configuración del control PD y
b) Diagrama a bloques del control PD en Matlab-Simulink

La acción derivativa responde a la rapidez de cambio del error, corrigiéndolo antes de que éste sea elevado; por lo tanto, se dice que es un controlador que se adelanta frente a una tendencia de error (expresado en forma de derivada). El lugar de las raíces del control PD se muestra en la figura 2.24.

```

1  % POLOS Y CEROS DE CONTROLADOR PD
2  % Claudia Mena Huerta
3  % Instituto Tecnológico de Puebla
4
5  clear all;
6  clc;
7
8  Kp = 2;
9  Kd = 5;    Td = 5;
10 num = [Kp*Td (Kp^2)*Td/Kd];
11 den = [1];
12 % Se crea la función de transferencia del control PD con Kp=2 y Kd=Td=5
13 PD = tf(num,den)
14 % Se obtiene el lugar de las raíces del control PD
15 figure(1)
16 rlocus(PD);

```

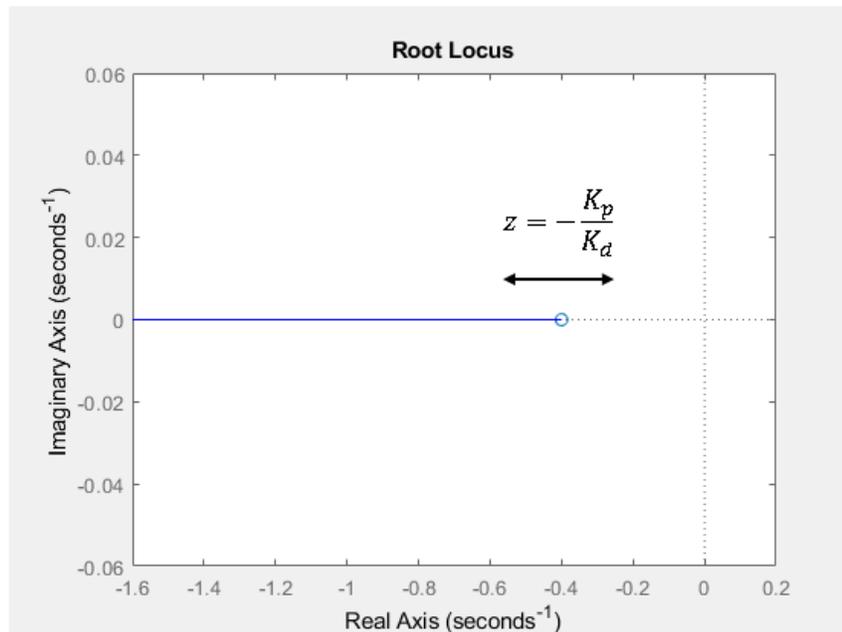
Command Window

```

PD =
10 s + 4

```

a)



b)

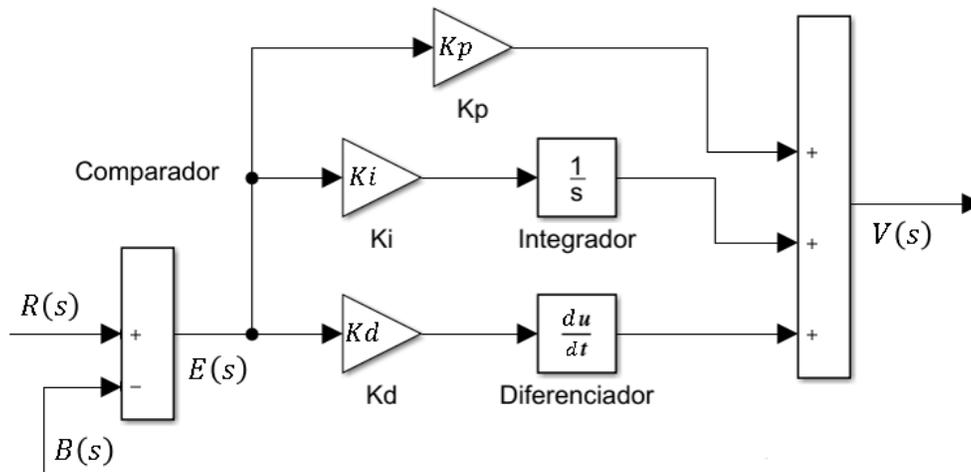
Figura 2.24 a) Programación del lugar de las raíces del PD y
b) Gráfica del lugar de las raíces del control PD en Matlab

2.5.4 Control Proporcional Integral Derivativo (PID)

El controlador PID (figura 2.25) es la solución más común a los problemas prácticos de control. El PID clásico surgió en los años 30's a partir de los controladores neumáticos. Conformado por la suma de los tres controladores proporcional-integral-derivativo, aumenta sus ventajas con respecto a los controladores vistos de forma aislada.

La salida $v(t)$ del controlador PID es proporcional al error $e(t)$, sumado a una cantidad proporcional a la integral del error $e(t)$ más una cantidad proporcional a la derivada del error $e(t)$, tal como se muestra en la ecuación 2.22.

$$v(t) = K_p e(t) + K_i \int e(t) dt + K_d \frac{de(t)}{dt} \quad \text{ecuación 2.22}$$



Control Proporcional-Integral-Derivativo

Figura 2.25 Diagrama a bloques del control PID en Matlab-Simulink

La expresión en el dominio s del PID se muestra en la ecuación 2.23 y el lugar de las raíces en la figura 2.26.

$$V(s) = K_p E(s) + \frac{K_i}{s} E(s) + K_d s E(s)$$

$$\therefore G_c(s) = \frac{V(s)}{E(s)} = K_p + \frac{K_i}{s} + K_d s$$

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

ecuación 2.23

```

1  % POLOS Y CEROS DE CONTROLADOR PID
2  % Claudia Mena Huerta
3  % Instituto Tecnológico de Puebla
4
5  clear all;
6  clc;
7
8  Kd = 3; Kp = 3; Ki = 3;
9  num = [Kd Kp Ki];
10 den = [1 0];
11 % Se crea la función de transferencia del control PID con Kd=Kp=Ki=3
12 PID = tf(num,den);
13 % Se obtiene el lugar de las raíces del control PID
14 figure(1)
15 rlocus(PID);

```

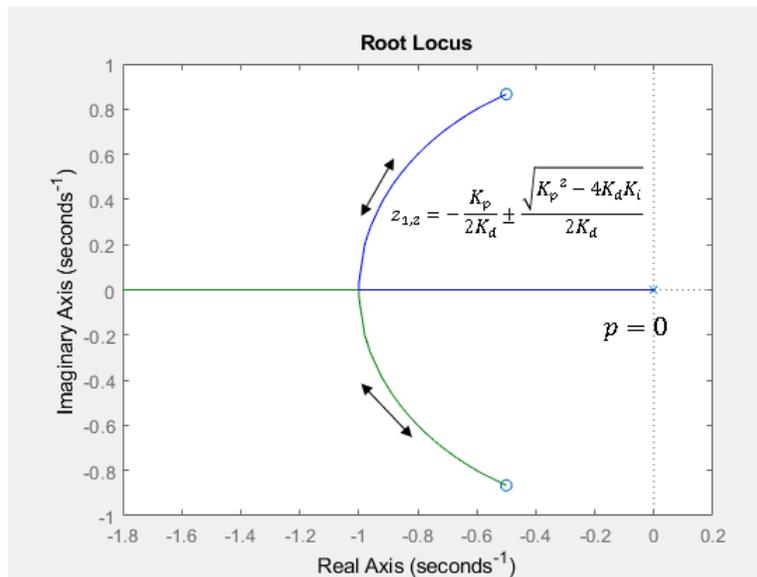
Command Window

```

PID =
      3 s^2 + 3 s + 3
      -----
              s

```

a)



b)

Figura 2.26 a) Programación del lugar de las raíces del PID y
b) Gráfica del lugar de las raíces del control PID en Matlab

La expresión del control PID mostrada en la ecuación 2.23 es la más común, porque se puede obtener acción proporcional o integral pura con parámetros finitos; sin embargo, es la forma donde los parámetros tienen poca interpretación física (Astrom & Tore, 2009). Por consiguiente, existen otras interpretaciones del control PID

- La forma estándar o no interactuante (ecuación 2.24).
- La forma serie o interactuante (ecuación 2.25).

$$G_{cestándar}(s) = k \left(1 + \frac{1}{s\tau_i} + s\tau_d \right) \quad \text{ecuación 2.24}$$

$$G_{cserie}(s) = k \left(1 + \frac{1}{s\tau_i} \right) (1 + s\tau_d) \quad \text{ecuación 2.25}$$

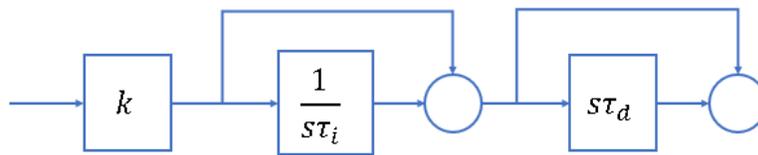


Figura 2.27 Control PID forma interactuante

Como se puede ver, el control PID paralelo y en serie (figura 2.27), matemáticamente son equivalentes. Desde un controlador en serie, siempre es posible cambiar a una forma paralela, gracias a las ecuaciones 2.26 a – c.

$$k_p = k \left(1 + \frac{\tau_d}{\tau_i} \right) \quad k_i = \frac{k}{\tau_i} \quad k_d = k\tau_d \quad \text{ecuación 2.26 a) – c)}$$

La principal diferencia entre esas estructuras es el efecto de establecer los coeficientes en el comportamiento del controlador. La forma paralela permite un desacoplamiento completo de acciones proporcionales, integrales y derivadas, mientras que, con una estructura estándar, una modificación del coeficiente k_p afectará las acciones proporcionales, integrales y derivadas. Es la razón por la que las personas que configuran manualmente el controlador prefieren usar la forma paralela.

2.6 AUTOMATIZACIÓN

Antes de la era industrial, la elaboración de distintos productos que eran llevados al mercado tenía un proceso en su mayor parte artesanal y completamente laborioso. Con la llegada de las propuestas de revolucionarios científicos y tecnológicos, como Frederick W. Taylor, se cambió la forma de generar un producto o tarea en gran escala, pues comenzaron a optimizarse los procesos productivos. Las tareas complejas ahora eran divididas en celdas de trabajo; se aplicaban dispositivos electromecánicos y electrónicos generando un nuevo panorama en la nueva época industrial: los procesos automatizados. En la industria se adquirió la automatización como una de las tareas más importantes con la intención de optimizar tiempos, procesos y disminuir costos al crear un producto.

Como menciona Daneri (2008), la expresión “automatización” hace referencia a la ejecución de procesos con la menor intervención posible del ser humano. La automatización comprende tres etapas importantes en los procesos industriales: medición, evaluación y control. Su principal enfoque va encausado a la liberación de recursos humanos y la eliminación de riesgos de trabajo. En contraparte, las desventajas de la automatización son el incremento de costes fijos, el incremento de mantenimiento y la reducción de flexibilidad de los recursos.

2.6.1 Arquitectura del proceso de automatización

La arquitectura de automatización se presenta en la figura 2.28 donde se muestran los principales elementos que la conforman. A continuación, se detallan sus principales características:

- Nivel de supervisión HMI: La interfaz hombre-máquina (HMI por sus siglas en inglés) se ubica en la parte de mando, puesto que el HMI es un elemento de entrada y salida de datos. El HMI es un programa que al ejecutarse permite al usuario tener el control del proceso automatizado. Dependiendo del tipo de proceso, el sistema a controlar y los distintos elementos que

conforman la parte operativa del proceso, serán programados los comandos necesarios para ingresar los datos y para su lectura e interpretación.

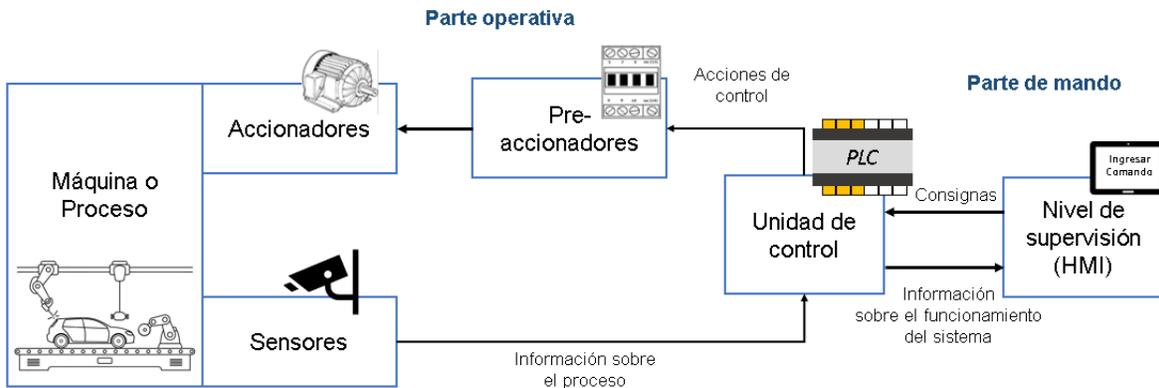


Figura 2.28 Arquitectura del proceso de automatización

- **Unidad de control:** Como su nombre lo dice, es el elemento que controlará los datos de entrada y salida del HMI y del proceso automatizado, además mandará la información del comportamiento del sistema hacia el HMI para su interpretación. Ejemplo de unidad de control es el Controlador Lógico Programable (PLC) el cual es usado frecuentemente en los procesos industriales.
- **Pre-accionadores:** También llamados preactuadores, son elementos que funcionan como interfaz entre el controlador y el actuador; toma como entrada una señal eléctrica que, al ser detectada, procede a mandar la señal hacia el actuador para iniciar su funcionamiento. Ejemplo de preactuadores son la electroválvula, el relé/contactor y los arrancadores estáticos.
- **Accionadores:** Los actuadores toman la señal proveniente de los pre-accionadores y posteriormente se encargan de generar su función, produciendo cambios energéticos en el proceso. Algunos actuadores ya se han presentado en la tabla 2.5 de este capítulo.

- **Sensores:** También nombrados transductores; éstos son dispositivos que tienen la capacidad de detectar ciertas variables del proceso, como temperatura, humedad, presión, etc., y la transforman en una señal de salida de tipo eléctrica de tensión o corriente.
- **Máquina o proceso:** La máquina es un conjunto de piezas que, acopladas entre sí, logran transformar una forma de energía en otra para hacer una tarea definida. El término “proceso” es muy complejo, puesto que se puede analizar en pequeña escala (compuesto de diversas máquinas) o como un sistema más desarrollado como las celdas de trabajo o líneas de producción.

2.6.2 Dispositivos de automatización

Las nuevas tecnologías se incorporaron en el campo de la robótica, entre ellos los dispositivos de automatización, los cuales son útiles en aplicaciones robustas y por ello es su preferencia en el área industrial. A continuación, se verá la descripción de algunos dispositivos de automatización necesarios para el desarrollo del proyecto de tesis.

2.6.2.1 Controlador Lógico Programable (PLC)

El Controlador Lógico Programable (PLC) es un dispositivo encargado del control en el sistema (véase figura 2.29); detecta las señales de entrada y activa las salidas a partir de la información recibida por uno o varios sensores; cuando tiene integrado un control de movimiento (MC), se puede comunicar con éste internamente.

Prácticamente, el PLC es una computadora digital cuya finalidad es controlar un sistema mecánico, contiene interfaces especiales de entrada / salida y un lenguaje de programación de control; el PLC está diseñado para operar en el entorno industrial, razón por la cual es muy popular en el ámbito tecnológico. El controlador

lógico programable tiene la capacidad de disponer entradas y salidas múltiples, además posee rangos de temperatura extendidos, inmunidad al ruido eléctrico y resistencia a la vibración y al impacto. El PLC pertenece a los sistemas de tiempo real porque las salidas del sistema controlado dependen de las condiciones de entrada.



Figura 2.29 Controlador Lógico Programable marca Delta DVP20SX2

El inicio del PLC se caracteriza porque éste se limitaba a ejecutar procesos secuenciales repetitivos, por lo tanto, no tenía una gran capacidad para controlar procesos con una complejidad mayor. Fue gracias a la creación de los microprocesadores que comenzaron a tener mejores funciones para controlar los sistemas, además de que se integraron memorias a estos dispositivos. El poder guardar datos del proceso controlado ayudó a determinar cuál es el comportamiento pasado y prever el comportamiento futuro del sistema.

Los elementos que componen la constitución del PLC son: Fuente de alimentación, unidad de operación y visualización, entradas y salidas, CPU y una interfaz para la conexión a PC y módulos de programa.

Las ventajas de los controladores programables lógicos, de acuerdo a Álvarez (2004), son:

- Una automatización de bajo costo y alta confiabilidad.
- Una complejidad relativamente sencilla.
- Permite la grabación, copia e impresión de programas desde un PC o desde los módulos de memoria.
- Programación e instalación factibles.
- Tienen salida a un relé con una gran capacidad de corte.
- Alta velocidad de control.
- Compatibilidad de red.
- Mantenimiento nulo.
- El programa de usuario puede ser protegido con contraseña.

Por su parte, el Controlador de movimiento (MC) es un módulo de control que ocasionalmente viene integrado en el PLC. Su manejo de altas velocidades permite que se pueda aplicar ampliamente en la industria de control de automatización, como embalaje, impresión, encapsulado, corte, máquina de control digital, etc.

2.6.2.2 Servo drive y servomotor

El controlador es responsable de calcular la ruta o trayectoria requerida y enviar señales de comando de bajo voltaje al servo drive y éste último tiene como tarea principal enviar el voltaje y la corriente necesarios al servomotor para lograr el movimiento requerido, por esa razón, el servo drive a veces es llamado “amplificador”. Las funciones del servo drive son controlar el par, la velocidad o la posición de los servomotores para posteriormente enviar la información al controlador lógico programable. Existen distintos tipos de servo drives, como el tipo amplificador de modo de par (la corriente es proporcional a la fuerza) y el de accionamiento directo (existe un control directo de la salida de fuerza del servomotor).

Por su parte, el servomotor (véase figura 2.30) es un tipo especial de motor, por lo tanto, es un dispositivo electromecánico que consiste en un motor eléctrico, un juego de engranes y una tarjeta de control. El eje del motor se acopla a una caja de engranajes similar a una transmisión, cuya finalidad es potenciar el torque del motor y permitir mantener una posición fija cuando se requiera. Este dispositivo permite controlar la posición del eje en un momento dado y se conecta al servo drive. Sus principales características son una buena exactitud en el control de velocidad y posición, alta fiabilidad de funcionamiento y pérdidas en el rotor muy bajas.



Figura 2.30 Servomotor y servo drive marca Delta

2.6.2.3 Otros dispositivos de automatización

Otros elementos importantes para la realización del tema de tesis son:

Computador: El PLC puede grabar funciones y no requerir la conexión de la computadora, sin embargo, el computador ofrece ciertas ventajas en su aplicación industrial, tal como tener una mayor eficacia de las operaciones, mayor seguridad y una reducción drástica de las operaciones manuales gracias a la adquisición de datos, el sistema de supervisión y el análisis de datos.

Panel de control o de mando: Es el elemento imprescindible de un sistema de control pues indica al usuario la función de cada uno de los mandos, pulsadores o indicadores del equipo para que el usuario pueda manejar la máquina.

2.6.3 Redes de comunicación industriales

Las redes de comunicación cubren una amplia gama de aplicaciones, permitiendo la vinculación y comunicación entre los distintos dispositivos del sistema automatizado. Existen diferentes clases de redes industriales, como son Profibus, Fieldbus Foundation, etc., y cada red soporta diferentes medios físicos, utilizando un protocolo de comunicación específico, el cual constituye el conjunto de reglas y convenciones que establecerán el idioma de comunicación entre los interlocutores (Daneri, 2008). La jerarquización de los niveles de las redes industriales se muestra en la figura 2.31.

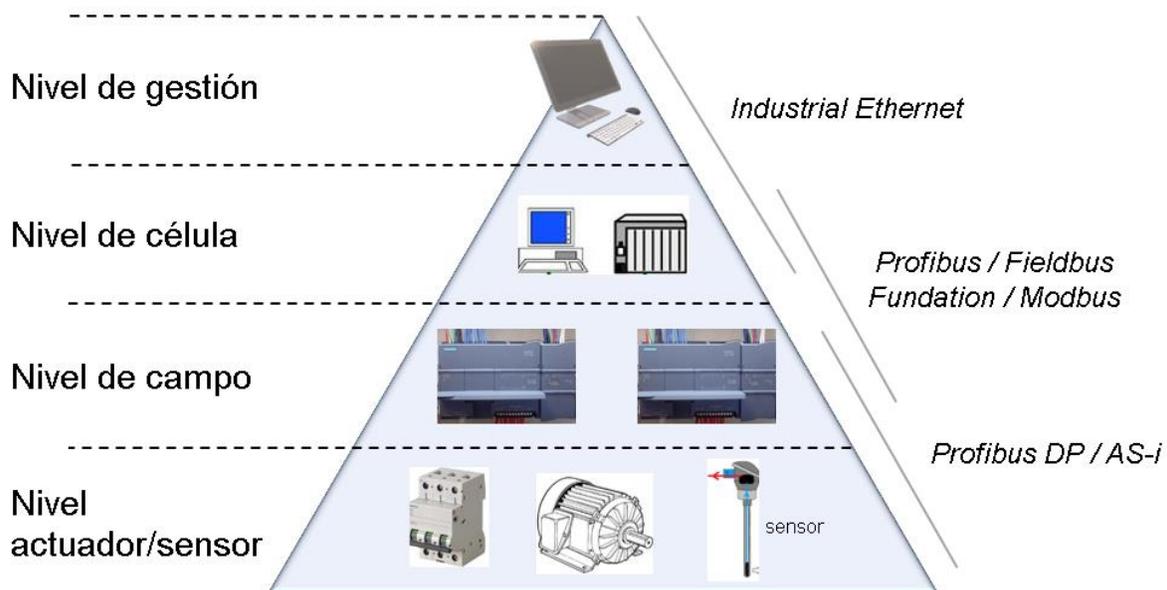


Figura 2.31 Niveles de jerarquía de redes de comunicación

En el nivel de gestión se manejan tareas dedicadas al proceso en su forma global; por su parte, el nivel de célula es donde se administran todas las tareas de automatización y optimización del proceso, es donde se encuentran conectados los dispositivos de automatización entre sí; un peldaño más bajo se ubica el nivel de campo el cual sirve de nexo entre las instalaciones y los autómatas programables; finalmente en el nivel de actuador/sensor se da la comunicación de estos elementos con un dispositivo maestro.

2.6.4 Protocolo de comunicación

Un protocolo de comunicaciones es un conjunto de normas para permitir la comunicación entre dos o más entidades que intervienen en un sistema automatizado, transmitiendo información por medio de cualquier tipo de variación de una magnitud física. El protocolo contempla controles sobre los datos transmitidos, dando así seguridad al sistema ante interferencias en el cable. A continuación, se presenta la descripción del protocolo de comunicación CANopen, puesto que se utilizó para la comunicación entre el Controlador lógico programable y los servo drives marca Delta.

De acuerdo a Guerrero, Yuste y Martínez (2010), en el nivel físico del modelo OSI (Open System Interconnection), al igual que otros aspectos de la comunicación, el enlace de datos también se encuentra normalizado. Ejemplos de normas de comunicaciones industriales son la RS-232, RS-422, RS-485, entre otros.

2.6.4.1 Protocolo CANopen

Para entender el funcionamiento de este protocolo, se requiere analizar el funcionamiento del protocolo CAN, el cual es antecesor del CANopen. El Controlador de Área Local (por sus siglas en inglés CAN), es un protocolo donde las unidades de mando están interconectadas. Fue desarrollado por la industria Bosch y utiliza dos conducciones llamadas Bus para transmitir los datos de forma rápida (código binario) en sistema bidireccional.

A pesar de que todas las unidades de control reciben las mismas señales, sólo reaccionan las unidades para las que están destinados los mensajes correspondientes, por consiguiente, el sistema codifica cada mensaje con un identificador especial (ID) y solo la unidad receptora adecuada podrá identificarlo. Los mensajes codificados están divididos en 7 sectores y contienen un segmento que indica su orden de prioridad. Los dos cables CAN son físicamente un par trenzado, donde uno es la línea 'H' y lleva datos de alto voltaje y, el otro es la línea

'L' que lleva datos de bajo voltaje. Ambas líneas van a todas las unidades conectadas.

El protocolo fue diseñado originalmente para sistemas de control de máquina orientados al movimiento, como sistemas de manejo (máquinas con sistema de red integrado como se muestra en la figura 2.32). Hoy se usa en diversos campos de aplicación, como equipos médicos, vehículos todoterreno, electrónica marítima, aplicaciones ferroviarias o de automatización.

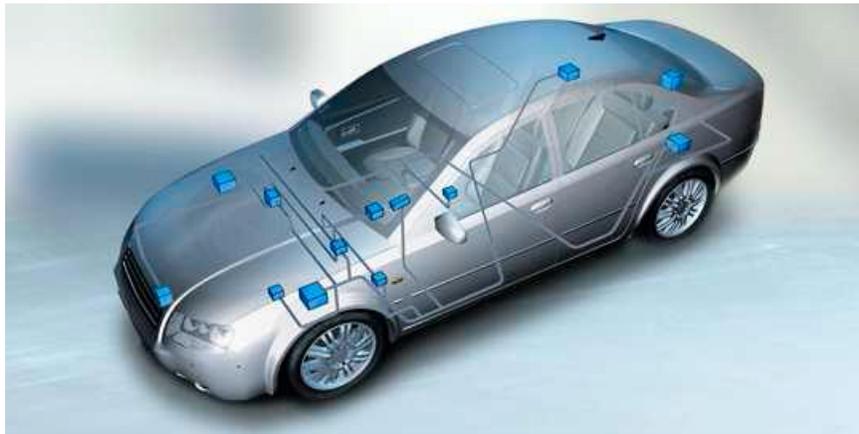


Figura 2.32 Aplicación de CAN-bus para reducir la cantidad de cableado
(Instituto Schneider Electric de Formación, 2008)

Más tarde, tomando las bases del protocolo CAN se creó CANopen, el cual es un protocolo de comunicaciones de alto nivel y de especificación del perfil del dispositivo, para uso industrial, basado en el bus CAN (Controller Area Network) ISO 11898 y recogido en la norma EN 50325-4.

El paraguas de CANopen cubre un marco de programación de red, descripciones de dispositivos, definiciones de interfaz y perfiles de aplicación. CANopen proporciona un protocolo que estandariza la comunicación entre dispositivos y aplicaciones de diferentes fabricantes. Se ha utilizado en una amplia gama de aplicaciones de automatización y movimiento. Además, se utiliza como un sistema de comunicación industrial general con una velocidad de transferencia de datos de hasta 1 Mbit / s.

Sus características principales se enlistan a continuación:

- Máxima velocidad: 1Mbit (40m)
- Multi-Master
- Las colisiones se resuelven, arbitraje por prioridad
- Hasta 8 bytes por mensaje
- Los mensajes tienen un único identificador (ID)
- Robusto, detecciones de errores y algoritmos de recuperación
- CAN2.0A: alrededor de 2k ID's
- La mayoría de transceivers aceptan par trenzado
- Implementa parcialmente la capa uno y dos del modelo OSI
- Finales de línea, 120 ohmios a cada extremo
- Máxima distancia: 1000m (50kBit)
- Los mensajes son de difusión
- Escribir 0 es dominante sobre escribir 1 que es recesivo
- Hasta 10,000 mensajes por segundo
- Cada nodo debe usar un ID diferente
- El ancho de banda se reduce a un 50% debido al control de errores
- CAN2.0B: 500 millones IDs

Con respecto a las capas del protocolo, en términos del modelo de sistemas de comunicación OSI, CAN cubre los dos primeros niveles: la capa física y la capa de enlace de datos. La capa física define las líneas utilizadas, los voltajes, la naturaleza de alta velocidad, etc. La capa de enlace de datos incluye el hecho de que CAN es un protocolo basado en cuadros (mensajes). CANopen cubre las cinco capas principales: red (direccionamiento, enrutamiento), transporte (confiabilidad de extremo a extremo), sesión (sincronización), presentación (datos codificados de manera estándar, representación de datos) y aplicación (véase la figura 2.33). La capa de aplicación describe cómo configurar, transferir y sincronizar dispositivos CANopen.

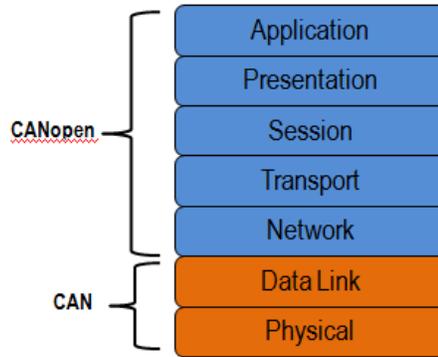


Figura 2.33 Niveles del protocolo CANopen (Instituto Schneider Electric de Formación, 2008)

A continuación, se enlistan las ventajas del protocolo CANopen y se muestran una comparación entre CANopen y otros protocolos en la figura 2.34:

- Garantiza una intercambiabilidad de dispositivos.
- Permite la comunicación entre dispositivos de diferentes fabricantes.
- Transmisión de datos de proceso de tiempo crítico.
- Monitoreo del dispositivo.
- Señalización de error.
- Administración de redes y red de conexión predefinida.
- CAN es insensitivo a las interferencias electromagnéticas debido al uso de una señal diferencial.

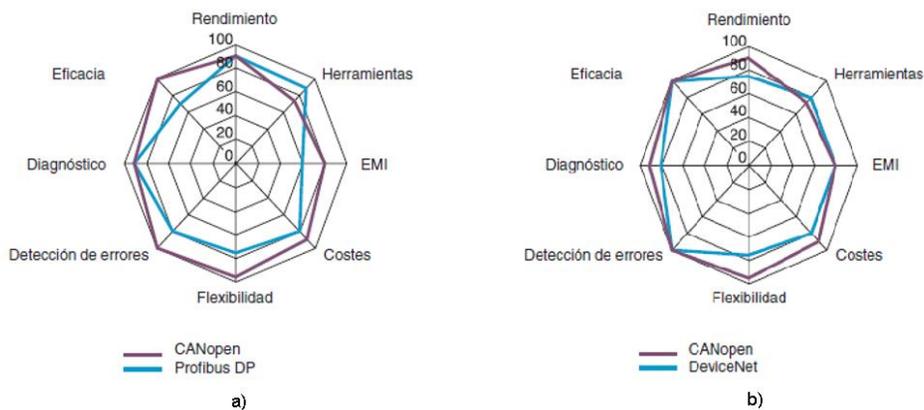


Figura 2.34 Comparación entre protocolos de comunicación a) CANopen contra Profibus DP y b) CANopen contra DeviceNet (Instituto Schneider Electric de Formación, 2008)

El protocolo Modbus es fuerte con una larga y exitosa historia de aplicaciones, pero cuando se necesita interpolación, debe necesitar algo más. CANopen es fuerte, fácil de usar y, más importante, totalmente compatible con muchos dispositivos de marca diferentes en el mercado.

Para la implementación del protocolo CANopen en el proyecto junto con los elementos Delta (como se muestra en la figura 2.35), se deben tener en cuenta los siguientes puntos:

- Es necesario usar cables CANOpen; los cables Ethernet tienen el mismo pasador y funcionarán en distancias cortas, pero pueden ocasionar problemas debido a que no tienen la misma impedancia interna.
- Se debe recordar encender el interruptor de resistencia de terminal de TAP-CN03. Si no se quiere usar, se debe poner una resistencia terminal de 120 Ohm.
- No es necesario proporcionar conexiones V+ y V- para distancias cortas, para las largas es recomendable.

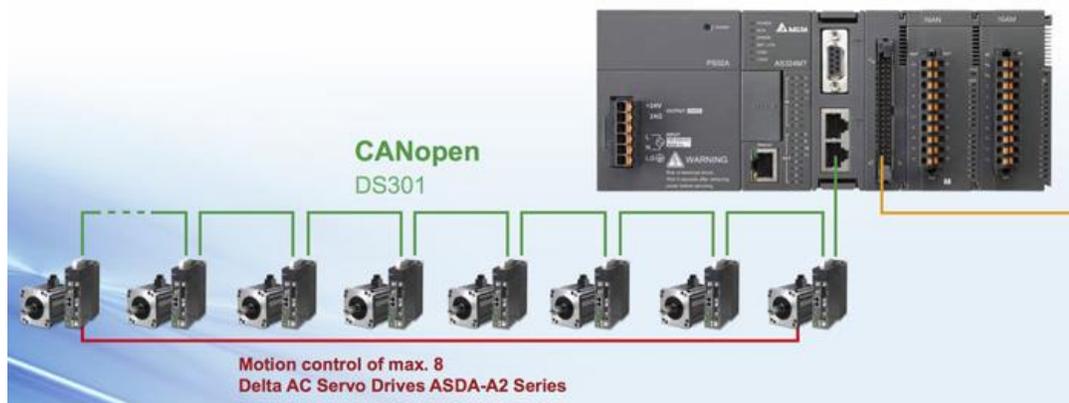


Figura 2.35 CANopen usando dispositivos Delta

CAPÍTULO III DESARROLLO

En esta sección se presentan las tareas realizadas referentes a la cinemática del robot y un sistema de control de posición en un robot paralelo activado por cables usando el control del servo drive que se describirá más adelante. Además, se presentan los diversos softwares donde es programado el algoritmo para controlar la posición del efector final.

3.1 ANÁLISIS MECÁNICO DEL ROBOT

Antes de generar el modelo matemático del robot, se debe analizar la estructura mecánica y la geometría del CDPR y así obtener datos que son adecuados para la obtención de la cinemática del robot.

3.1.1 Clasificación y descripción del sistema mecánico

El robot paralelo activado por cables estudiado en el proyecto de tesis presenta una estructura cinemática cerrada (paralelo) y un tipo de movimiento planar con 3 grados de libertad: traslación sobre ejes 'x' y 'y', rotación sobre el eje 'z'. Su tipo de actuación es eléctrica con una transmisión directa (físicamente generada por un tambor acoplado) y finalmente, tiene un tipo de espacio irregular.

Físicamente se conforma por una plataforma fija de perfiles de aluminio con un espesor de 80 mm cada uno; las dimensiones externas de esta plataforma son 2500 mm de ancho por 1355 mm de alto, mientras que su área interna mide 1480 mm de ancho por 900 mm de altura. Por otra parte, como todo robot, posee un efector final (plataforma móvil) y se conforma por perfiles de aluminio de 20 mm de espesor; las dimensiones de esta pequeña plataforma son de 82 mm de ancho por 46 mm de alto (ver figura 3.1).

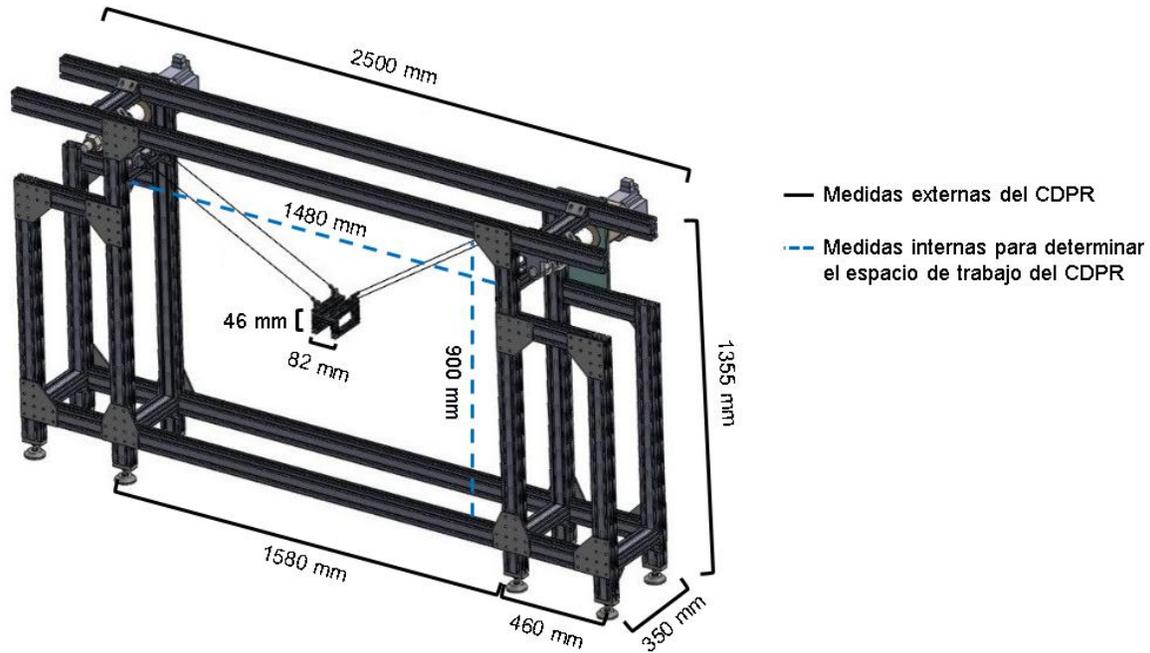


Figura 3.1 Medidas del robot planar paralelo
 (Cortesía del CIM del Instituto Tecnológico de Puebla)

Los actuadores del robot paralelo cableado son dos servomotores fijados en sus extremos superiores como se observa en la figura 3.1; de marca Delta 220VCA modelo ECMA Series EΔ13 10. El eje o flecha de cada servomotor (considerando la salida de la caja de engrane cuya relación es 5:1) tiene acoplado un tambor como se presenta en la figura 3.2 Son 2 tambores en total, cuyo perímetro para cada uno de ellos es de 237 milímetros. La relación para cada actuación se presenta de la siguiente forma:

Para el actuador número 1 (izquierdo):

$$1 \text{ vuelta}_{\text{caja de engranes1}} = 360^\circ = 2\pi = 237 \text{ mm} = 10000 \text{ pulsos}_{\text{encoder1}}$$

Para el actuador número 2 (derecho):

$$1 \text{ vuelta}_{\text{caja de engranes2}} = 360^\circ = 2\pi = 237 \text{ mm} = 10000 \text{ pulsos}_{\text{encoder2}}$$

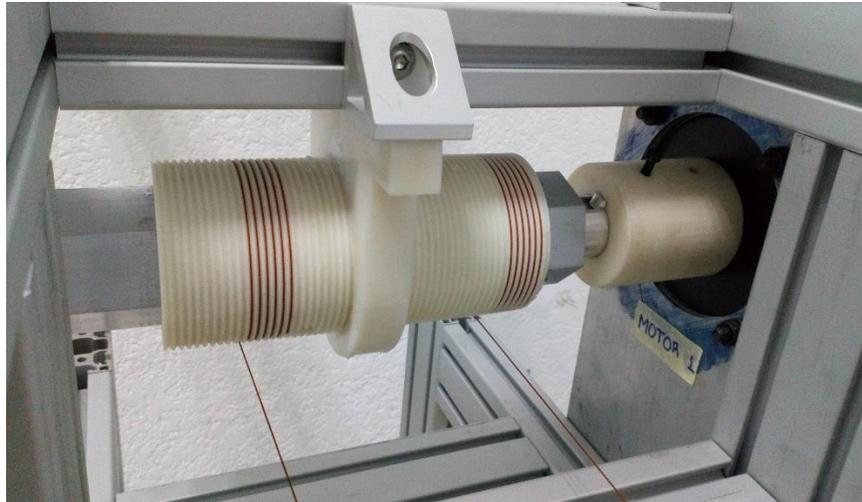


Figura 3.2 Tambor acoplado a la flecha del servomotor

Finalmente, el robot contiene 4 cables; cada tambor tiene enrollado un par de cables por sus extremos, mientras que la otra punta de los cables se sujeta con la esquina correspondiente de la plataforma móvil. En la Figura 3.3 se muestra el robot paralelo activado por cables tipo planar y sus componentes principales.

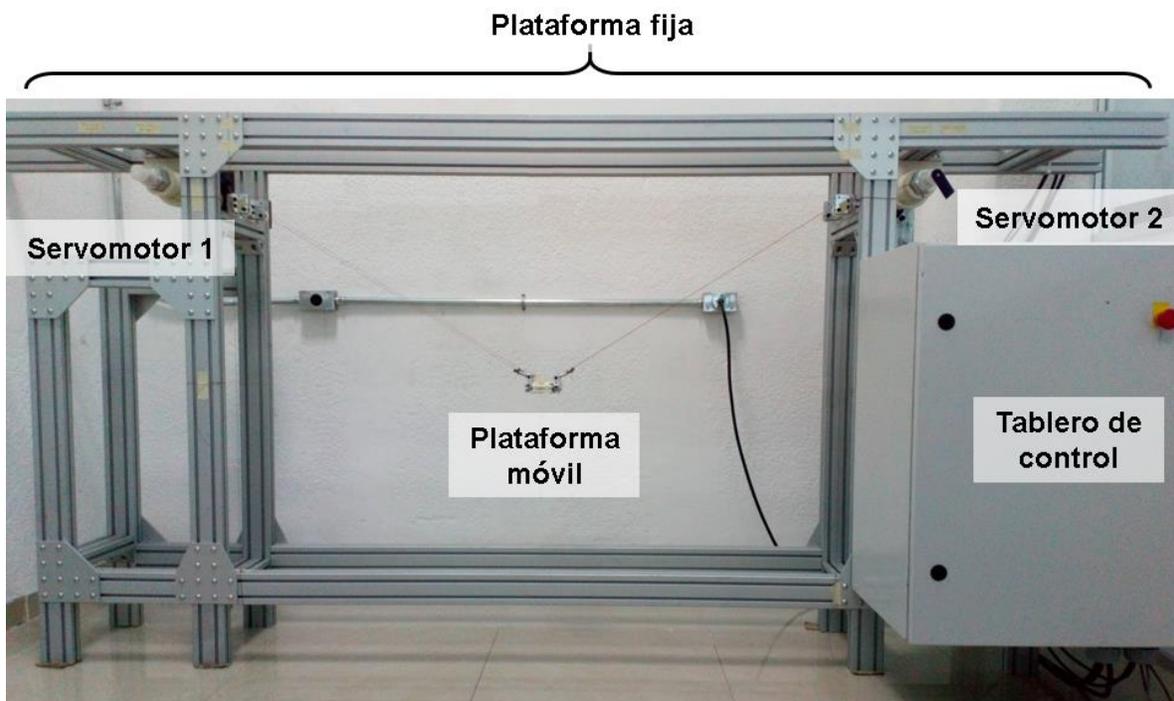


Figura 3.3 Robot activado por cables estudiado

3.1.2 Descripción de los grados de libertad

Inicialmente, la configuración del esquema inicial del CDPR presenta 6 g. d. l: tres rotaciones y tres traslaciones sobre los ejes 'x', 'y' y 'z'; En la figura 3.4 se presenta el diagrama esquemático inicial del robot cableado paralelo tipo planar:

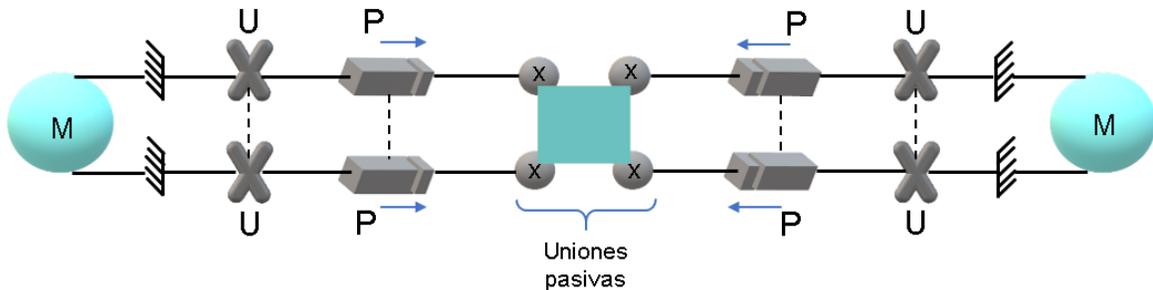


Figura 3.4 Diagrama esquemático inicial del robot planar paralelo

Si el paralelogramo se mantiene, se eliminan 3 grados de libertad de la plataforma móvil: dos rotaciones, una sobre el eje 'x' y otra sobre el eje 'y', además de una traslación sobre el eje 'z'; resultando un CDPR de tipo planar, por lo tanto:

$$\lambda = (6 - 3) \text{ g.d.l.} = 3 \text{ g.d.l.}$$

Y su esquema final se presenta como en la figura 3.5:

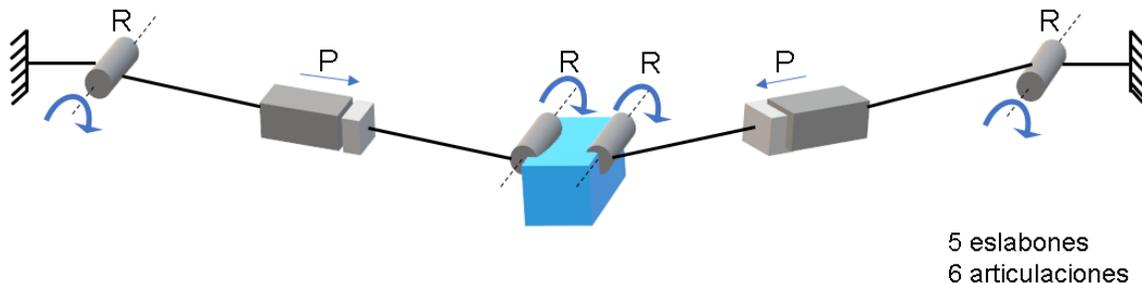


Figura 3.5 Diagrama esquemático final del robot planar paralelo

Al obtener el valor de λ , se procede a obtener el número de grados de libertad de este robot paralelo activado por cables a través de la ecuación 2.1:

Datos:

$$\lambda = 3$$

$$L = 5$$

$$J = 6 \rightarrow \begin{cases} 4R: f_R = 1 \\ 2P: f_P = 1 \end{cases}$$

Ecuación 2.1:

$$n = \lambda L - \sum_{i=1}^J (\lambda - f_i)$$

$$n = \{(3)(5) - [6(3 - 1)]\} \text{ g.d.l.}$$

$$n = (15 - 12) \text{ g.d.l.}$$

$$n = 3 \text{ g.d.l.}$$

De esa forma se comprueba que el número de grados de libertad del CDPR estudiado es de 3 *g.d.l.*, presentándose movimiento rotacional en el eje 'z' y movimiento traslacional sobre los ejes 'x' y 'y'; por lo tanto, se define un tipo de robot planar. Posteriormente, se determina su clasificación respecto al número de grados de libertad (n) y el número de cables (m) que conforman al robot. Si este CDPR tiene 3 *g.d.l.* y 2 cables (cabe recordar que en total son 4 cables, pero al tener una configuración de paralelogramo éstos pueden analizarse como 2).

$$n + 1 = 3 + 1 = 4 \text{ g.d.l.}$$

$$m = 2 \text{ cables}$$

Entonces se cumple que:

$$4 > 2$$

$$\therefore \mathbf{n + 1 > m}$$

Por ende, el CDPR estudiado cae en la clasificación "incompletamente restringido".

3.2 ARQUITECTURA DEL SISTEMA

La arquitectura del robot (véase figura 3.6) se divide en dos partes: la de control y la eléctrica. Esta arquitectura permite entender la distribución de los distintos elementos de control del robot y saber cuál es la relación y conexión entre ellos, así como su función.

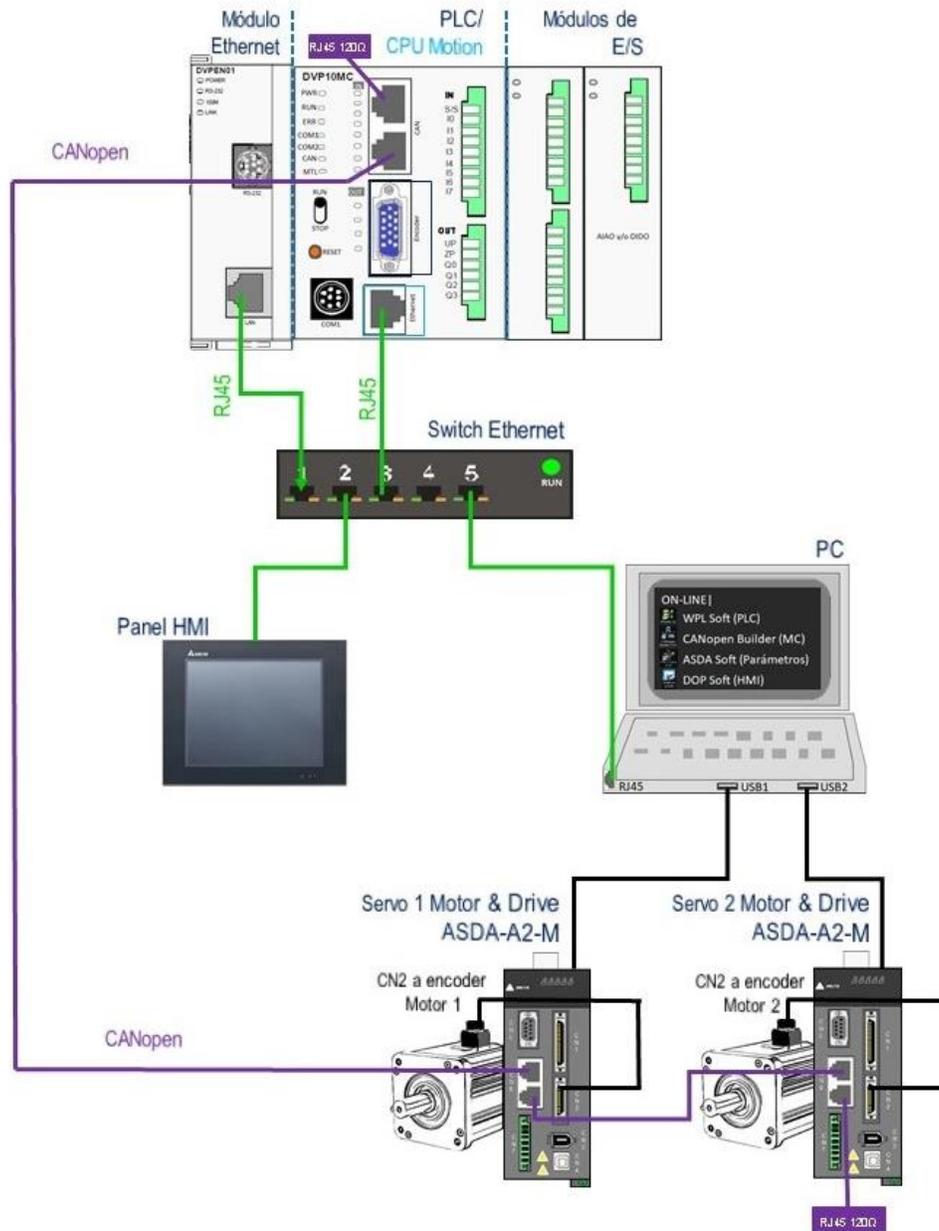


Figura 3.6 Arquitectura del sistema de control usando Hardware Delta

3.2.1 Arquitectura de control

El robot paralelo activado por cables, conjuntamente tiene otros dispositivos que conforman una arquitectura para controlar y establecer la comunicación del controlador a los distintos dispositivos que a continuación se describen:

- Controlador DVP10MC11T

El DVP10MC11T es un tipo de controlador de movimiento multieje generado por Delta. Cumple con el protocolo de comunicación básico CANopen DS301 y el protocolo de control de movimiento DSP402. Este controlador multifunción DVP10MC11T consta de un módulo PLC estándar y un módulo de control de movimiento MC. Para su programación, se requiere del software de programación ISPSOFT, el cual se describirá más adelante. El DVP10MC11T se aplica principalmente para controlar el servoaccionamiento con precisión a través del bus CANopen para cumplir las funciones como control de velocidad, control de posición, etc. (Delta Electronics, Inc., 2012).

El PLC tiene la posibilidad de expandirse por medio de sus puertos; Por el costado izquierdo se encuentra un puerto de extensión paralelo que se puede conectar con un máximo de 7 módulos maestros de bus de campo, como maestro DeviceNet / CANopen, módulos Ethernet y módulos de cantidad analógica de alta velocidad, mientras que por el costado derecho se puede conectar la serie DVP-S de módulos de extensión PLC, tales como módulos de cantidad analógica y cantidad digital de baja velocidad.

Debido a la adaptación de un MC (control de movimiento) en el DVP10MC11T, se obtienen altas velocidades para el sistema de accionamiento. En las figuras 3.6 y 3.7 se presentan las partes que integran el hardware del controlador Delta DVP10MC11T, las cuales se enlistan en las tablas 3.1 y 3.2 respectivamente.

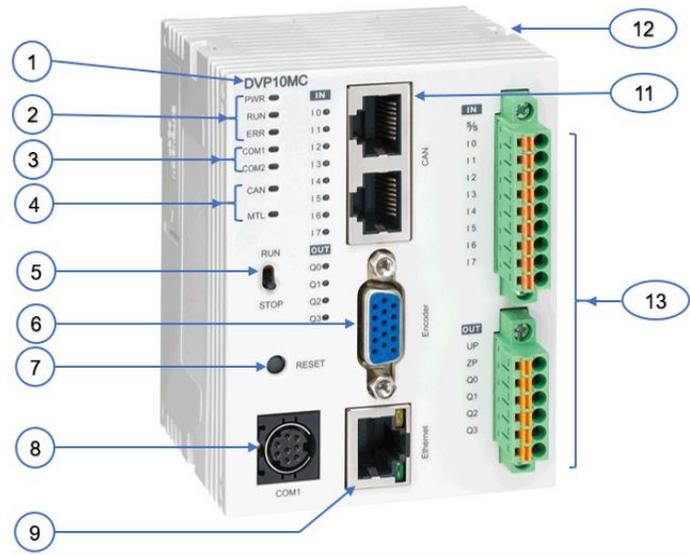


Figura 3.7 Parte frontal del PLC / MC Delta DVP10MC11T (Delta Electronics, Inc., 2012)

Tabla 3.1 Configuración Hardware PLC Delta (Indicadores de figura 3.6)

1	Nombre del modelo	7	Botón de RESET
2	Indicadores POWER / RUN / ERR	8	Puerto de comunicación COM1
3	Indicadores comunicación	9	Puerto de comunicación Ethernet
4	Indicadores CAN, MTL	11	Puerto de comunicación CANopen
5	Interruptor RUN / STOP	12	Clip para fijar módulo de extensión
6	Interface para encoder	13	Terminales de Entradas/Salidas

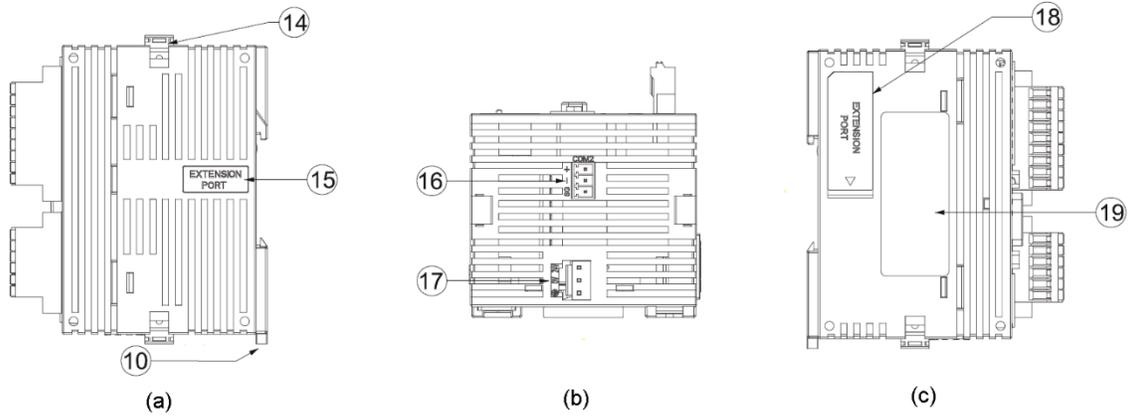


Figura 3.8 a) Parte lateral derecha, b) Parte superior y c) Parte lateral izquierda del DVP10MC11T (Delta Electronics, Inc., 2012)

Tabla 3.2 Configuración Hardware PLC Delta (Indicadores de figura 3.8)

10	Clip para Carril DIN
14	Clip para fijar módulo de extensión
15	Puerto de extensión en el lado derecho
16	Puerto de comunicación COM2
17	Interface para alimentación a 24V
18	Puerto de extensión en el lado izquierdo
19	Placa de nombre

- Módulo de Ethernet DVPEN01-SL

El módulo de Ethernet se requiere para lograr comunicar el PLC con varios dispositivos al mismo tiempo.

DVPEN01-SL es un módulo de comunicación Ethernet para configuración remota y comunicación a través del ISPSOft, software que se utiliza para la programación del PLC y del cual se explicará su proceso más adelante. DVPEN01-SL puede intercambiar datos. Admite el protocolo de comunicación MODBUS TCP y puede realizar monitoreo remoto mediante el uso del software SCADA (Control de Supervisor y Adquisición de Datos) o HMI (Interfaces Hombre-Máquina). DVPEN01-SL puede ser el maestro de MODBUS TCP, enviando instrucciones MODBUS TCP y controlando el equipo periférico. Además, bajo la detección automática de MDI / MDI-X, no necesita usar un cable cruzado.

- Servo drive ASD-A2 1021M y Servomotor ECMA-EΔ1310ES

Los dos servomotores cuya marca es ECMA-EΔ1310ES (ver figura 3.8), forman parte del robot paralelo activado por cables y configuran el sistema de accionamiento de este sistema robótico. En general, la gama de estos servomotores tiene alta precisión de posicionamiento porque cuentan con codificadores incrementales cuya resolución es de 20 bits (1280000 pulsos/revolución) y pueden eliminar comandos inestables a baja velocidad, además de suavizar el funcionamiento del servomotor. Una ventaja de los servomotores es que, al cortar la alimentación, la posición de los servomotores de 17 bits no se perderá.

El servomotor posee dos filtros de supresión de vibraciones (en baja frecuencia) para un sistema de brazo largo y logra minimizar la vibración en los bordes de la máquina de manera efectiva. Por otra parte, cuando se presenta alta frecuencia, el servomotor tiene adecuados dos filtros de muesca automáticos y un filtro de muesca manual para suprimir la resonancia mecánica de manera eficiente. Las características técnicas del ECMA-EΔ1310ES se muestran en el anexo A.

Por su parte, el servo drive ASD-A2 1021 M detecta la ubicación en el eje de referencia, tiene un tiempo de respuesta inferior a 5 microsegundos, puede ejecutar un seguimiento de marcas y como máximo tiene 800 registros. Sus especificaciones se muestran en el anexo B.

La conexión física del sistema de control se generó en un tablero de medidas 640 mm de ancho x 800 mm de alto x 440 mm de profundidad, cuyo propósito es ordenar y organizar los diferentes dispositivos, como se observa en la figura 3.9.



Figura 3.9 Distribución de dispositivos de potencia y control en tablero

3.2.2 Arquitectura eléctrica del sistema

Con respecto a la conexión eléctrica de los mecanismos de accionamiento del CDPR y del controlador DVP10MC11T, en la figura 3.10 se observa un diagrama de la arquitectura eléctrica de este sistema. Se presenta como alimentación principal una fuente trifásica, cuya función es alimentar una fuente de 24 Vcd y ésta a su vez es necesaria para la alimentación del PLC Delta; por otro lado, los 220 Vca mantienen encendidos los servo drives, a menos que se interrumpa la alimentación por medio de los distintos interruptores y guardamotores correspondientes; los servo drives, por su parte, accionan a los servomotores. En el anexo C se presenta el diagrama eléctrico generado en el software de programación “E-plan”.

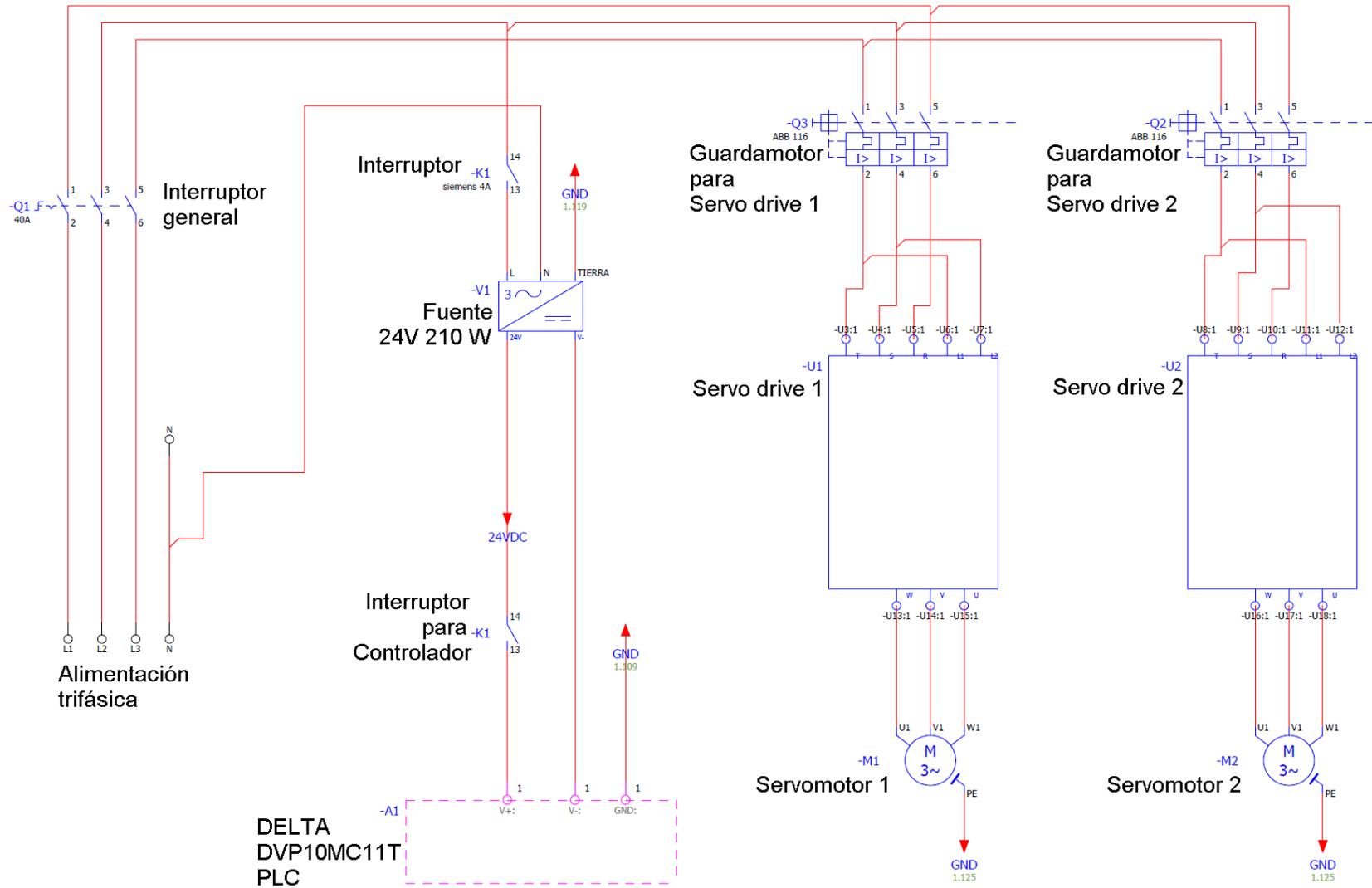


Figura 3.10 Arquitectura eléctrica del sistema de potencia y control

3.3 SISTEMA DE CONTROL

El sistema de control que se implementó en este proyecto (presentado en la figura 3.11) tiene distintas etapas necesarias para su análisis y posterior implementación del sistema de control en el CDPR.

Inicialmente se tiene el bloque de la obtención de la cinemática inversa, cuya entrada es la posición cartesiana (x , y) del efector final y que es deseada por el usuario; su salida serán las longitudes (mm) de los cables l_1 y l_2 las cuales deben ser transformadas en posición angular ($^\circ$) o cantidad de pulsos de referencia (puu). Posteriormente, esas variables obtenidas para el cable l_1 y el cable l_2 (ahora en grados o pulsos) se envían cada uno a distintos puntos de suma donde el sistema realiza la diferencia de la posición deseada contra la posición real de cada servomotor, obteniendo un error de posición para el servomotor 1 y otro error de posición para el servomotor 2. La señal de error (para cada caso) se envía al controlador del servo drive, el cual tiene la función de corregir ese error disminuyéndolo, enviando una señal de control que acciona el servomotor haciéndolo girar la cantidad de grados necesarios y hacia la dirección correcta. El servo drive y el servomotor conforman la etapa de accionamiento de este sistema.

El encoder de cada servomotor detectará las vueltas que gira la flecha de salida de caja de engranaje del servomotor (relación 5:1); las vueltas se interpretarán en grados o pulsos reales y se envían al punto de suma para poder generar la señal de error $e_{(t)}$.

Al accionar los servomotores, gira un tambor que se ubica en la flecha y enrolla o desenrolla el cable. Una revolución del tambor 1 y 2 es igual a 237 mm de cable. Finalmente, los cables se conectan a la plataforma móvil, la cual dará la posición cartesiana real. Se analiza el diagrama y se determina un tipo de control indirecto, puesto que los encoders no detectan directamente la posición cartesiana real de la plataforma, sino que se hace a través de la detección de la posición angular real de los servomotores.

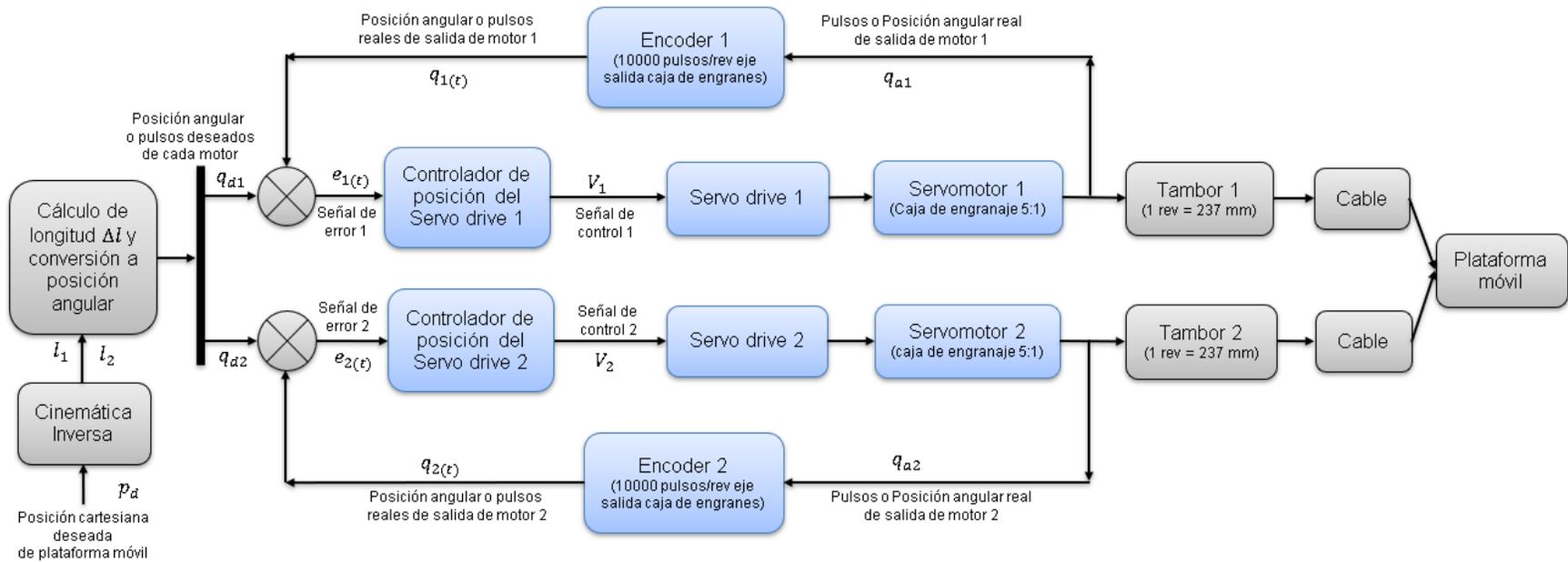


Figura 3.11 Esquema a bloques del sistema de control del CDPR

3.4 ANÁLISIS CINEMÁTICO DEL ROBOT CABLEADO

Se presenta a continuación el análisis cinemático del robot paralelo activado por cables, incluyendo en éste su cinemática inversa, cinemática directa y cinemática diferencial. A partir de la cinemática inversa se obtiene la directa; recordando el apartado del capítulo 2, se menciona la facilidad de la obtención de la cinemática inversa para un robot paralelo; de esa forma se explica el por qué se comienza el análisis de la cinemática inversa de este CDPR y posteriormente la directa.

3.4.1 Cinemática inversa

Para la obtención de la cinemática inversa, se plantea el siguiente hecho: Dadas las posiciones P_x , P_y en el robot paralelo activado por cables, se obtienen las ecuaciones de las longitudes de los cables l_1 y l_2 (ver figura 3.12).

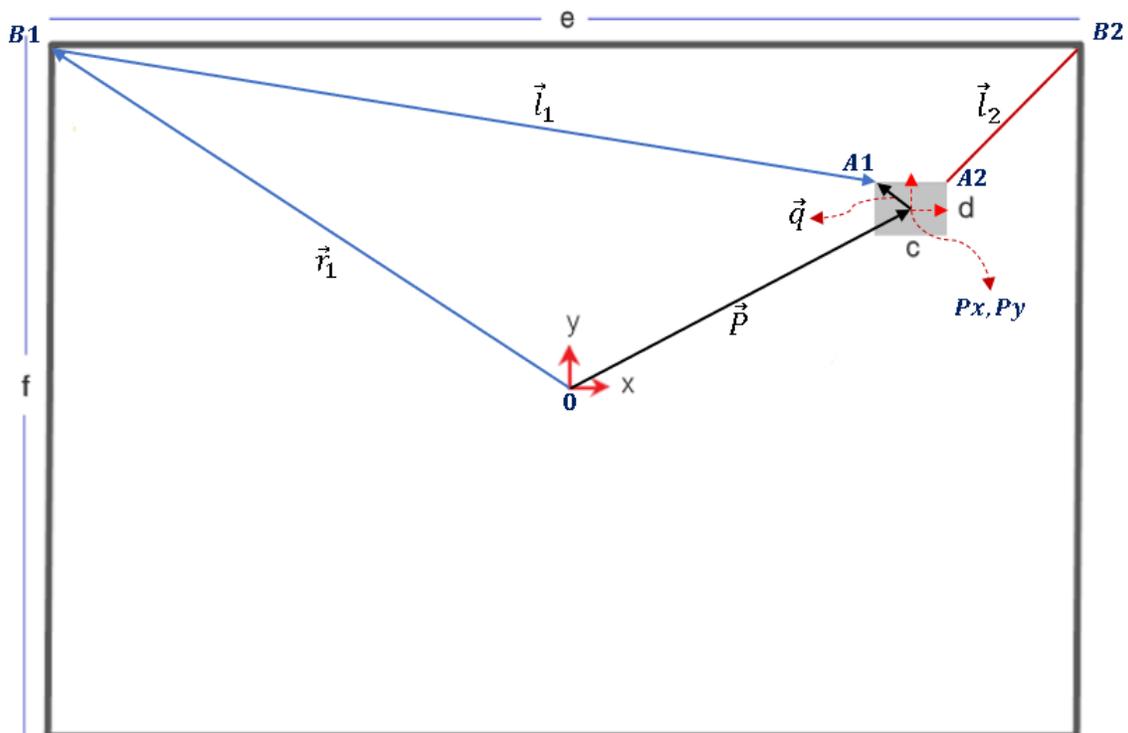


Figura 3.12 Diagrama para obtención de la cinemática inversa del Robot paralelo cableado planar

Del análisis de la figura 3.12, se tiene que:

$$l_1 := \overline{A1B1}$$

$$l_2 := \overline{A2B2}$$

La ecuación vectorial para el cable 1 es:

$$\vec{P} + \vec{q}_1 = \vec{r}_1 + \vec{l}_1$$

Mientras que la ecuación vectorial para el cable 2 es:

$$\vec{P} + \vec{q}_2 = \vec{r}_2 + \vec{l}_2$$

En resumen, se tiene que la ecuación vectorial para cada cable del CDPR es:

$$\vec{P} + \vec{q}_i = \vec{r}_i + \vec{l}_i$$

Al despejar, se obtiene la ecuación vectorial para obtener la cinemática inversa:

$$\vec{l}_i = \vec{P} + \vec{q}_i - \vec{r}_i$$

Para el caso general donde $\theta \neq 0$ es decir, existe rotación sobre el eje 'z'; en forma matricial:

$$l_i = \mathbf{P} + \mathbf{R}_{(z,\theta)} \mathbf{q}'_i - \mathbf{r}_i \quad \forall i = 1, 2$$

Donde los valores de las matrices \mathbf{q}'_i y $\mathbf{r}_i \forall i = 1, 2$ son:

$$\mathbf{q}'_1 = \begin{bmatrix} -c/2 \\ d/2 \\ 0 \end{bmatrix} \quad y \quad \mathbf{r}_1 = \begin{bmatrix} -e/2 \\ f/2 \\ 0 \end{bmatrix}$$

$$\mathbf{q}'_2 = \begin{bmatrix} c/2 \\ d/2 \\ 0 \end{bmatrix} \quad y \quad \mathbf{r}_2 = \begin{bmatrix} e/2 \\ f/2 \\ 0 \end{bmatrix}$$

- Entonces la forma matricial para el cable 1 (actuado por el servomotor 1) es:

$$\begin{bmatrix} l_{1x} \\ l_{1y} \\ l_{1z} \end{bmatrix} = \begin{bmatrix} Px \\ Py \\ Pz \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -c/2 \\ d/2 \\ 0 \end{bmatrix} - \begin{bmatrix} -e/2 \\ f/2 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} l_{1x} \\ l_{1y} \\ l_{1z} \end{bmatrix} = \begin{bmatrix} Px \\ Py \\ Pz \end{bmatrix} + \begin{bmatrix} -\frac{c \cos \theta}{2} - \frac{d \sin \theta}{2} \\ -\frac{c \sin \theta}{2} + \frac{d \cos \theta}{2} \\ 0 \end{bmatrix} - \begin{bmatrix} -e/2 \\ f/2 \\ 0 \end{bmatrix}$$

Por lo tanto, la ecuación de la cinemática inversa para l_1 resulta:

$$l_{1x} = Px - \frac{1}{2}(c \cos \theta + d \sin \theta - e)$$

$$l_{1y} = Py - \frac{1}{2}(c \sin \theta - d \cos \theta + f)$$

$$l_{1z} = 0$$

Se obtiene la magnitud de l_1 usando la norma ℓ_2 o norma euclídea:

$$l_1 = \|\vec{l}_1\| = \sqrt{l_{1x}^2 + l_{1y}^2}$$

Sustituyendo los valores de l_{1x} y l_{1y} :

$$l_1 = \sqrt{\left[Px - \frac{1}{2}(c \cos \theta + d \sin \theta - e) \right]^2 + \left[Py - \frac{1}{2}(c \sin \theta - d \cos \theta + f) \right]^2}$$

$$l_1 = \sqrt{Px^2 - Px(c \cos \theta + d \sin \theta - e) + \frac{1}{4}(c^2 \cos^2 \theta + d^2 \sin^2 \theta + e^2 \dots}$$

$$\sqrt{\dots + 2cd \cos \theta \sin \theta - 2ce \cos \theta - 2de \sin \theta) + Py^2 - Py(c \sin \theta - d \cos \theta + f) \dots}$$

$$\sqrt{\dots + \frac{1}{4}(c^2 \sin^2 \theta + d^2 \cos^2 \theta + f^2 - 2cd \cos \theta \sin \theta + 2cf \sin \theta - 2df \cos \theta)}$$

Simplificando, se obtiene la ecuación general 3.1 para l_1 :

$$l_1 = \sqrt{Px^2 + Py^2 - Px(c \cos \theta + d \sin \theta - e) - Py(c \sin \theta - d \cos \theta + f) \dots}$$

ecuación 3.1

$$\sqrt{\dots + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce \cos \theta - 2de \sin \theta + 2cf \sin \theta - 2df \cos \theta)}$$

Sin embargo, si se buscan orientaciones constantes y paralelas al piso, es decir, donde $\theta = 0$ se tiene:

$$l_1 = \sqrt{Px^2 + Py^2 - Px(c - e) - Py(-d + f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)}$$

Su forma simplificada, cuando $\theta = 0$ se muestra en la ecuación 3.2:

$$l_1 = \sqrt{\left(Px - \frac{c - e}{2} \right)^2 + \left(Py + \frac{d - f}{2} \right)^2}$$

ecuación 3.2

- Pasando al caso de l_2 , la forma matricial para el cable 2 (actuado por el servomotor 2) es:

$$\begin{bmatrix} l_{2x} \\ l_{2y} \\ l_{2z} \end{bmatrix} = \begin{bmatrix} Px \\ Py \\ Pz \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c/2 \\ d/2 \\ 0 \end{bmatrix} - \begin{bmatrix} e/2 \\ f/2 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} l_{2x} \\ l_{2y} \\ l_{2z} \end{bmatrix} = \begin{bmatrix} Px \\ Py \\ Pz \end{bmatrix} + \begin{bmatrix} \frac{c \cos \theta}{2} - \frac{d \sin \theta}{2} \\ \frac{c \sin \theta}{2} + \frac{d \cos \theta}{2} \\ 0 \end{bmatrix} - \begin{bmatrix} e/2 \\ f/2 \\ 0 \end{bmatrix}$$

Por lo tanto, la ecuación de la cinemática inversa para l_2 resulta:

$$l_{2x} = Px + \frac{1}{2}(c \cos \theta - d \sin \theta - e)$$

$$l_{2y} = Py + \frac{1}{2}(c \sin \theta + d \cos \theta - f)$$

$$l_{2z} = 0$$

Se obtiene la magnitud del cable l_2 usando la norma ℓ_2 :

$$l_2 = \|\vec{l}_2\| = \sqrt{l_{2x}^2 + l_{2y}^2}$$

Sustituyendo los valores de l_{2x} y l_{2y} se obtiene:

$$l_2 = \sqrt{\left[Px + \frac{1}{2}(c \cos \theta - d \sin \theta - e) \right]^2 + \left[Py + \frac{1}{2}(c \sin \theta + d \cos \theta - f) \right]^2}$$

$$l_2 = \sqrt{Px^2 + Px(c \cos \theta - d \sin \theta - e) + \frac{1}{4}(c^2 \cos^2 \theta + d^2 \sin^2 \theta + e^2 \dots}$$

$$\sqrt{\dots - 2cd \cos \theta \sin \theta - 2ce \cos \theta + 2de \sin \theta) + Py^2 + Py(c \sin \theta + d \cos \theta - f) \dots}$$

$$\sqrt{\dots + \frac{1}{4}(c^2 \sin^2 \theta + d^2 \cos^2 \theta + f^2 + 2cd \cos \theta \sin \theta - 2cf \sin \theta - 2df \cos \theta)}$$

Simplificando, se obtiene la ecuación general 3.3 para l_2 :

$$l_2 = \sqrt{Px^2 + Py^2 + Px(c \cos \theta - d \sin \theta - e) + Py(c \sin \theta + d \cos \theta - f) \dots}$$

ecuación 3.3

$$\sqrt{\dots + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce \cos \theta + 2de \sin \theta - 2cf \sin \theta - 2df \cos \theta)}$$

De igual forma, si se buscan orientaciones constantes y paralelas al piso, donde $\theta = 0$, se tiene que:

$$l_2 = \sqrt{Px^2 + Py^2 + Px(c - e) + Py(d - f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)}$$

Su forma simplificada, cuando $\theta = 0$ se muestra en la ecuación 3.4.

$$l_2 = \sqrt{\left(Px + \frac{c - e}{2}\right)^2 + \left(Py + \frac{d - f}{2}\right)^2}$$

ecuación 3.4

3.4.2 Cinemática directa

A través de las ecuaciones 3.2 y 3.4 de la cinemática inversa, se obtiene la cinemática directa del robot paralelo activado por cables estudiado:

$$l_1 = \sqrt{Px^2 + Py^2 - Px(c - e) - Py(-d + f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)}$$

$$l_2 = \sqrt{Px^2 + Py^2 + Px(c - e) + Py(d - f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)}$$

Todos los términos de las ecuaciones 3.2 y 3.4 se elevan al cuadrado y se obtiene:

$$l_1^2 = Px^2 + Py^2 - Px(c - e) - Py(-d + f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)$$

$$l_2^2 = Px^2 + Py^2 + Px(c - e) + Py(d - f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)$$

- Posteriormente, en la ecuación donde l_1 está elevada al cuadrado, se despeja Px^2 :

$$Px^2 = l_1^2 - Py^2 + Px(c - e) + Py(-d + f) - \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)$$

El valor anterior de Px^2 se sustituye en la ecuación en la ecuación donde l_2 está elevada al cuadrado:

$$l_2^2 = l_1^2 - Py^2 + Px(c - e) + Py(-d + f) - \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df) \\ + Py^2 + Px(c - e) - Py(-d + f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)$$

Simplificando:

$$l_2^2 = l_1^2 + 2Px(c - e)$$

Se despeja Px y se obtiene el primer elemento de la coordenada:

$$Px = \frac{l_2^2 - l_1^2}{2(c - e)} \quad \text{ecuación 3.5}$$

- Para encontrar la ecuación de la coordenada Py en relación con las longitudes l_1 y l_2 se suman las ecuaciones 3.2 y 3.4 elevadas al cuadrado:

$$l_1^2 + l_2^2 = 2Px^2 + 2Py^2 - 2Py(f - d) + \frac{2}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)$$

$$l_1^2 + l_2^2 = 2Px^2 + 2Py^2 - 2Py(f - d) + \frac{2}{4}(f^2 - 2df + d^2) + \frac{2}{4}(c^2 - 2ce + e^2)$$

Se tiene que:

$$2Py^2 - 2Py(f - d) + \frac{2}{4}(f^2 - 2df + d^2) = 2 \left[Py - \frac{(f - d)}{2} \right]^2$$

Entonces al factorizar toda la ecuación se reescribe como:

$$2 \left[Py - \frac{(f-d)}{2} \right]^2 = l_1^2 + l_2^2 - 2Px^2 - \frac{2}{4}(c^2 - 2ce + e^2)$$

Sustituyendo el valor de $Px = \frac{l_2^2 - l_1^2}{2(c-e)}$:

$$2 \left[Py - \frac{(f-d)}{2} \right]^2 = l_1^2 + l_2^2 - 2 \left[\frac{l_2^2 - l_1^2}{2(c-e)} \right]^2 - \frac{1}{2}(c-e)^2$$

Finalmente despejando Py de la última ecuación, se tiene:

$$Py = -\sqrt{\frac{l_1^2 + l_2^2 - 2 \left[\frac{l_2^2 - l_1^2}{2(c-e)} \right]^2 - \frac{1}{2}(c-e)^2}{2}} + \frac{(f-d)}{2} \quad \text{ecuación 3.6}$$

3.4.3 Cinemática diferencial y singularidades

La cinemática diferencial inversa del robot paralelo activado por cables estudiado se obtiene a partir de la ecuación 2.6.

$$\dot{\mathbf{q}} = \mathbf{J}_q \dot{\mathbf{p}}$$

Donde:

$$\dot{\mathbf{q}} = \begin{bmatrix} \dot{l}_1 \\ \dot{l}_2 \end{bmatrix}$$

$$\dot{\mathbf{p}} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

$$\mathbf{J} = \begin{bmatrix} \frac{\partial l_1}{\partial P_x} & \frac{\partial l_1}{\partial P_y} \\ \frac{\partial l_2}{\partial P_x} & \frac{\partial l_2}{\partial P_y} \end{bmatrix}$$

De las ecuaciones de cinemática inversa del robot paralelo activado por cables planar, se obtiene:

$$l_1 = \sqrt{Px^2 + Py^2 - Px(c - e) - Py(-d + f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)}$$

Para términos de simplificación se le da una variable a la siguiente ecuación de l_1 :

$$r = Px^2 + Py^2 - Px(c - e) - Py(-d + f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)$$

Se obtienen los primeros elementos del Jacobiano con respecto a l_1 .

$$\frac{\partial l_1}{\partial P_x} = \frac{1}{2} r^{-1/2} [2P_x - (c - e)]$$

$$\frac{\partial l_1}{\partial P_y} = \frac{1}{2} r^{-1/2} [2P_y - (-d + f)]$$

En el caso de la ecuación de C.I. de la ecuación l_2 :

$$l_2 = \sqrt{Px^2 + Py^2 + Px(c - e) + Py(d - f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)}$$

Para términos de simplificación se le da una variable a la siguiente ecuación de l_2 :

$$s = Px^2 + Py^2 + Px(c - e) + Py(d - f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)$$

Se obtienen los últimos elementos del Jacobiano con respecto a l_2 .

$$\frac{\partial l_2}{\partial P_x} = \frac{1}{2} s^{-1/2} [2P_x + (c - e)]$$

$$\frac{\partial l_2}{\partial P_y} = \frac{1}{2} s^{-1/2} [2P_y + (d - f)]$$

Finalmente, los términos del Jacobiano se sustituyen en la ecuación 2.6 en forma matricial:

$$\begin{bmatrix} \dot{l}_1 \\ \dot{l}_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{2} r^{-1/2} [2P_x - (c - e)] & \frac{1}{2} r^{-1/2} [2P_y - (-d + f)] \\ \frac{1}{2} s^{-1/2} [2P_x + (c - e)] & \frac{1}{2} s^{-1/2} [2P_y + (d - f)] \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

$$\dot{l}_1 = \frac{1}{2}r^{-1/2}[2P_x - (c - e)]v_x + \frac{1}{2}r^{-1/2}[2P_y - (-d + f)]v_y$$

$$\dot{l}_2 = \frac{1}{2}s^{-1/2}[2P_x + (c - e)]v_x + \frac{1}{2}s^{-1/2}[2P_y + (d - f)]v_y$$

Se simplifican de la siguiente forma:

$$\dot{l}_1 = \frac{1}{2}r^{-1/2}[(2P_x - c + e)v_x + (2P_y + d - f)v_y]$$

$$\dot{l}_2 = \frac{1}{2}s^{-1/2}[(2P_x + c - e)v_x + (2P_y + d - f)v_y]$$

Y se sustituyen los valores de las variables 'r' y 's', quedando las ecuaciones 3.7 y 3.8 de la cinemática diferencial inversa de la siguiente forma:

ecuación 3.7

$$\dot{l}_1 = \frac{[(2P_x - c + e)v_x + (2P_y + d - f)v_y]}{2\sqrt{Px^2 + Py^2 - Px(c - e) - Py(-d + f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)}}$$

ecuación 3.8

$$\dot{l}_2 = \frac{[(2P_x + c - e)v_x + (2P_y + d - f)v_y]}{2\sqrt{Px^2 + Py^2 + Px(c - e) + Py(d - f) + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce - 2df)}}$$

Finalmente, se obtienen las singularidades del sistema robótico presentado. Para su cálculo se necesita la ecuación 2.7 presentada en el segundo capítulo.

$$\det[\mathbf{J}(\mathbf{q})] = 0$$

Por lo tanto, se obtiene el determinante del siguiente Jacobiano:

$$\mathbf{J}(\mathbf{q}) = \begin{bmatrix} \frac{1}{2}r^{-1/2}[2P_x - (c - e)] & \frac{1}{2}r^{-1/2}[2P_y - (-d + f)] \\ \frac{1}{2}s^{-1/2}[2P_x + (c - e)] & \frac{1}{2}s^{-1/2}[2P_y + (d - f)] \end{bmatrix}$$

$$\begin{aligned} \det[\mathbf{J}(\mathbf{q})] &= \frac{1}{4}r^{-1/2}s^{-1/2}(2P_x - c + e)(2P_y + d - f) \\ &\quad - \frac{1}{4}r^{-1/2}s^{-1/2}(2P_x + c - e)(2P_y + d - f) \end{aligned}$$

Se factoriza y se obtiene:

$$\det[\mathbf{J}(\mathbf{q})] = \frac{1}{4}r^{-1/2}s^{-1/2}(2P_y + d - f)(2e - 2c)$$

Entonces se iguala a cero para obtener una singularidad:

$$\frac{1}{4}r^{-1/2}s^{-1/2}(2P_y + d - f)(2e - 2c) = 0$$

Lo anterior se cumple cuando:

$$(2P_y + d - f)(2e - 2c) = 0 \quad \therefore \quad 2P_y + d - f = 0$$

Y se obtiene la singularidad presentada en la ecuación 3.9, la cual demuestra que en esos puntos la plataforma del robot debe pasar con cuidado o a gran velocidad.

$$P_y = \frac{f - d}{2}$$

ecuación 3.9

3.5 PROGRAMACIÓN EN SOFTWARE DELTA

Para entender cómo se implementó el programa de sistema de control en los controladores marca Delta, se requiere analizar el proceso de programación de los softwares usados en este proyecto. Por una parte, se usó el software “ISPSOft” para la programación del PLC, “CANopen” Builder para la programación del Motion Control, el software “DOPSoft” para programar la HMI y “ASDA Soft” para la configuración de los parámetros del servo drive.

- Nociones básicas y requerimientos del sistema

Para programar en los distintos softwares se requieren conocimientos básicos de automatización; los requisitos básicos de la herramienta de configuración de red para la instalación de las aplicaciones se detallan en la tabla 3.3.

Tabla 3.3 Requerimientos para configuración de red

<i>Ítem</i>	<i>Requerimientos del Sistema</i>
<i>Sistema operativo</i>	Windows® XP/VISTA/7(32-bit/64-bit)/8(32-bit/64-bit)/10(64-bit)
<i>Memoria</i>	256 MB o superior (512 MB o superior es recomendado)
<i>Disco duro</i>	Capacidad: más de 500 MB de espacio o superior
<i>Interfaz de la computadora</i>	Para conectar al PLC: Puerto RJ45, puerto USB y puerto Ethernet. El usuario debe seleccionar uno de ellos de acuerdo a la interfaz de comunicación prevista por el PLC o el módulo usado.
<i>Software de comunicación</i>	COMMGR, un gestor de comunicación, debe ser instalado en la computadora.

3.5.1 Programación del Controlador Lógico Programable Delta

El ciclo de trabajo del PLC Delta DVP10MC11T (véase figura 3.13) utiliza un método de exploración estándar al momento de evaluar el programa de usuario, tal como se muestra en la tabla 3.4.

Tabla 3.4 Proceso de ciclo de trabajo del PLC Delta

<i>Explorar estado de entrada</i>	Leer el estado de entrada física y guardar los datos en la memoria interna.
<i>Evaluar el programa de usuario</i>	Evaluar el programa de usuario con datos almacenados en la memoria interna. La exploración de programa inicia de arriba hacia abajo y de izquierda a derecha hasta llegar al final del programa.
<i>Actualizar las salidas</i>	Escribir los datos evaluados a las salidas físicas.

El PLC lee el estado de cada entrada (encendido/apagado) y almacena ese valor en la memoria antes de evaluar el programa. Posteriormente, si se hace un cambio en las entradas externas, no se actualizará en la memoria hasta el inicio del próximo ciclo de ejecución.

Se ejecutan las instrucciones (arriba a abajo – izquierda a derecha) y se almacenan en la memoria interna los datos.

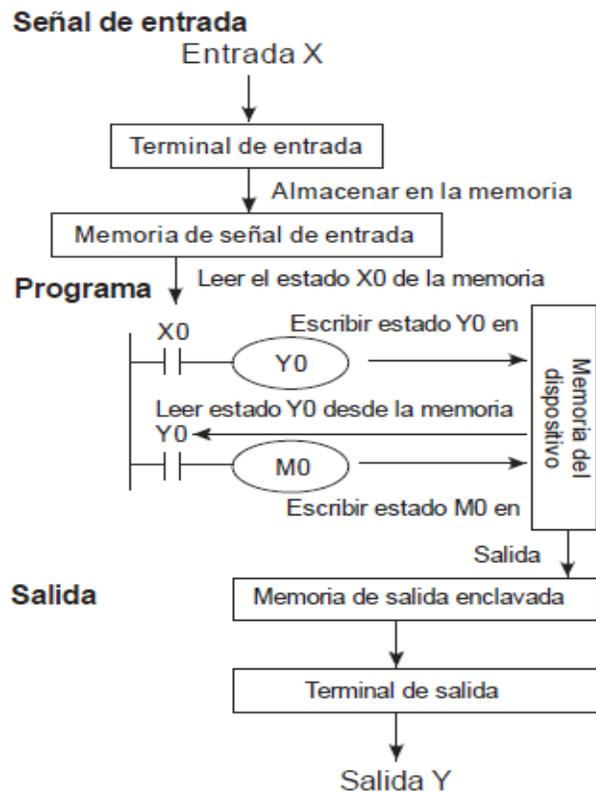


Figura 3.13 Ciclo de trabajo del PLC Delta

3.5.1.1 Lenguaje de Programación en ISPSOft

Existen tres lenguajes de programación disponibles para el tipo de PLC DVP10MC11T: Ladder Diagram (LD), Sequential Function Chart (SFC) e Instruction List (IL).

Para realizar la programación del PLC Delta en este proyecto de tesis, se optó por la programación en diagrama escalera por dos razones: la experiencia que se tiene con respecto a este tipo de lenguaje y las características factibles del mismo que se describen a continuación.

El lenguaje Ladder Diagram (LD), es uno de los lenguajes de programación definidos por IEC 61131-3 y es usado frecuentemente al crear un programa para PLC. Un diagrama de escalera LD es un lenguaje de programación que representa un programa por diagrama gráfico basado en los diagramas de circuitos del hardware de lógica de relé. Se muestran los elementos del LD en la figura 3.14.

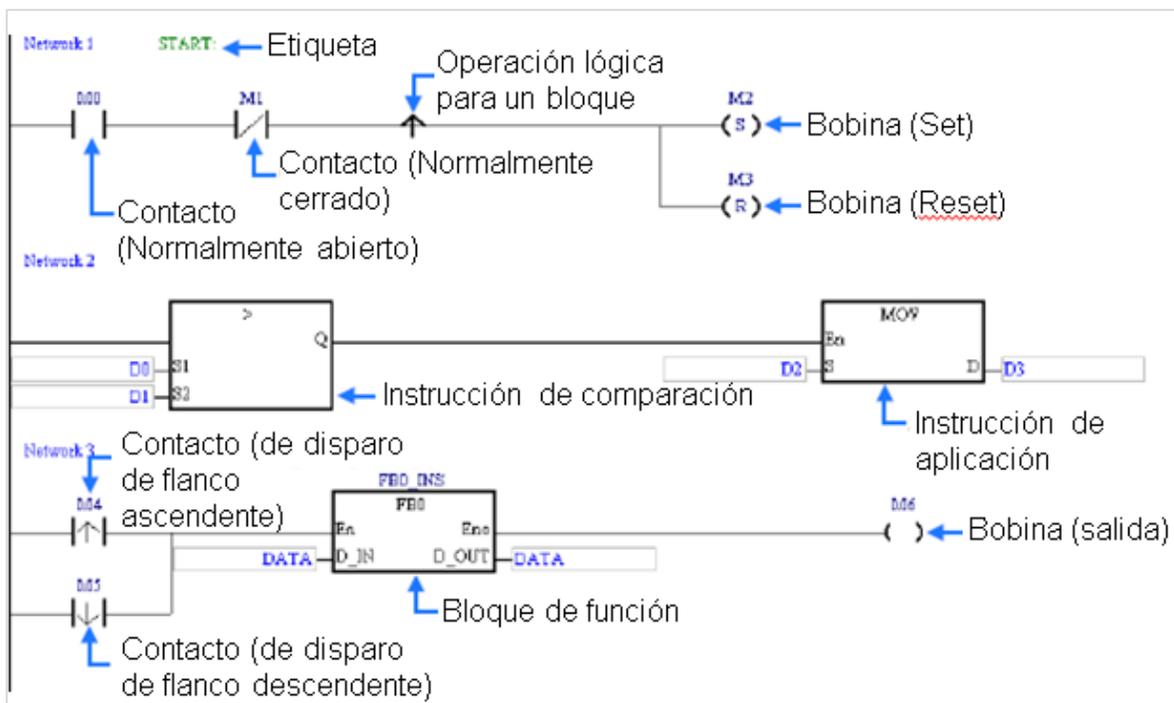


Figura 3.14 Lenguaje de Programación "Ladder Diagram" para el PLC

3.5.1.2 Ayuda en pantalla

En el software ISPSOft v3.04 se puede recurrir a la Ayuda integrada en el mismo (ver figura 3.15) y se accede seleccionando la pestaña “Help” → “PLC Instruction and Special Registers Reference”.

Los capítulos correspondientes a la ayuda son:

- Contenido
- Instrucciones Básicas
- Instrucciones de Aplicación
- Registros Especiales (M)
- Registros Especiales (D)

Si se requiere una descripción más detallada de las instrucciones del PLC Delta también se puede consultar el manual que se titula “DVP-ES2/EX2/SS2/SA2/SX2/SE&TP Operation Manual - Programming” (Delta Electronics, Inc., 2014).

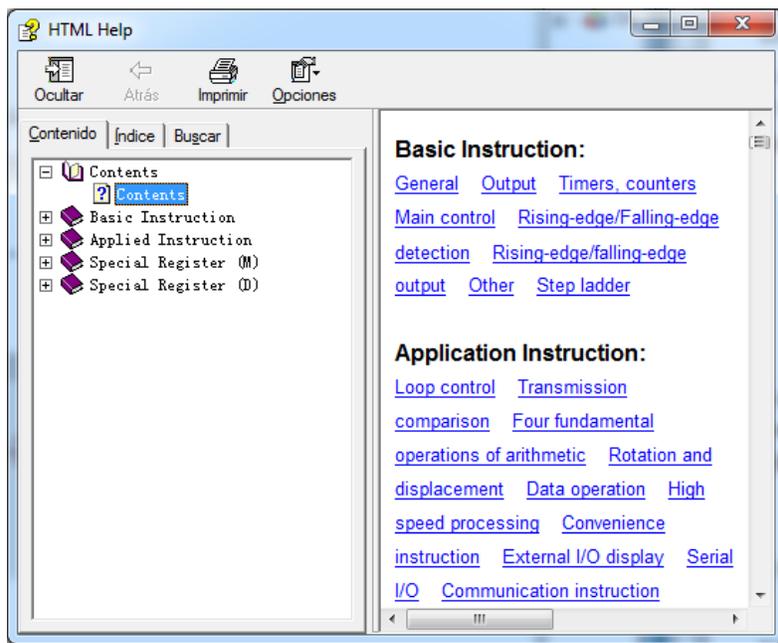


Figura 3.15 Ayuda del ISPSOft

3.5.1.3 Generar un programa en ISPSOft v3.04

Se ejecuta el programa ISPSOft v3.07; para crear un nuevo archivo de debe dirigir a “Menu bar” y se selecciona la opción “File” → “New” (observar figura 3.16). Se abrirá una ventana donde se nombrará el proyecto y se configurarán sus características. En el proyecto de tesis se seleccionará “Controller Type” → “DVP” y en “PLC Type” → “MC”, porque se usará el PLC Delta DVP10MC11T. Después de configurar se da clic en “OK”.

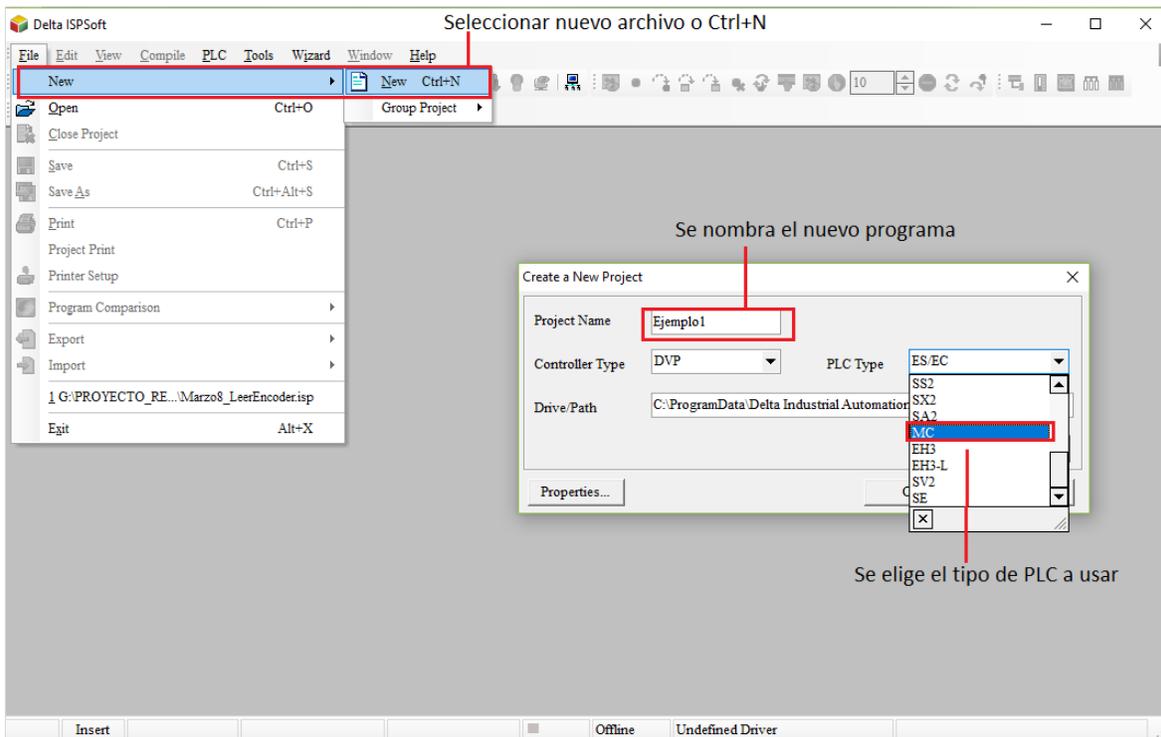


Figura 3.16 Creación de un nuevo programa en ISPSOft

Dentro de la ventana del software se abrirán nuevas áreas de trabajo (véase la figura 3.17). A continuación, se describe el entorno de programación del software ISPSOft v3.04.

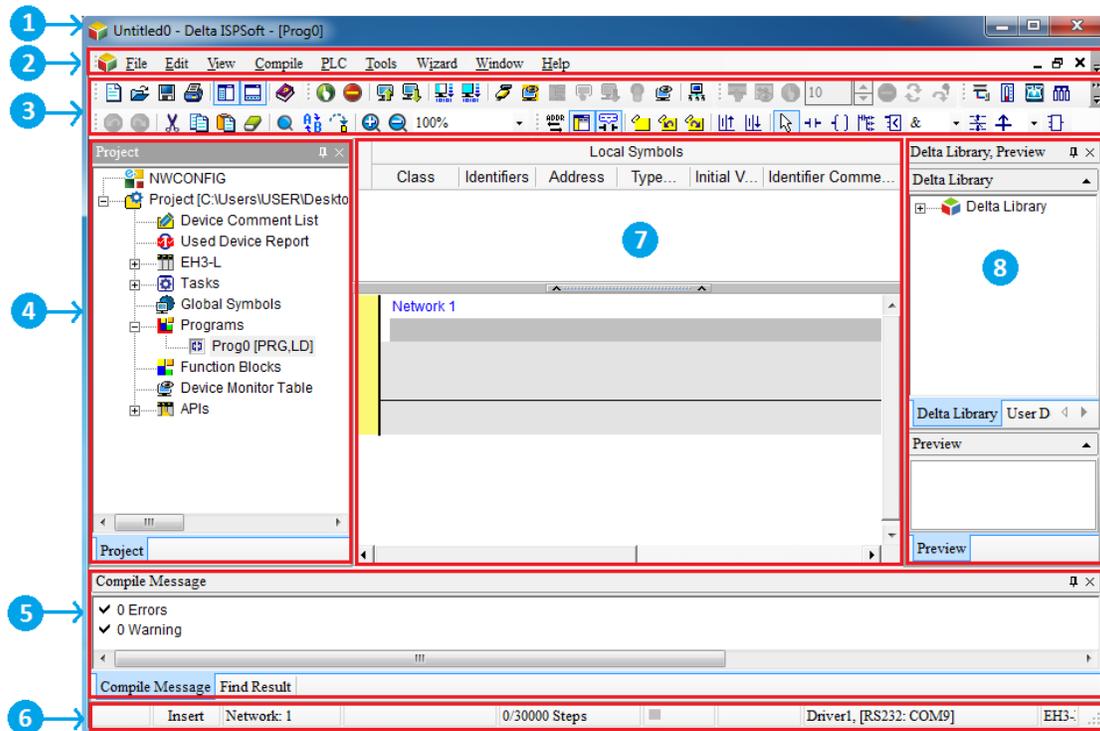


Figura 3.17 Ventana principal de ISPSOFT

Window title (1): Muestra un nombre de proyecto y un título de programa.

Menu bar (2): Tiene 8 menús, el primero es “File” el cual proporciona la función de acceder a proyectos; “Edit” da la función de editar un proyecto; “View” proporciona la función de visualización de información sobre un proyecto y la función de establecer un ambiente de trabajo; “Compile” da la función de verificar la sintaxis y la función de compilar un programa en un código de ejecución; “PLC” proporciona la función de operar o configurar un PLC a través de ISPSOFT; “Tools” da algunas funciones convenientes, como configuración de la comunicación, cambiar tipo de PLC, entre otras; “Window” proporciona la función de administrar ventanas en el área de trabajo; finalmente “Help” da la ayuda al usuario.

Toolbar (3): Hay 5 barras de herramientas: “File toolbar” el cual proporciona las funciones relacionadas con el procesamiento de un proyecto; “Editing toolbar” da las funciones relacionadas con el trabajo de edición; “PLC toolbar” proporciona las funciones relacionadas con el funcionamiento de un PLC; “Debugging toolbar” da

las funciones relacionadas con la depuración de un programa; “Programming toolbar” proporciona las funciones relacionadas con la edición de un programa.

Project management área (4): Adopta una interfaz que utiliza una estructura de árbol jerárquica.

Message display área (5): Los mensajes relacionados con la compilación de un programa y la búsqueda de datos en un proyecto se muestran en esta área.

Status bar (6): Se muestra información sobre la edición de un programa y la conexión.

Working area (7): En esta parte se muestran un área de edición de programas, una tabla de símbolos, una tabla de dispositivos, etc.

Delta Library (8): El contenido de la biblioteca Delta y la biblioteca definida por el usuario se muestran en esta área.

Posteriormente, para generar un nuevo programa se da clic derecho en “Project management area” (véase figura 3.18) y se elige “Programs”→“New”.

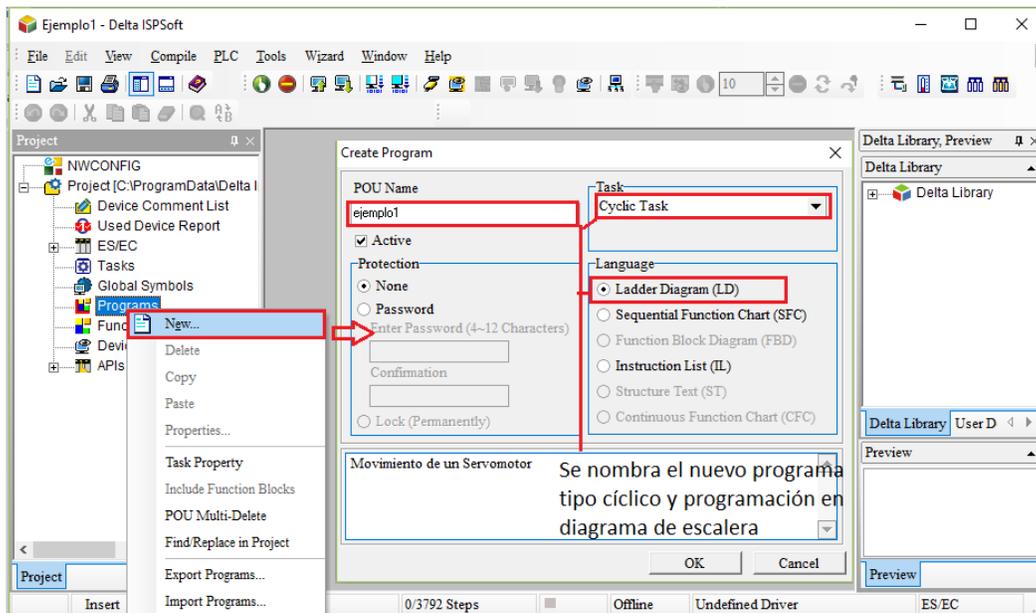
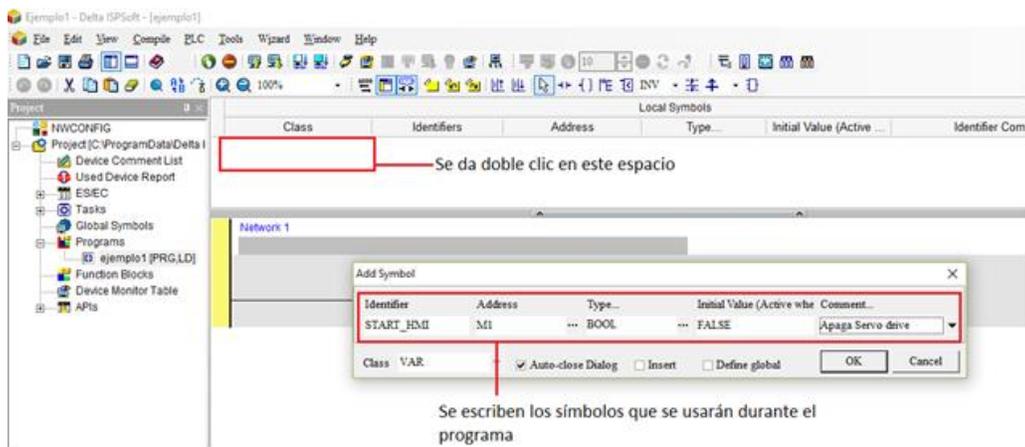


Figura 3.18 Configuración del nuevo programa

Se abrirá una nueva ventana “Create Program”; ahí se nombra el programa principal y se elige tipo de tarea y lenguaje de programación de acuerdo a los requerimientos y habilidades del usuario. El tipo de tarea a elegir en el proyecto es “Cyclic Task” con un lenguaje de programación “Ladder Diagram (LD)”.

El siguiente paso es generar una tabla de símbolos para su uso durante el programa. La tabla se abre por default al generar el nuevo programa; se ubica en la parte de arriba del área de programación en “Working area”. Se da clic en el espacio indicado en la figura 3.19 y se abrirá una nueva ventana donde se editarán los símbolos a usar. La configuración de los símbolos dependerá de los requisitos de programación del usuario, por ejemplo: tipos de datos, tipos de registros, tamaño del registro, datos locales o globales; se debe considerar la existencia de registros especiales, los cuales tienen su propia función y esos no se deben usar como registros normales.



a)

Local Symbols						
Class	Identifiers	Address	Type...	Initial Value (Active ...	Identifier Comment...	
VAR	START_MC	D6250	WORD	0	Envía a MC para encender drive	
VAR	STOP_MC	D6251	WORD	0	Envía a MC para apagar drive	
VAR	MOT_REL_MC	D6253	WORD	0	Envía a MC para mov relativo del motor	
VAR	MOT_ABS_MC	D6267	WORD	0	Envía a MC para mov home del motor	
VAR	START_HMI	M0	BOOL	FALSE	Botón Start HMI	
VAR	STOP_HMI	M1	BOOL	FALSE	Botón Stop HMI	
VAR	ACTIVE_FLAG	M10	BOOL	FALSE	Marca Siempre Activa	
VAR	MOT_ABS_HMI	M16	BOOL	FALSE	Botón home del motor HMI	
VAR	MOT_REL_HMI	M2	BOOL	FALSE	Botón movimiento relativo del motor HMI	
VAR	MOV_D2650	Y0	BOOL	FALSE	Envía Y0 para encender drive	
VAR	MOV_D6251	Y4	BOOL	FALSE	Envía Y4 para apagar drive	

b)

Figura 3.19 a) Edición de símbolos y b) Tabla de símbolos creada

Ya generada la tabla de símbolos, se comienza a programar en LD. Se selecciona la primera línea de programación; en “Programming toolbar” se da clic en el ícono “Contact” como se muestra en la figura 3.20.

Ese contacto se arrastra hasta la línea de programación dando clic en esa área. El contacto aparecerá y luego se da doble clic sobre el contacto para configurarlo como: Normalmente abierto, Normalmente cerrado, Disparador de flanco ascendente o Disparador de flanco descendente, según lo requiera el programador.

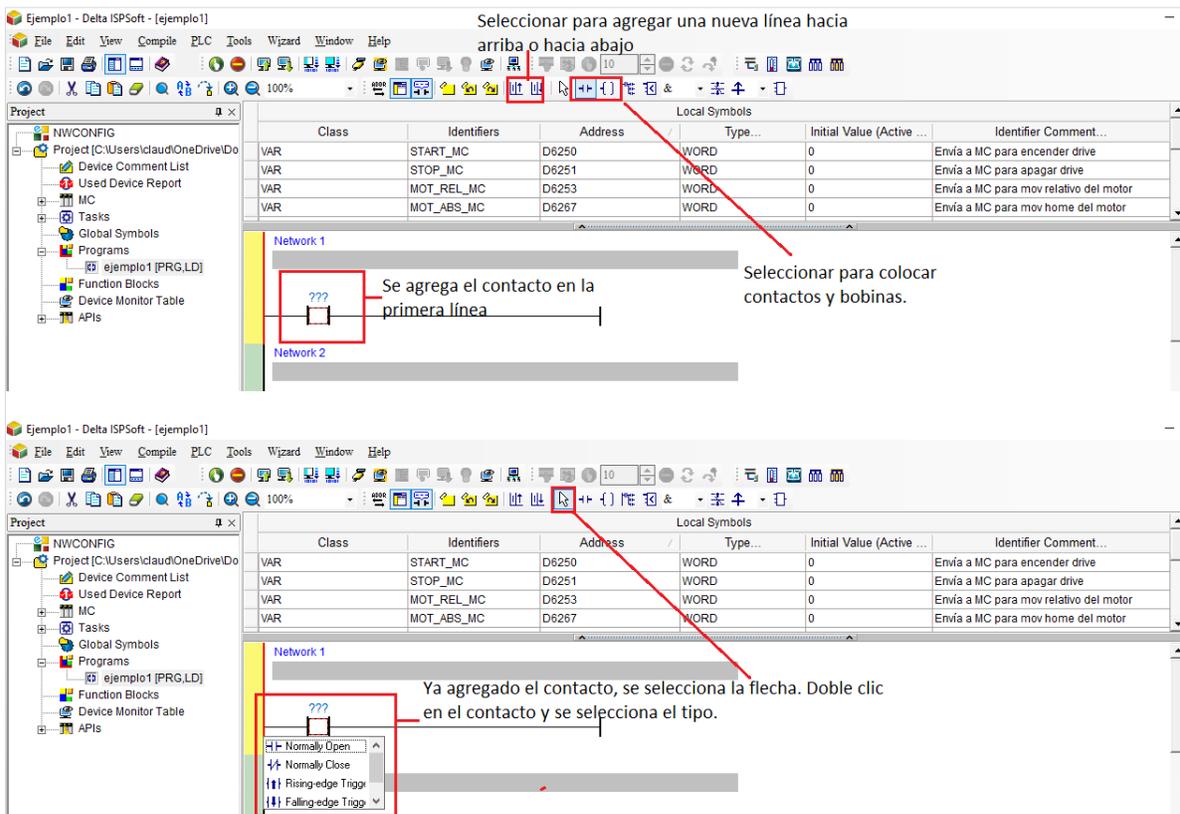


Figura 3.20 Colocación de contactos y bobinas.

En la figura 3.21 se observa la primera línea de programación finalizada con contactos que activan o desactivan la bobina Y0; en el área de comentarios (parte superior de la línea de programación) se escribe una leyenda, por ejemplo “Activar Y0 desde la marca M2 (START_HMI)”; esta etiqueta describe la función que desempeña la línea en el programa.

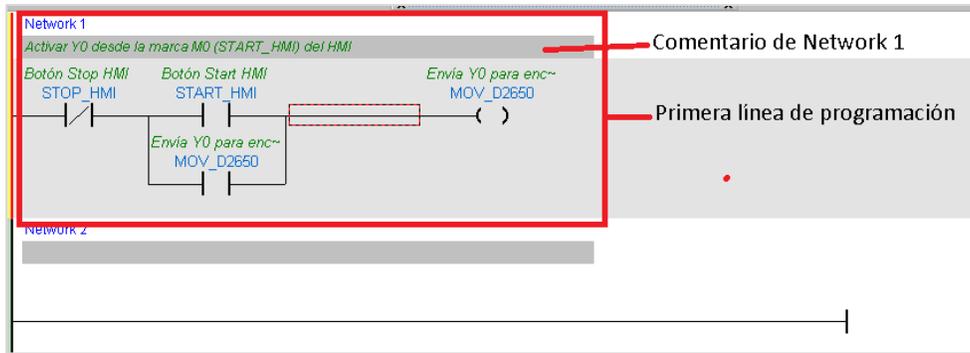


Figura 3.21 Ejemplo de una línea de programación

También existen bloques de programación (instrucciones) de tipo “Comparison Instruction”, API’s (Application Programming Interface) y FB’s (Block Functions, bloques generados por el usuario).

En la figura 3.22 se muestra cómo programar usando un bloque de instrucción. Dar clic en el ícono de “Selección de API”; se abrirá una ventana “API FB” y se selecciona el bloque de la función requerida por el programador.

Figura 3.22 Uso de Bloques/Instrucciones de Programación

Posteriormente se debe definir el tipo de dato de entrada “S” y el de destino “D”. Se hace doble clic sobre el bloque de instrucción “MOV” y se hará como muestra la figura 3.23. En el recuadro “API List” se observa qué tipo de dato es válido para cada instrucción.

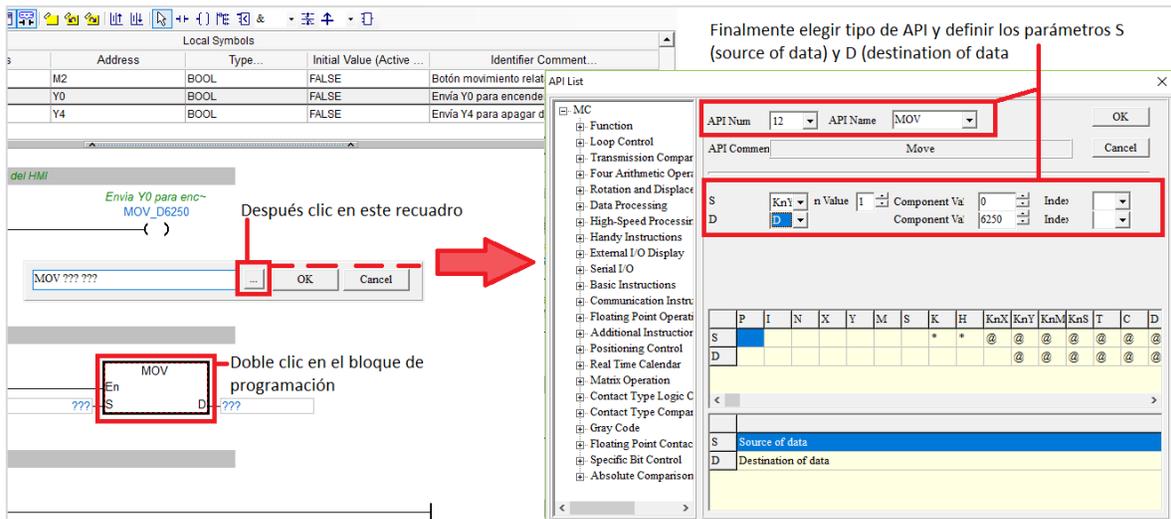


Figura 3.23 Configurar datos de entrada y de destino para una API

Tanto en ISPSOft como en CANopen Builder existen unos registros especiales para el envío y recibo de datos entre el PLC y MC; éstos son: registros de lectura del D6000 ~ D6226 y registros de escritura de D6250 ~ D6476 (véase figura 3.24).

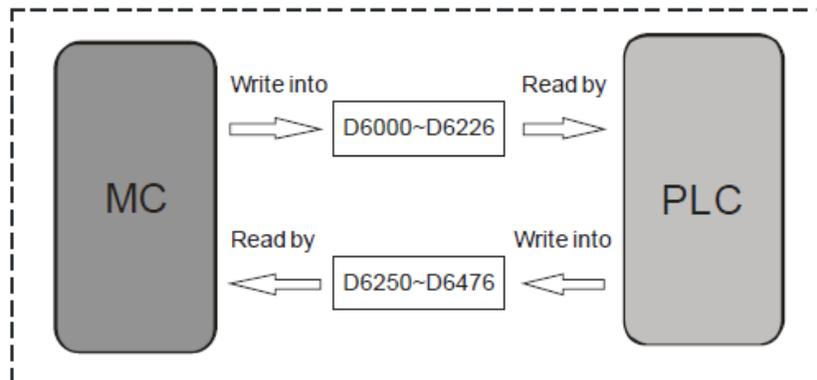


Figura 3.24 Registros especiales de lectura y escritura entre PLC – MC

3.5.1.4 Configuración de la red en ISPSOft v3.04

Después de terminar el programa en ISPSOft, se requiere conectarse a la red para poder cargar el programa al PLC. El primer paso es hacer la configuración del driver de comunicación en COMMGR; cuando se ubica en la pantalla COMMGR, se da clic en el botón “Add” y se prosigue a nombrar el driver (Driver1), posteriormente se elige el tipo RJ45 y se selecciona la ubicación del protocolo COM. Finalmente se autodetecta y se visualiza si se logró hacer la detección del primer driver, como se muestra en la figura 3.25.

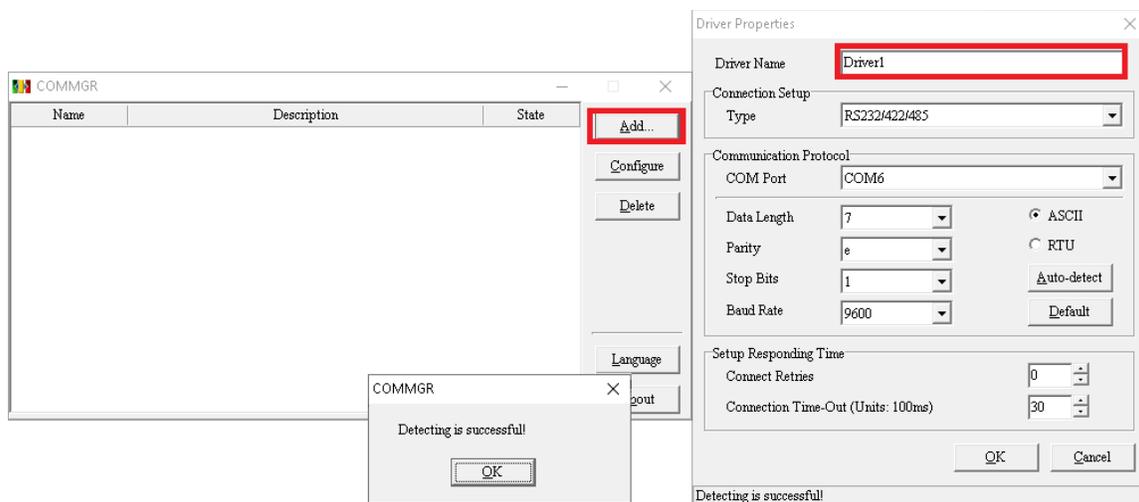


Figura 3.25 Configuración del driver 1 Conexión RJ45

Posteriormente, en “ISPSOft” se selecciona “Menu bar” y después la opción “Tools” → “Communication Setting” y se abre una nueva ventana. Se debe elegir el driver de comunicación configurado en COMMGR y aceptar la comunicación del driver1, como se muestra en la figura 3.26.

De esa forma ya se encuentra comunicado el PLC con la computadora para cargar los programas y monitorearlos. Primero se debe compilar el programa para detectar si no existen errores de sintaxis o programación seleccionando el ícono “Compile the Project”; finalmente se transfiere el archivo al PLC seleccionando la opción “Transfer”

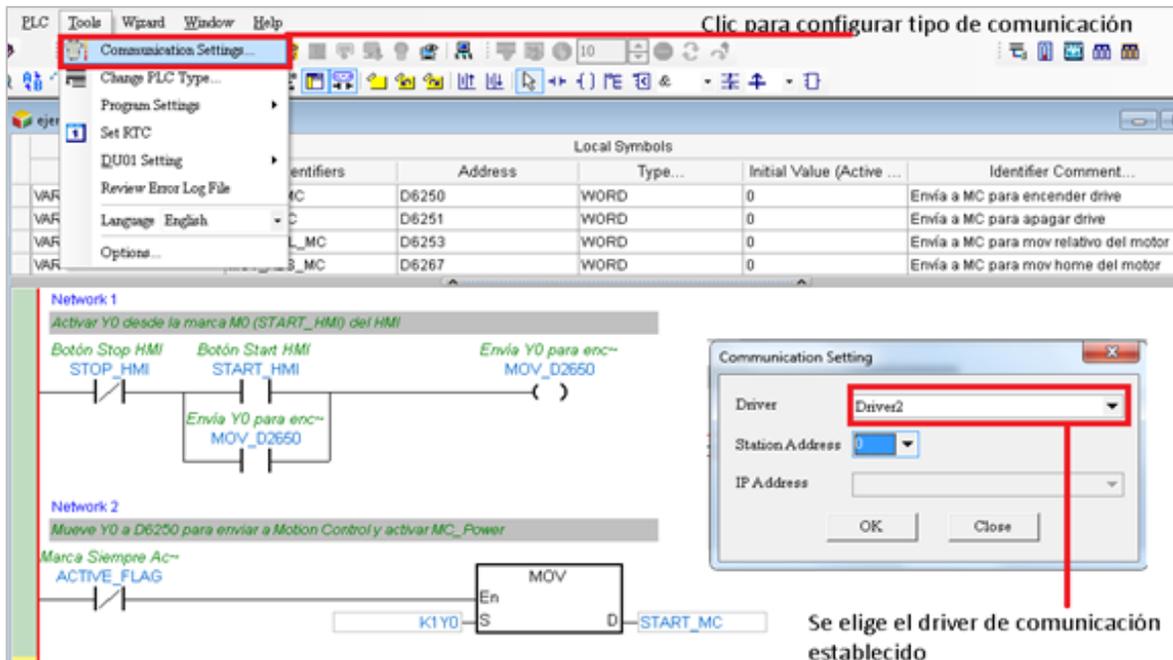


Figura 3.26 Configuración de comunicación por RJ45

Una vez presionado "Transfer", se espera a que el progreso de "Object Code" llegue a un 99%; una vez terminada la carga del programa se debe confirmar si el usuario desea ejecutar el PLC en estado "RUN". Se selecciona "Aceptar" y se finaliza la transferencia del programa al PLC como se muestra en la figura 3.27.

Para visualizar o monitorear en línea el progreso del programa ya cargado (figura 3.28), se debe seleccionar "Online Mode" o teclear el comando Ctrl+F4. Así se corroborarán los estados actuales de cada entrada y salida de datos.

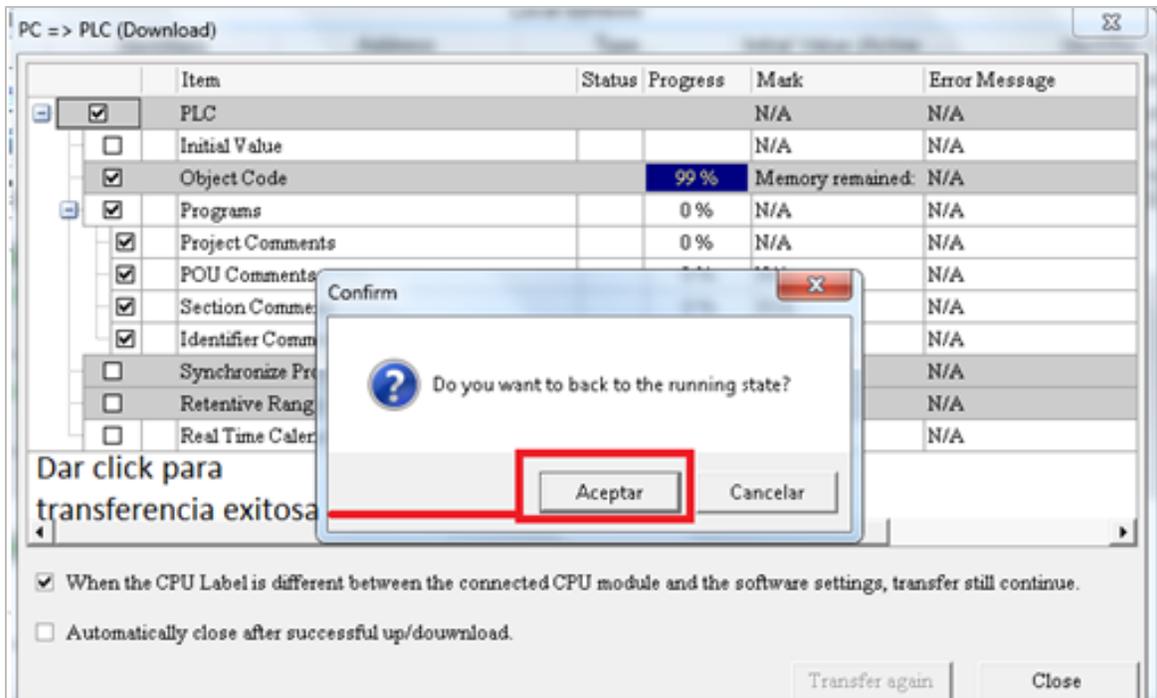


Figura 3.27 Proceso de transferencia del programa al PLC

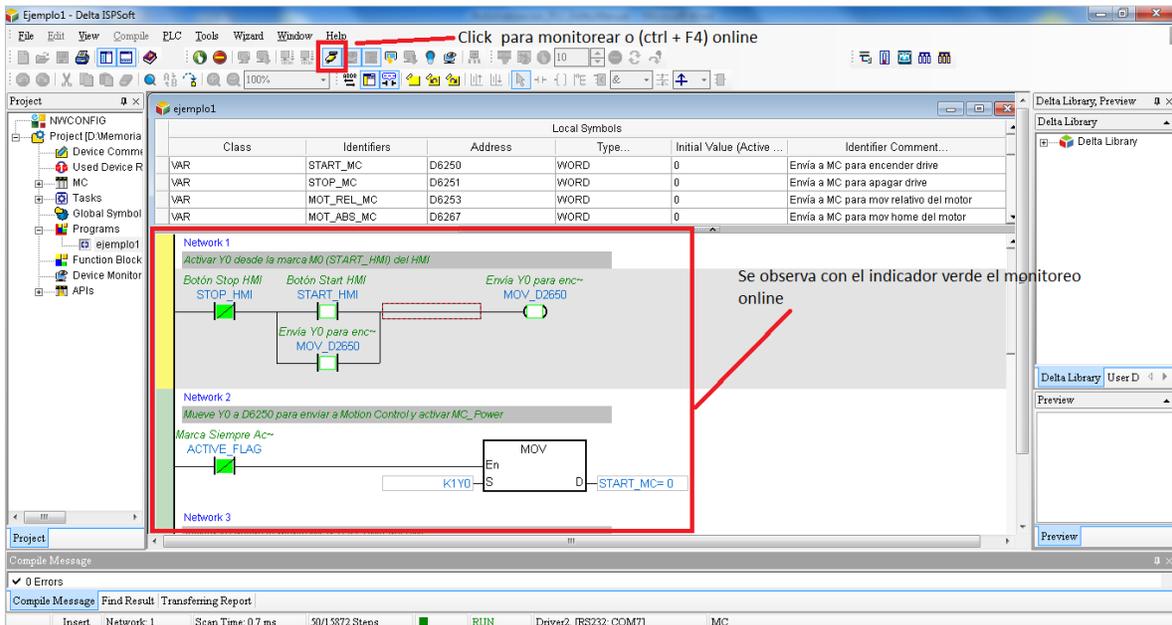


Figura 3.28 Modo online

3.5.2 Programación del Control de Movimiento Delta

Para programar el motion control Delta se debe realizar el proceso que se describe a continuación. Cabe mencionar que el programa “CANopen Builder” tiene integrada una ayuda de programación, la cual se visualiza oprimiendo la tecla F1; también se puede obtener información detallada en el manual “DVP_MC Bus-Type Multi-Axis Motion Controller Operating Manual” (Delta Electronics, Inc., 2012).

Primero se ejecuta el programa “CANopen Builder” versión 6.02; Builder después de la pantalla de arranque se abrirá la ventana principal del software de configuración de CANopen. A continuación, se describen los elementos de la pantalla principal mostrados en la figura 3.29.

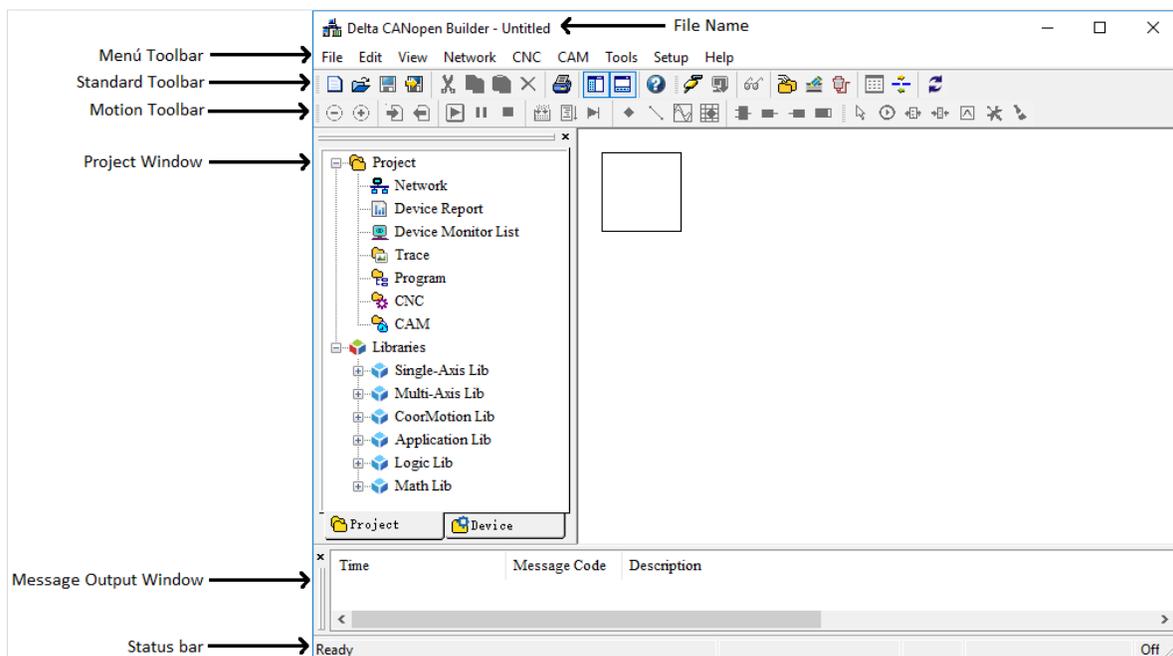


Figura 3.29 Ventana principal de CANopen Builder

- *File Name*: Es el nombre del archivo actual editado con el software CANopen Builder.

- *Menu Toolbar*: Hay nueve menús principales en la ventana del software CANopen Builder para que pueda seleccionar las funciones; son "File", "Edit", "View", "Network", "CNC", "CAM", "Tools", "Setup" y "Help".
- *Standard Toolbar*: Se elige el ícono de la función requerida directamente en lugar de seleccionar la función en la barra de herramientas del menú.
- *Motion Toolbar*: Integra las funciones que se ubican en la barra de herramientas del menú principal.
- *Project Window*: Integra la lista de dispositivos CANopen y lista de proyectos que tienen sistema EDS importado.
- *Message Output Window*: Muestra la información de salida de red CANopen (información de diagnóstico) y la información de compilación del programa.
- *Status bar*: Muestra el estado de la configuración en CANopen Builder (estado en línea o fuera de línea), la unidad número del módulo de detección, la configuración del puerto de conexión serial (velocidad de comunicación y formato) o la dirección IP del DVP10MC. La unidad número para el primer módulo de exploración cerca del lado izquierdo del PLC es 1 y la unidad número para el segundo módulo de exploración del lado izquierdo del PLC es 2. De esta forma, el otro número de unidad del módulo de exploración se definirá como 3,4,5,6,7,8, respectivamente.

3.5.2.1 Configuración de la red en CANopen Builder v6.02

Para hacer la comunicación por Ethernet del MC con el PLC y la computadora se requiere generar un segundo driver como se hizo con la comunicación del PLC. Primero se configura la conexión de red por Ethernet en la computadora (véase figura 3.30); En la ventana "Conexiones de red" se da clic derecho en "Ethernet" y se selecciona "Propiedades"; posteriormente se elige "Protocolo de Internet versión 4(TCP/IPv4)" y se da clic en "Propiedades"; por último se cambia la dirección IP a 192.168.1.12; ésta debe concordar con la dirección de red, mientras que en la dirección de host solo debe coincidir el primer dígito.

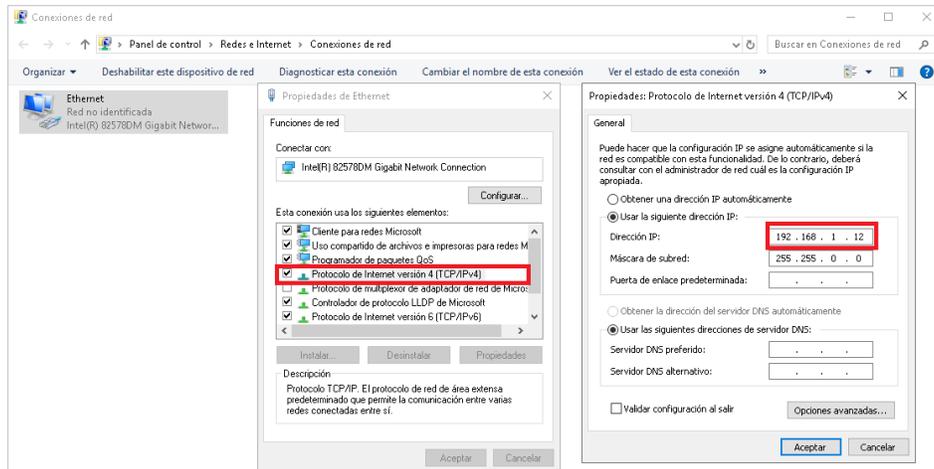


Figura 3.30 Configuración de IP Address

Se prosigue a configurar el segundo driver desde COMMGR; se nombra el driver 2 y se selecciona tipo Ethernet, después se da clic en “Search” para buscar la dirección IP de la conexión por Ethernet. Finalmente, ya detectada la dirección, se acepta la configuración, como se muestra en la figura 3.31.

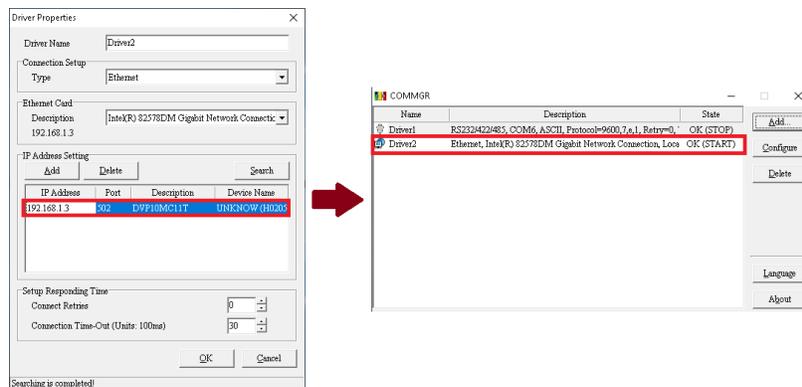


Figura 3.31 Configuración segundo drive “Ethernet”

Se configura el tipo de comunicación que se hará de la PC al dispositivo Motion Control que está integrado en el PLC. En el “Menu toolbar”, se da clic en la pestaña “Setup” → “Communication Setting” → “Ethernet Channel”; se abrirá una ventana nueva y se busca la transmisión por Ethernet dando clic en el botón “Broadcast Search”, como lo muestra la figura 3.32.

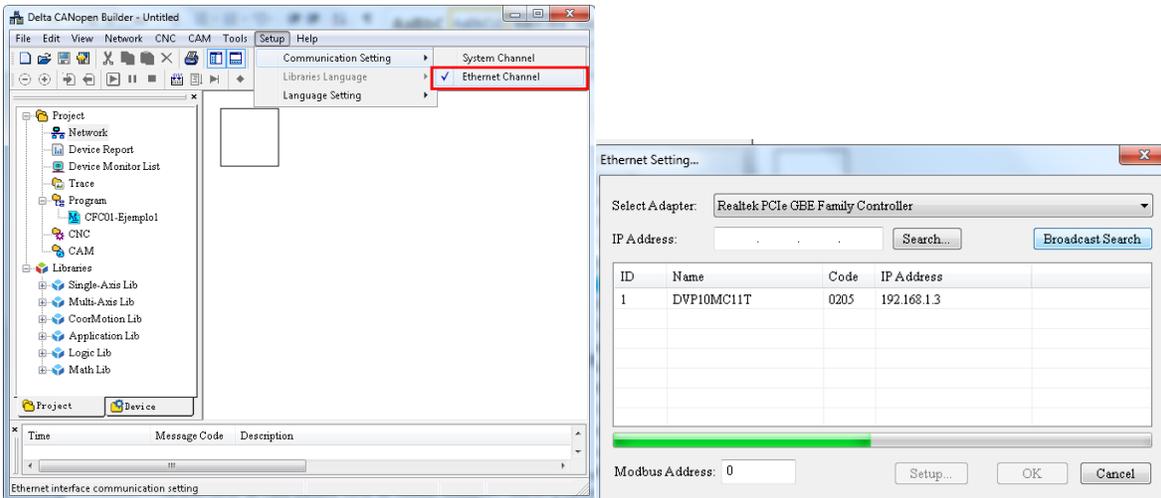


Figura 3.32 Configuración de la red Ethernet en CANopen Builder

Ya localizada la IP del Motion Control, en la opción IP Address se tecldea la dirección IP localizada. Se da clic en “Search” y se abrirá el recuadro “Input Password”. Se debe teclear la contraseña configurada (para este caso es “0000”) como se observa en la figura 3.33. Se finaliza la configuración.

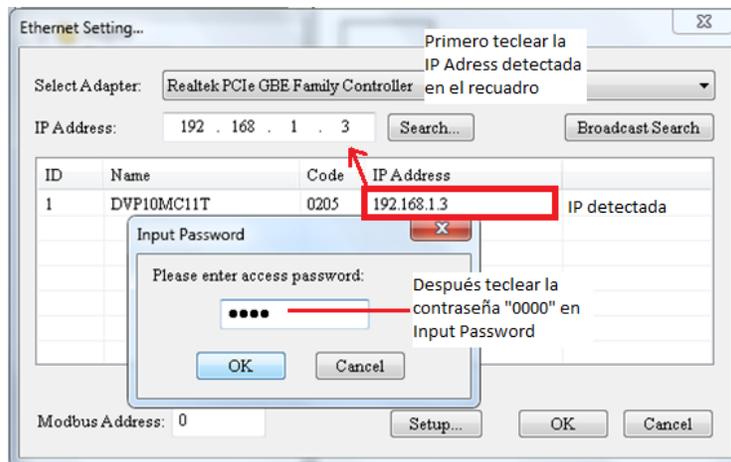


Figura 3.33 Ajuste de la comunicación por ethernet

Se abrirá una ventana llamada “Ethernet Config” con las características de la comunicación por ethernet y se selecciona “Ok” (véase figura 3.34). Se abre una

pequeña ventana “CANopen Builder”, donde finalmente se muestra que la configuración por ethernet se ha logrado satisfactoriamente.

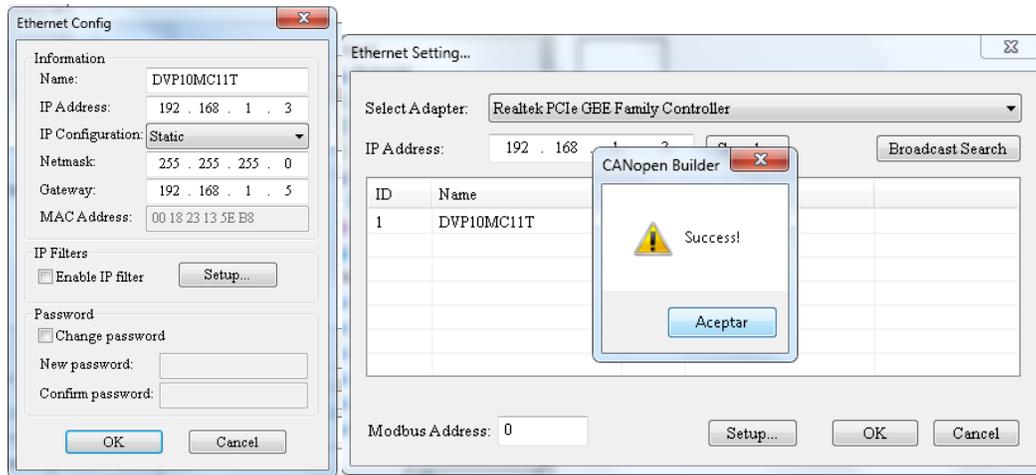


Figura 3.34 Configuración de ethernet lograda

Después de configurar la comunicación, se requieren detectar en la red CANopen los Servo drives y el DVP10MC11T que se encuentran comunicados físicamente entre ellos. En “Standard toolbar” se da clic en el ícono “Online mode” o se tecldea F10 (ver figura 3.35). Se cargarán los dispositivos en “Network” ubicado en “Project Window”.

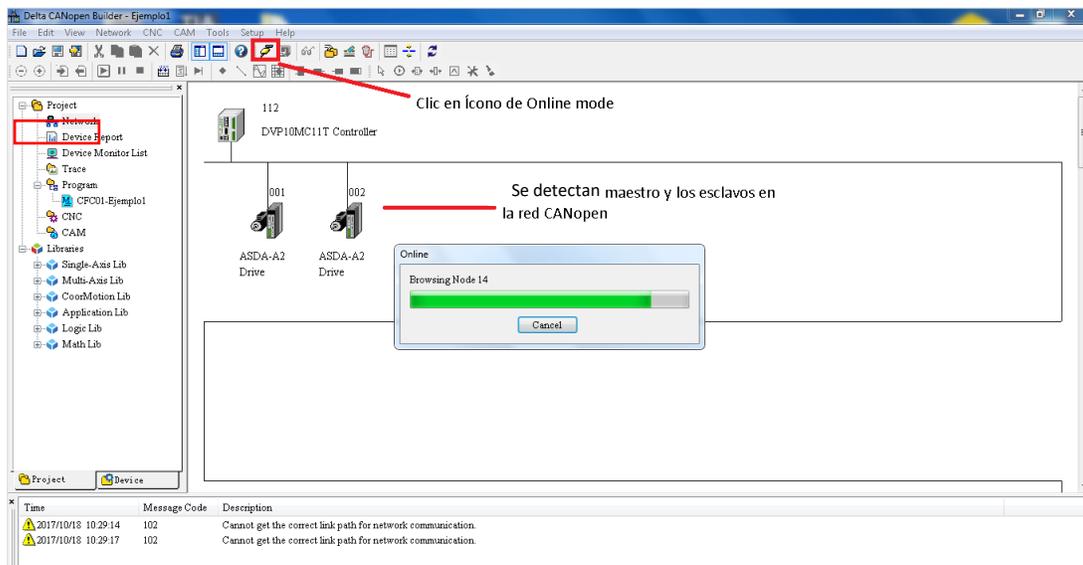


Figura 3.35 Detección del servo drive en la red CANopen

A continuación, se ajusta la lista de nodos con los Servo drives o los dispositivos que el usuario requiera en su proyecto; en este caso se usará el servo drive con dirección 1 y el servo drive con dirección 2. Para realizar este paso observe la figura 3.36. Se da doble clic en la imagen del DVP10MC11T Controller y se selecciona el dispositivo que se vaya a usar; se da clic en la flecha derecha para pasar los dispositivos a la Node List.

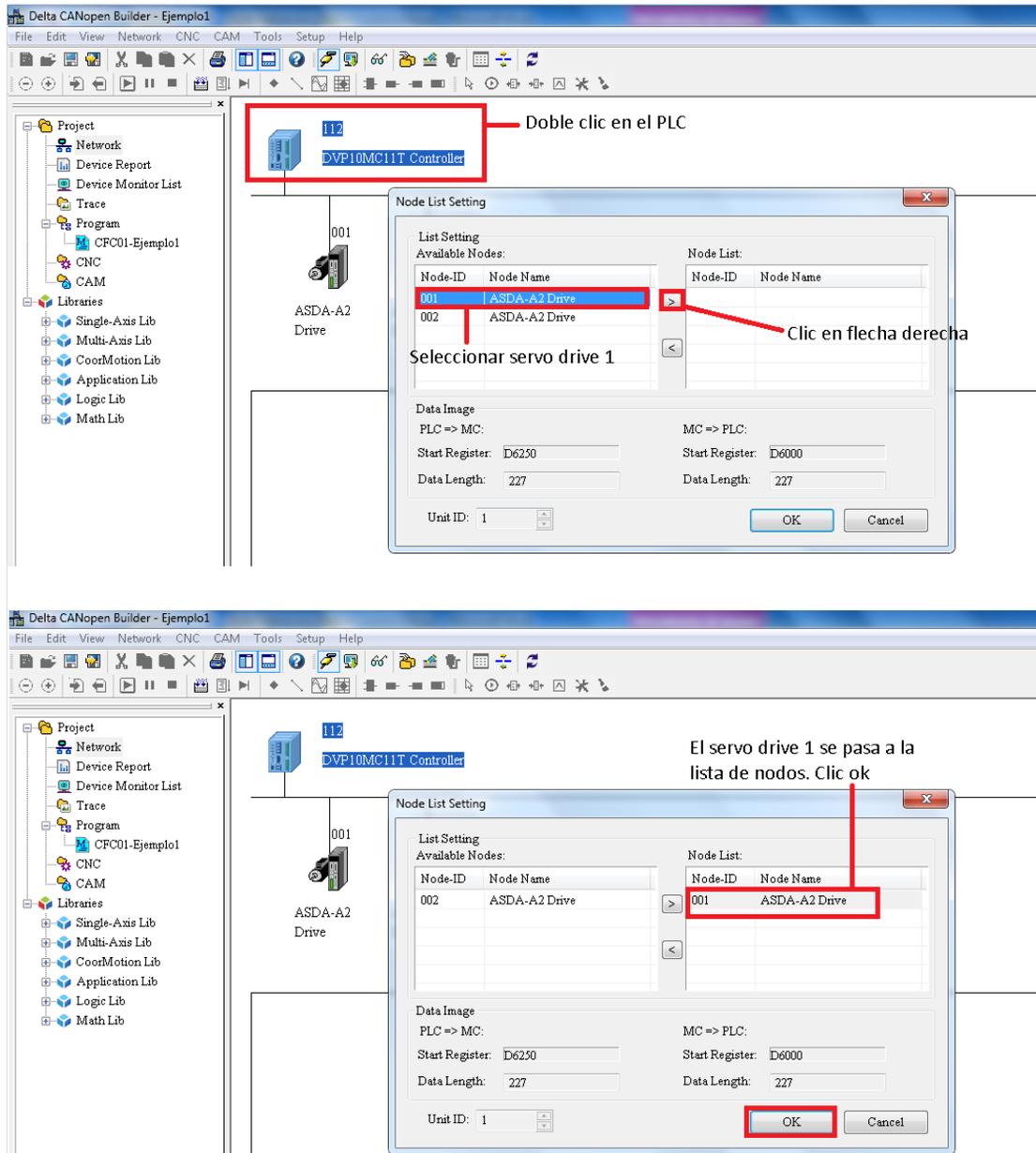


Figura 3.36 Configuración de "Node List" en el Controlador DVP10MC11T

Para terminar la configuración de la Network, se da doble clic en el dispositivo ASDA-A2 Drive y se configurarán los valores de acuerdo a los requerimientos de la aplicación del usuario. En este caso los valores a modificar son los que se observan en la figura 3.37.

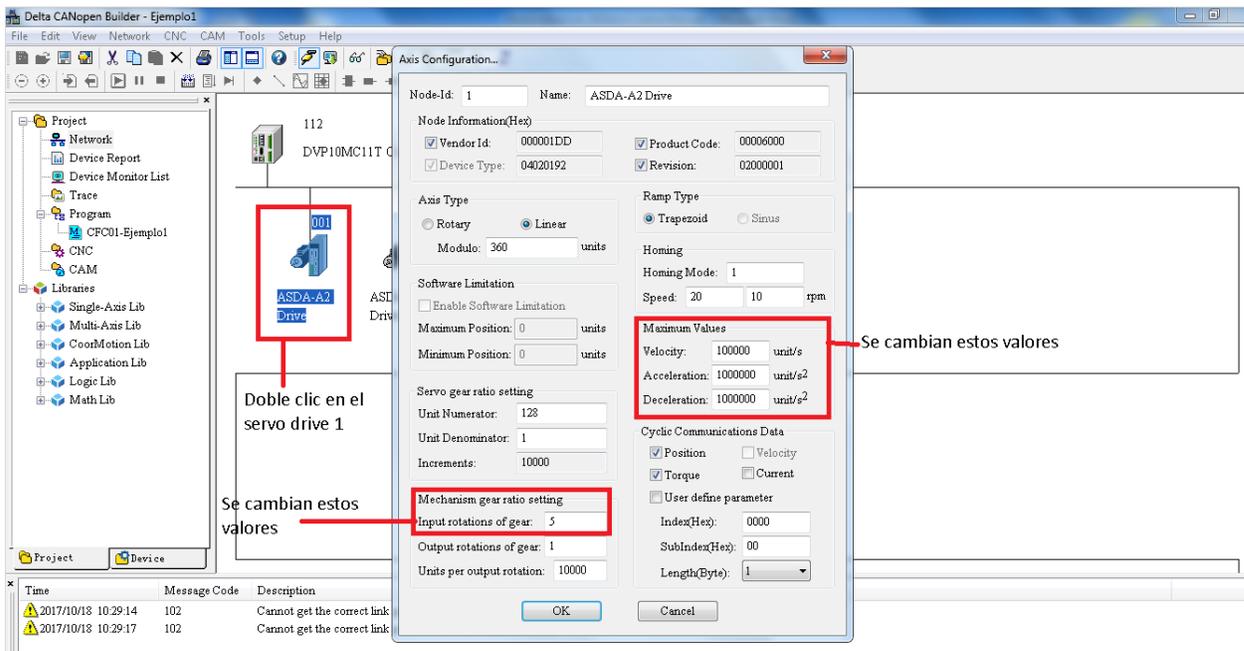


Figura 3.37 Configuración del servo drive con dirección 1

En “Mechanism gear ratio setting” se modifica solamente Input rotation of gear, cuyo valor por default era 1; para el proyecto se modifica a 5. Se cambiarán los valores de acuerdo al mecanismo físico del motor, en este caso el servomotor tiene una caja de engranes que genera esa relación de rotación 1:5.

En el caso “Maximum Values”, se modifican todos los parámetros de la siguiente manera porque la aplicación necesita manejar altas velocidades:

- Velocity: De 10'000 a 100'000 unit/s.
- Acceleration: De 10'000 a 1'000'000 unit/s².
- Deceleration: De 10'000 a 1'000'000 unit/s².

3.5.2.2 Generar un programa en CANopen Builder v6.02

En “Project Window” (parte izquierda de la ventana principal), se selecciona la pestaña “Project” → “Program” (ver figura 3.38), se da clic derecho y se selecciona la única opción disponible “New Program”.

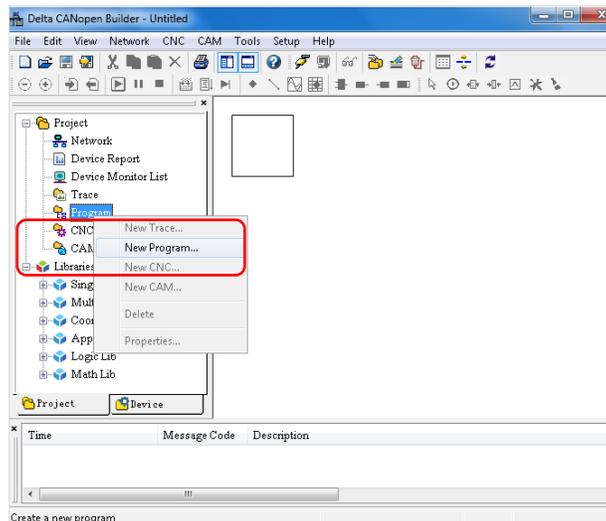


Figura 3.38 Generar nuevo programa en CANopen Builder v6.02

Se nombra el nuevo programa. Se presentan las opciones “Program Type” y “Task Type” y se configuran dependiendo del requerimiento del usuario (ver figura 3.39).

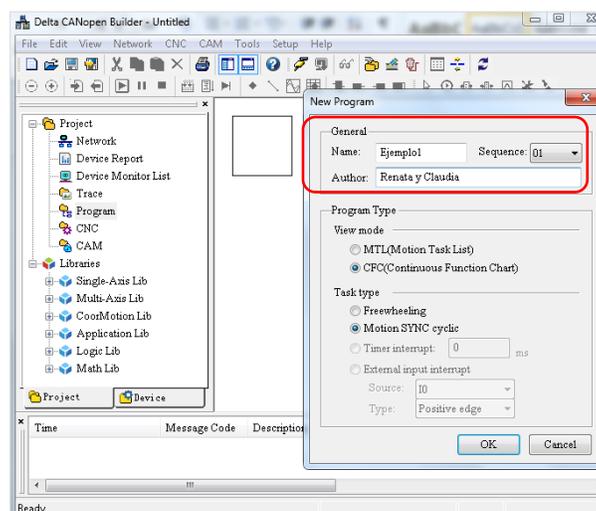


Figura 3.39 Configuración del nuevo Programa en MC

Para empezar a programar, en CANopen Builder, en “Project Window” se encuentran las instrucciones clasificadas en 6 librerías: “Single-Axis Lib”, “Multi-Axis Lib”, “CoorMotion Lib”, “Application Lib”, “Logic Lib” y “Math Lib”. En la figura 3.40 se observan las instrucciones correspondientes a las librerías ya mencionadas.

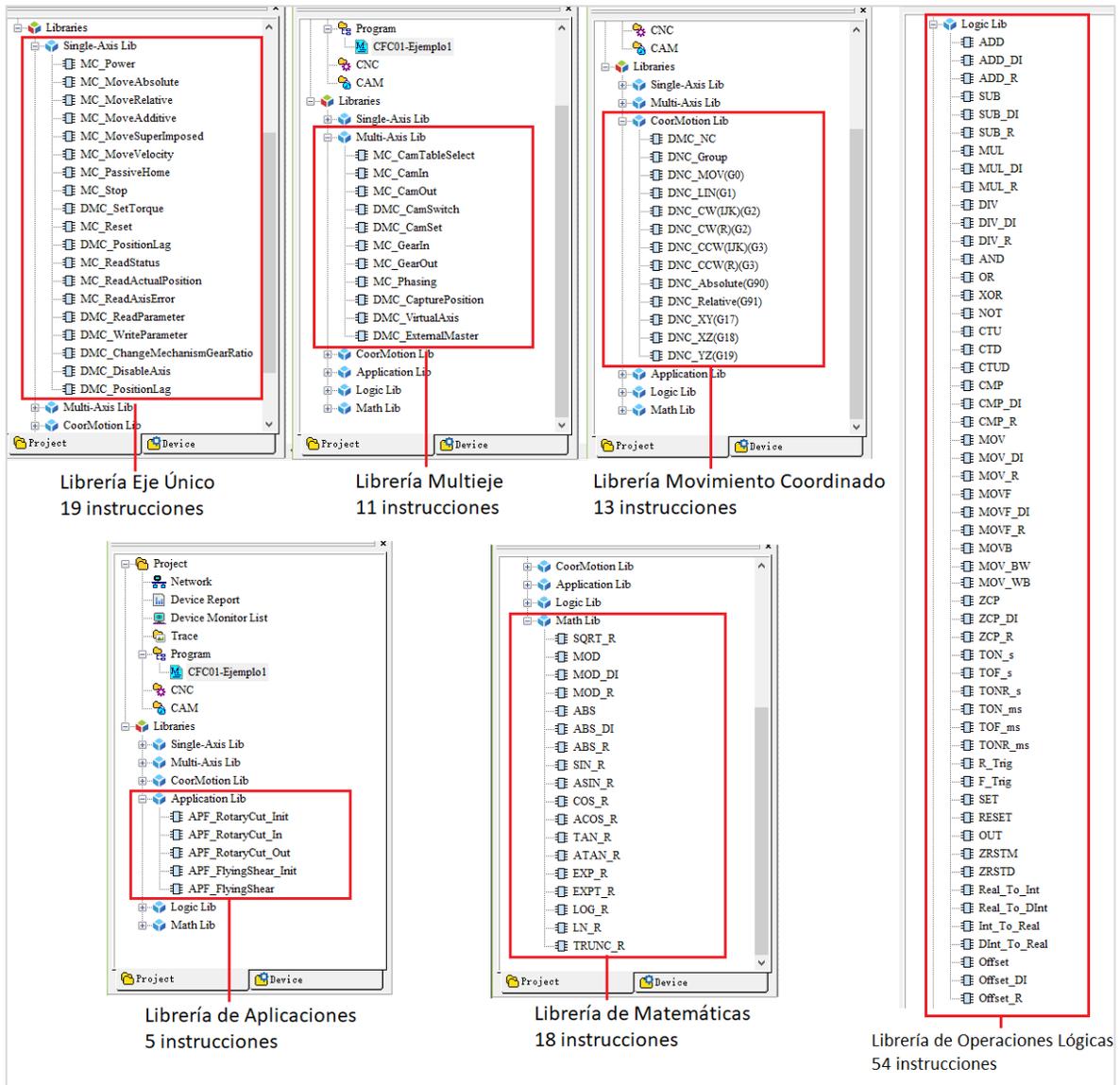


Figura 3.40 Instrucciones correspondientes a las librerías en Motion Control

Para anexar los bloques de programación, se dirige a alguna librería, después se da doble clic en la instrucción elegida, por ejemplo “MC_Power” y se arrastra el bloque al área de programación. Posteriormente, en las terminales del bloque se colocan las entradas (lado izquierdo) y las salidas (lado derecho), como lo muestra la figura 3.41. Además, es posible colocar comentarios por encima de los bloques para indicar su funcionamiento.

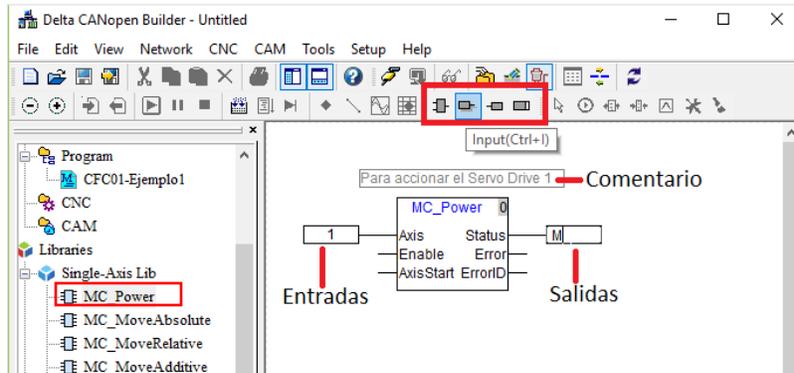


Figura 3.41 Entradas, salidas y comentarios en MC

Para el bloque MC_Power, el tipo de dato de las entradas y salidas se define en el manual o en la ayuda (ver figura 3.42). Se observa que en la terminal “Axis” se debe colocar la dirección del servo drive, en este caso corresponde con 1.

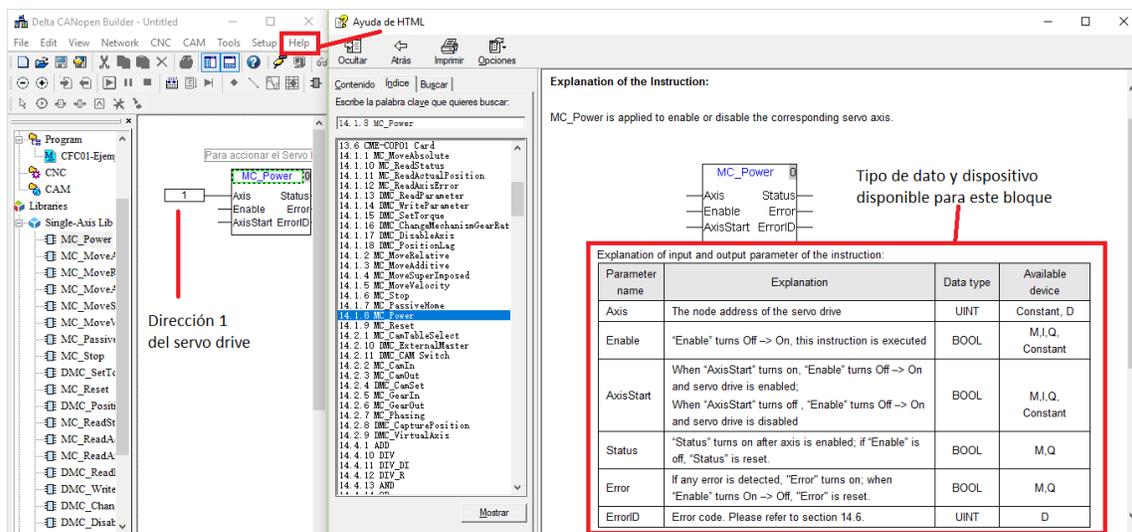


Figura 3.42 Tipos de datos para cada bloque en sección “Ayuda”

3.5.2.3 Compilación, carga y monitoreo en CANopen Builder

Ya finalizado el programa, en “Program Toolbar” (localizado en “Motion Toolbar”) se da clic en el ícono “Check all” o se tecldea F7 para compilar el programa (ver figura 3.43). En “Message Output Window” se visualiza si el programa presenta errores.

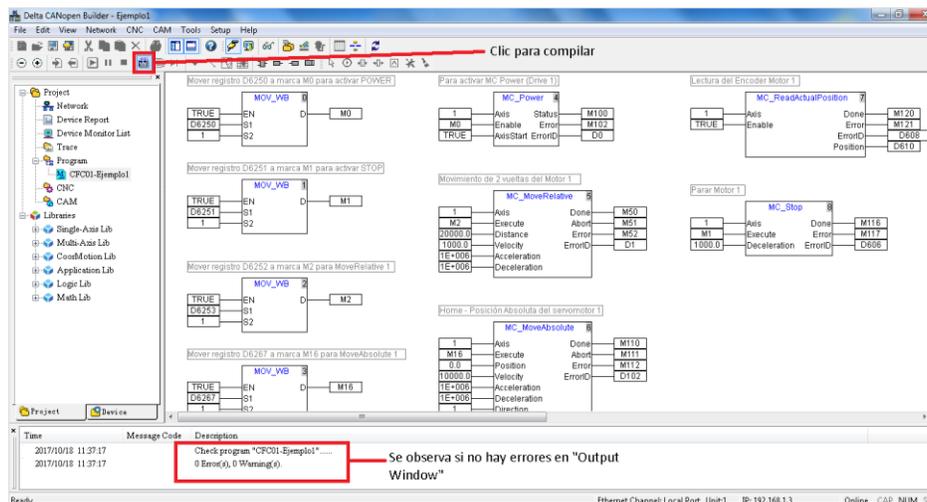


Figura 3.43 Compilación del programa en CANopen Builder

Posteriormente se da clic en el ícono “Download” ubicado en “Standard Toolbar” como se muestra en la figura 3.44; se cargará el programa al Motion Control.

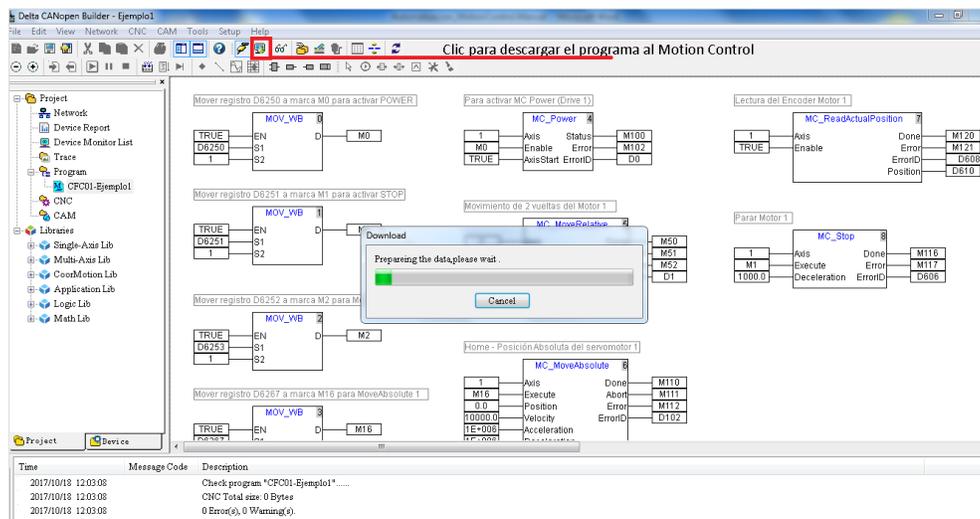


Figura 3.44 Descarga del programa al MC

Finalmente, ya cargado el programa al MC se da clic en “Programs Monitoring” (icono de anteojos) para monitorear el estado de cada instrucción del programa en tiempo real (ver figura 3.45). Así se validará si el programa se ejecuta como requiere el usuario.

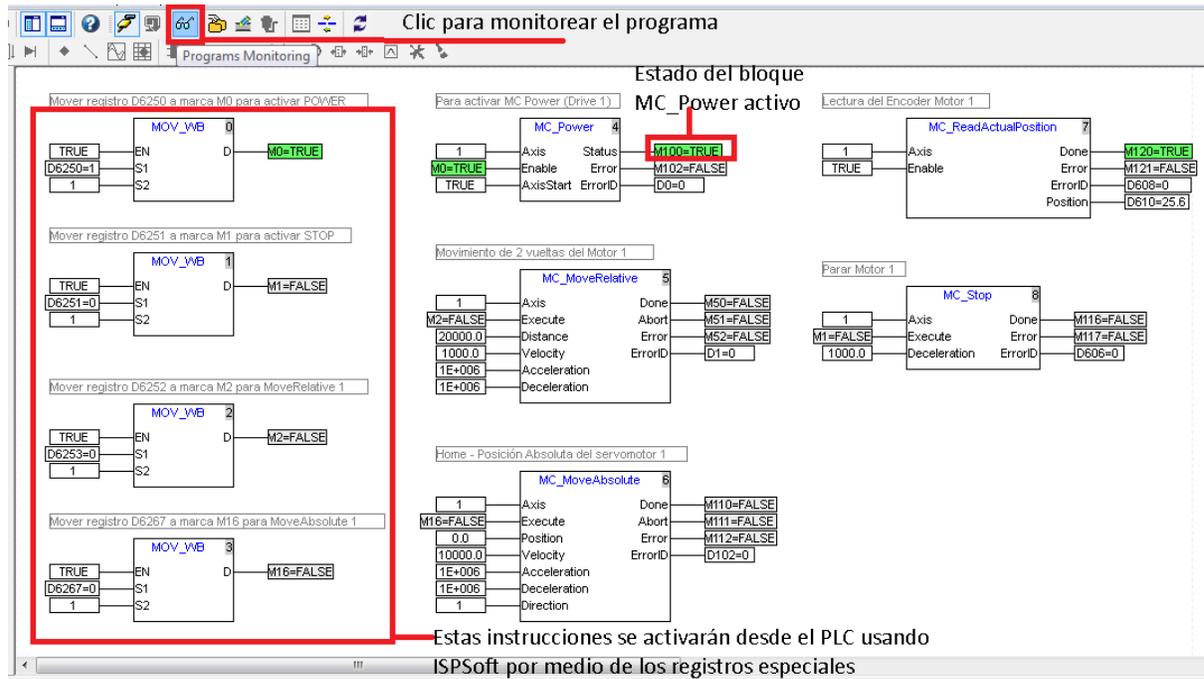


Figura 3.45 Monitoreo del programa y su ejecución en MC

En este caso es necesario que los dos programas estén cargados en sus respectivos dispositivos de control, porque el funcionamiento de uno dependerá del otro. Generalmente el programa principal se genera en ISPSOFT (PLC), mientras en CANopen Builder se realiza el programa de los comandos para el movimiento de los dispositivos conectados en red.

3.5.3 Programación de la Interfaz Usuario – Máquina Delta

El programa para la HMI del proyecto se realizó en el software “DOPSoft 2.00.07”. Las series de paneles de operación DOP-B, HMC, DOP-W y DOP-H de Delta se fabrican adoptando un software fácil de usar y hardware de alta velocidad para proporcionar una interfaz programable poderosa y estable. Además, DOPSoft admite más de cuarenta marcas de PLC, incluidos Allen Bradley, Festo, Moeller, Siemens, Toshiba, Mitsubishi, GE Fanuc, etc.

3.5.3.1 Generar un programa en DOPSoft v2.00.07

Primero se ejecuta DOPSoft v2.00.07 y después de visualizar la pantalla de arranque, se abrirá la ventana del software de configuración de Screens de DOPSoft. En la opción “Menu bar” se da clic en la pestaña “File” → “New” o se tecldea la combinación “Ctrl+N”. Se abrirá una ventana donde se elige la serie de panel a usar; también se observan las características de cada pantalla como su modelo, resolución, rotación y cantidad de colores. Para el proyecto se seleccionó la pantalla B07E411, tal como se muestra en la figura 3.46.

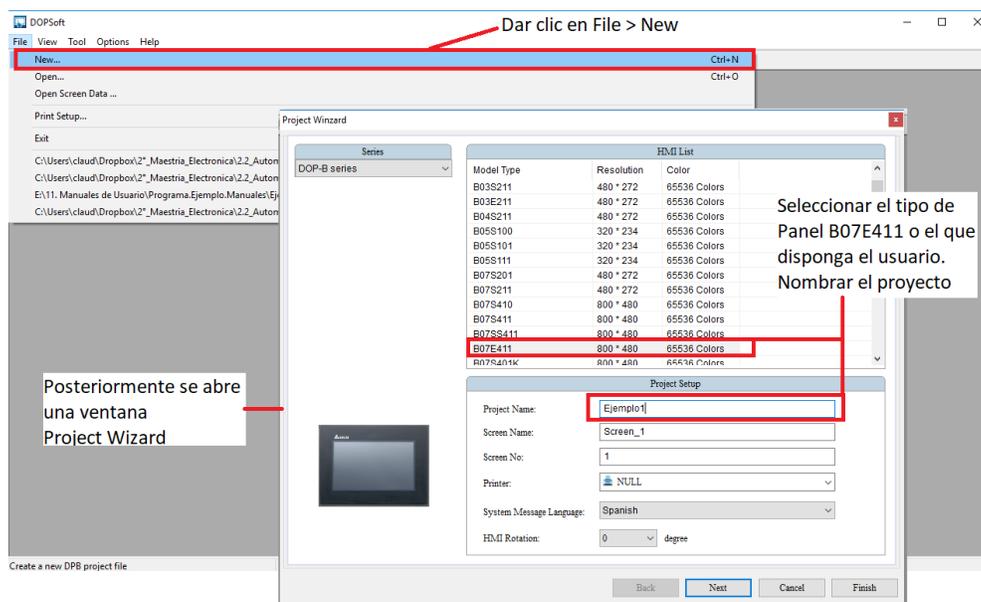


Figura 3.46 Creación y configuración de la pantalla HMI

Luego se selecciona el puerto COM donde se comunicará la HMI como se muestra en la figura 3.47; el fabricante (para el proyecto es la opción “Delta”) y en “series” seleccionar “Delta 10MC ASCII”. Finalmente se da clic en “Finish”.

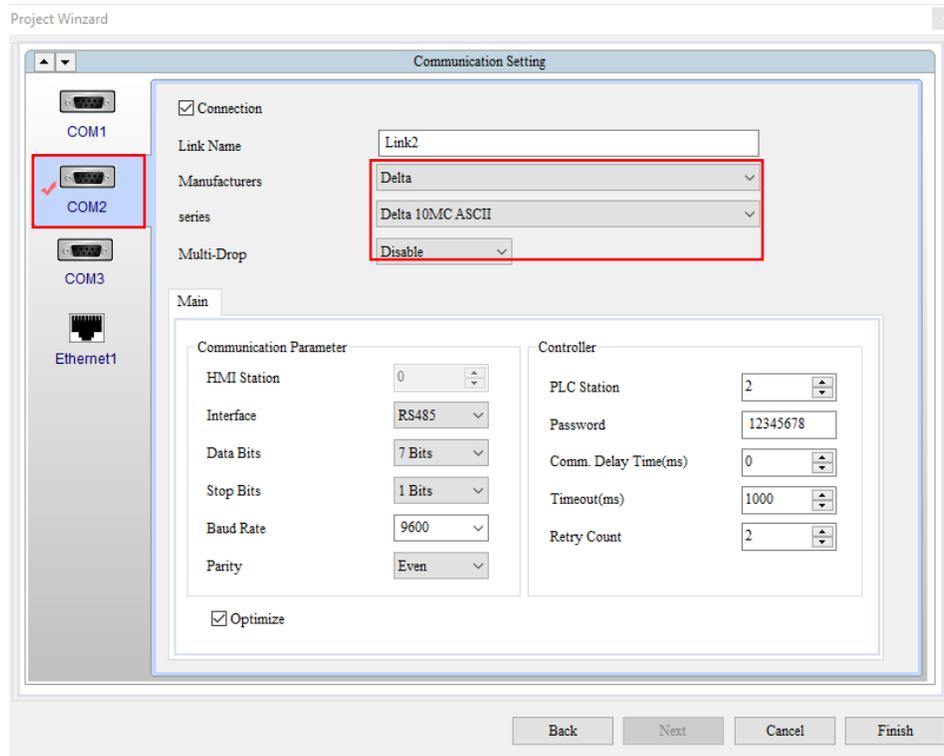


Figura 3.47 Comunicación por RJ45

Después de configurar la comunicación como se mostró con anterioridad, se procede a abrir la plantilla de la HMI en donde se insertarán los elementos necesarios para la operación del robot paralelo activado por cables. Los distintos elementos gráficos de programación se ubican en la opción “Element Toolbar” del menú principal.

A continuación, en la figura 3.48 se muestran los elementos existentes en DOPSoft. Éstos se clasifican en 17 grupos: Botones, medidores, gráficos de barras, de tubería, de pastel, indicadores, monitores, para movimiento, entradas, curvas, historial, alarmas, teclados, listas, marcos, slide y formas básicas. De acuerdo a los

requerimientos, el usuario seleccionará los elementos, los diseñará y dará eventos dependiendo de la aplicación.

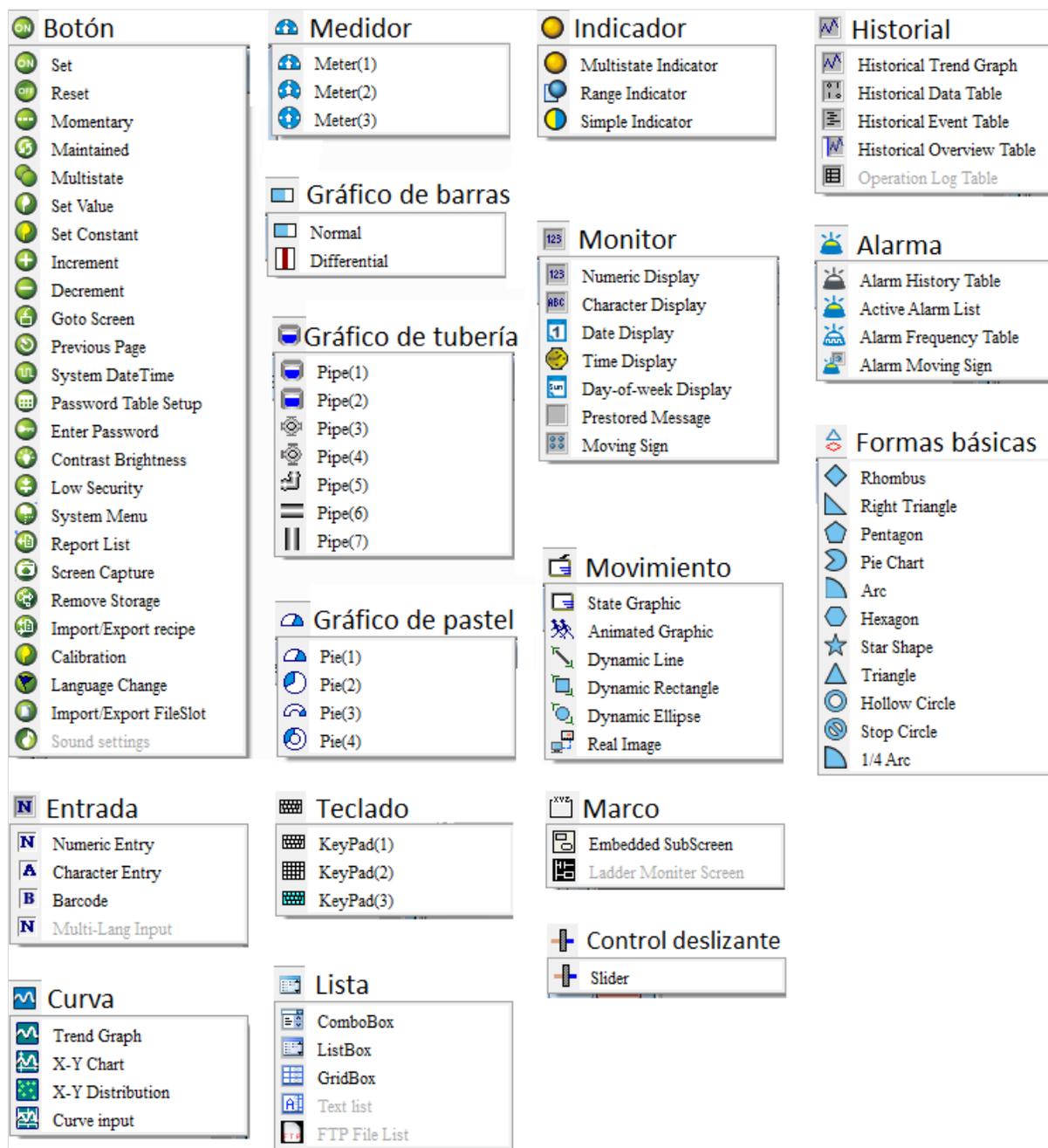


Figura 3.48 Opciones de "Element Toolbar"

Se agregan los elementos gráficos, por ejemplo, un botón; se prosigue configurando el botón en la pestaña "Picture" como lo muestra la figura 3.49. Después se selecciona en "Stretch Mode" la opción "Stretch All" y se selecciona "Transparent Color". En "Element description" se nombra al botón "Activar_MC_Momentary" tal como se muestra en la figura 3.50. Los botones tienen dos estados (on/off), por lo tanto, se debe realizar la configuración para ambos estados.

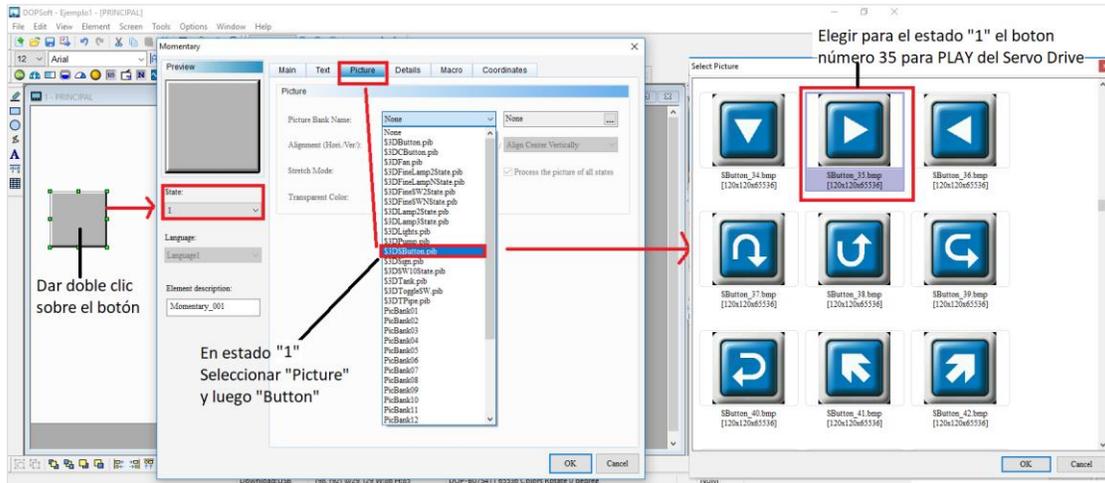


Figura 3.49 Configuración del botón en HMI

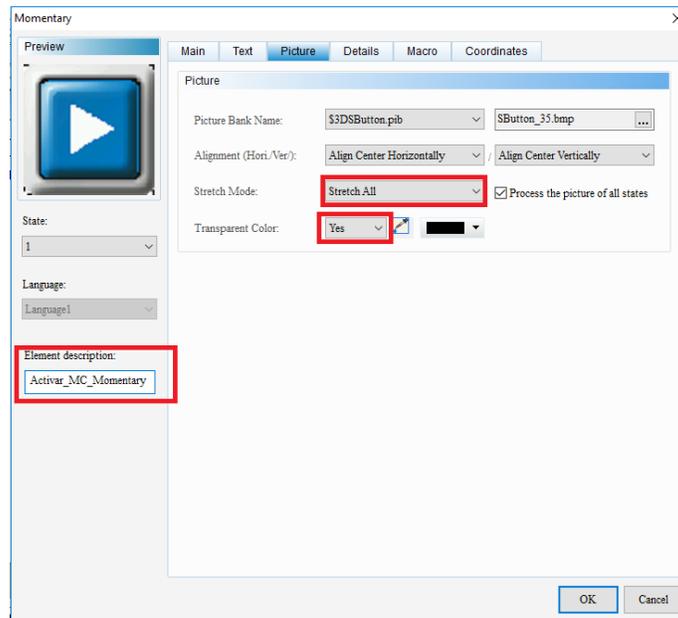


Figura 3.50 Finalizar la apariencia del botón

Posteriormente, se selecciona la pestaña “Main” y ahí (para ambos estados 0/1) se vincula al botón con una marca programada en el software ISPSOft del PLC; en este ejemplo se vincula el botón con la marca M0 como lo muestra la figura 3.51.

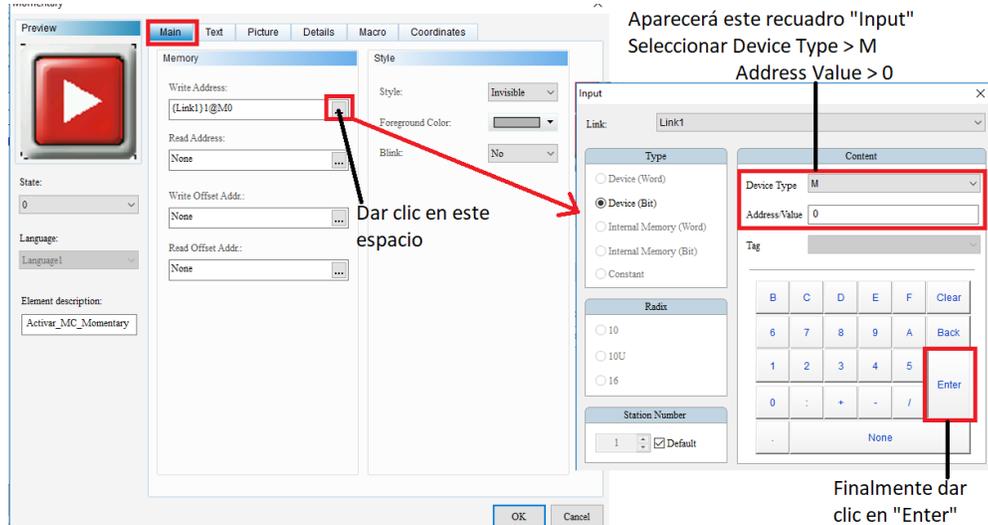


Figura 3.51 Dirección de escritura para el botón (marca M0)

Para terminar de editar el primer botón (ver figura 3.52), en la pestaña “Coordinates” se ajustan las medidas y las coordenadas de posición del botón.

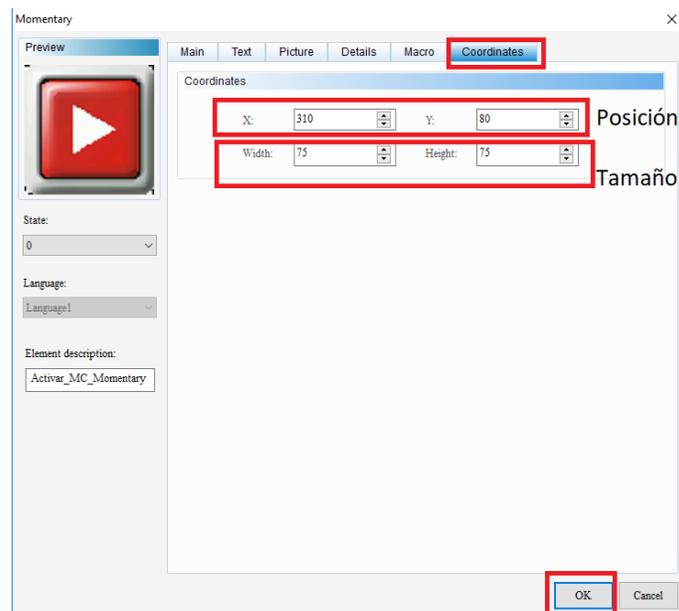


Figura 3.52 Coordenadas y tamaño del botón

Algunos ejemplos de otros elementos que se agregaron en la pantalla de la HMI para este proyecto son presentados tabla 3.5 junto con sus características.

Tabla 3.5 Botones y sus características para la HMI

<i>Nombre del Botón</i>	<i>Estado 0</i>	<i>Estado 1</i>	<i>Marca asignada</i>	<i>Coordenadas</i>	<i>Tamaño</i>
<i>Desactivar_MC_Momentary</i>	 \$3DSButton SButton_78	 \$3DSButton SButton_32	M2	X: 420 Y: 80	75 * 75
<i>Home_Momentary</i>	 \$3DSButton SButton_11	 \$3DSButton SButton_02	M3	X: 490 Y: 240	90 * 90
<i>Relative_Momentary</i>	 \$3DSButton BUTTON_05	 \$3DSButton BUTTON_06	M4	X: 240 Y: 240	90 * 90

Ya que están distribuidos los botones en sus lugares correspondientes, se requiere agregar texto para indicar el funcionamiento de cada botón, por lo tanto, se debe seleccionar el ícono de texto “A” que se ubica en la barra “Drawing” (parte izquierda de la pantalla). Se colocará el texto en la pantalla y se dará doble clic para abrir una ventana “Text”. Se escribirá en el recuadro blanco el texto a visualizar, así como se configurará el formato del texto (tipo de letra, color, alineación, etc.); ver figura 3.53.

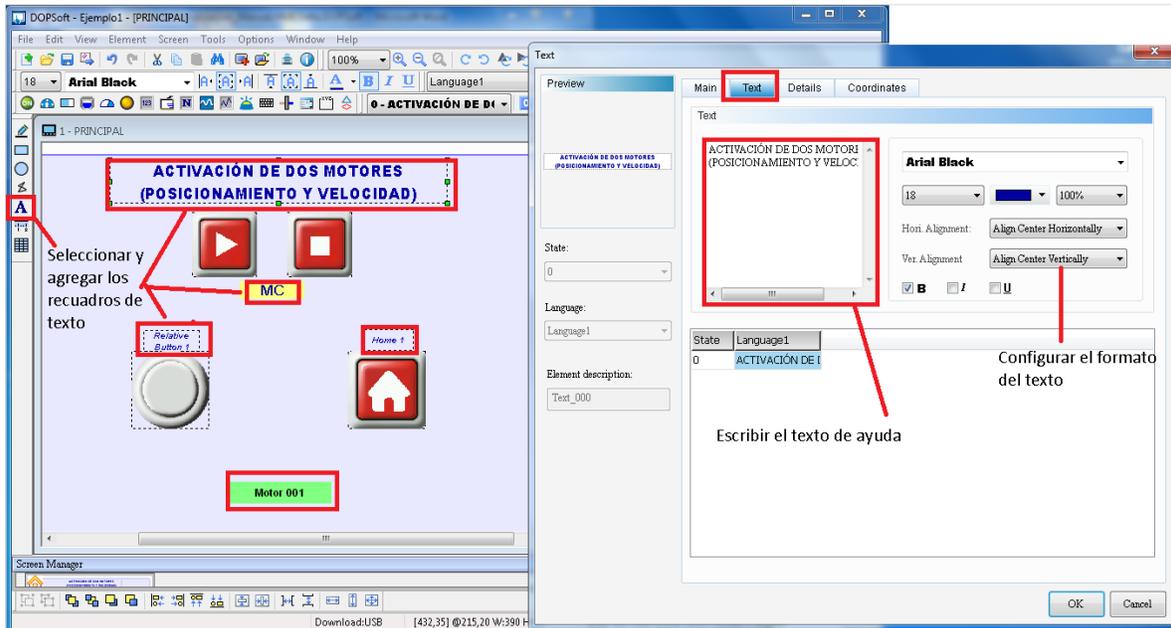


Figura 3.53 Agregar texto en la pantalla HMI

Adicionalmente, si se requiere agregar elementos para introducir datos (caracteres o datos numéricos), se debe dar clic en el símbolo “N” ubicado en la sección “Element toolbar”, como lo muestra la figura 3.54. Se despliega una lista de elementos y se debe seleccionar “Numeric Entry”.

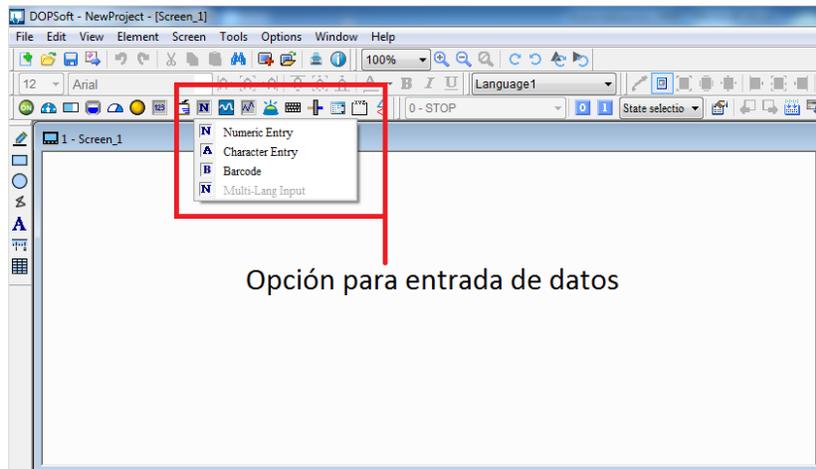


Figura 3.54 Elemento para introducir datos

Después se da doble clic en el elemento de tipo “Entrada de Dato” y se abrirá una ventana. En la pestaña “Main”, se seleccionará el tipo de dato a introducir en la caja, así como otras características tales como valor máximo y mínimo permitidos, dirección y cantidad de dígitos enteros y fraccionarios. En la figura 3.55 se observan todos los elementos a modificar por el usuario según los requerimientos.

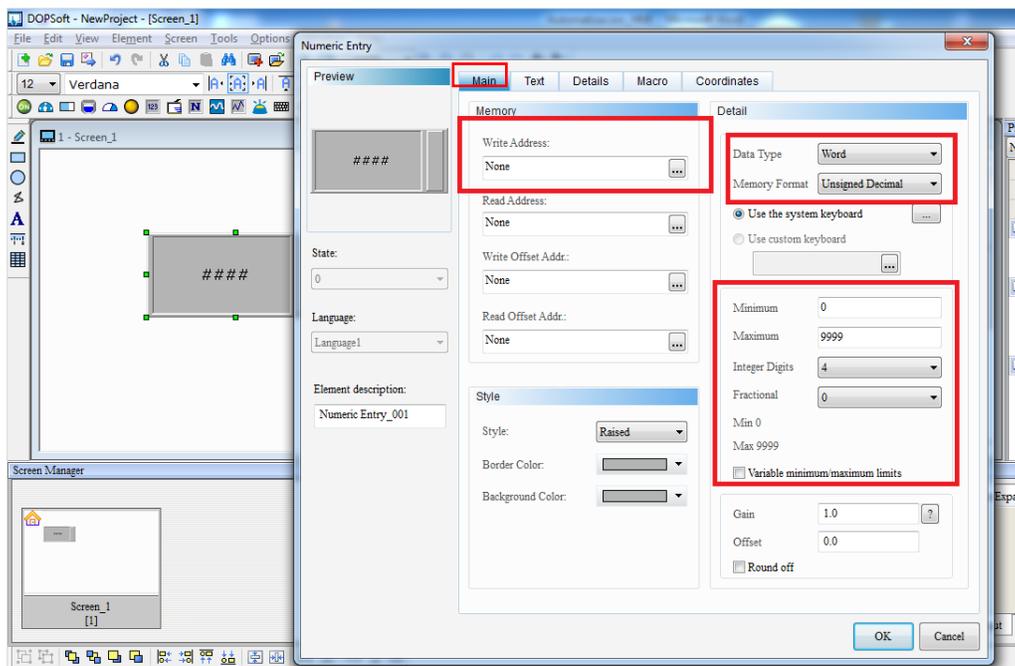


Figura 3.55 Configuración de Numeric Entry

3.5.3.2 Compilación, carga y monitoreo en DOPSoft

Para la realización de la compilación de la HMI (detectar errores de programa), se da clic en “Compile” ubicado en “Layout toolbar”. En el recuadro “Output” ubicado abajo de la ventana, se observará si existió algún error de compilación (ver figura 3.56).

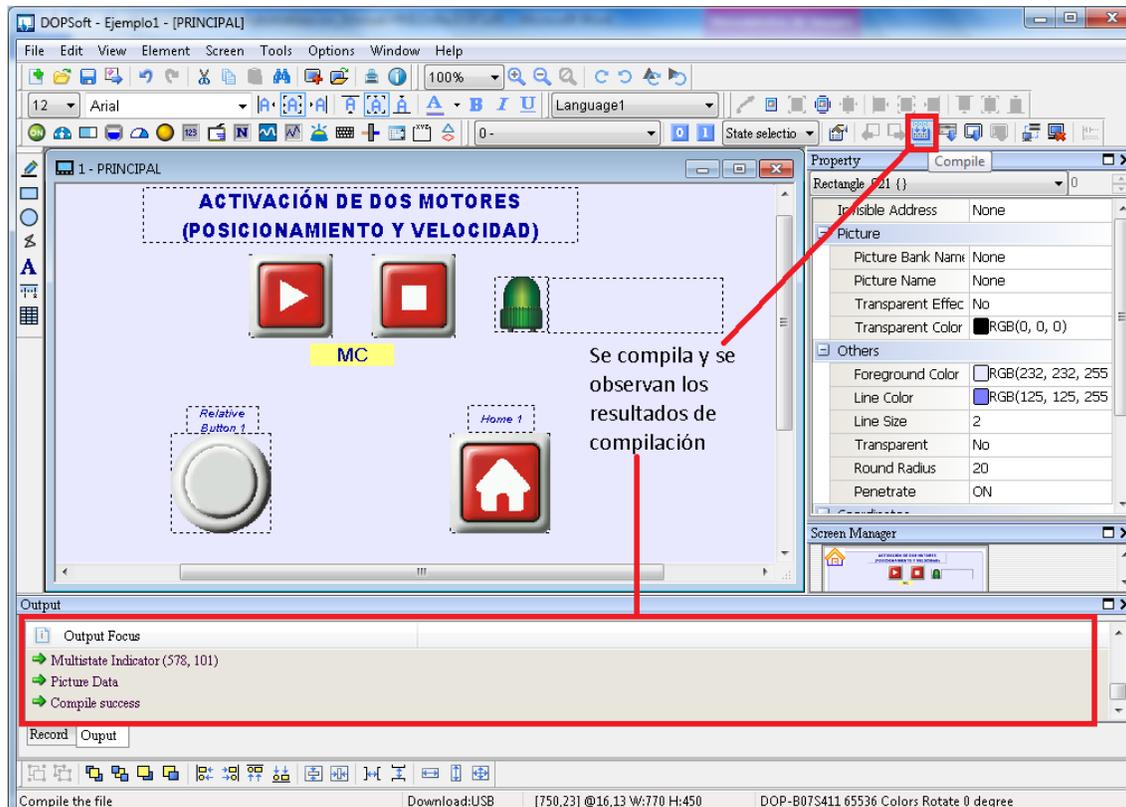


Figura 3.56 Compilación de la HMI terminada

Para simular en línea se da clic en “On-line Simulation” como lo muestra la figura 3.57. Es importante mencionar que, si solamente se tiene un cable de comunicación RJ45 de la PC al PLC, no se podrá monitorear en línea a la vez el programa del PLC (ISPSOft) y la HMI (DOPSoft). Por lo tanto, solamente se debe monitorear un programa en tiempo real.

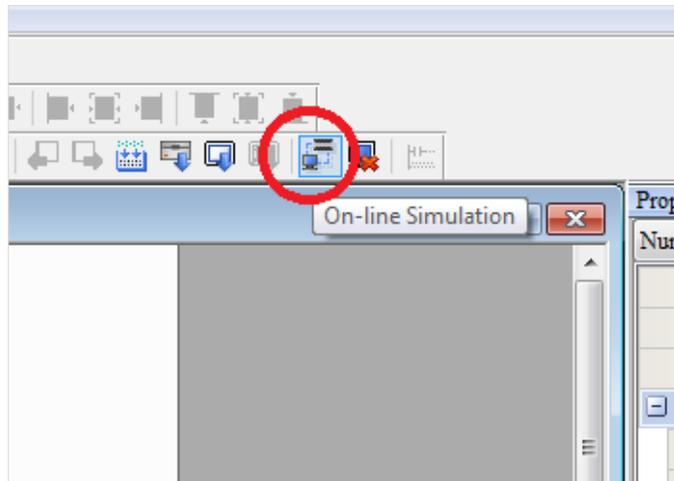


Figura 3.57 HMI on-line Simulation

Ya lograda la comunicación por el puerto RJ45, se abrirá otra ventana con la simulación. El usuario podrá manipular los distintos elementos programados, introducir datos, visualizar texto e indicadores, entre otras funciones. En el ejemplo de la figura 3.58 se observa un cambio de la imagen del botón cuando éste es presionado y la simulación del led al ser activado.

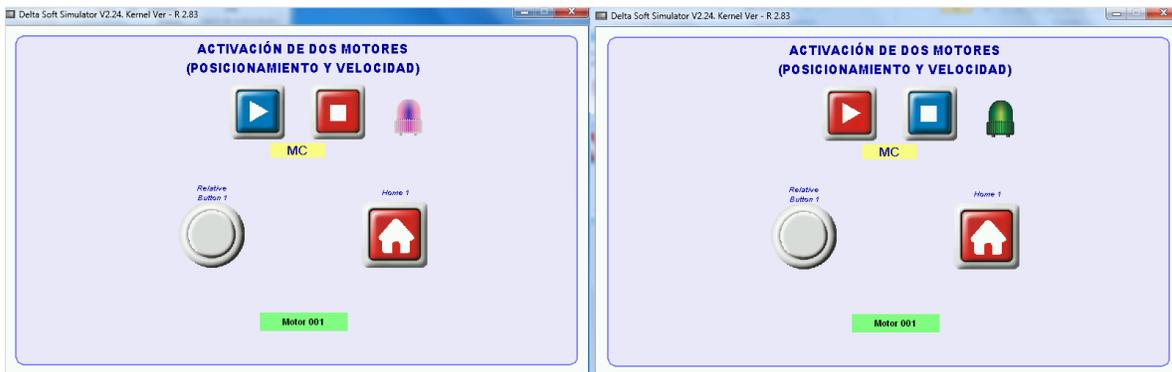


Figura 3.58 HMI para completar el proyecto de MC y PLC terminado

3.6 PARAMETRIZACIÓN DEL SERVO DRIVE

Para configurar los parámetros del servo drive se requirió del software ASDA Soft versión 5.3.3.0 distribuido de manera gratuita por la marca Delta para realizar las modificaciones en el control del servo. Para lograr modificar los parámetros es necesario considerar los símbolos que aparecen en la figura 3.59, ya que para modificar algunos parámetros se requiere apagar el servo drive y así harán efecto los cambios.

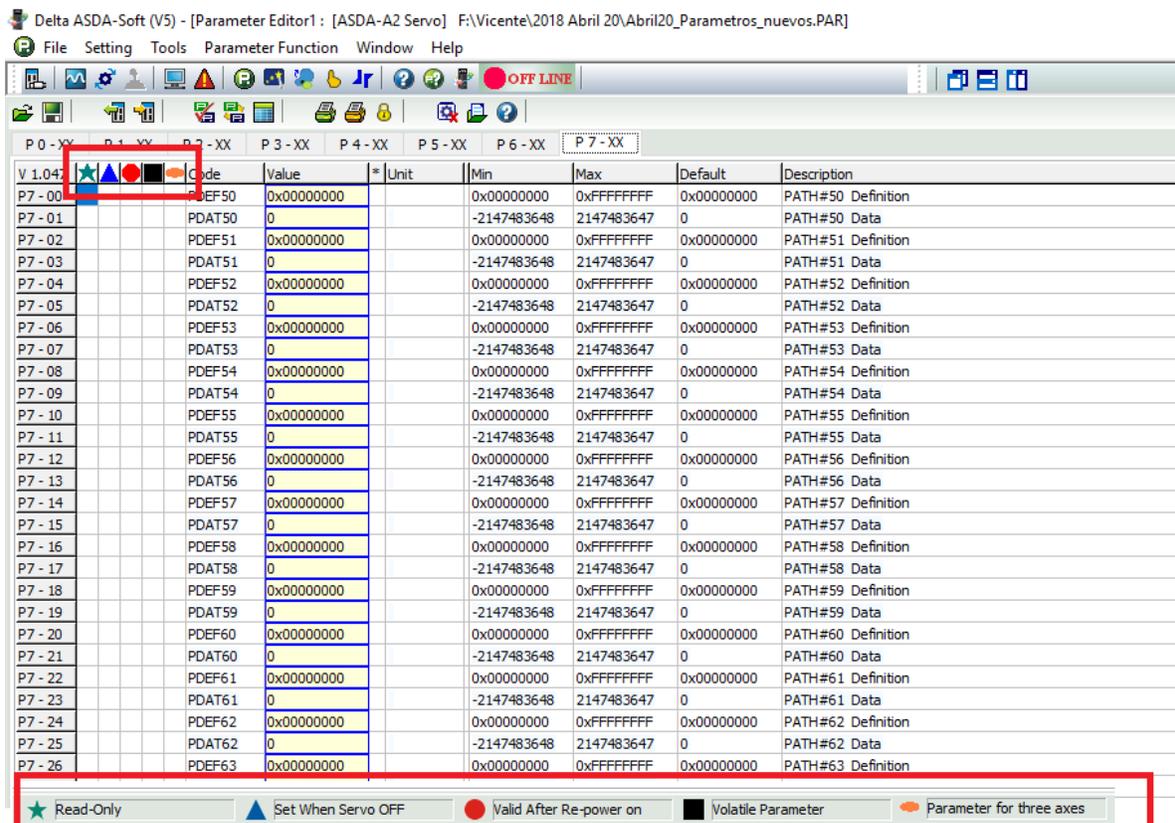


Figura 3.59 Tipos de parámetros en el servo drive

Para hacer alguna de estas modificaciones, primero se debe entrar al programa, ASDA Soft versión 5.3.3.0 y se abrirá una pequeña ventana donde se hará clic en “Start auto detect” (véase figura 3.60). De esta forma el programa detectará si puede usarse el servo drive.

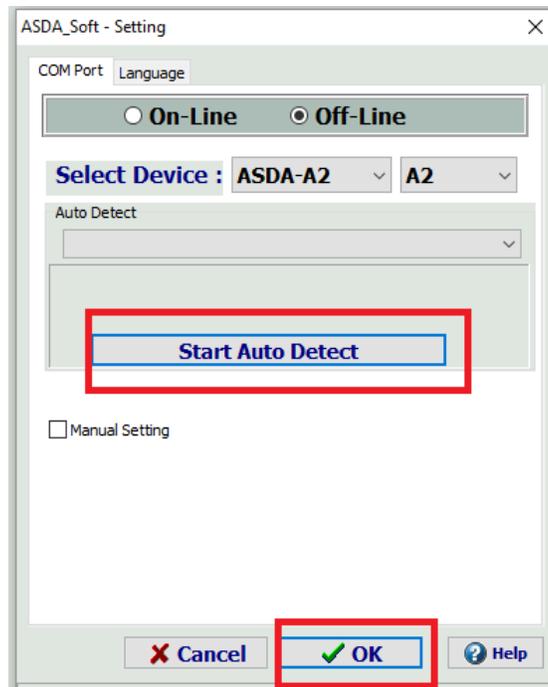


Figura 3.60 Conexión de servo drive al software

Posteriormente, el cuadro rojo que se muestra en la figura 3.61a, cuya etiqueta dice “OFF LINE” cambiará a color verde y se seleccionará la opción “Parámetro editor” (se visualiza con la letra P). Después se da clic al ícono mostrado en la figura 3.61b, que lleva por nombre “Read Parameters”; éste tomará la lectura de los parámetros actuales del servo drive.

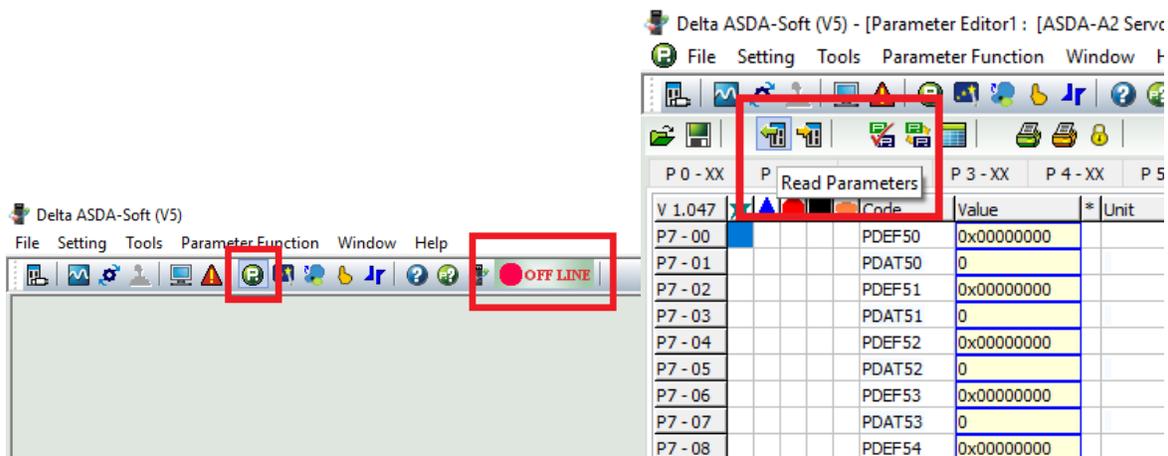


Figura 3.61 a) Entorno de ASDA Soft y b) Conexión del servo drive al software

En la figura 3.62 se observa la primera columna correspondiente al valor del parámetro. La columna que corresponde a “Value” nos indica el valor actual del parámetro. Este valor se puede modificar haciendo doble clic en la casilla correspondiente a “Value” y escribiendo el nuevo valor. Dicho valor debe estar dentro de los rangos permitidos que indican las siguientes columnas “Min” y “Max”.

La columna que lleva por nombre “Default” indica el valor que trae de fábrica cada parámetro del servodrivo y en la última columna se presenta la descripción de la función de cada parámetro específico.

V. 1.047	P 0 - XX	P 1 - XX	P 2 - XX	P 3 - XX	P 4 - XX	P 5 - XX	P 6 - XX	P 7 - XX	Description
		Code	Value	Unit	Min	Max	Default		
P1-01		CTL	0x0002		0x0000	0x111F	0x0000		Input Setting of Control Mode and Control Command
P1-02		PSTL	0x0000		0x0000	0x0011	0x0000		Speed and Torque Limit Setting
P1-03		AOUT	0x0000		0x0000	0x0013	0x0000		Polarity Setting of Encoder Pulse Output
P1-04		MON1	100	%	0	100	100		MON1 Analog Monitor Output Proportion
P1-05		MON2	100	%	0	100	100		MON2 Analog Monitor Output Proportion
P1-06		SFLT	0	ms	0	1000	0		Analog Speed Command (Low-pass Filter)
P1-07		TFLT	0	ms	0	1000	0		Analog Torque Command (Low-pass Filter)
P1-08		PFLT	0	10ms	0	1000	0		Smooth Constant of Position Command (Low-pass Filter)
P1-09		SP1	500	0.1r/min	-60000	60000	1000		Internal Speed Command 1 / Internal Speed Limit 1
P1-10		SP2	2000	0.1r/min	-60000	60000	2000		Internal Speed Command 2 / Internal Speed Limit 2
P1-11		SP3	3000	0.1r/min	-60000	60000	3000		Internal Speed Command 3 / Internal Speed Limit 3
P1-12		TQ1	100	%	-300	300	100		Internal Torque Command 1 / Internal Torque Limit 1
P1-13		TQ2	100	%	-300	300	100		Internal Torque Command 2 / Internal Torque Limit 2

Figura 3.62 Columnas de parámetros de ASDA Soft

En la tabla 3.6 se presenta un resumen de los parámetros del servo drive.

Tabla 3.6 Parámetros modificados para el proyecto “Control de un CDPDR”

Parámetro	Valor predeterminado y actual	Descripción
P1-01	0x000B	Configuración de entrada del modo de control y control de comandos: Control del servo drive en modo CANopen
P1-09	1000	Comando de velocidad interna

Continuación de tabla 3.6 Parámetros modificados para el proyecto “Control de un CDPR”

<i>Parámetro</i>	<i>Valor predeterminado y actual</i>	<i>Descripción</i>
<i>P2-15</i>	122	Ajuste el interruptor de límite de reversa abierto normalmente
<i>P2-16</i>	123	Ajuste el interruptor de límite de avance abierto normalmente
<i>P2-17</i>	121	Ajuste el interruptor de parada de emergencia del motor abierto normalmente
<i>P2-27</i>	0x0000	Ganancia de swicheo y selección de swicheo
<i>P3-00</i>	1 y 2 para cada servomotor	Ajustar la dirección de nodo para cada servo drive
<i>P3-01</i>	403	Ajustar la velocidad de comunicación del servo drive a 1Mbit/s en CANopen

Los parámetros mostrados en la tabla anterior deben ser configurados como se muestra, ya que de hacer lo contrario, no se podrá utilizar la red por CANopen y no habrá comunicación entre los distintos dispositivos de control del robot.

3.6.1 Análisis del Control de posición del Servo drive ASDA A2-M

En esta sección, se aborda el análisis del control de posición de lazo cerrado interno que contiene el servo drive ASDA A2-M. De igual forma se presentan los parámetros que se ajustan para el control de posición.

De acuerdo a la figura 3.63 el tipo de control que tiene integrado el servo drive ASDA A2-M es control Feedback con Feedforward (o anticipativo).

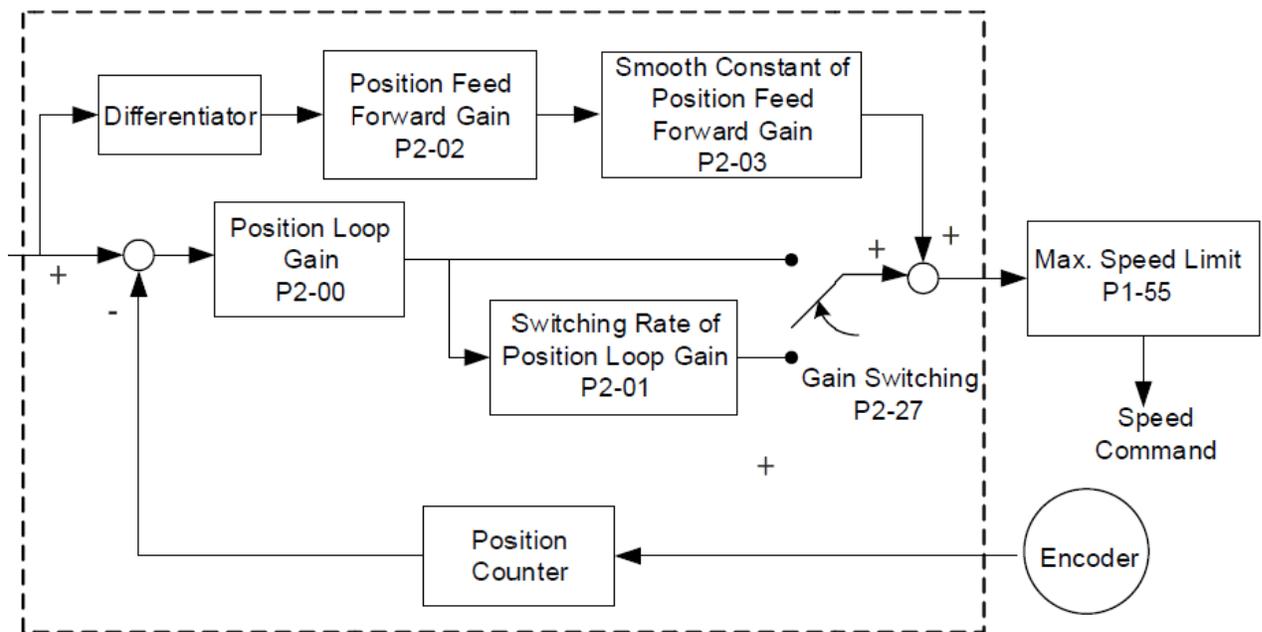


Figura 3. 63 Diagrama a bloques del Control de posición Feedforward y Feedback del Servo drive (Delta Electronics, Inc., 2014)

El principal motivo por el cual existe la configuración anticipativa en un sistema de control, es para rechazar las perturbaciones medibles dentro del lazo de control y ayudar al controlador principal a regular la variable de interés. En la tabla 3.7 se presenta la comparación del control anticipativo contra el control de realimentación. La mejor estrategia para el control de un sistema, es usar ambos controles conjuntamente, ya que la parte feedforward corrige gran parte del error y feedback termina de corregir el error faltante.

Tabla 3. 7 Comparación de control feedback contra feedforward

<i>Realimentación</i>	<i>Anticipativo</i>
Lazo cerrado	Lazo abierto
Lento	Rápido
Robusto	No es robusto
Afecta la estabilidad	No afecta la estabilidad
Correctivo	Preventivo

En el caso particular del control del servo drive se aprecia a los parámetros P2-02 y P2-03 que modifican este lazo feedforward, además, se observa que se presenta una parte deriva antecediendo esos parámetros.

En el software “ASDA Soft v.5” para la configuración del servo drive, se monitorean los parámetros P2-02 y P2-03 como se muestra en la figura 3.64.

Ganancia de Prealimentación de Posición y Constante Suave de Ganancia de Prealimentación

P2 - 02		PFG	50	%	0	100	50	Position Feed Forward Gain
P2 - 03		PFF	5	ms	2	100	5	Smooth Constant of Position Feed Forward Gain

Parameter Name	Unit	Minimum ~ Maximum	Default	16/32 bit
P2 - 02	%	0 ~ 100	50	16bit
Position Feed Forward Gain				
Value <input type="text" value="50"/>				

Parameter Name	Unit	Minimum ~ Maximum	Default	16/32 bit
P2 - 03	ms	2 ~ 100	5	16bit
Smooth Constant of Position Feed Forward Gain				
Value <input type="text" value="5"/>				

Figura 3.64 Position Feed Forward Gain y P2-03 Smooth Constant of Position Gain

Se observa que, a través de un diferenciador, se toma la señal del comando de posición deseada y si el comando de posición se cambia suavemente, aumentar el valor de ganancia en P2-02 y el valor de P2-03 puede reducir el error de posición y acortar el tiempo de establecimiento. Por otra parte, si el comando de posición no cambia suavemente, la disminución del valor de ganancia y la constante suave de ganancia pueden generar un problema de vibración mecánica en el servomotor. Si el valor de PFG (P2-02) se establece demasiado grande, podría causar un sobreimpulso.

Por otro lado, en la etapa del lazo de control feedback se presenta la etapa proporcional del control de posición, donde actúa la ganancia KPP sobre el error. Este parámetro se describe a continuación:

KPP (P2-00): Este parámetro determina la respuesta del lazo de posición (Ganancia proporcional). Cuanto mayor sea el valor de KPP, mayor será la frecuencia de respuesta del lazo de posición. Y causará un mejor seguimiento del error, un error de posición más pequeño y un tiempo de establecimiento más corto. Sin embargo, si KPP se establece demasiado grande, el ancho de banda de respuesta del lazo de posición aumentará y disminuirá el margen de fase. El rotor del servomotor girará con vibración en dirección hacia adelante y hacia atrás en todo momento (rebasará la posición deseada); por lo tanto, KPP debe reducirse hasta que el rotor deje de vibrar. Cuando el torque externo interrumpe, el KPP excesivamente bajo no puede satisfacer la demanda de desviación de posición. En esta situación, el parámetro P2-02 (mencionado en la etapa anterior) puede reducir efectivamente el error de posición (véase la figura 3.65). Los valores de la ganancia proporcional (para el lazo de posición) van de 0 a 2047 rad/s.

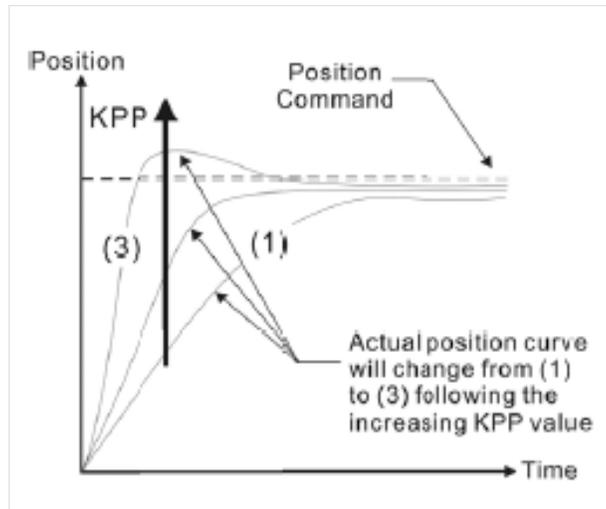


Figura 3.65 Efectos de KPP (ganancia proporcional)

Como se observa en la figura 26, después de la ganancia proporcional del control de posición KPP hay dos caminos, el directo al punto de suma y otro que conecta a una etapa llamada “Switching Rate of Position Loop Gain” (P2-01) que entra en acción solamente cuando el parámetro de cambio de ganancia P2-27 hace esa conmutación.

Se aprecia que Switching Rate of Position Loop Gain tiene un rango entre 10% a 500%. Este parámetro es una velocidad cambiante de la “Ganancia del lazo de posición” de acuerdo con la condición de cambio de ganancia. Como se observa, la opción Gain Switching Mode debe establecerse como cero y cuando se hace la conmutación, entra en acción la etapa “Switching Rate of Position Loop Gain”.

Después de las etapas anteriores, la señal de salida será la que controlará al servomotor (señal de control) pero no sin antes pasar por una etapa de Límite de velocidad máxima (figura 29), para que no sobrepase una velocidad que pueda afectar el funcionamiento del servomotor. Como se muestra en la figura 30, el rango de velocidad va de 10 a 6000 r/min, sin embargo, en el parámetro P1-55 se establecerá la velocidad máxima dependiendo de las características del servomotor que se use. Finalmente, después de esa etapa, se obtiene el comando de velocidad.

Para finalizar con el análisis de control de posición interno del servo drive, se observa en la figura 3.63 que el lazo de retroalimentación del sistema está dado por la lectura del encoder que contiene el servomotor. Como tal el Encoder es el dispositivo y a través de la etapa "Position Counter", se hace la lectura de los pulsos generados. Esa señal sensada llega al primer punto de suma que determinará el error entre la posición deseada y la real del servomotor.

Por lo tanto, se concluye que el tipo de control integrado en el servo drive es de tipo proporcional anticipativo, para la disminución de vibraciones o perturbaciones mecánicas en el sistema robótico.

CAPÍTULO IV RESULTADOS

En este capítulo se muestran las pruebas experimentales para validar las teorías respecto a la cinemática y control del robot paralelo activado por cables, además, se presentan los resultados del tema que comprenden:

- Cálculo de la posición inicial del robot.
- Programación y modelación en Matlab-Simulink.
- Simulaciones de diferentes posiciones del robot usando la cinemática inversa en Matlab-Simulink.
- Programación en PLC Delta para la Cinemática inversa del robot.
- Programación en MC Delta para el movimiento del robot.
- Programación del HMI Delta para la manipulación del robot.
- Tablas de resultados de las diferentes posiciones del robot usando C.I.
- Revisión de primeros resultados.
- Ajustes y resultados finales.

4.1 CÁLCULO DE POSICIÓN INICIAL

En primera estancia, se obtienen los resultados de l_1 y l_2 en la posición cero o posición Home, es decir, cuando P_x , P_y y el ángulo de inclinación θ valen cero. Se toman en cuenta las medidas de la plataforma móvil (base 'c' y altura 'd') y de la plataforma fija (base 'e' y altura 'f') y se sustituyen en las ecuaciones 3.2 y 3.4 para obtener l_1 y l_2 respectivamente:

$$c = 82 \text{ mm} \quad d = 46 \text{ mm} \quad e = 1480 \text{ mm} \quad f = 900 \text{ mm}$$

Como se mencionó anteriormente, se usó la ecuación 3.2 para calcular l_{1Home} :

$$l_1 = \sqrt{\left(P_x - \frac{c - e}{2}\right)^2 + \left(P_y + \frac{d - f}{2}\right)^2} = \sqrt{\left(0 - \frac{82 - 1480}{2}\right)^2 + \left(0 + \frac{46 - 900}{2}\right)^2} \text{ mm}$$

Por lo tanto, cuando $P_x = P_y = 0$ (posición central "Home") se obtiene que l_1 vale:

$$l_{1Home} = 819.103 \text{ mm}$$

Por otra parte, se sabe que en la posición Home el valor de $l_{2Home} = l_{1Home}$. A continuación, se comprueba esa suposición usando la ecuación 3.4:

$$l_2 = \sqrt{\left(P_x + \frac{c - e}{2}\right)^2 + \left(P_y + \frac{d - f}{2}\right)^2} = \sqrt{\left(0 + \frac{82 - 1480}{2}\right)^2 + \left(0 + \frac{46 - 900}{2}\right)^2} \text{ mm}$$

Por lo tanto, cuando $P_x = P_y = 0$ (posición central "Home") se obtiene que l_2 vale:

$$l_{2Home} = 819.103 \text{ mm}$$

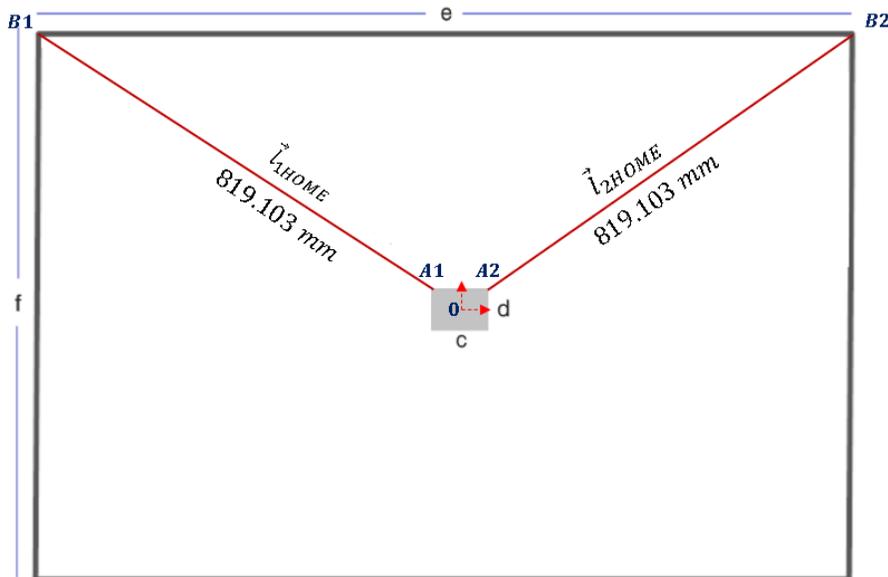


Figura 4.1 Posición Home del CDPR

Se comprobó que en la posición cero ('Home') la longitud de los cables l_1 y l_2 es la misma en ese punto (ver figura 4.1), entonces:

$$\text{Si } P_x = 0 \text{ \& } P_y = 0 \rightarrow l_{1Home} = l_{2Home} = 819.103 \text{ mm}$$

4.2 PROGRAMA INICIAL PARA SIMULAR EN MATLAB-SIMULINK

El programa generado en Matlab-Simulink es necesario para simular el comportamiento del robot y comparar los resultados simulados con los resultados reales. Este programa se conforma de dos partes: el modelo virtual del CDPR y el modelo de la cinemática inversa del robot.

4.2.1 Modelo del robot paralelo activado por cables

Empleando datos como las dimensiones del robot y los valores extraídos de las ecuaciones anteriores (longitudes de cables l_{1Home} y l_{2Home}), se generó el modelo del robot por medio de la librería Simscape Multibody First Generation, en SimMechanics de Matlab-Simulink. Se consideró la figura 3.5 “Diagrama esquemático final del robot planar paralelo” para definir las articulaciones. Una parte del modelo se presenta en la figura 4.2; de acuerdo al color del marco que rodea al objeto, se describen los elementos junto con su función en el modelo del CDPR.

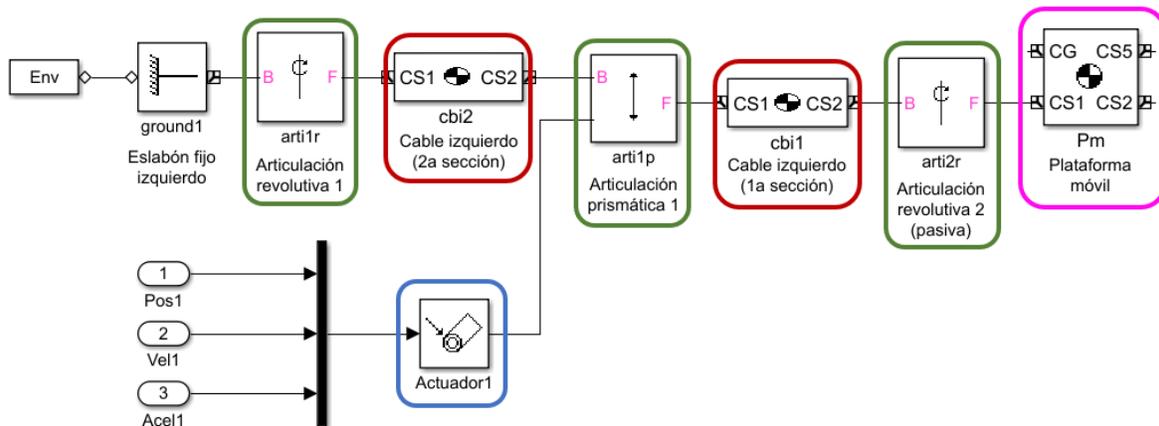


Figura 4. 2 Modelo del CDPR sección izquierda

- Plataforma móvil (—): Se ubica en la posición central del modelo virtual. En este bloque se definen las dimensiones del efector final (base y altura). Sin embargo, se debe tomar en cuenta que el sistema de referencia de la

plataforma (punto central) se ubica inicialmente en el punto de referencia absoluto del sistema, por ende, los parámetros de la plataforma móvil se introducen como se observa en la figura 4.3.

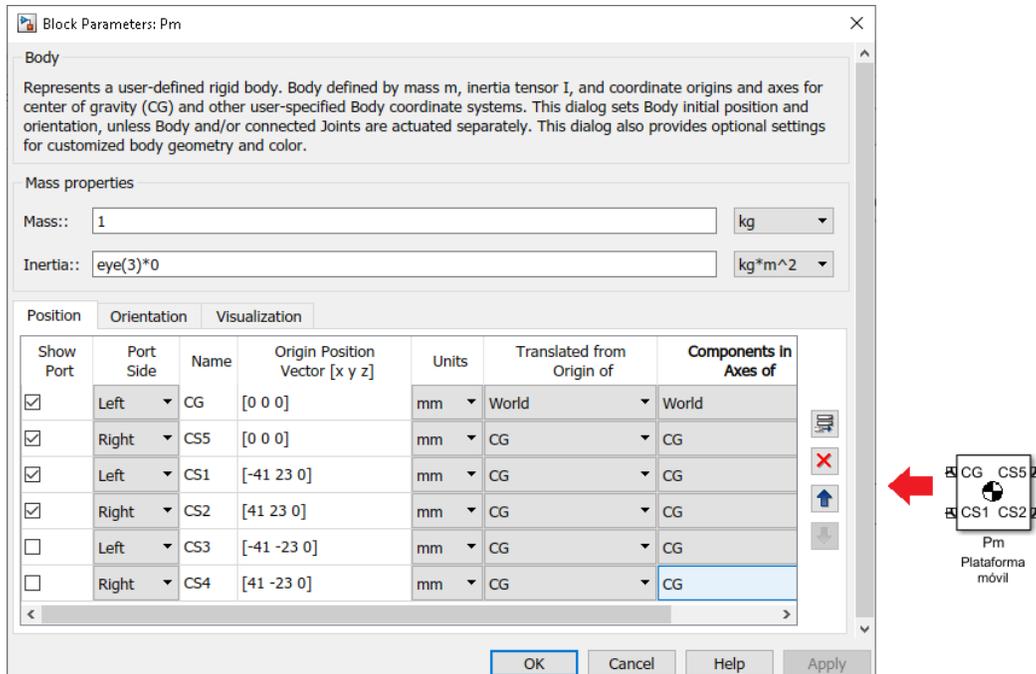


Figura 4. 3 Configuración de dimensiones de la plataforma móvil

- Cable izquierdo (—): Se encuentra enmarcado de color rojo; de acuerdo al diagrama esquemático del CDPDR, cada cable se puede representar como la unión de dos eslabones rígidos (dos secciones por cable). Además, en su configuración se debe tomar en cuenta la pendiente del cable.
- Articulaciones (—): De acuerdo al diagrama esquemático, se determinaron 6 articulaciones en total, cuatro de tipo rotacional y dos prismáticas. En este caso, se observa que las articulaciones que actúan pasivamente son las rotacionales, mientras que las articulaciones actuadas son las prismáticas. Las articulaciones se encuentran unidas a los cables, al efector final y a los eslabones fijos también llamados “tierra”.
- Actuador (—): Los actuadores se conectan a las articulaciones prismáticas; sus entradas son posición, velocidad y aceleración articulares (medidas en grados o pulsos).

Finalmente, en la figura 4.4 se muestra el modelo virtual completo del CDPR. En el modelo se presenta un bloque llamado “Entorno de máquina” para poder simular el sistema; por otra parte, la plataforma móvil tiene como salidas unos sensores de posición y de inclinación de la plataforma, para medir el comportamiento del robot al simular la cinemática inversa.

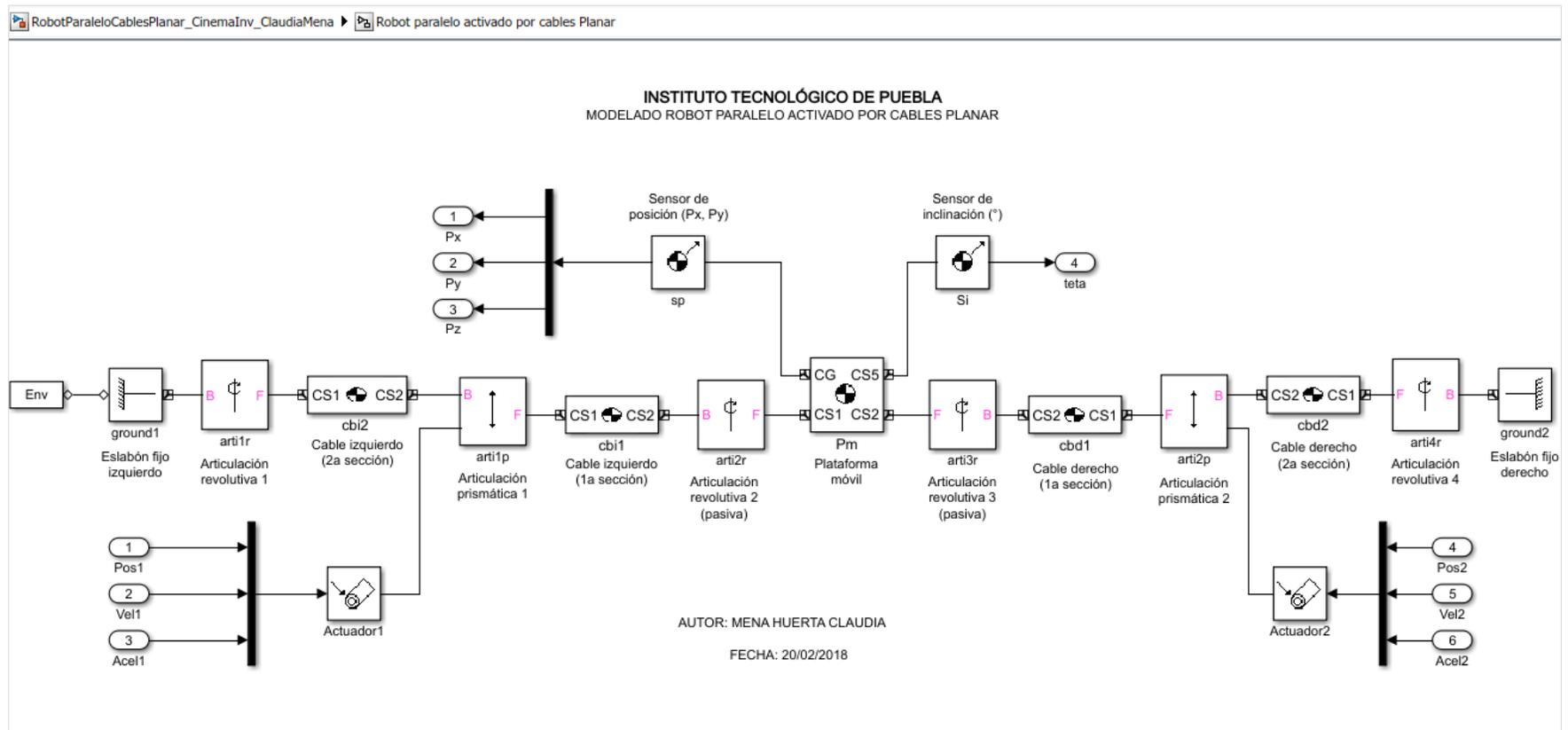


Figura 4. 4 Modelo completo del Robot Paralelo activado por cables planar en Matlab-Simulink

4.2.2 Modelo de la cinemática inversa del CDPR

Otro aspecto programado en Simulink es el modelo cinemático inverso, el cual se presenta en la figura 4.5.

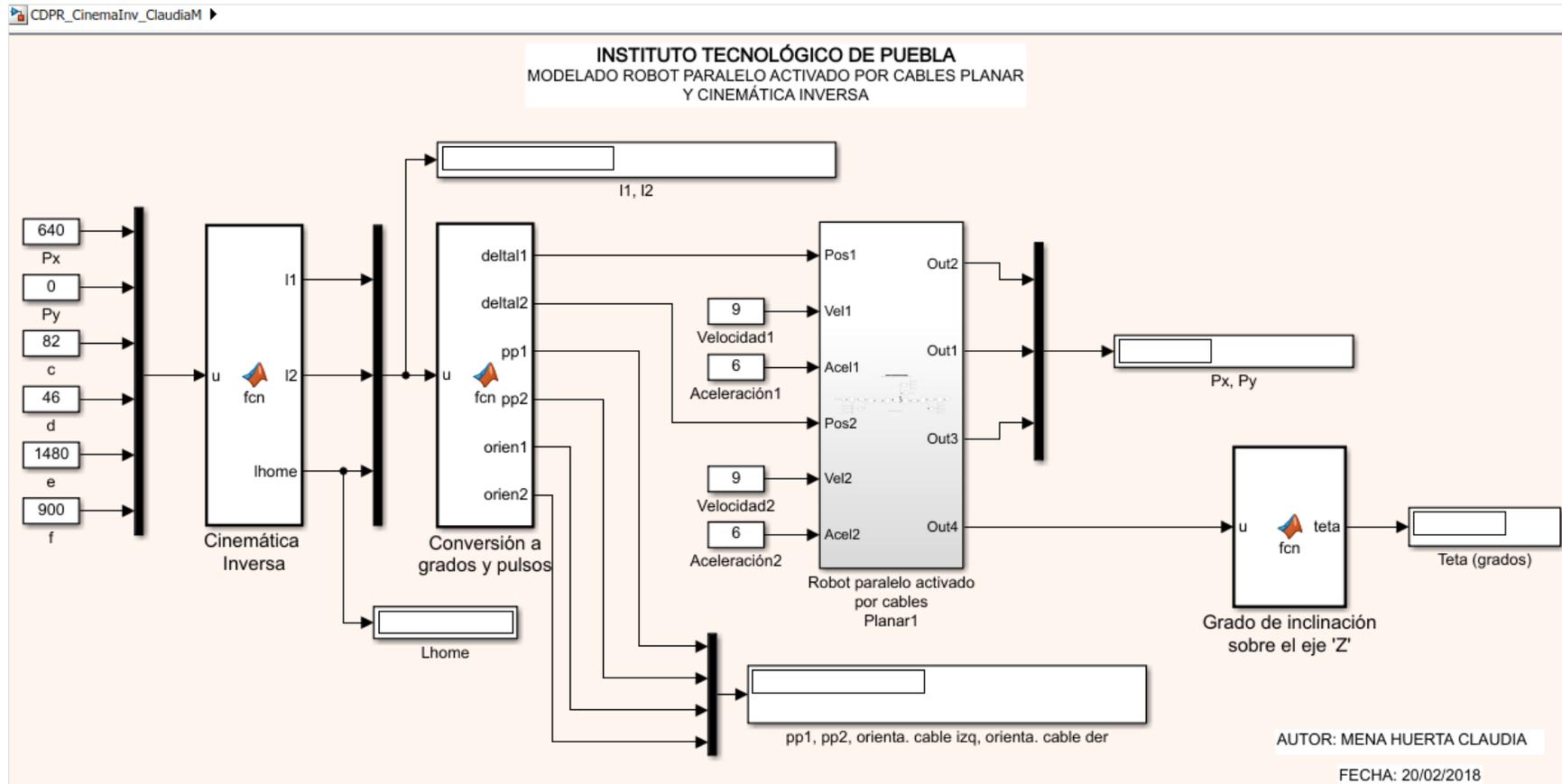


Figura 4. 5 Modelación virtual de la cinemática inversa con $\theta = 0$

El primer bloque llamado “Cinemática Inversa” tiene como entradas la posición deseada en coordenadas cartesianas P_x y P_y en milímetros y las dimensiones de las plataformas fija y móvil del robot (medidas en milímetros). También tiene tres variables como salida: l_1 , l_2 y l_{Home} . Las ecuaciones de cinemática inversa se programaron en Matlab como se observa en la figura 4.6.

```

1 function [l1,l2,lhome] = fcn(u)
2 % Instituto Tecnológico de Puebla
3 % Alumna: Mena Huerta Claudia
4 % Cinemática Inversa Robot Planar Paralelo Cableado
5 % 28-ago-18
6
7 % Entradas
8 Px = u(1); % Coordenada X de un punto dado
9 Py = u(2); % Coordenada Y de un punto dado
10 c = u(3); % Base de plataforma móvil
11 d = u(4); % Altura de plataforma móvil
12 e = u(5); % Base de plataforma fija
13 f = u(6); % Altura de plataforma fija
14
15 % Cálculos Cinemática Inversa
16 l1 = sqrt((Px - (c-e)/2)^2 + (Py + (d-f)/2)^2); % Longitud variable de l1
17 l2 = sqrt((Px + (c-e)/2)^2 + (Py + (d-f)/2)^2); % Longitud variable de l2
18 lhome = sqrt(((c-e)/2)^2 + ((d-f)/2)^2); % Longitud constante de lhome

```

Figura 4. 6 Primer bloque de programación en Matlab: Cinemática Inversa

Es necesario clarificar el por qué se usaron las ecuaciones 3.2 y 3.4 (cuando $\theta = 0$) para simular la cinemática inversa y no se tomaron en cuenta las ecuaciones 3.1 y 3.3 (cuando $\theta \neq 0$). Las ecuaciones para la obtención de l_1 y l_2 tienen como variables de entrada la posición P_x , P_y y el ángulo de inclinación θ . Teóricamente se sabe que el robot es incompletamente restringido, por lo tanto, la resolución de este sistema es más compleja. En la realidad no se sabe exactamente cómo será el comportamiento del robot al posicionar su plataforma móvil en distintos puntos del plano, por consiguiente, se supone en un principio que al mover la plataforma en cualquier punto no tendrá inclinación alguna, entonces, se define el ángulo de inclinación de la plataforma θ con un valor igual a cero.

Posteriormente, en la simulación los valores de salida del primer bloque se envían a un segundo bloque llamado “Conversión a grados y pulsos”, tal como se expone en la figura 4.7. En este pequeño programa se obtiene primero la diferencia entre l_{Home} menos la longitud de l_1 deseada; se realiza el mismo cálculo para el segundo cable. Por último, esa diferencia de longitudes se transforma en pulsos para comparar con los resultados físicos y en grados para la simulación.

```

1  function [deltal1, deltal2, pp1, pp2, orien1, orien2] = fcn(u)
2  % Instituto Tecnológico de Puebla
3  % Alumna: Mena Huerta Claudia
4  % Cinemática Inversa Robot Planar Paralelo Cableado
5  % 28-ago-18
6
7  % Entradas
8  l1 = u(1);
9  l2 = u(2);
10 lhome = u(3);
11
12 % Cálculos Cinemática Inversa
13 deltal1 = l1 - lhome;
14 deltal2 = l2 - lhome;
15 pp1 = (deltal1*10000)/237;
16 pp2 = (deltal2*10000)/237;
17
18 orien1 = 360 - acosd(699/lhome); % ángulo para 1° orientación del cable derecho 2a sección
19 orien2 = 180 + acosd(699/lhome); % ángulo para 2° orientación del cable derecho 2a sección
20 % 699 = e/2 - c/2 = 900/2 - 46/2 = 740 - 41

```

Figura 4. 7 Segundo bloque de programación en Matlab: Conversión a grados

La cantidad de grados que debe girar cada actuador se envía al modelo virtual del robot paralelo activado por cables; los datos de salida arrojados por el modelo del CDPR son la posición simulada y el ángulo de inclinación obtenidos por los sensores.

Los datos de posición $P_{x_{real}}$ y $P_{y_{real}}$ se visualizan en un display; por su parte, el ángulo de inclinación del efector final no se envía directamente al otro display, porque el sensor arroja una matriz rotacional, por lo tanto, de esa matriz se calcula el ángulo θ de la siguiente forma: $teta = \text{acosd}(\text{Valor1MatrizRotación})$. Finalmente, el dato obtenido de θ se muestra en un segundo display.

4.3 SIMULACIONES

A través del modelo de la cinemática inversa generado en Matlab, se realizaron diversas simulaciones ingresando puntos del plano 'xy'. En la figura 4.8 se observan los resultados obtenidos de la simulación cuando se mueve la plataforma a la posición deseada 640 mm en 'x' y 0 en 'y'.

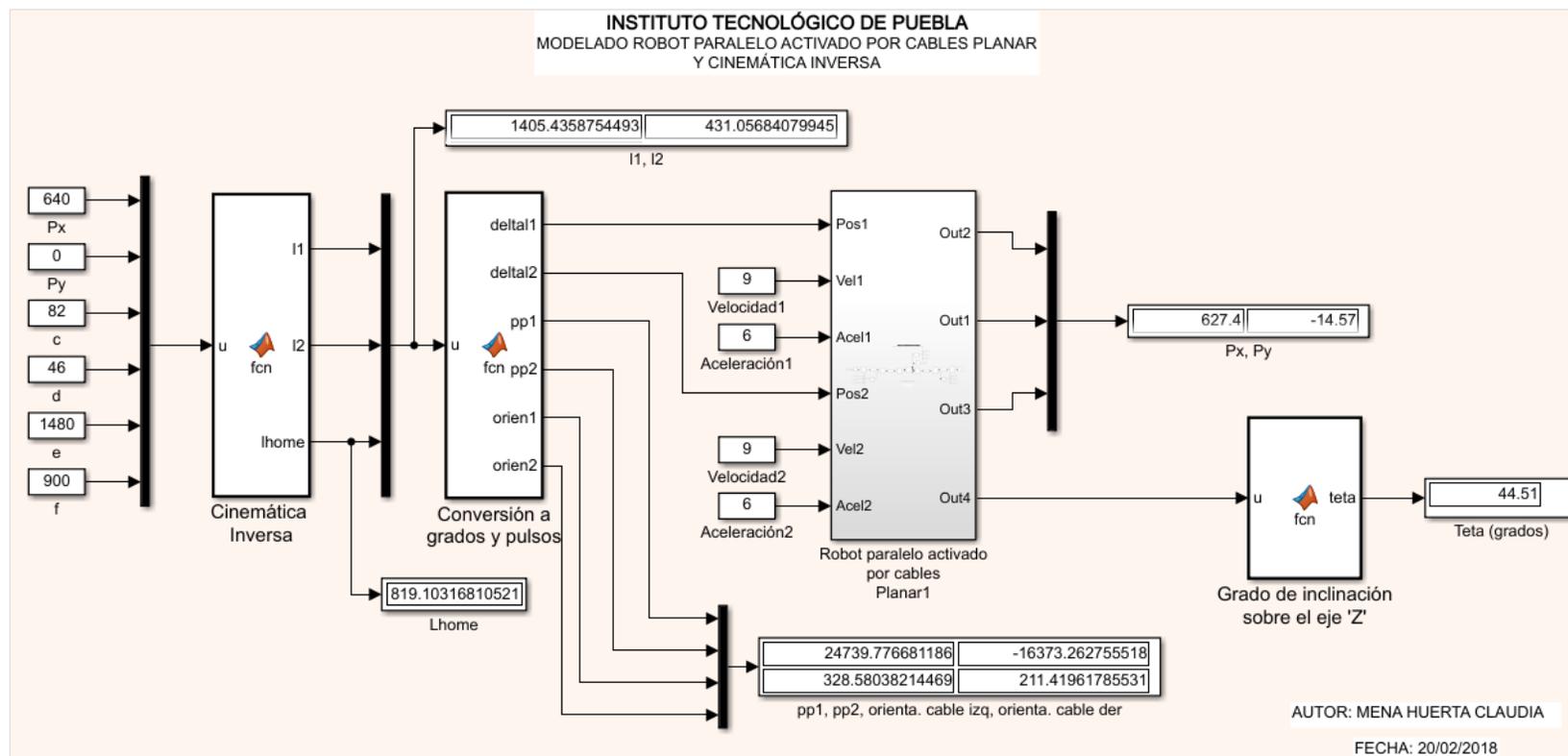


Figura 4. 8 Simulación de tercera coordenada

Se usaron ocho puntos de prueba en el espacio. A continuación, se muestran las simulaciones para algunos puntos deseados; además, los resultados se ingresaron en las tablas 4.1 – 4.3 para compararlos con los resultados reales del robot físico.

En la figura 4.9 se observa la simulación de la posición inicial, mientras que en la figura 4.10 se presenta la simulación del movimiento en la primera coordenada (0 en 'x' y 320 milímetros en 'y').

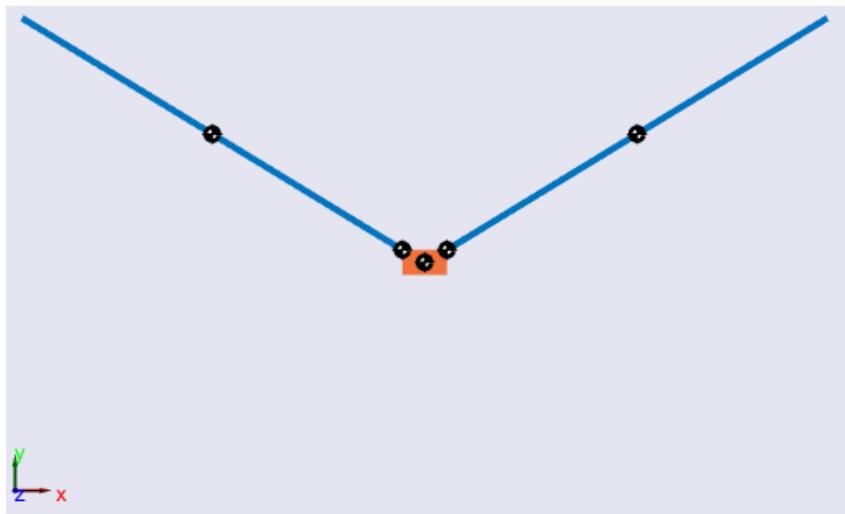


Figura 4.9 Simulación de posición 'Home'

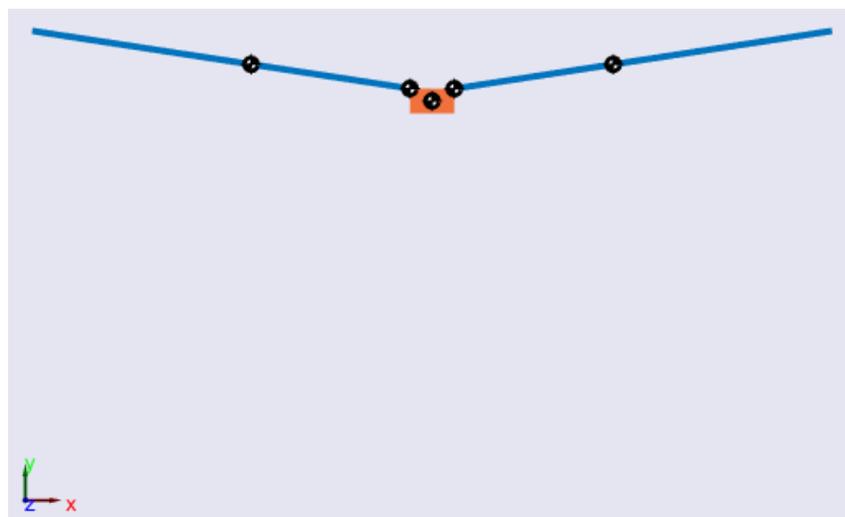


Figura 4.10 Simulación de la primera coordenada (0, 320)

El punto simulado 0 en 'x' y -450 mm en 'y', así como el punto 640 mm en 'x' y 0 en 'y', son mostrados en las figuras 4.11 y 4.12 respectivamente.

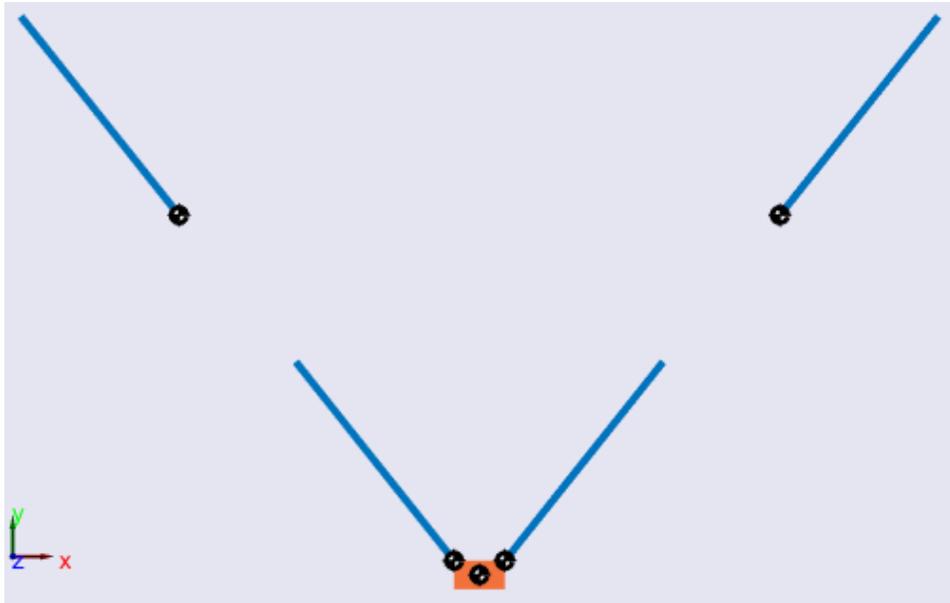


Figura 4.11 Simulación de la segunda coordenada (0, -450)

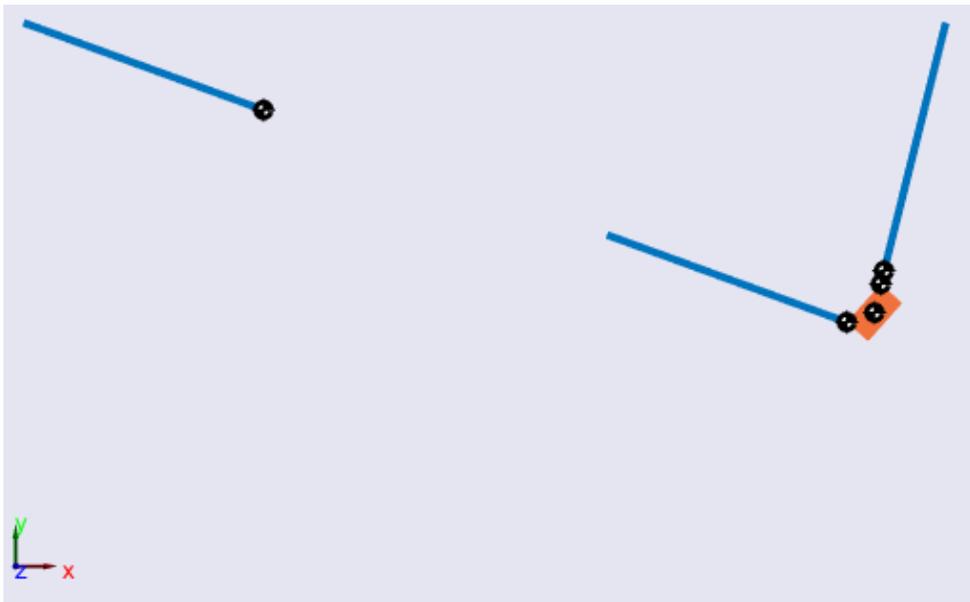


Figura 4.12 Simulación de la tercera coordenada (640, 0)

Finalmente, se simularon las coordenadas no. 6 y no. 8, correspondientes a -600 mm en 'x' con 300 mm en 'y' y -640 mm en 'x' con -450 mm en 'y'. Se muestran en las figuras 4.13 y 4.14 respectivamente.

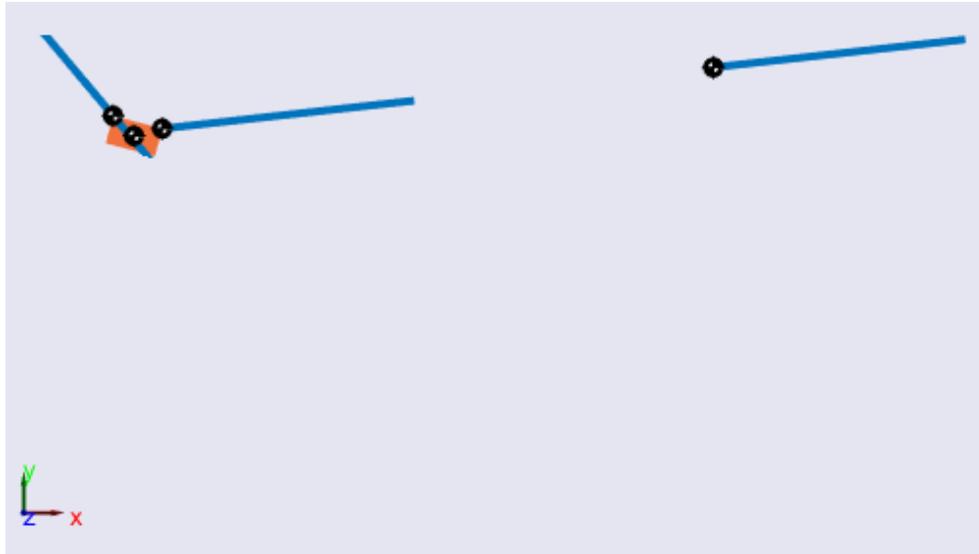


Figura 4.13 Simulación de la sexta coordenada (-600, 300)

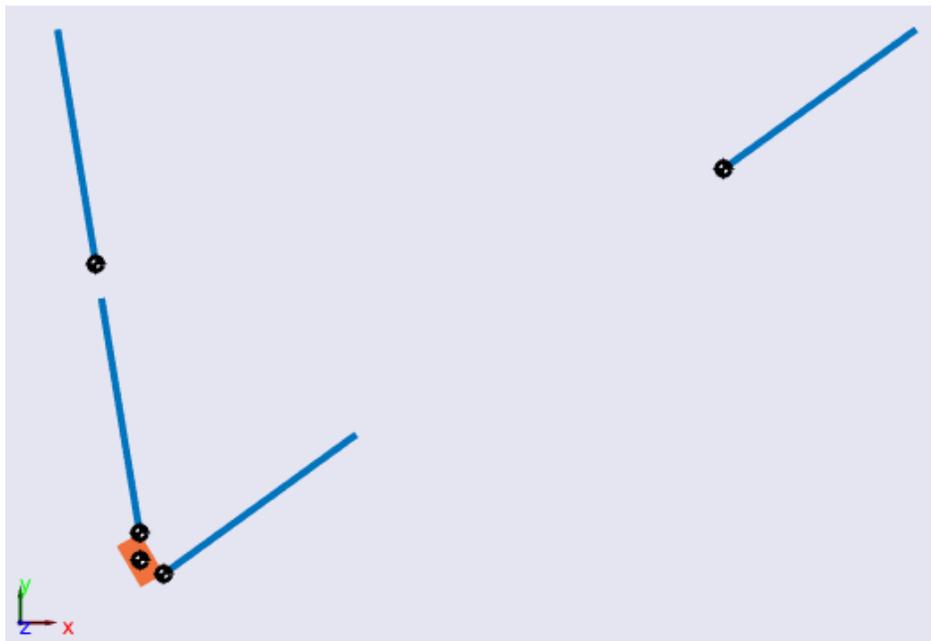


Figura 4.14 Simulación de la octava coordenada (-640, -450)

4.4 PROGRAMA EN PLC/MC/HMI

El programa de la cinemática inversa en el robot se realizó en el PLC/MC Delta DVP10MC11T junto con su HMI y en la figura 4.15 se muestra su resumen.

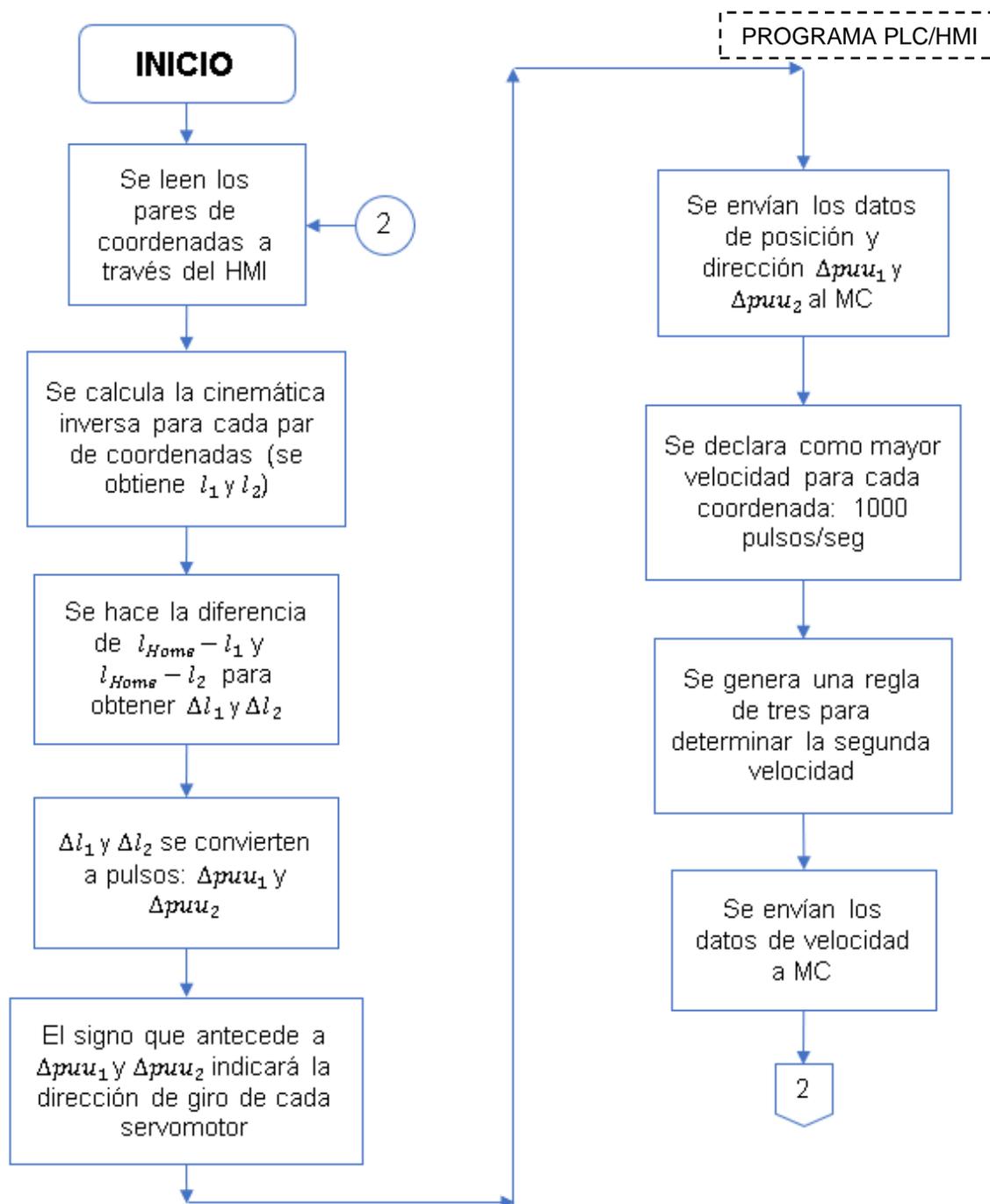
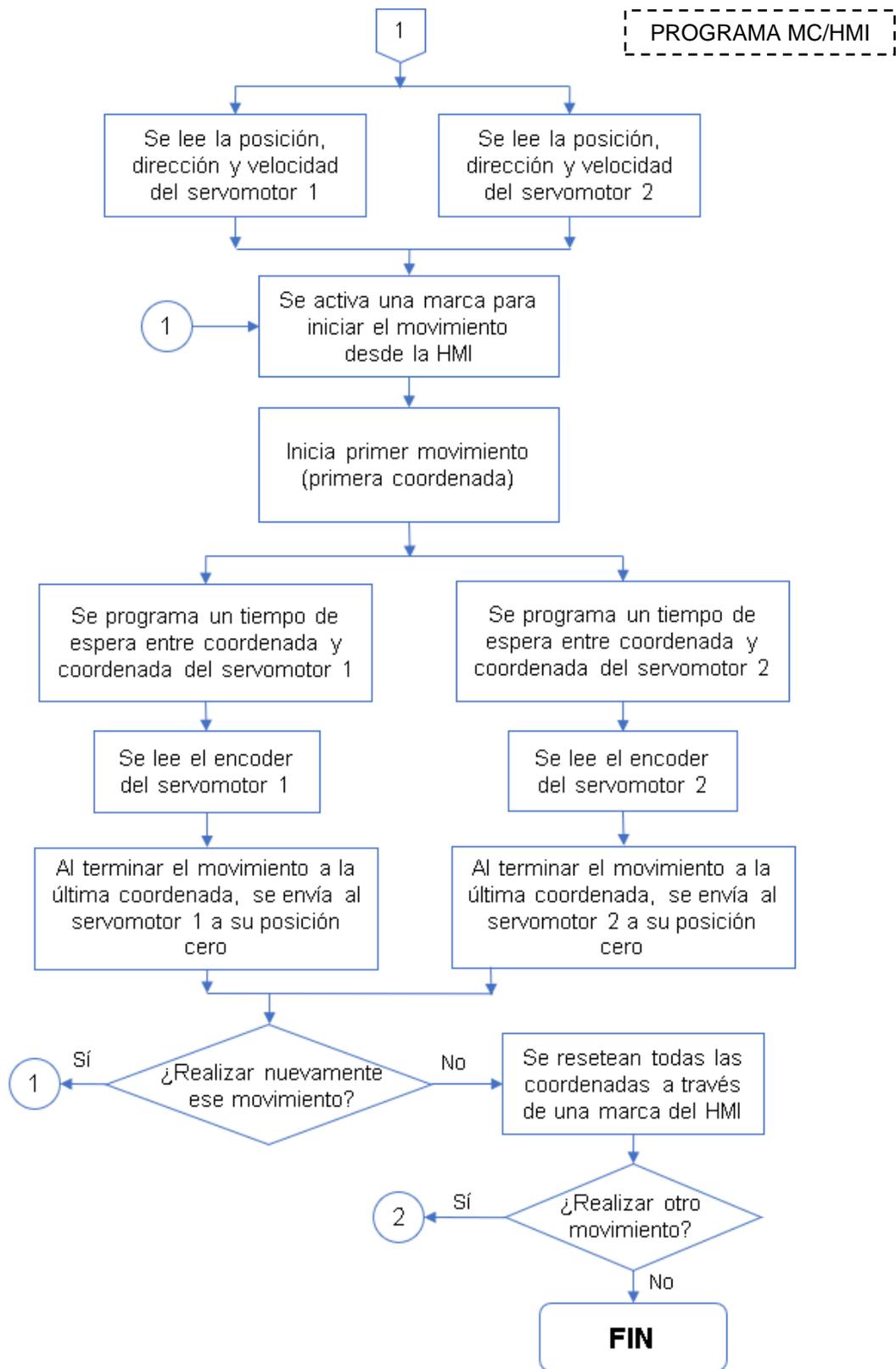


Figura 4.15 Diagrama de Programa en PLC/HMI



Continuación de figura 4.15 Diagrama de Programa en MC/HMI

Como se observa en el diagrama de flujo, se generó un programa para el movimiento del robot paralelo activado por cables que consta de tres partes: programación en PLC Delta, programación en MC Delta y programación de Interfaz Máquina-Usuario.

4.5 PRUEBAS INICIALES

Se generaron pruebas en el robot usando el programa anterior y se tomaron distintas mediciones como se muestran en las tablas 4.1, 4.2 y 4.3. Es importante mencionar que el algoritmo de la cinemática inversa tiene como principal característica la condición inicial $\theta = 0$.

Tabla 4.1 Primeras pruebas: Mediciones de puntos en el plano (Px, Py) y ángulo de inclinación de la plataforma

COORDENADAS							ÁNGULO	
N°	Deseadas		Matlab		Reales		Matlab	Real
	Px (mm)	Py (mm)	Px (mm)	Py (mm)	Px (mm)	Py (mm)	θ	θ
1	0	320	0.001	320	0	295	0.005°	0
2	0	-450	0.2	-450	0	-450	0.4°	0
3	640	0	627.4	-14.57	627	-15	44.51°	42°
4	-640	0	-622.5	-12.91	-625	-15	54.3°	43°
5	600	300	602.7	293.9	603	295	15.11°	21°
6	-600	300	-602.7	293.9	-603	295	14.99°	21°
7	640	-450	615.7	-465.5	615	-465	36.18°	47°
8	-640	-450	-586.2	-458.7	-615	-465	75.5°	47°

Como se observa en la tabla 4.1, las coordenadas reales del robot difieren en pequeña cantidad con respecto a las coordenadas simuladas en Matlab, sin embargo, al comparar la coordenada real contra la coordenada deseada se observa que mientras más lejano sea el punto o coordenada de medición el error va incrementando.

Por otra parte, en la tabla 4.2 se presentan los resultados de la cinemática inversa para cada coordenada (valores de l_1 y l_2). Se observa que los resultados en Matlab en comparación con los del MC son iguales; la comparación contra las longitudes de cables reales es muy cercana, siendo el error medido en décimas. Este pequeño error se debe a las mediciones realizadas de forma indirecta.

Tabla 4.2 Primeras pruebas: Mediciones de longitudes de cables l_1 y l_2 en cada punto

Punto		LONGITUD DE CABLE						
N°	Deseadas		Matlab		PLC/MC		Reales	
	Px	Py	l_1 (mm)	l_2 (mm)	l_1 (mm)	l_2 (mm)	l_1 (mm)	l_2 (mm)
1	0	320	707.1421	707.1421	707.142	707.142	707	707
2	0	-450	1121.4856	1121.4856	1121.49	1121.49	1121	1122
3	640	0	1405.4358	431.0568	1405.44	431.057	1405	431
4	-640	0	431.0568	1405.4358	431.057	1405.44	431	1405
5	600	300	1305.1934	161.0279	1305.19	161.028	1305	161
6	-600	300	161.0279	1305.1934	161.028	1305.19	161	1305
7	640	-450	1600.6404	878.9823	1600.64	878.982	1600	878
8	-640	-450	878.9823	1600.6404	878.982	1600.64	878	1600

Finalmente, en la última tabla 4.3 se muestran las comparaciones de los pulsos o eventos que deben girar cada servomotor. En las primeras dos columnas se muestra el punto deseado; posteriormente en las columnas tres y cuatro se muestran los pulsos a girar medidos en Matlab-Simulink; en las últimas dos columnas se muestran los pulsos calculados por el PLC/MC Delta. Se observa que los resultados son iguales, por lo cual no existe un error de medición en Matlab ni en el PLC/MC Delta.

Tabla 4.3 Primeras pruebas: Mediciones de pulsos que debe girar cada servomotor en cada punto

Punto		EVENTOS				
N°	Deseadas		Matlab		PLC/MC	
	Px	Py	$\Delta Ppu1$ (pulsos)	$\Delta Ppu2$ (pulsos)	$\Delta Ppu1$ (pulsos)	$\Delta Ppu2$ (pulsos)
1	0	320	-4724.0941	-4724.0941	-4724.09	-4724.09
2	0	-450	12758.7533	12758.7533	12758.8	-12758.8
3	640	0	24739.7766	-16373.2627	24739.8	16373.3
4	-640	0	-16373.2627	24739.7766	-16373.3	-24739.8
5	600	300	20510.1394	-27766.8869	20510.1	27766.9
6	-600	300	-27766.8869	20510.1394	-27766.9	-20510.1
7	640	-450	32976.2585	2526.5484	32976.3	-2526.55
8	-640	-450	2526.5484	32976.2585	2526.55	-32976.3

En las figuras 4.16 a 4.19 se muestran las posiciones de la plataforma móvil de robot en los puntos $(0, 0)$, $(0, -450)$, $(-600, 300)$ y $(640, -450)$. En la posición cero (figura 4.16) se midieron las longitudes de los cables y correspondían aproximadamente a 819 milímetros; también se midió el ángulo y se detectó en cero grados. Para realizar las mediciones posteriores, se colocó una guía de referencia con cables en forma de cuadrante.

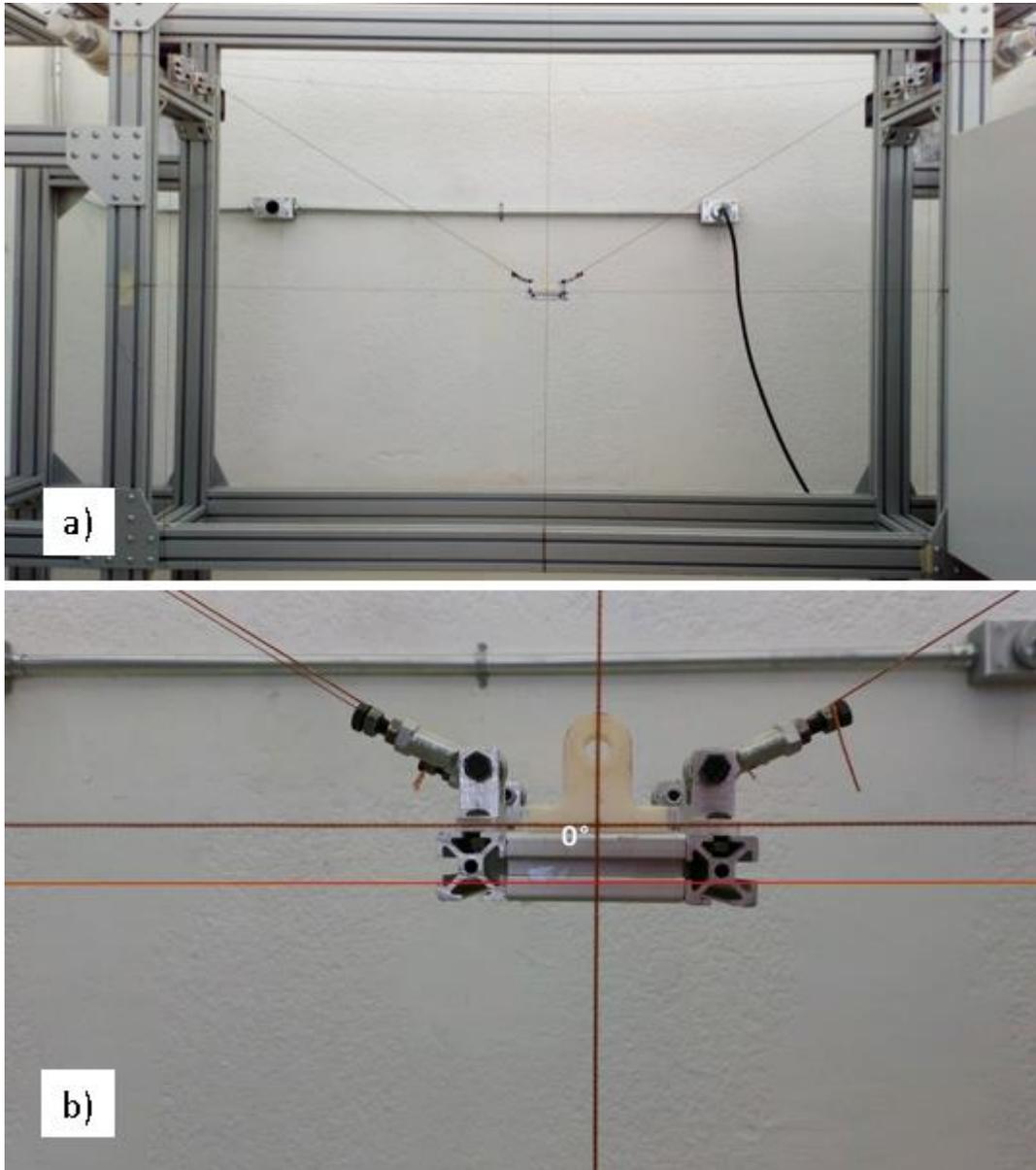


Figura 4.16 Prueba inicial: a) Posición Home $(0, 0)$.
b) Plataforma móvil con 0°

En la figura 4.17 se movió el robot a la posición 0 en el eje 'x' y -450 milímetros en el eje 'y'. Se midió la longitud de los cables y cada uno tuvo una medida de 1121 milímetros. Se midió la coordenada real con respecto al sistema de referencia y se obtuvo 0 sobre el eje 'x' y -450 milímetros sobre el eje 'y' con un ángulo de inclinación 0, tal como se muestra en la figura 4.17.

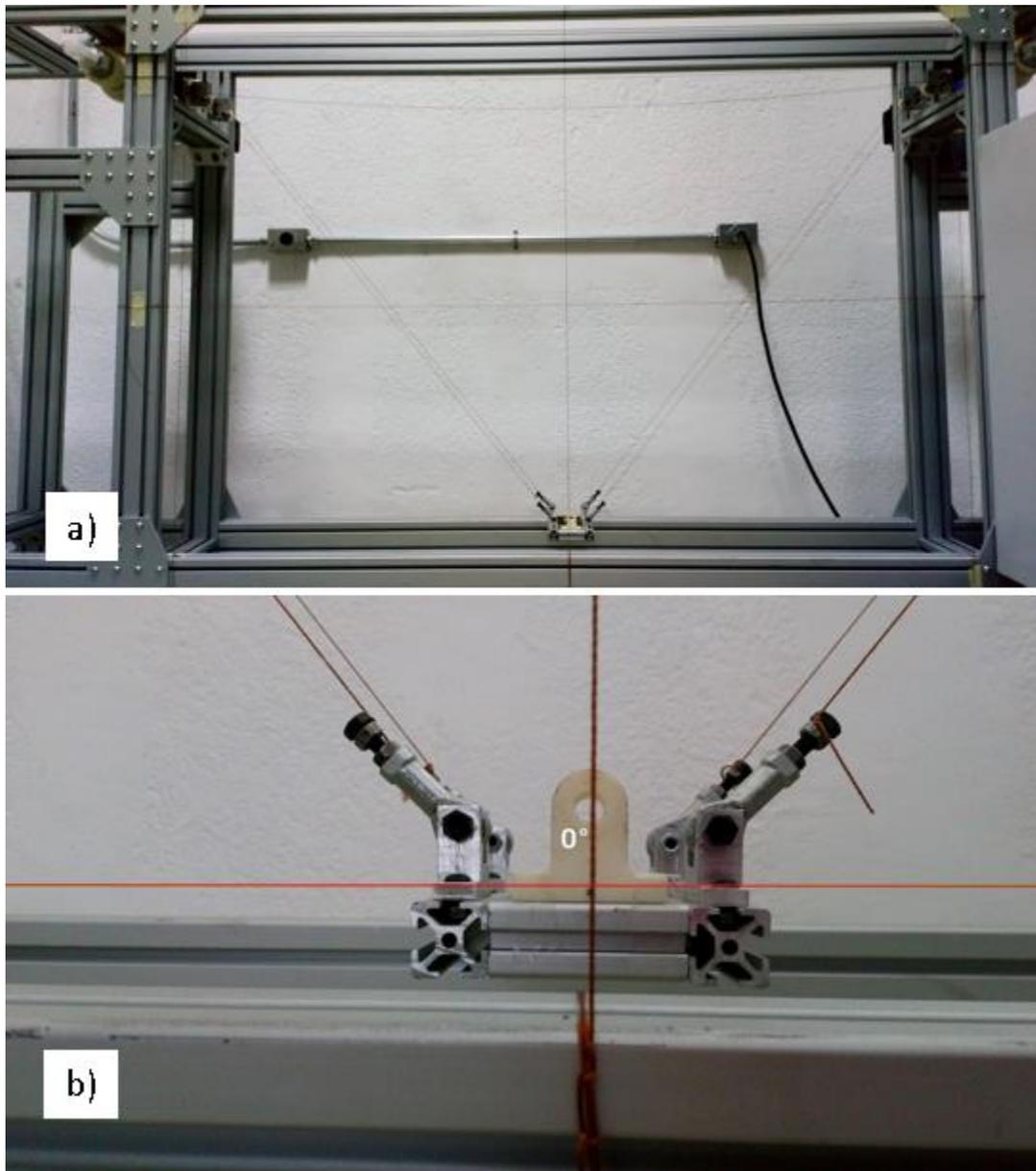


Figura 4.17 Prueba inicial: a) Segunda coordenada (0, -450).
b) Plataforma móvil con inclinación 0

Posteriormente, la sexta coordenada deseada correspondió a -600 milímetros en el eje 'x' y 300 milímetros en el eje 'y'. De igual forma, se midió la longitud de los cables; el cable l_1 (izquierdo) tuvo una medida de 161 milímetros, mientras que el cable l_2 (derecho) midió 1305 milímetros. Se midió la coordenada real con respecto al sistema de referencia y se obtuvo -603 milímetros sobre el eje 'x' y 295 milímetros sobre el eje 'y' con un ángulo de inclinación de 21° (véase la figura 4.18).

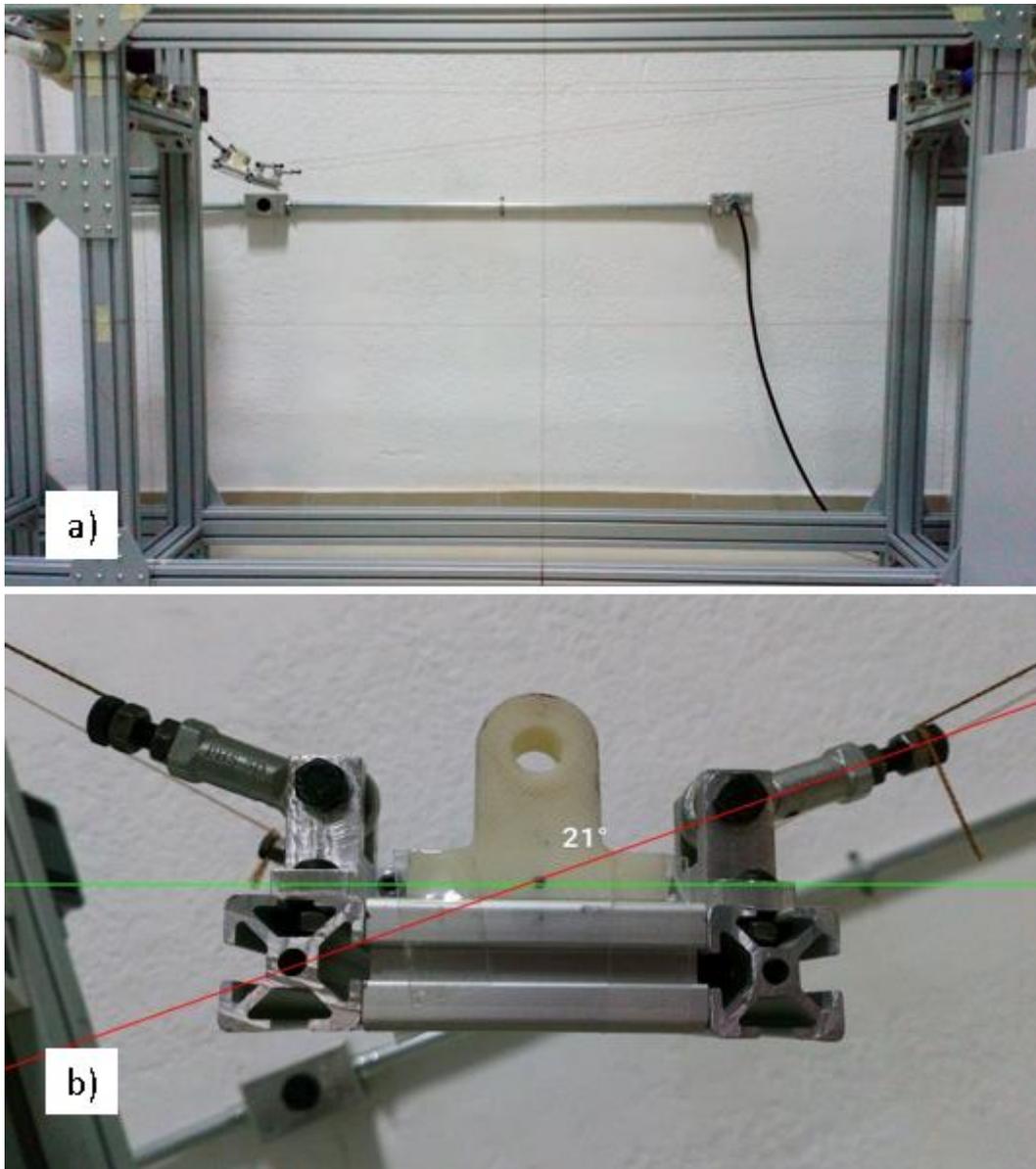


Figura 4.18 Prueba inicial: a) Sexta coordenada (-600, 300)
b) Plataforma móvil con inclinación de 21°

La figura 4.19 muestra la séptima coordenada deseada: 640 milímetros en el eje 'x' y -450 milímetros en el eje 'y'. Al medir la longitud de los cables, se obtuvo que el cable l_1 (izquierdo) tenía una longitud de 1600 milímetros y el cable l_2 (derecho) tenía una longitud de 878 milímetros. Se midió la coordenada real con respecto al sistema de referencia y se obtuvo 615 milímetros sobre el eje 'x' y -465 milímetros sobre el eje 'y' con un ángulo de inclinación de 47° .

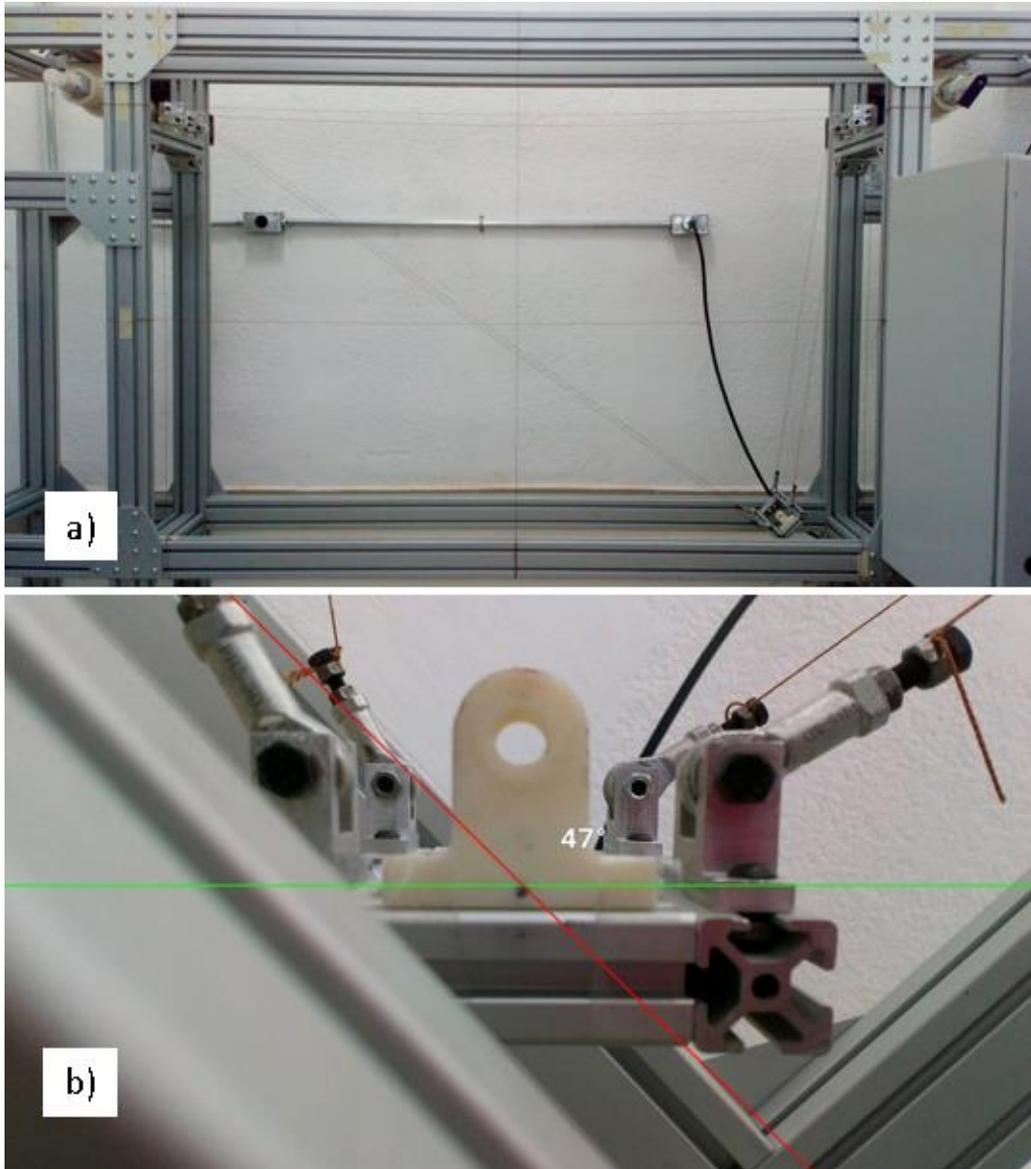


Figura 4.19 Prueba inicial: a) Séptima coordenada (640, -450)
b) Plataforma móvil con inclinación de 47°

4.5.1 Análisis y revisión de primeros resultados

En el programa de Matlab mostrado en la figura 4.20 se obtuvieron las gráficas de tres posiciones deseadas contra las posiciones reales de la plataforma móvil.

```
Figura420GraficasCoordenadas.m x +
1   % Instituto Tecnológico de Puebla
2   % Alumna: Mena Huerta Claudia
3   % Gráficas de tres posiciones del sistema de Control en el CDPR
4   % 14-ago-19
5
6 -  clc;
7 -  clear all;
8
9 -  C2d =[0; -450];
10 - C2r =[0; -450];
11 - figure(1)
12 - plot(0, -450, 'b*')
13 - axis([-640 640 -450 0])
14 - hold on
15 - plot(0, -450, 'r*')
16 - plotv(C2d, 'b-')
17 - plotv(C2r, 'r-')
18 - title('Gráfica de segunda coordenada: deseada vs real')
19 - xlabel('Eje X (mm)')
20 - ylabel('Eje Y (mm)')
21 - legend({'Coord. deseada (0,-450)', 'Coord. real (0,-450)', ...
22         '\surd(Xd^2+Yd^2)', '\surd(Xr^2+Yr^2)'}, 'Location', 'northeast')
23
24 - C6d =[-600; 300];
25 - C6r =[-603; 295];
26 - figure(2)
27 - plot(-600, 300, 'b*')
28 - axis([-640 640 -450 450])
29 - hold on
30 - plot(-603, 295, 'r*')
31 - plotv(C6d, 'b-')
32 - plotv(C6r, 'r-')
33 - title('Gráfica de sexta coordenada: deseada vs real')
34 - xlabel('Eje X (mm)')
35 - ylabel('Eje Y (mm)')
36 - legend({'Coordenada deseada (-600,300)', 'Coordenada real (-603,295)', ...
37         '\surd(Xd^2+Yd^2)', '\surd(Xr^2+Yr^2)'}, 'Location', 'southwest')
38
39 - C7d =[640; -450];
40 - C7r =[615; -465];
41 - figure(3)
42 - plot(640, -450, 'b*')
43 - axis([-640 640 -460 460])
44 - hold on
45 - plot(615, -465, 'r*')
46 - plotv(C7d, 'b-')
47 - plotv(C7r, 'r-')
48 - title('Gráfica de séptima coordenada: deseada vs real')
49 - xlabel('Eje X (mm)')
50 - ylabel('Eje Y (mm)')
51 - legend({'Coordenada deseada (640,-450)', 'Coordenada real (615,-465)', ...
52         '\surd(Xd^2+Yd^2)', '\surd(Xr^2+Yr^2)'}, 'Location', 'southwest')
```

Figura 4.20 Programa para graficar posición de la plataforma móvil en Matlab

En la figura 4.21 se observa la gráfica de la segunda posición (0, -450); el color azul representa la posición deseada, mientras que el color rojo muestra la posición real. Como se observa en la gráfica, no existe alguna desviación de las coordenadas reales comparadas con las coordenadas deseadas 'x' y 'y'.

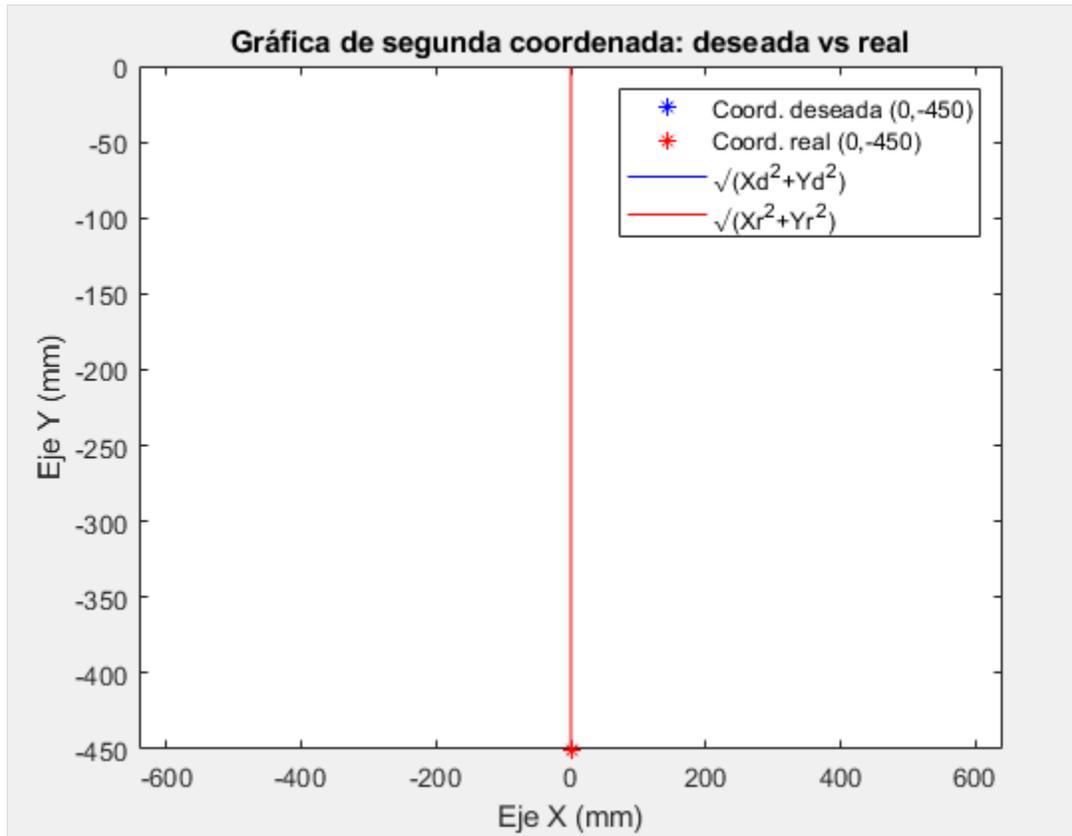


Figura 4.21 Gráfica de segundo par de coordenadas

En las figuras 4.22 y 4.23 se muestran las gráficas del sexto par de coordenadas y del séptimo par de coordenadas respectivamente. En el sexto par no se visualiza una fuerte desviación de las coordenadas reales contra las coordenadas deseadas, sin embargo, no ocurre la misma situación en la figura 4.23, donde se aprecia un error grande entre las coordenadas reales respecto a las coordenadas deseadas.

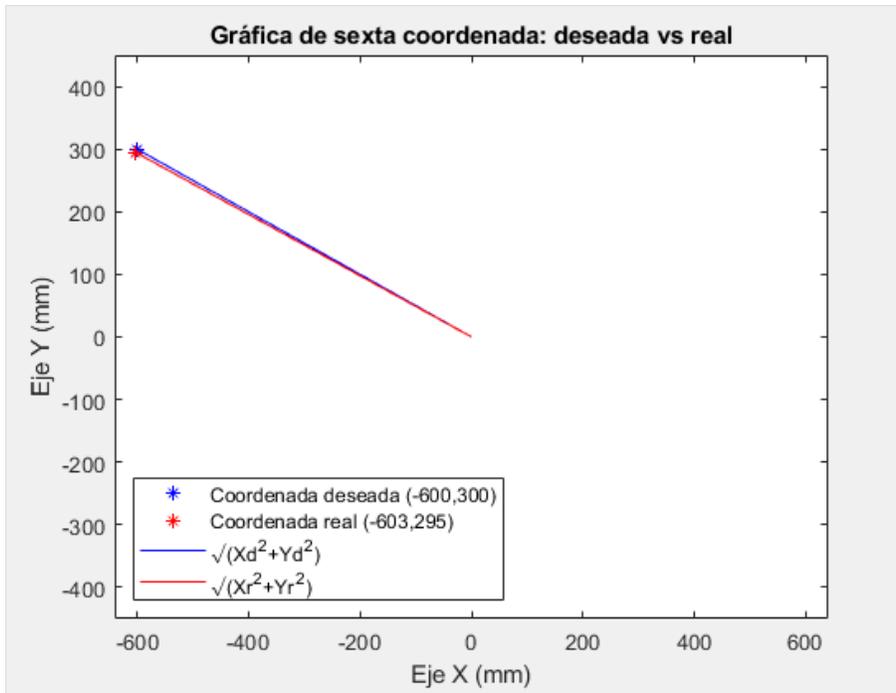


Figura 4.22 Gráfica de sexto par de coordenadas

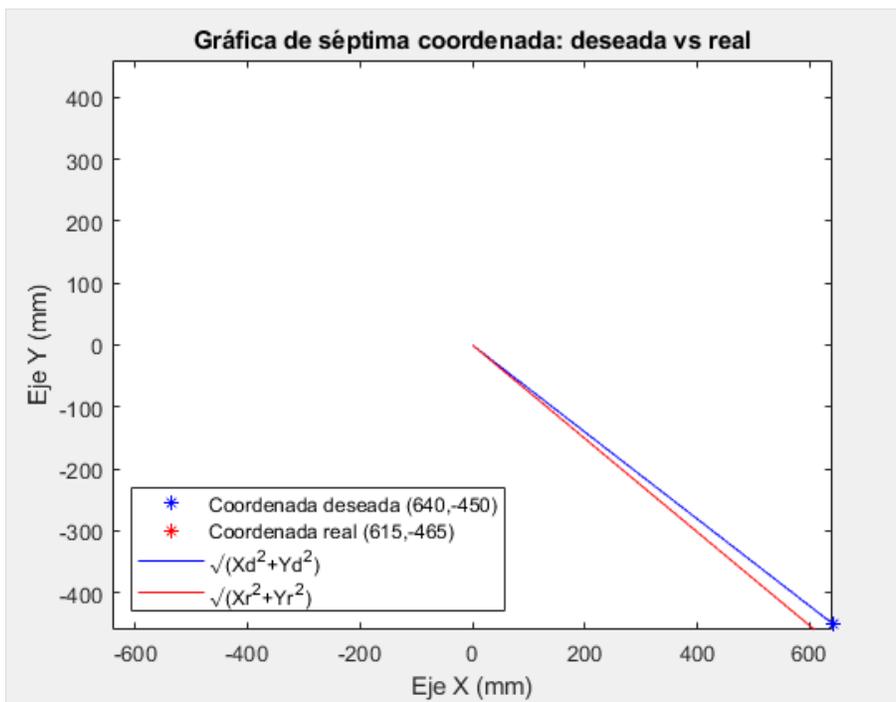


Figura 4.23 Gráfica de séptimo par de coordenadas

Finalmente se realizó el programa para graficar el módulo de cada vector de posición y el error de posición (según su módulo) como se muestra en la figura 4.24.

```

Figura424GraficaError.m x +
1   % Instituto Tecnológico de Puebla
2   % Alumna: Mena Huerta Claudia
3   % Gráfica de error de posición del sistema de Control en el CDPR
4   % 14-ago-19
5
6   clc;
7   clear all;
8
9   C1d = [0 320]; C1r = [0 295]; Mod1d = norm(C1d); Mod1r = norm(C1r);
      :
15  C7d = [640 -450]; C7r = [615 -465]; Mod7d = norm(C7d); Mod7r = norm(C7r);
16  C8d = [-640 -450]; C8r = [-615 -465]; Mod8d = norm(C8d); Mod8r = norm(C8r);
17
18  V = [1,2,3,4,5,6,7,8];
19  ModDes = [Mod1d, Mod2d, Mod3d, Mod4d, Mod5d, Mod6d, Mod7d, Mod8d];
20  ModReal = [Mod1r, Mod2r, Mod3r, Mod4r, Mod5r, Mod6r, Mod7r, Mod8r];
21  figure(1)
22  plot(V, ModDes, 'b*')
23  axis([0 9 0 800])
24  hold on
25  plot(V, ModReal, 'r*')
26  title('Gráfica de módulo de cada coordenada: deseada vs real')
27  xlabel('N° de coordenada')
28  ylabel('Módulo de cada coordenada')
29  legend({'Norma L2 de Coordenadas deseadas','Norma L2 de Coordenadas reales'},...
30         'Location','southwest')
31
32  Error1 = 100 - (Mod1r*100/Mod1d);
      :
39  Error8 = 100 - (Mod8r*100/Mod8d);
40  Error = [Error1, Error2, Error3, Error4, Error5, Error6, Error7, Error8];
41  figure(2)
42  bar(Error)
43  title('Gráfica porcentaje de error de posición')
44  xlabel('N° de coordenada')
45  ylabel('Porcentaje de error')
46  text(1, Error1, num2str(Error1), 'HorizontalAlignment', 'center',...
47       'VerticalAlignment', 'bottom')
      :
60  text(8, Error8, num2str(Error8), 'HorizontalAlignment', 'center',...
61       'VerticalAlignment', 'bottom')

```

Figura 4.24 Programa para graficar la norma Euclídea y el error de posición de cada coordenada en Matlab

La gráfica de la norma l^2 o norma euclídea para cada par de coordenadas se muestra en la figura 4.25. En ella se puede observar que en el punto cuya coordenada es (0, -450) no se presenta un error de posición, sin embargo, en las posiciones donde 'x' es diferente de cero o 'x' es mayor que 150, comienza a presentarse una desviación considerable de la posición deseada. Ejemplo de ello se presenta en las coordenadas número 1, 7 y 8, donde la longitud del vector de posición real de la plataforma es más corto comparado con el vector de posición deseada.

En el caso de la coordenada número 5 (600 mm, 300 mm) se presenta una pequeña variación entre las longitudes del vector de posición, con la diferencia de que el módulo de la posición real es ligeramente más grande que el módulo de la posición deseada.

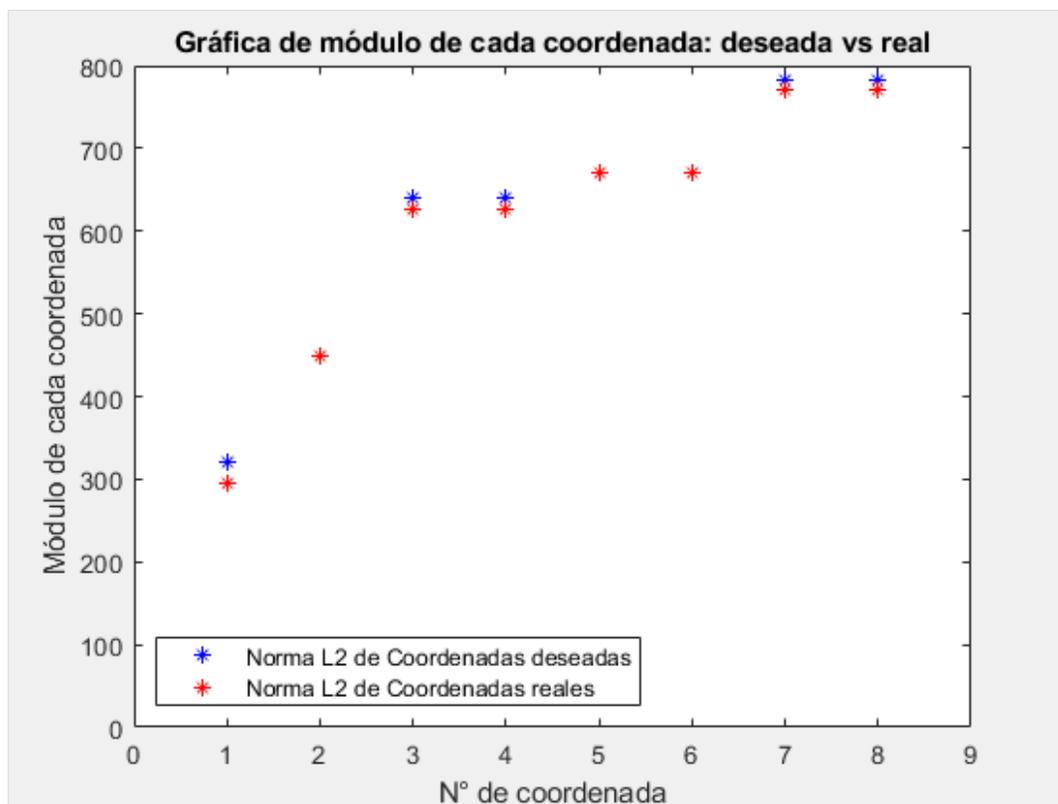


Figura 4.25 Gráfico del módulo de los vectores de posición deseados y reales

En la gráfica de porcentaje de error de posición de la plataforma móvil (véase figura 4.26) se consideró la siguiente relación del error:

$$error = posición_{referencia} - posición_{salida}$$

Es decir, si la posición real no alcanza a la posición de referencia, el error será positivo, en caso contrario, si la posición real está por encima de la posición de referencia, entonces se habla de la existencia de un error negativo.

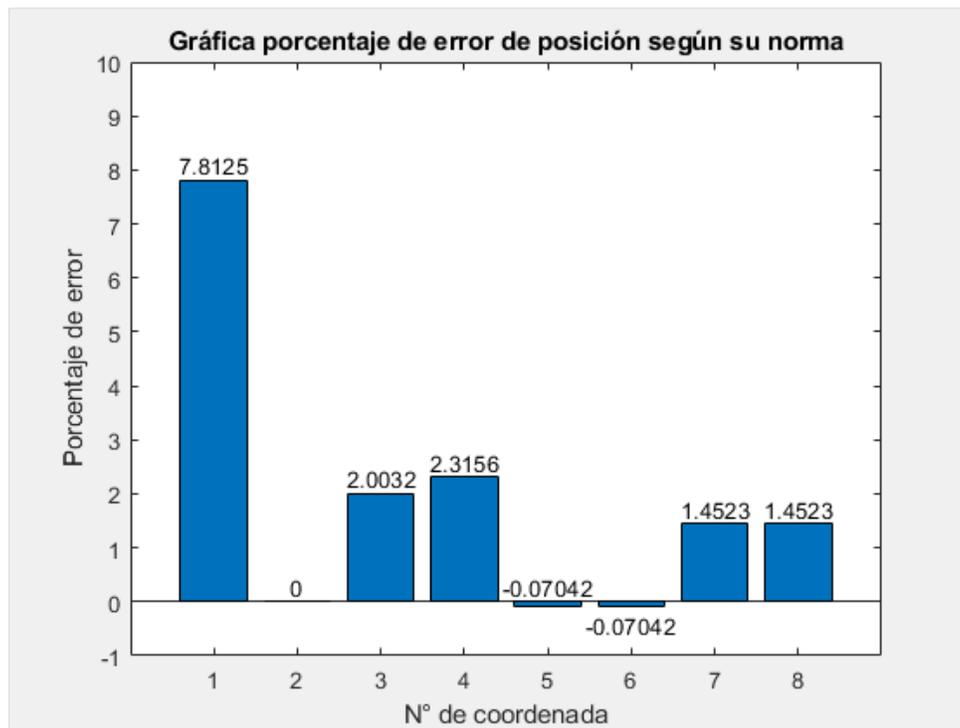


Figura 4.26 Gráfico del error de posición de la plataforma móvil

En la posición 2 (0, -450) no existe un error de posición, sin embargo, el mayor error obtenido entre la posición real contra la deseada se presenta en la primera coordenada (0, 320) y esto se debe a la configuración del sistema mecánico, el cual tiene acoplado unas poleas con tornillos, los cuales modifican el vector de la longitud de los cables l_1 y l_2 y la tensión de los cables no es distribuida de la misma forma (véase figura 4.27), sin embargo, esto no sucede cuando la plataforma se posiciona en las partes media y baja del espacio de trabajo.

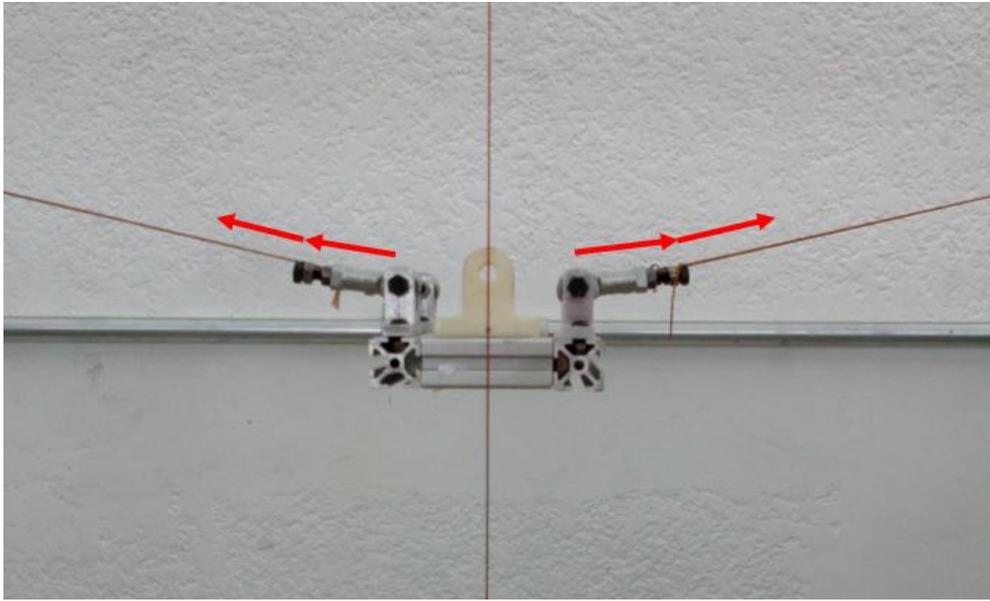


Figura 4.27 Error de posición en primera coordenada

En las figuras 4.28 y 4.29 se presenta las gráficas del error en la posición sobre el eje 'x' y en la posición sobre el eje 'y' respectivamente.

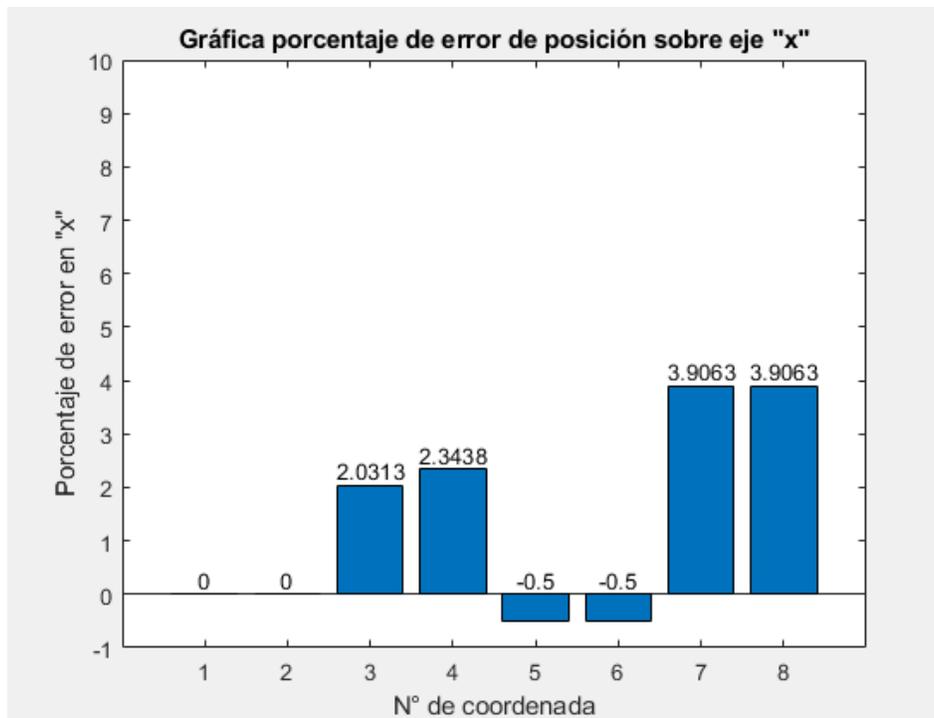


Figura 4.28 Gráfica porcentaje de error de posición en eje x

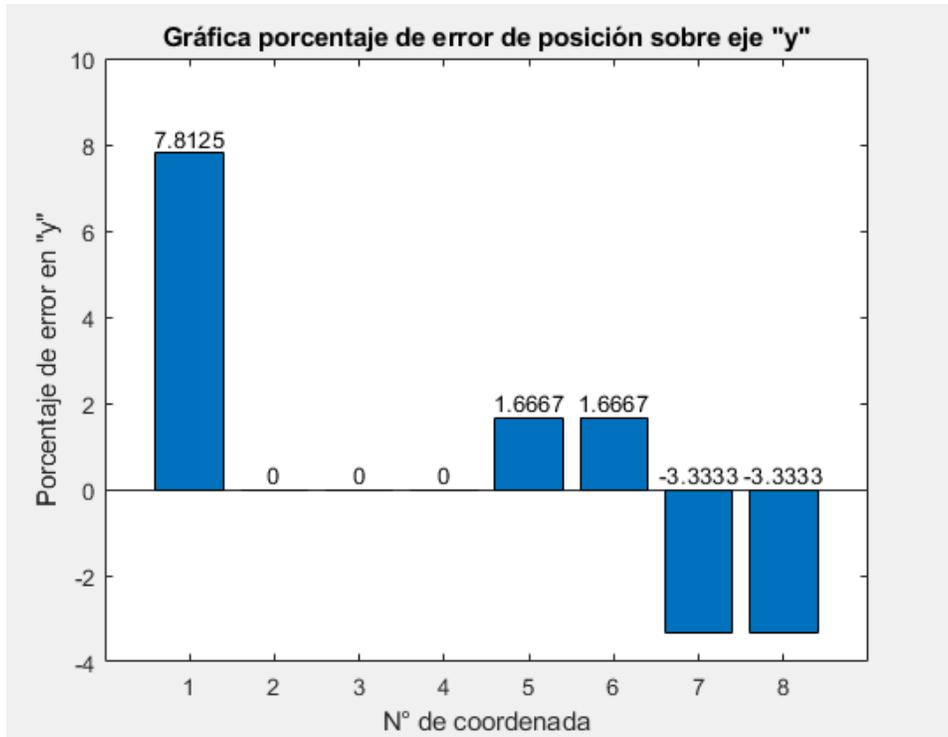


Figura 4.29 Gráfica porcentaje de error de posición en eje 'y'

Se observa que el mayor error sobre el eje 'x' es presentado en las últimas coordenadas, porque es donde existe una mayor inclinación de la plataforma móvil, por lo tanto, al ser el ángulo mucho mayor que cero ($\theta \gg 0$) presenta un error grande de posición; en el caso del error de posición sobre el eje 'y', el mayor error se presenta en la primera coordenada por el defecto mecánico del robot explicado anteriormente

Como se ha mostrado en las primeras pruebas, la plataforma móvil no consigue alinearse cuando $x \neq 0$; mientras el robot era manipulado en la recta donde $x = 0$ no existía rotación de la plataforma móvil sobre el eje 'z', sin embargo, fuera de esta línea, la plataforma comenzaba a inclinarse gradualmente. Este efecto se debe a que el robot es incompletamente restringido, pues su número de g.d.l. es mayor al número de cables que éste posee, por lo tanto, no se tiene una fuerza adicional que asegure un ángulo de inclinación $\theta = 0$ en cualquier punto del espacio de trabajo.

4.6 AJUSTES FINALES

De acuerdo al análisis derivado de las gráficas del error de posicionamiento de la plataforma móvil, se determina generar otras pruebas en el robot paralelo activado por cables con ajustes al modelo cinemático del robot; por consiguiente, el valor del ángulo de inclinación θ será considerado en el nuevo modelo tanto en la simulación en Matlab-Simulink, así como en el programa de control de movimiento en el PLC/MC Delta.

La mejora considera tomar como valores deseados (posición y orientación) los resultados reales obtenidos en las primeras pruebas del robot. El valor de l_{1Home} y l_{2Home} es el mismo porque en esa posición la plataforma no tiene ninguna orientación sobre el eje 'z'. Por consiguiente, ahora se hará uso de las ecuaciones 3.1 y 3.3 de la cinemática inversa del robot y los valores de las dimensiones del robot serán introducidos en las ecuaciones.

$$l_1 = \sqrt{Px^2 + Py^2 - Px(c \cos \theta + d \sin \theta - e) - Py(c \sin \theta - d \cos \theta + f) \dots}$$

$$\sqrt{\dots + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce \cos \theta - 2de \sin \theta + 2cf \sin \theta - 2df \cos \theta)}$$

La nueva ecuación 4.1 para la obtención de l_1 (cuando $\theta \neq 0$) será programada en la simulación de Matlab y en el código para el PLC Delta.

$$l_1$$

$$= \sqrt{Px^2 + Py^2 - Px(82 \cos \theta + 46 \sin \theta - 1480) - Py(82 \sin \theta - 46 \cos \theta + 900) \dots}$$

$$\sqrt{\dots + \frac{1}{4}(30009240 - 242720 \cos \theta - 136160 \sin \theta + 147600 \sin \theta - 82800 \cos \theta)}$$

Para el caso de l_2 se realiza el mismo procedimiento:

$$l_2 = \sqrt{Px^2 + Py^2 + Px(c \cos \theta - d \sin \theta - e) + Py(c \sin \theta + d \cos \theta - f) \dots}$$

$$\sqrt{\dots + \frac{1}{4}(c^2 + d^2 + e^2 + f^2 - 2ce \cos \theta + 2de \sin \theta - 2cf \sin \theta - 2df \cos \theta)}$$

La nueva ecuación 4.2 para la obtención de l_2 (cuando $\theta \neq 0$) será programada en la simulación de Matlab y en el código para el PLC Delta.

$$l_2$$

$$= \sqrt{Px^2 + Py^2 + Px(82 \cos \theta - 46 \sin \theta - 1480) + Py(82 \sin \theta + 46 \cos \theta - 900) \dots}$$

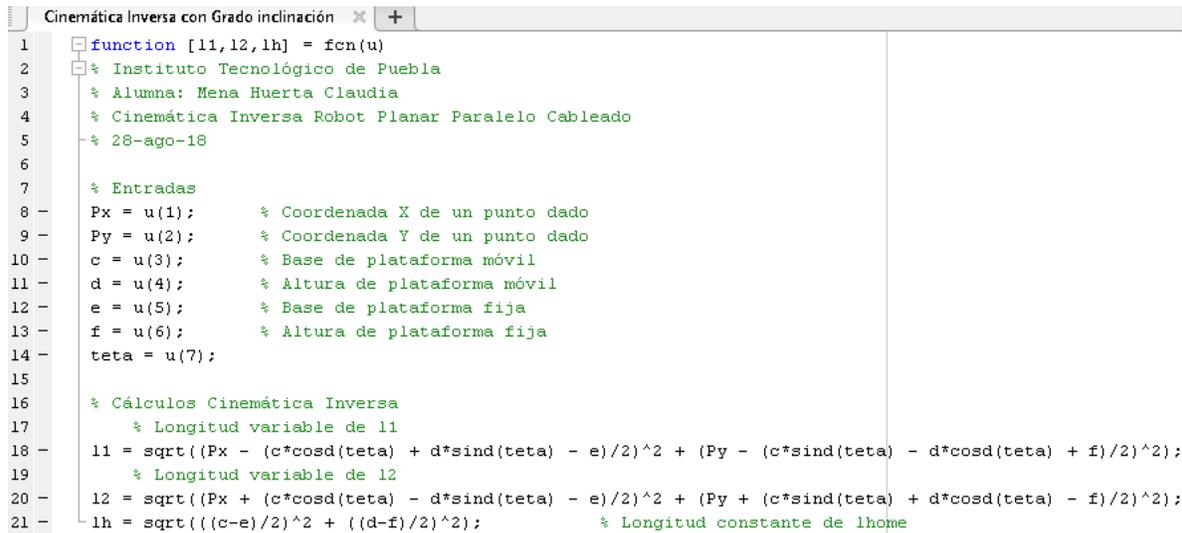
$$\sqrt{\dots + \frac{1}{4}(30009240 - 242720 \cos \theta + 136160 \sin \theta - 147600 \sin \theta - 82800 \cos \theta)}$$

La interpretación de las ecuaciones 4.1 y 4.2 es la siguiente: Los valores tal como seno de $\theta = 0$ eran anulados en las ecuaciones anteriores y el coseno de $\theta = 0$ no modificaba los valores constantes. Sin embargo, al declarar que $\theta \neq 0$ se conservan los términos de seno y coseno de toda la ecuación, porque existe un valor de θ que antes no era considerado.

Los términos donde aparece seno y coseno de θ no representan por sí solos una variación significativa, en cambio, la suma de todos esos términos genera una diferencia que debe considerarse, puesto que ahora alterará el valor de l_1 y l_2 y, por ende, cambiará la ubicación de la plataforma. De esta forma se espera lograr obtener una posición más cercana a la deseada.

4.7 PROGRAMA FINAL PARA SIMULAR

Para el último modelo virtual de la cinemática inversa del CDPR, se tomó en cuenta el grado de inclinación θ , razón por la cual se cambió la programación del bloque “Cinemática Inversa” en Matlab-Simulink, tal como se muestra en la figura 4.30.



```
1 function [l1,l2,lh] = fcn(u)
2 % Instituto Tecnológico de Puebla
3 % Alumna: Mena Huerta Claudia
4 % Cinemática Inversa Robot Planar Paralelo Cableado
5 % 28-ago-18
6
7 % Entradas
8 Px = u(1); % Coordenada X de un punto dado
9 Py = u(2); % Coordenada Y de un punto dado
10 c = u(3); % Base de plataforma móvil
11 d = u(4); % Altura de plataforma móvil
12 e = u(5); % Base de plataforma fija
13 f = u(6); % Altura de plataforma fija
14 teta = u(7);
15
16 % Cálculos Cinemática Inversa
17 % Longitud variable de l1
18 l1 = sqrt((Px - (c*cosd(teta) + d*sind(teta) - e)/2)^2 + (Py - (c*sind(teta) - d*cosd(teta) + f)/2)^2);
19 % Longitud variable de l2
20 l2 = sqrt((Px + (c*cosd(teta) - d*sind(teta) - e)/2)^2 + (Py + (c*sind(teta) + d*cosd(teta) - f)/2)^2);
21 lh = sqrt(((c-e)/2)^2 + ((d-f)/2)^2); % Longitud constante de lhome
```

Figura 4.30 Bloque cinemática inversa con grado de inclinación

Los bloques de programación “Conversión a grados y pulsos” y “Grado de inclinación sobre el eje ‘z’”, así como el modelo virtual del CDPR no presentan alguna modificación, por lo tanto, las variaciones del nuevo modelo y de los resultados de l_1 , l_2 , Px_{real} y Py_{real} serán causados por el bloque de “Cinemática inversa con Grado de inclinación”.

4.7.1 Simulaciones finales

A través del modelo modificado de la cinemática inversa generado en Matlab considerando el grado de inclinación θ , se realizaron diversas simulaciones ingresando puntos del plano ‘xy’. En la figura 4.31 se observan los resultados obtenidos de la simulación cuando se mueve la plataforma a la posición deseada 627.4 mm en ‘x’ y -14.57 mm en ‘y’, con un ángulo de inclinación de 44.52°.

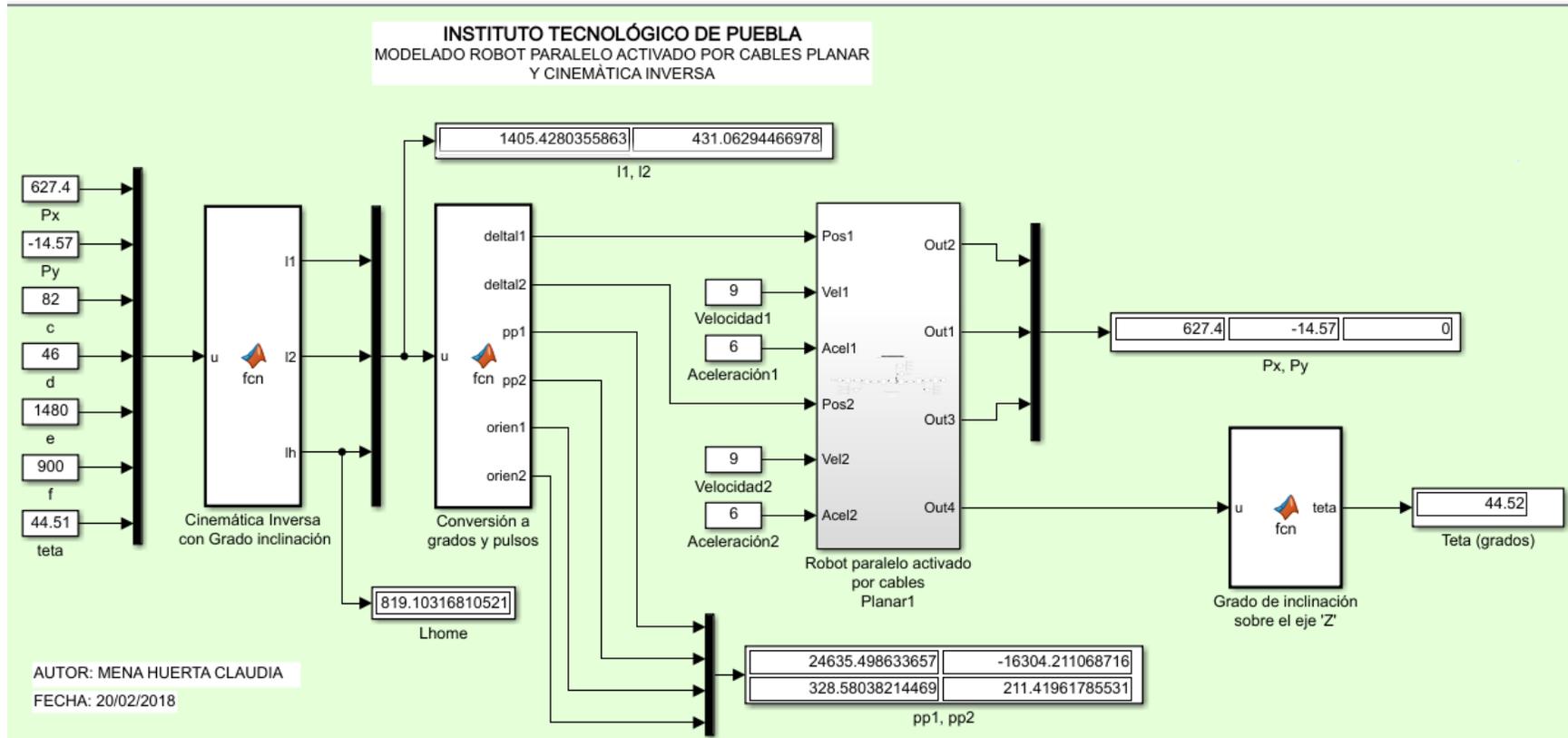


Figura 4.31 Simulación de coordenada (627.4, -14.57) con $\theta = 44.51^\circ$

Como se observa en la figura anterior, al considerar el valor medido con anterioridad θ , se observa que, al ingresar ese valor en el nuevo modelo cinemático de simulación, los valores de salida de la simulación P_x, P_y, θ son iguales a los valores de entrada (deseados). Por lo tanto, el nuevo modelo cinemático asegura ubicar la plataforma móvil en una posición más exacta con respecto al modelo cinemático inverso donde θ siempre vale cero.

4.8 RESULTADOS EXPERIMENTALES FINALES

De acuerdo a los resultados obtenidos en las simulaciones finales, se generó un nuevo programa en “ISPSOft” para la cinemática inversa en el PLC; con respecto a los programas para el MC y la HMI, éstos también se modificaron porque cambiaron los datos de entrada y aumentaron los datos durante la modificación del programa principal.

En los Anexos D – F se presentan los programas titulados “TrayectoriaAnguloTesis.isp” para la programación de la cinemática inversa en PLC, “TrayectoriaAnguloTesis.elc” para la programación del movimiento en MC y “TrayectoriaAnguloTesis.dps” para la programación de la interfaz usuario – máquina. También se presenta un Anexo L como archivo de vídeo titulado “Control de un Robot paralelo activado por cables_Mena Huerta Claudia” donde se presenta el sistema de control de posición del CDPR.

Se realizaron nuevas pruebas y mediciones correspondientes a la ejecución del nuevo programa. Por ejemplo, en la figura 4.32 se muestra una nueva prueba con las coordenadas deseadas (-627, -15) e ingresando el ángulo de inclinación -42° . Se tomaron las medidas correspondientes y se observó que la plataforma sí llega a la coordenada deseada. Las longitudes de los cables medidos sí corresponden a las longitudes calculadas en Matlab.

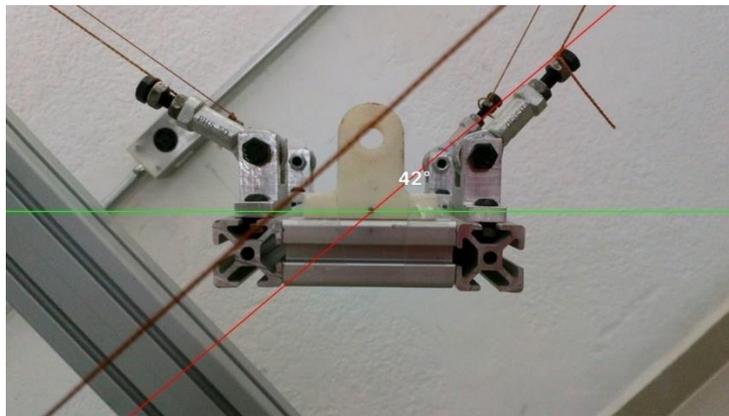


Figura 4.32 Nueva prueba en (-627, -15) y un ángulo de inclinación de -42°

En las figuras 4.33 y 4.34 se presentan los resultados para las coordenadas 603 mm en 'x' y 295 mm en 'y', así como -603 mm en 'x' y 295 mm en 'y' (posición contraria) y también se logró ubicar a la plataforma en la posición deseada correspondiente.

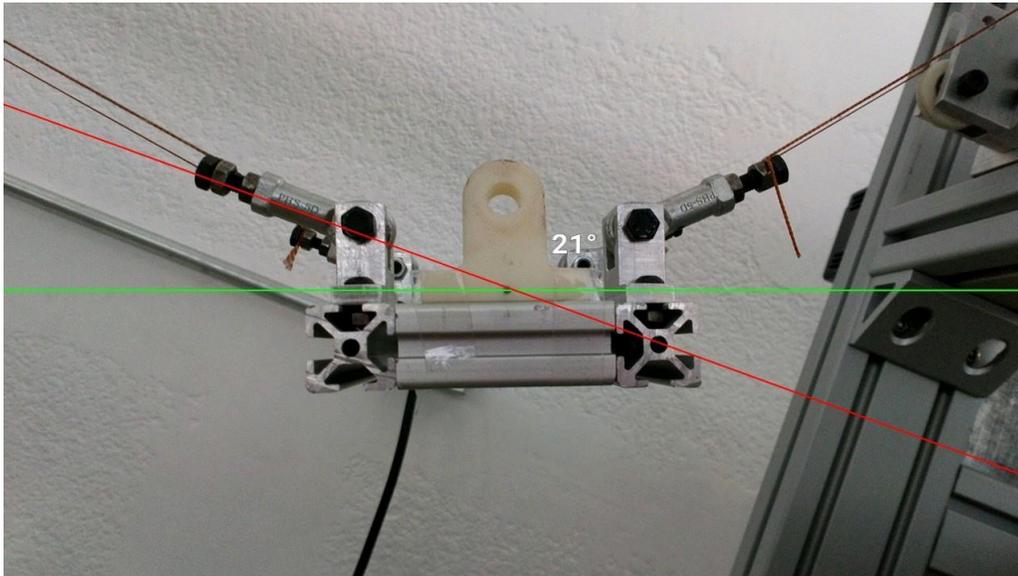


Figura 4.33 Nueva prueba en (603, 295) y un ángulo de inclinación de 21°

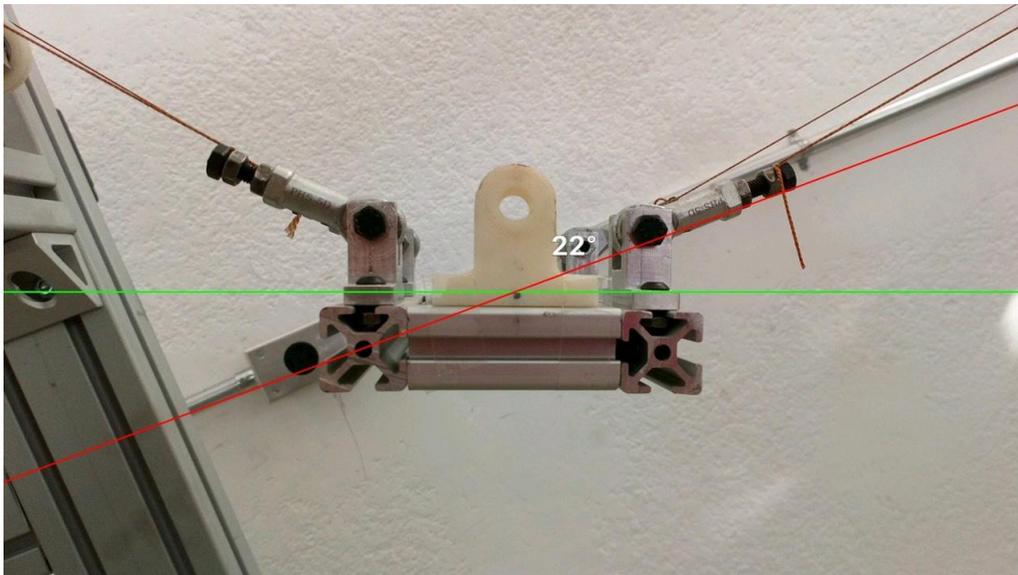


Figura 4.34 Nueva prueba en (-603, 295) y un ángulo de inclinación de -21°

4.9 TABLAS DE RESULTADOS FINALES

En la tabla 4.4 se muestran los resultados de la posición y orientación de la plataforma móvil obtenidos de la aplicación del nuevo programa de cinemática inversa donde el ángulo de inclinación θ es considerado en los cálculos.

Tabla 4.4 Resultados finales: Mediciones de puntos en el plano (Px, Py) y ángulo de inclinación de la plataforma

COORDENADAS							ÁNGULO		
N°	Deseadas		Matlab		Reales		Deseado	Matlab	Real
	Px (mm)	Py (mm)	Px (mm)	Py (mm)	Px (mm)	Py (mm)	θ	θ	θ
1	0	295	0.003	295	0	280	0	0.01	0
2	0	-450	0.2	-450	0	-450	0	0.004°	0
3	627	-15	623.9	-14.43	626	-15	42°	48.38°	42°
4	-627	-15	-629.5	-14.82	-627	-15	-42°	36.25°	-42°
5	603	295	602.6	295.5	603	294	21°	14.88°	21°
6	-603	295	-602.6	295.5	-603	295	-21°	14.77°	-22°
7	615	-465	620.4	-464.7	616	464	47°	40.26°	47°
8	-615	-465	-606.7	-463.9	-614.5	-465	-47°	57.65°	-47°

En la tabla 4.5 se presentan los resultados finales de la cinemática inversa para cada coordenada (valores de l_1 y l_2). Se observa que los resultados en Matlab en comparación con los del MC son iguales; la comparación contra las longitudes de cables reales es muy cercana, siendo el error medido en décimas. Este pequeño error se debe a las mediciones realizadas de forma indirecta, como ocurrió con los resultados iniciales.

Tabla 4.5 Resultados finales: Mediciones de longitudes de cables l_1 y l_2 en cada punto

Punto			LONGITUD DE CABLE					
N°	Deseadas		Matlab		PLC/MC		Reales	
	Px	Py	l_1 (mm)	l_2 (mm)	l_1 (mm)	l_2 (mm)	l_1 (mm)	l_2 (mm)
1	0	295	711.3543	711.3543	711.354	711.354	711	711
2	0	-450	1121.4856	1121.4856	1121.49	1121.49	1121	1122
3	627	-15	1404.0526	431.7248	1404.05	431.725	1404	432
4	-627	-15	431.7248	1404.0526	431.725	1404.05	432	1404
5	603	295	1304.9259	159.8852	1304.93	159.885	1305	160
6	-603	295	159.8852	1304.9259	159.885	1304.93	160	1305
7	615	-465	1606.3206	876.7531	1606.32	876.753	1606	876
8	-615	-465	876.7531	1606.3206	876.753	1606.32	877	1606

Finalmente, en la última tabla 4.6 se muestran las comparaciones de los pulsos o eventos que deben girar cada servomotor, correspondientes a los resultados finales medidos. En las primeras dos columnas se muestra el punto deseado; posteriormente en las columnas tres y cuatro se muestran los pulsos a girar medidos en Matlab-Simulink; en las últimas dos columnas se muestran los pulsos calculados por el PLC/MC Delta. Se observa que los resultados son iguales, por lo cual no existe un error de medición en Matlab ni en el PLC/MC Delta.

Tabla 4.6 Resultados finales: Mediciones de pulsos que debe girar cada servomotor en cada punto

Punto		EVENTOS				
N°	Deseadas		Matlab		PLC/MC	
	Px	Py	$\Delta Ppu1$ (pulsos)	$\Delta Ppu2$ (pulsos)	$\Delta Ppu1$ (pulsos)	$\Delta Ppu2$ (pulsos)
1	0	295	-4546.3639	-4546.3639	-4546.36	4546.36
2	0	-450	12758.7533	12758.7533	12758.8	-12758.8
3	627	-15	24681.4111	-16345.0768	24681.4	16345.1
4	-627	-15	-16345.0768	24681.4111	-16345.1	-24681.4
5	603	295	20498.8509	-27815.1001	20498.8	27815.1
6	-603	295	-27815.1001	20498.8509	-27815.1	-20498.8
7	615	-465	33215.9276	2432.4871	33215.9	-2432.49
8	-615	-465	2432.4871	33215.9276	2432.49	-33215.9

4.9.1 Análisis y discusión de resultados finales

Para determinar cuál fue el error de posición para cada par de coordenadas, primero se realizó un programa en Matlab donde se grafican las posiciones deseadas contra las posiciones reales en el plano cartesiano (coordenadas número 3, 6 y 7 presentadas en la tabla 4.4). El programa final para la obtención de las figuras 4.35 – 4.37 se presenta en el Anexo G “Código de programa para la obtención de las coordenadas de referencia contra las reales en Matlab”.

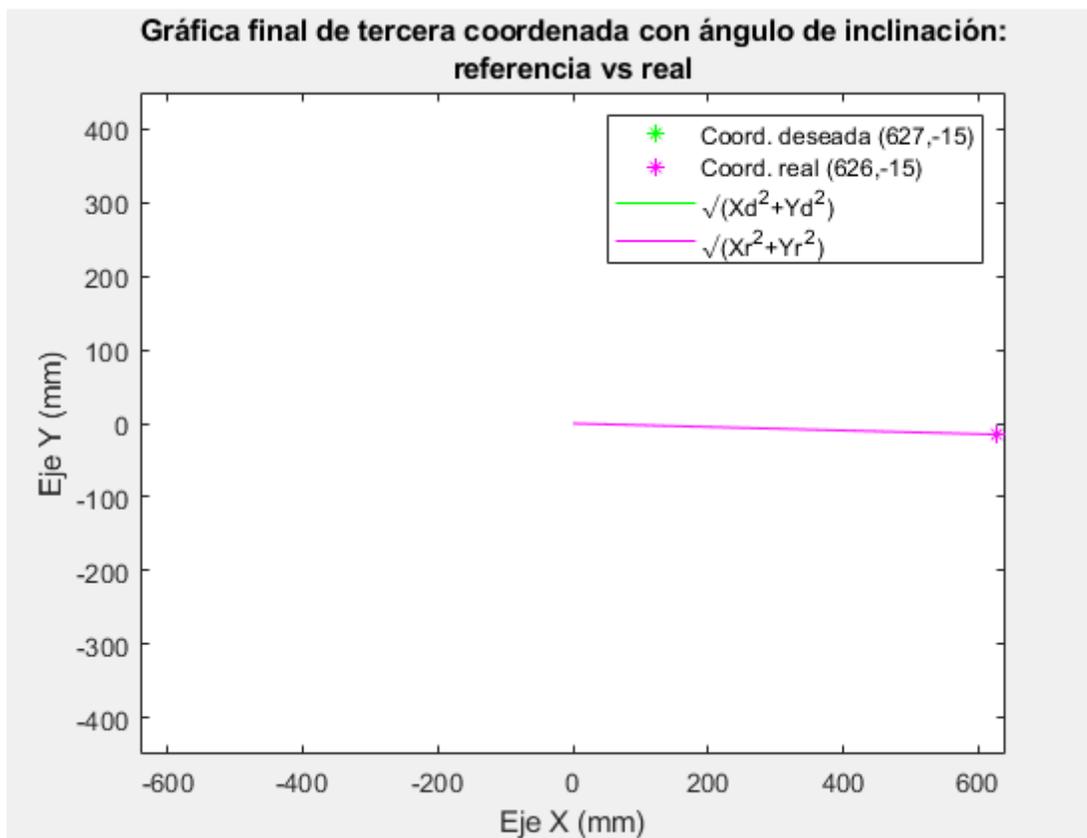


Figura 4.35 Gráfica de tercer par de coordenadas en prueba final

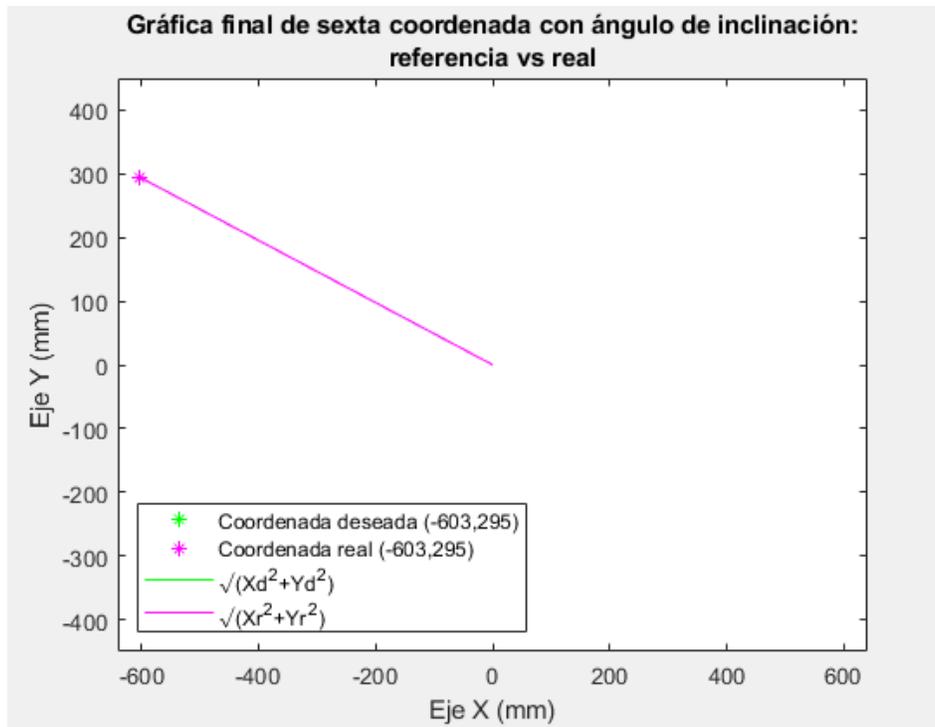


Figura 4.36 Gráfica de sexto par de coordenadas en prueba final

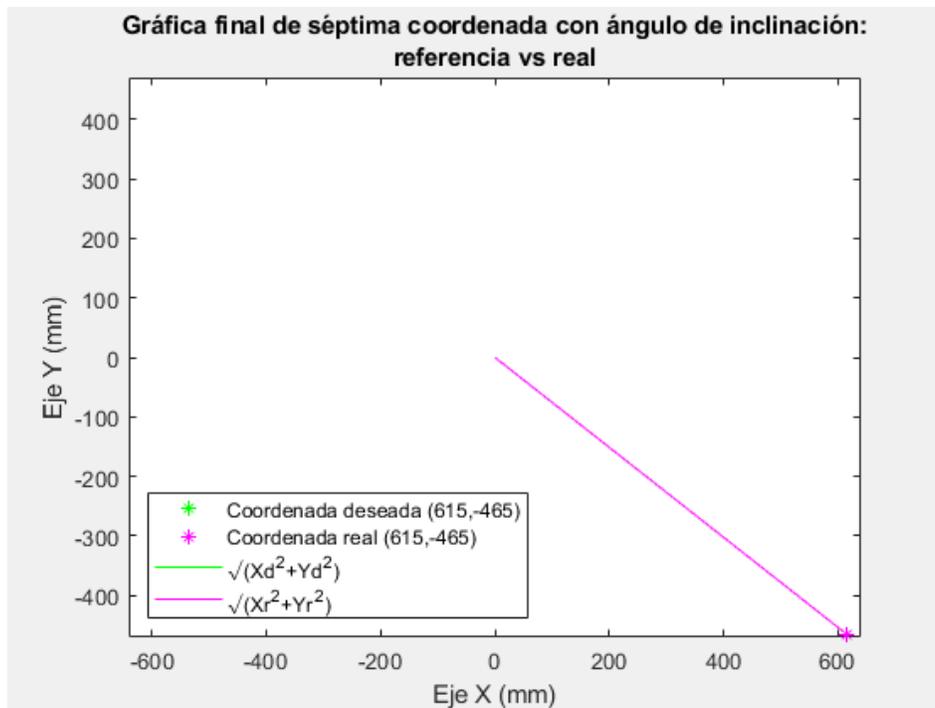


Figura 4.37 Gráfica de séptimo par de coordenadas en prueba final

Analizando los gráficos anteriores, se deduce un decremento del error de posición entre la coordenada de posición de referencia con la coordenada de posición real, ya que no se percibe alguna desviación en las tres posiciones de ejemplo. Sin embargo, se requiere analizar el error de posición para definir el comportamiento del robot con el nuevo algoritmo de control programado.

Por consiguiente, se graficó el módulo de cada vector de posición y se compararon los reales contra los deseados, como se muestra en la figura 4.38.

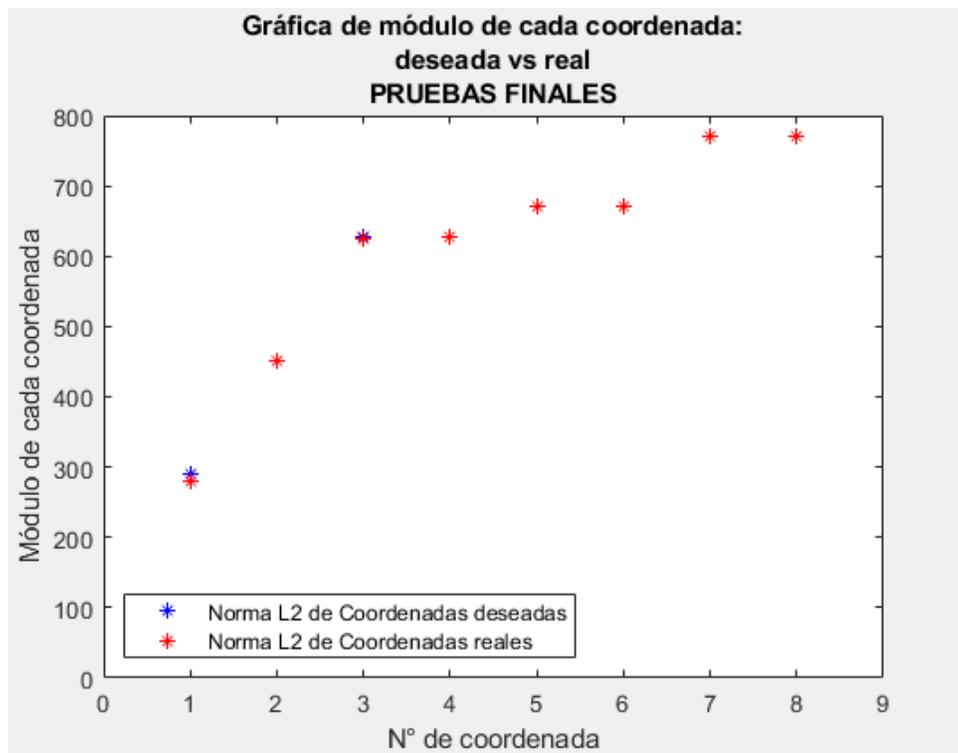


Figura 4.38 Gráfica prueba final de módulo de cada coordenada

De la gráfica anterior se puede inferir un error mayor en la primera coordenada, tal como ocurrió en la prueba inicial. Sin embargo, los errores de las demás posiciones han disminuido en gran medida.

En la figura 4.39 se muestra la gráfica del porcentaje de error de posición con respecto a la norma de los vectores de posición.



Figura 4.39 Gráfica prueba final porcentaje de error de posición según su módulo

De acuerdo a los datos graficados en la figura anterior, el mayor porcentaje de error está presente en la primera posición, es decir cuando se desea ubicar el centro de la plataforma móvil en la posición (0, 295). Ese error grande se presenta por la configuración mecánica de la plataforma móvil y los cables enlazados a través de los tornillos, los cuales modifican la dirección de los vectores de los cables l_1 y l_2 y, por consiguiente, se modifica la posición de la plataforma cuando la plataforma sube en el eje 'x'. No obstante, las posiciones número 2 – 8 muestran un mejor comportamiento comparado con el gráfico de la figura 4.26. Finalmente, en las gráficas 4.40 y 4.41 se muestra el error de posición con respecto al eje 'x' y al eje 'y'.

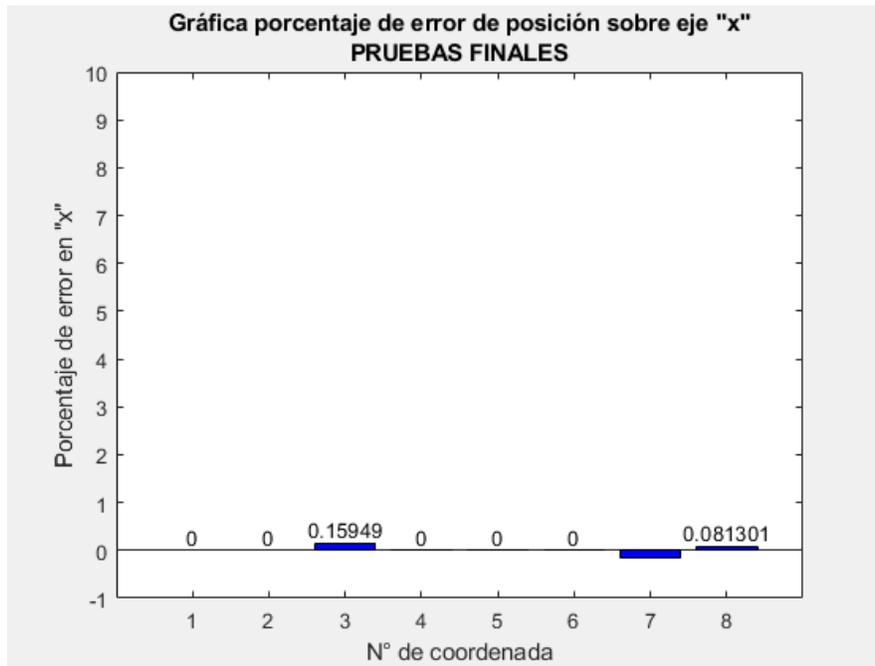


Figura 4.40 Gráfica prueba final porcentaje de error de posición en eje 'x'

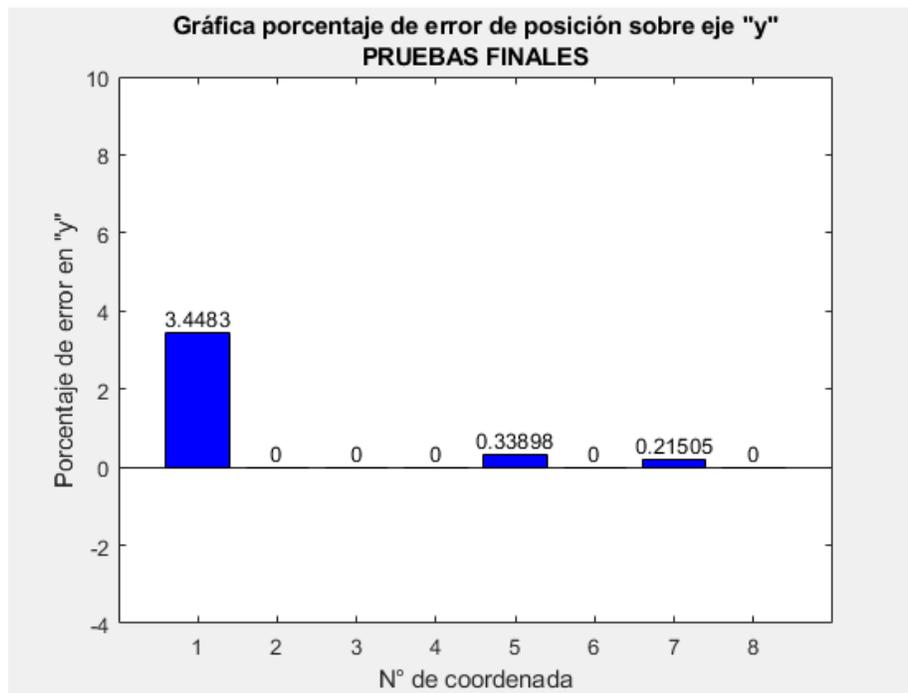


Figura 4.41 Gráfica prueba final porcentaje de error de posición en eje 'y'

Las gráficas anteriores presentan una mejoría comparadas con las gráficas de las figuras 4.28 y 4.29. En el caso del error de posición referente al módulo de los vectores de posición, se observa un cambio positivo en las últimas coordenadas, sin embargo, en el primer caso se muestra que pasó de un porcentaje de error de 7.8% a 3.4%, aunque todavía existe un error significativo.

Mientras que el error en las últimas posiciones pasó de 1.4% a un 0.05% de error. En la tabla 4.7 se muestra el promedio del porcentaje de error para los casos “módulo de vector de posición”, “posición en eje x” y “posición en eje y” y se comparan los valores de las primeras pruebas (sin considerar el ángulo de inclinación) contra los valores de las pruebas finales (considerando el ángulo de inclinación θ).

Tabla 4.7 Comparación de promedio de error de las pruebas iniciales contra las pruebas finales

	<i>Pruebas iniciales</i>	<i>Pruebas finales</i>
<i>Promedio de porcentaje de error de módulo de posición</i>	1.897%	0.4687%
<i>Promedio de porcentaje de error de posición en eje 'x'</i>	1.6484%	0.0504%
<i>Promedio de porcentaje de error de posición en eje 'y'</i>	2.2265%	0.5002%

Entonces, se puede deducir que el último algoritmo de control donde se toma en cuenta el ángulo de inclinación de la plataforma, reduce el error de posición de la plataforma en cualquier punto del espacio de trabajo del robot, mejorando así el sistema de control de posición del CDPR.

CAPÍTULO V CONCLUSIONES

5.1 CONCLUSIONES

Se comprendieron distintos conceptos teóricos en los primeros capítulos. De acuerdo a los conceptos estudiados, se concluye que un robot paralelo activado por cables es un dispositivo mecánico cuyas características combinadas de un robot paralelo convencional con un robot cableado, proporcionan mejoras como manejar un mayor espacio de trabajo, poseer estructuras ligeras comparadas con los manipuladores seriales y es bajo en su costo de producción.

El robot paralelo activado por cables también llamado por sus siglas en inglés CDPR, es un sistema mecánico de cadena cinemática cerrada conformado por una plataforma móvil (efector final) unida a una plataforma fija de mayores dimensiones a través de un número de cables, los cuales son accionados a través de motores permitiendo que los cables se enrolen o desenrolen modificando su longitud.

En el primer capítulo se presentaron los objetivos específicos y general del proyecto “Control de un robot activado por cables”, los cuales fueron analizados para determinar si éstos se efectuaron con éxito.

Por consiguiente, se investigó la teoría de robótica necesaria para el desarrollo del proyecto de tesis, por ejemplo, se indagó sobre las diversas clasificaciones de un robot paralelo cableado, así como se entendió la importancia de estudiar el sistema robótico para su posterior análisis cinemático. Se analizaron los diversos algoritmos de sistema de control de un robot, optando por un algoritmo de control basado en la cinemática del robot.

Se clasificó el CDPR del tema de tesis de la siguiente forma: Robot cableado de cadena cinemática de lazo cerrado (paralelo) con un movimiento rotacional sobre el eje 'z' y dos movimientos traslacionales, un desplazamiento en el eje 'x' y otro en el

eje 'y', por lo tanto, se afirma que el robot tiene 3 g.d.l. Posee una actuación eléctrica dada por los servomotores y una transmisión por medio de dos dispositivos sin fin.

A través de un diagrama del robot donde se muestran los distintos vectores de posición que influyen en la plataforma móvil (revisar figura 3.12) se logró obtener el modelo cinemático inverso, directo y diferencial del robot, así como también se obtuvo la singularidad para este CDPR.

Se diseñó un algoritmo de posición con retroalimentación por encoder basándose en el modelo de la cinemática inversa del robot y las trayectorias coordinadas o isócronas para lograr posicionar y orientar la plataforma móvil en la ubicación dada por el usuario.

Tomando como referencia la metodología propuesta en el Capítulo I (revisar figura 1.2), se implementó de forma exitosa cada paso de la metodología y fue una guía para el desarrollo del proyecto de tesis.

A través del diagrama esquemático del robot paralelo cableado y de su cinemática inversa, se logró modelar el robot en una plataforma virtual y se consiguió ejecutar simulaciones de diferentes posiciones de la plataforma móvil del robot en el plano 'x, y'. El modelo y la simulación virtual se generó en Matlab – Simulink.

De acuerdo a la metodología, se integró la arquitectura de comunicación en el robot, para poder enlazar los distintos dispositivos de control con un comando principal (computadora) a través de la red CANopen y la red Ethernet tomando como referencia la Norma Oficial Mexicana NOM-001-SEDE-2012. Se efectuaron las tareas de comunicación como: carga del programa al PLC y MC, lectura de parámetros del servo drive, operación del programa a través de la HMI y monitoreo en línea del proceso de ejecución del PLC y MC.

Por otra parte, se logró aplicar el sistema de control al CDPR, programando el algoritmo de control en dispositivos industriales marca Delta y se obtuvieron tres programas finales: un programa de cinemática inversa implementado en "ISPSoft"

para el Controlador lógico programable; un programa de comandos de movimiento en “CANopen Builder” para el dispositivo Control de movimiento; finalmente un programa con comandos de operación en “DOPSoft” para la Interfaz usuario-máquina.

Se definieron puntos de interés y se realizaron dos conjuntos de pruebas: las pruebas iniciales y las finales. De acuerdo al análisis de los primeros resultados, se realizaron ajustes al algoritmo de control por cinemática y finalmente se generaron pruebas finales obteniendo mejoras en los resultados con respecto al error de posición de la plataforma.

Evaluando el sistema de control basado en la cinemática del robot, se determinó que este método de control es un poco complicado para este tipo de robots; puesto que, al ser un robot de tipo incompletamente restringido, se complica la implementación por el faltante de uno o más cables que ejerzan una fuerza complementaria, logrando posicionar la plataforma en una posición sin rotación alguna.

A pesar de la configuración mecánica del robot, se implementó y evaluó satisfactoriamente el sistema de control de posición basado en la cinemática del robot y con retroalimentación por encoder en el robot paralelo activado por cables, logrando posicionar el centro de la plataforma móvil en la coordenada de referencia dada por el usuario.

5.2 PROBLEMAS ENCONTRADOS

Las dificultades presentadas durante el proceso del proyecto de tesis “Control de un robot activado por cables” fueron las siguientes:

Al adaptar los elementos mecánicos del robot, como son los servomotores, los cables, entre otros, se presentaron dificultades, por ejemplo, en la unión entre el eje del servomotor con el cople ubicados en el extremo superior derecho del robot, ya que al cople le falta unos milímetros para encajar correctamente con la flecha, por lo tanto, se presenta un pequeño juego mecánico entre estos dos elementos. Por otra parte, al integrar los cables del robot, se tuvo que realizar de forma minuciosa para que la plataforma no tuviera ninguna desalineación en su posición cero, idealmente la plataforma no debe tener inclinación alguna, sin embargo, por aspectos mecánicos no se logra esa alineación ideal por completo.

Uno de los tambores (posición derecha) también presenta un pequeño defecto mecánico, ya que está conformado por la unión de dos semicilindros. Así mismo, la plataforma tiene en sus extremos superiores unos tornillos acoplados, los cuales modifican el comportamiento de su cinemática lo y esto no es conveniente.

Otro aspecto muy importante que complicó el desarrollo del proyecto, fue que no se pudo deshabilitar el lazo de control de cada servo drive, ya que su arquitectura de control es cerrada y no permite modificar ese lazo para poder integrar algún otro algoritmo de control diferente al propio del servo. El mismo servo drive tiene un sistema que detecta cuando se quiere modificar la arquitectura de control, bloqueando su funcionamiento y no permitiendo que se haga alguna variación. La justificación de los desarrolladores de este dispositivo es que al ser un servo drive de aplicación industrial, éste no tiene la opción para ser modificado en su estructura de control. Finalmente, el último problema encontrado (explicado en el capítulo 4), es la presencia de un desnivel en cualquier posición donde $x \neq 0$, puesto que idealmente se esperaba que en todo punto del espacio de trabajo no existiera un ángulo de inclinación sobre el eje ‘z’.

5.3 TRABAJO FUTURO

Como trabajo futuro se propone diseñar una arquitectura del robot con un driver cuyo lazo interno de control pueda ser modificado sin dificultad alguna y motores de corriente alterna que puedan ser comunicados fácilmente con estos drivers.

Posteriormente, si lo anterior se cumple, se requiere obtener el modelo dinámico del robot para poder generar un algoritmo de control basado en la dinámica, por ejemplo, algún tipo de control visto en el Capítulo II (PI, PD, PID) o bien, un control cuya dificultad sea mayor como un control tipo robusto. Así, podrá implementarse en conjunto con el control basado en la cinemática del robot.

Con un diagrama de control cuya arquitectura pueda ser modificada, también se podrá detectar el torque de cada motor y generar ecuaciones dinámicas donde las tensiones de los cables estén asociadas con otras variables del motor y así poder delimitar el espacio de trabajo del robot conforme al modelo dinámico.

Por otra parte, se plantea modificar el número de cables del robot y de esa forma ya no se tendrá un robot incompletamente restringido, permitiendo tener un mejor control de posición sobre la plataforma móvil.

5.4 PUBLICACIONES

Se realizaron dos artículos los cuales ya fueron aceptados y/o publicados en revistas indexadas, con el propósito de compartir los trabajos que se realizaron durante la estancia en el posgrado.

El primer artículo tiene por nombre “Desarrollo de una arquitectura del sistema de actuación de un robot paralelo activado por cables”, fue aceptado y publicado por la revista “UTCJ Theorema Revista Científica” No. 9 Julio – Dic 2018, con ISSN: 2448-7007 e indizada en Latindex.

El segundo artículo es “Modelado y simulación de un robot planar paralelo activado por cables”, fue aceptado por la revista “Programación Matemática y Software PROGMAT” con ISSN: 2007-3283 e indizada en Latindex; además este artículo se presentó en el 12° Congreso Internacional de Cómputo en Optimización y Software – CICOS 2018 con Sede en Jiutepec, Morelos.

Los artículos publicados, así como el póster presentado en el Congreso, se muestran en los Anexos I, J y K respectivamente.

REFERENCIAS

- Álvarez, M. (2004). *Controladores lógicos*. Barcelona: S.A. Marcombo.
- Astrom, K. J., & Tore, H. (2009). *Control PID Avanzado*. (S. Dormido, Trans.) Madrid: Pearson Educación.
- Barrientos, A., Peñín, L. F., Balaguer, C., & Aracil, R. (2007). *Fundamentos de robótica* (Segunda ed.). Madrid: McGraw-Hill.
- Barry, N., Fisher, E., & Vaughan, J. (2016). Modeling and control of a cable-suspended robot for inspection of vertical structures. *Journal of Physics: Conference Series*, 744, 1-8. doi:10.1088/1742-6596/744/1/012071
- Bruckmann, T., & Pott, A. (2013). *Cable-Driven Parallel Robots* (Vol. 12). Berlin: Springer-Verlag Berlin Heidelberg. doi:10.1007/978-3-642-31988-4
- Daneri, P. A. (2008). *PLC. Automatización y control industrial*. Buenos Aires: Editorial Hispano Americana S.A.
- Delta Electronics, Inc. (2012, August 17). DVP_MC Bus-Type Multi-Axis Motion Controller Operating Manual. Taoyuan, Taiwan. doi:DVP-0191420-03
- Delta Electronics, Inc. (2014, Febrero 6). Delta AC Servo Drive ASDA-A2 Series. Taoyuan, Taiwan.
- Delta Electronics, Inc. (2014, Diciembre 12). *Delta High Resolution AC Servo Drive for Network Communication Applications. ASDA-A2 Series User Manual*. Taoyuan, Taiwan: IABU Headquarters.
- Delta Electronics, Inc. (2014, Agosto 29). DVP-ES2/EX2/SS2/SA2/SX2/SE&TP Operation Manual - Programming. Taoyuan, Taiwan.
- Gagliardini, L., Caro, S., Gouttefarde, M., Wenger, P., & Girin, A. (2014, August 15). A Reconfigurable Cable-Driven Parallel Robot for Sandblasting and Painting of Large Structures. *Mechanisms and Machine Science*, 32, 275-291. doi:10.1007/978-3-319-09489-2_20
- Guerrero, V., Yuste, R., & Martínez, L. (2010). *Comunicaciones industriales*. Alfaomega.
- Hernández, R. (2010). *Introducción a los sistemas de control: Conceptos, aplicaciones y simulación con MATLAB*. México: Prentice Hall.
- IFR International Federation of Robotics. (2019). *World Robotics 2019 edition. Executive Summary World Robotics 2019 Industrial Robots*. From IFR International Federation of Robotics:
<https://ifr.org/downloads/press2018/Executive%20Summary%20WR%202019%20Industrial%20Robots.pdf>
- Instituto Schneider Electric de Formación. (2008, Noviembre 11). Manual de formación CANopen. Bac de Roda, Barcelona, España.

- Jazar, R. N. (2010). *Theory of Applied Robotics Kinematics, Dynamics and Control* (2nd ed.). New York: Springer. doi:10.1007/978-1-4419-1750-8
- Khosravi, M. A., & Taghirad, H. D. (2014, March 4). Robust PID control of fully-constrained cable driven parallel robots. *Mechatronics*, 24(2), 87-97. doi:10.1016/j.mechatronics.2013.12.001
- Lahouar, S., Ottaviano, E., Zeghoul, S., Romdhane, L., & Ceccarelli, M. (2009, November 1). Collision free path-planning for cable-driven parallel robots. *Robotics and Autonomous Systems*, 57(11), 1083-1093. doi:10.1016/j.robot.2009.07.006
- Liu, P., Qiu, Y., Su, Y., & Chang, J. (2014, August 4). On the Minimum Cable Tensions for the Cable-Based Parallel Robots. *Hindawi Journal of Applied Mathematics*, 1-8. doi:http://dx.doi.org/10.1155/2014/350492
- Lynch, K. M., & Park, F. C. (2017). Kinematics of closed chains. In K. M. Lynch, & F. C. Park, *Modern Robotics: Mechanism, Planning and, Control* (pp. 245-261). Cambridge: Cambridge University Press.
- Massiero, S., Carraro, E., Ferraro, C., Gallina, P., Rossi, A., & Rosati, G. (2009, November 1). Upper Limb Rehabilitation Robotics After Stroke: A Perspective from the University of Padua, Italy. *Journal of rehabilitation medicine: official journal of the UEMS European Board of Physical and Rehabilitation Medicine*, 41, 981-985. doi:10.2340/16501977-0404
- Merlet, J.-P. (2016). On the workspace of suspended cable-driven parallel robots. *IEEE International Conference on Robotics and Automation* (pp. 841-846). Stockholm: IEEE. doi:10.1109/ICRA.2016.7487214
- Rodríguez, M., Baradat, C., Michelin, M., Olmos, J., Cañada, J., Gouttefarde, M., . . . Palomino, M. (2015, Septiembre 22). Validación del prototipo de robot de cables CoGiRo para tareas de manipulación y ensamblaje de piezas complejas de alto valor económico. *Interempresas Metalmecánica*, 1-15. From <http://www.interempresas.net/MetalMecanica/Articulos/144648-Validacion-prototipo-robot-cables-CoGiRo-tareas-manipulacion-ensamblaje-piezas-complejas.html>
- Rushton, M., & Khajepour, A. (2018, July). Transverse Vibration Control in Planar Cable-Driven Robotic Manipulators. *Cable-Driven Parallel Robots. Mechanisms and Machine Science*, 243-253. doi:10.1007/978-3-319-61431-1_21
- Siciliano, B., Sciavicco, L., Villani, L., & Oriolo, G. (2009). *Robotics. Modelling, Planning and Control*. London: Springer. doi:10.1007/978-1-84628-642-1
- Taghirad, H. D. (2013). *Parallel Robots. Mechanics and Control*. Boca Raton: CRC Press.

ANEXOS

ANEXO A

ESPECIFICACIONES DEL SERVOMOTOR

220V Series (Delta Electronics, Inc., 2014).

Modelo: ECMA Series	EΔ13 10
Potencia nominal de salida (kW)	1.0
Torque nominal (N-m) *1	4.77
Torque máximo (N-m)	14.32
Velocidad nominal (r/min)	2000
Velocidad máxima (r/min)	3000
Corriente nominal (A)	5.6
Corriente máxima (A)	16.8
Potencia nominal (kW/s)	27.1
Momento de inercia de rotor (x10 ⁻⁴ kg-m ²) (sin freno)	8.41
Constante de tiempo mecánica (ms)	1.51
Par constante-KT (N-m/A)	0.85
Voltaje constante-KE (mV/(r/min))	31.9
Resistencia de armadura (Ohm)	0.47
Inductancia de armadura (mH)	5.99
Constante de tiempo eléctrica (ms)	12.88

*Los valores de par de torsión son valores continuos permitidos a 0-40 ° C de temperatura ambiente cuando se conectan con los tamaños de disipadores de calor: 400mm x 400mm x 20mm. Aluminio F100.

Continuación Anexo A.

Modelo: ECMA Series	EΔ13 10
Resistencia de aislamiento	100MΩ, DC 500V
Fuerza de aislamiento	AC 1500 V, 60 segundos
Peso (kg) (sin freno)	7
Peso (kg) (con freno)	8.4
Carga máxima del eje radial (N)	490
Carga máxima del eje de empuje (N)	98
Potencia nominal (kW/s) (con freno)	24.9
Momento de inercia de rotor ($\times 10^{-4}$ kg-m ²) (con freno)	9.14
Constante de tiempo mecánica (ms) (con freno)	1.64
Par de retención del freno [Nt-m (min)]	10.0
Consumo de energía de freno (a 20°C) [W]	19.0
Tiempo de liberación del freno [ms (Máx.)]	10
Tiempo de frenado [ms (Máx.)]	70
Grado de vibración (μ m)	15
Temperatura de operación (°C)	0°C a 40°C
Humedad de operación	20 a 90% RH (sin condensación)
Capacidad de vibración	2.5G

ANEXO B

ESPECIFICACIONES DEL SERVO DRIVE

220V Series (Delta Electronics, Inc., 2014).

Modelo: ASDA-A2 Series		1kW
		10
Fuente de alimentación	Fase/Voltaje	Trifásico / Monofásico 220VCA
	Rango de voltaje permisible	Trifásico / Monofásico 200 ~ 230VCA
	Corriente continua de salida	7.3 adjustable-rate mortgage
Sistema de enfriamiento		Ventilador de refrigeración
Resolución de encoder/ Resolución de realimentación		Incremental: 20-bit (1280000 p/rev) Absoluto: 17-bit
Control del circuito principal		Control SVPWM (Modulación de ancho de pulso de espacio vectorial)
Modos de sintonización		Auto/Manual
Freno dinámico		Incorporado
Modo Control de Posición	Fuente de comandos	Tren de pulsos externos / Parámetros internos
	Operación de límite de torque	Ajustar por parámetros
	Compensación de avance	Ajustar por parámetros
	Comando de entrada analógica	Rango de voltaje
Resistencia de entrada		10KΩ
Constante de tiempo		2.2 μs

ANEXO C

ARQUITECTURA ELÉCTRICA DEL SISTEMA DE CONTROL

F26_001

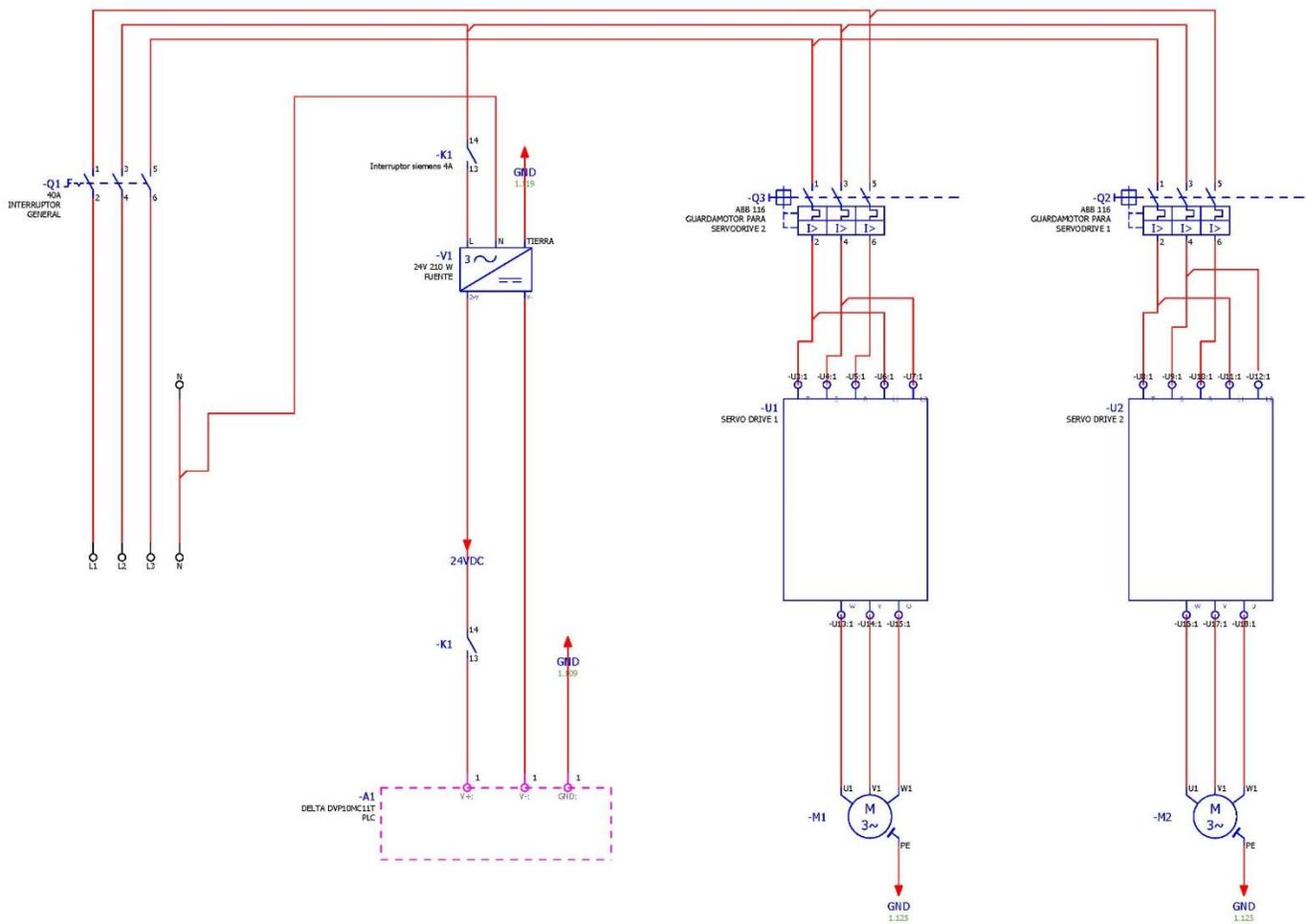


EPLAN Software & Service
 GmbH & Co. KG
 An der alten Ziegelei 2
 40789 Monheim am Rhein
 Tel +49 (0)2173 - 39 64 - 0

Empresa/cliente			
Descripción de proyecto	Plantilla de proyecto con estructura de identificación según la norma IEC: Estructura de página con instalación y lugar de montaje		
Número de diseño	IEC_tpl001		
Comisión	EPLAN		
Fabricante (empresa)	EPLAN Software & Service GmbH & Co. KG		
Circuito	Proyecto de ejemplo EPLAN		
Nombre de proyecto	Tablero_electrico		
Producto			
Tipo			
Lugar de instalación			
Responsable del proyecto			
Particularidad de pieza			
Creado	27/05/2018		
Modificado	28/05/2018	de (abreviatura) REENIISS	Número de páginas 2

Cambio	Fecha	Nombre	EPLAN	EPLAN Software & Service GmbH & Co. KG	Cambio	27/05/2018	PORTADA	ANL
			Tablero_electrico	An der alten Ziegelei 2		REENIISS		DBT
			Plantilla de proyecto con estructura de identificación según la norma IEC:		DBT		Número de diseño	1
							Responsable	
							Hoja	1
							Página	1/2

106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129



Cambio	Fecha	Nombre	EPLAN	EPLAN Software & Service GmbH & Co. KG	Cambio	28/05/2018	CTO PRINCIPAL		ANL
			Tablero_electrico	An der alten Ziegelei 2	SCP	REENISS	Número de diseño	Responsable	Hoja 1
			Plantilla de proyecto con estructura de identificación según norma IEC:				Estructura de página con instalación y lugar de montaje		Página 2/2

ANEXO D

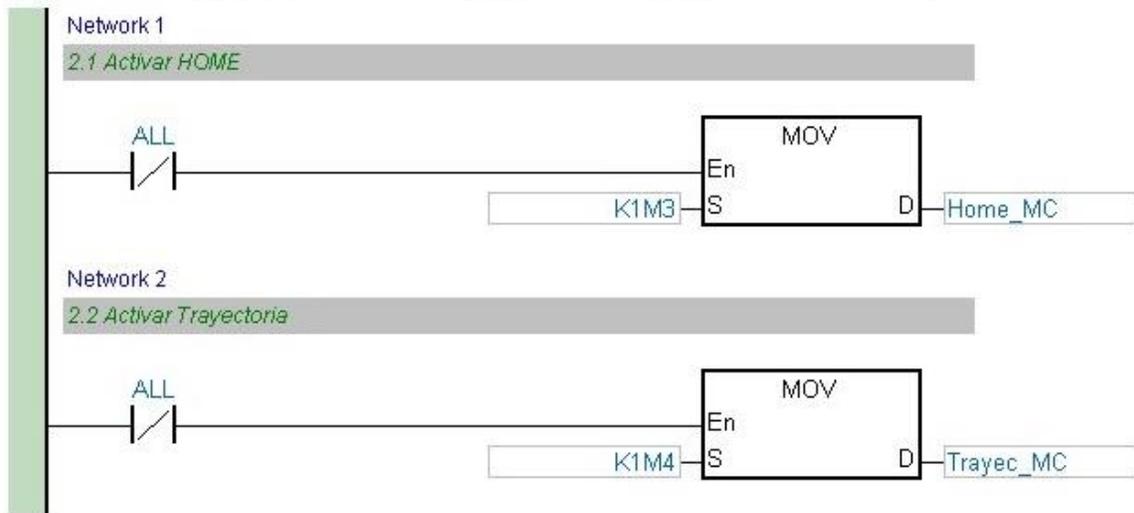
CÓDIGO DEL PROGRAMA CINEMÁTICO EN ISPSOFT

Global Symbols

Declaration Type	Identifiers	Address	Type	Initial Value	Identifier Comment
VAR	ALL	M10	BOOL	FALSE	
VAR	ResetAll_HMI	M7	BOOL	FALSE	
VAR	L1a	D6258	REAL	0.0	
VAR	L2a	D6260	REAL	0.0	
VAR	L1b	D6262	REAL	0.0	
VAR	L2b	D6264	REAL	0.0	
VAR	L1c	D6266	REAL	0.0	
VAR	L2c	D6268	REAL	0.0	
VAR	L1d	D6270	REAL	0.0	
VAR	L2d	D6272	REAL	0.0	
VAR	L1e	D6274	REAL	0.0	
VAR	L2e	D6276	REAL	0.0	
VAR	Puu1a_MC	D6300	REAL	0.0	
VAR	Puu2a_MC	D6302	REAL	0.0	
VAR	Puu1b_MC	D6304	REAL	0.0	
VAR	Puu2b_MC	D6306	REAL	0.0	
VAR	Puu1c_MC	D6308	REAL	0.0	
VAR	Puu2c_MC	D6310	REAL	0.0	
VAR	Puu1d_MC	D6312	REAL	0.0	
VAR	Puu2d_MC	D6314	REAL	0.0	
VAR	Puu1e_MC	D6316	REAL	0.0	
VAR	Puu2e_MC	D6318	REAL	0.0	

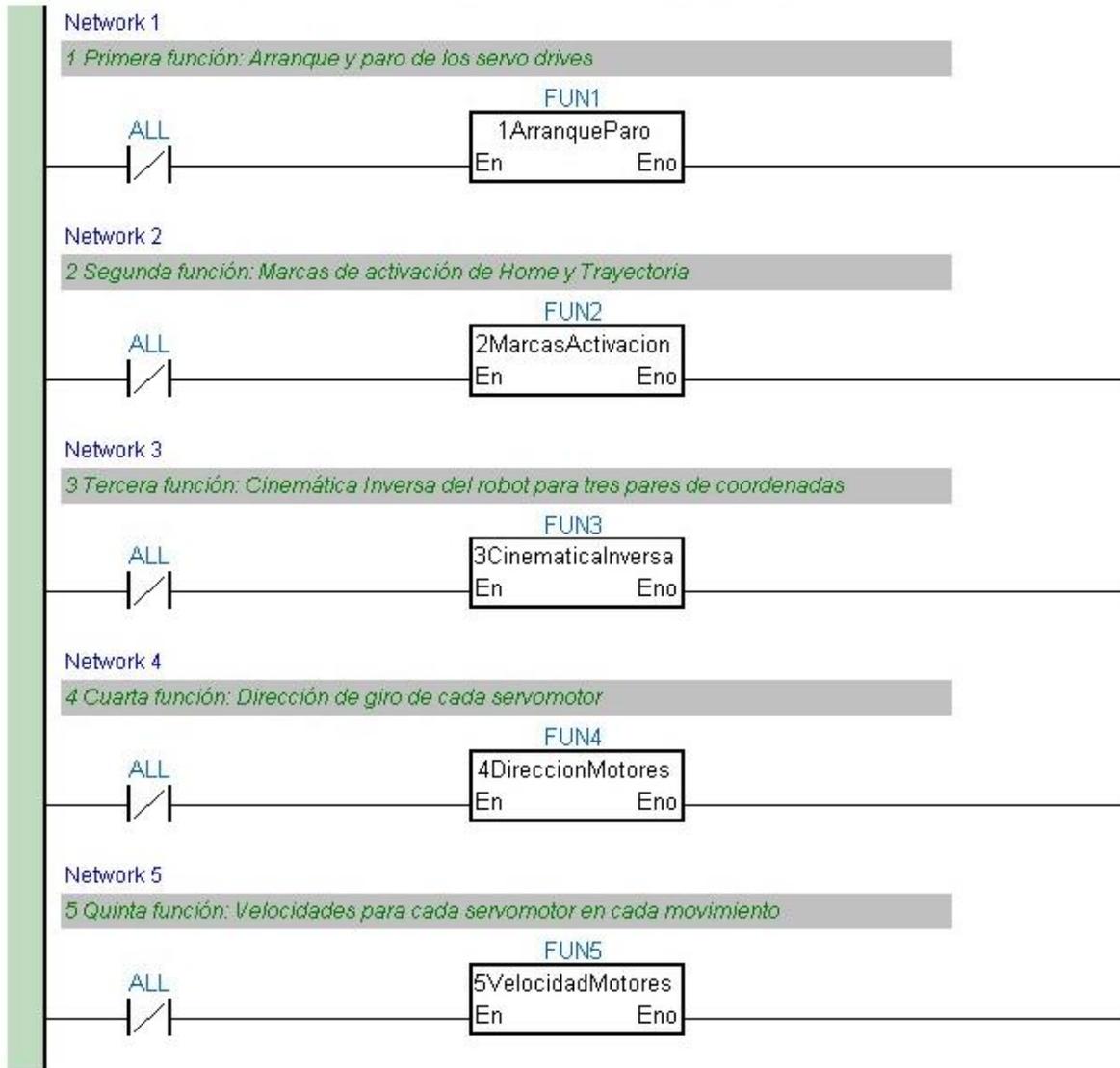
2MarcasActivacion

Declaration Type	Identifiers	Address	Type	Initial Value
VAR	Home_HMI	M3	BOOL	FALSE
VAR	Home_MC	D6253	WORD	0
VAR	Trayec_HMI	M4	BOOL	FALSE
VAR	Trayec_MC	D6254	WORD	0



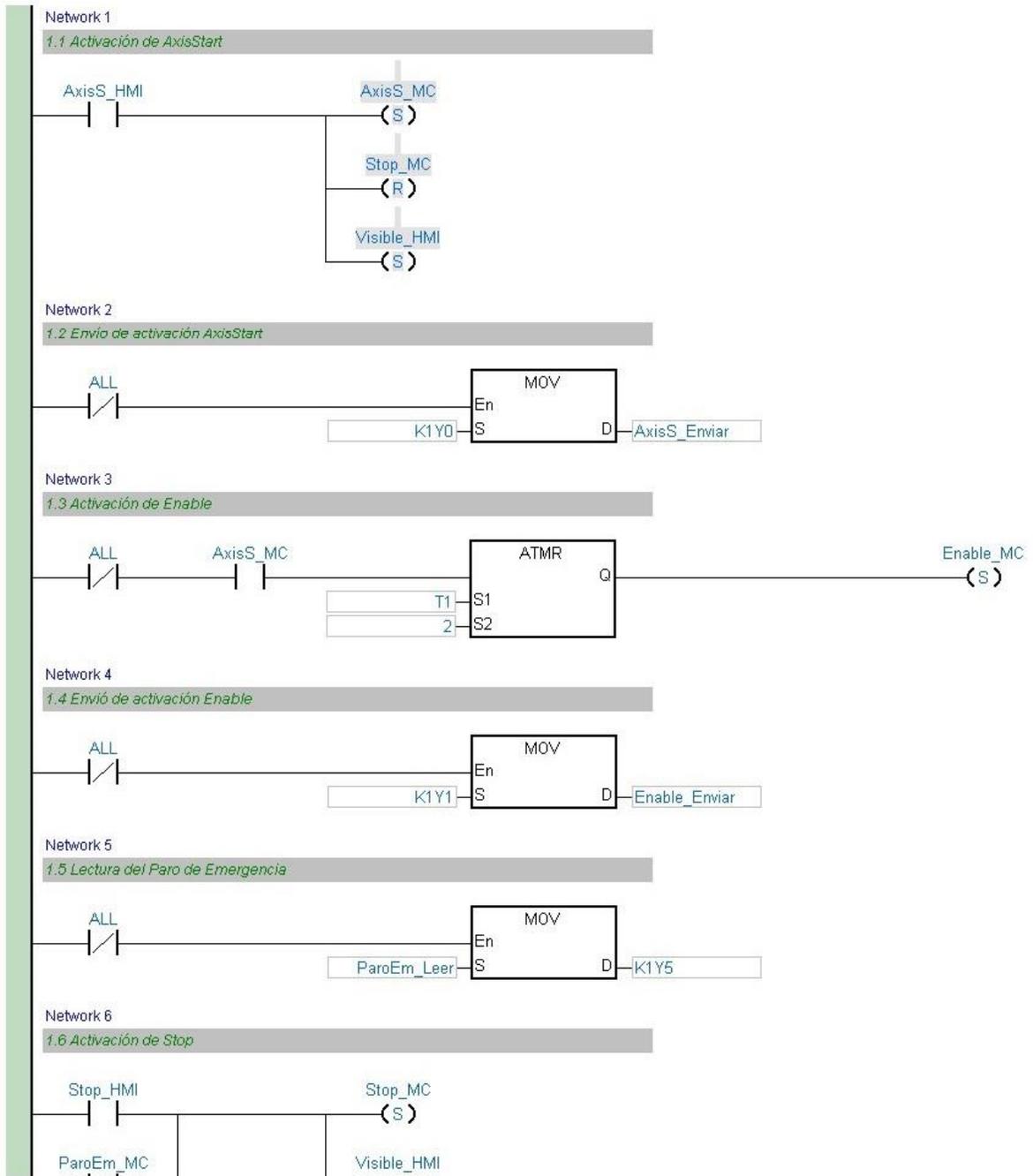
Main

Declaration Type	Identifiers	Address	Type	Initial Value
VAR	FUN1	N/A [Auto]	1ArranqueParo	N/A
VAR	FUN2	N/A [Auto]	2MarcasActivacion	N/A
VAR	FUN3	N/A [Auto]	3CinematicaInversa	N/A
VAR	FUN4	N/A [Auto]	4DireccionMotores	N/A
VAR	FUN5	N/A [Auto]	5VelocidadMotores	N/A

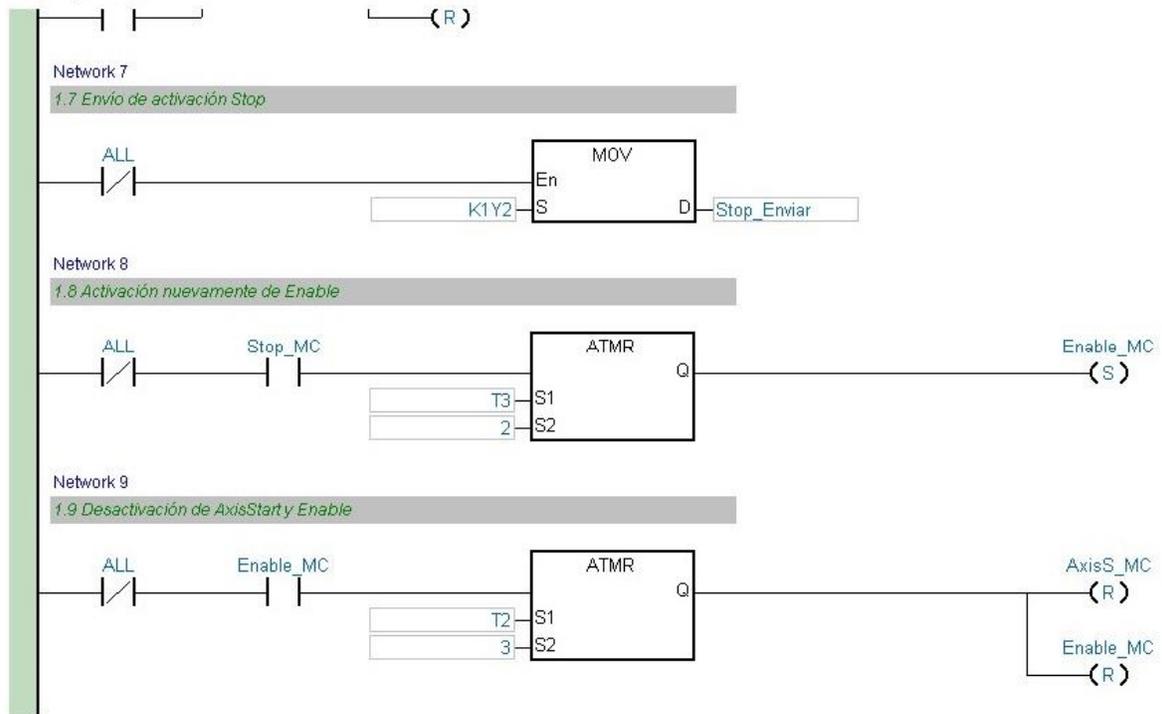


1ArranqueParo

Declaration Type	Identifiers	Address	Type	Initial Value	Identifier Comment
VAR	AxisS_HMI	M0	BOOL	FALSE	
VAR	AxisS_MC	Y0	BOOL	FALSE	
VAR	Enable_MC	Y1	BOOL	FALSE	
VAR	Stop_HMI	M2	BOOL	FALSE	
VAR	Stop_MC	Y2	BOOL	FALSE	
VAR	ParoEm_MC	Y5	BOOL	FALSE	
VAR	Visible_HMI	M6	BOOL	FALSE	
VAR	ParoEm_Leer	D6000	WORD	0	
VAR	AxisS_Enviar	D6250	WORD	0	
VAR	Enable_Enviar	D6251	WORD	0	
VAR	Stop_Enviar	D6252	WORD	0	



1ArranquePera

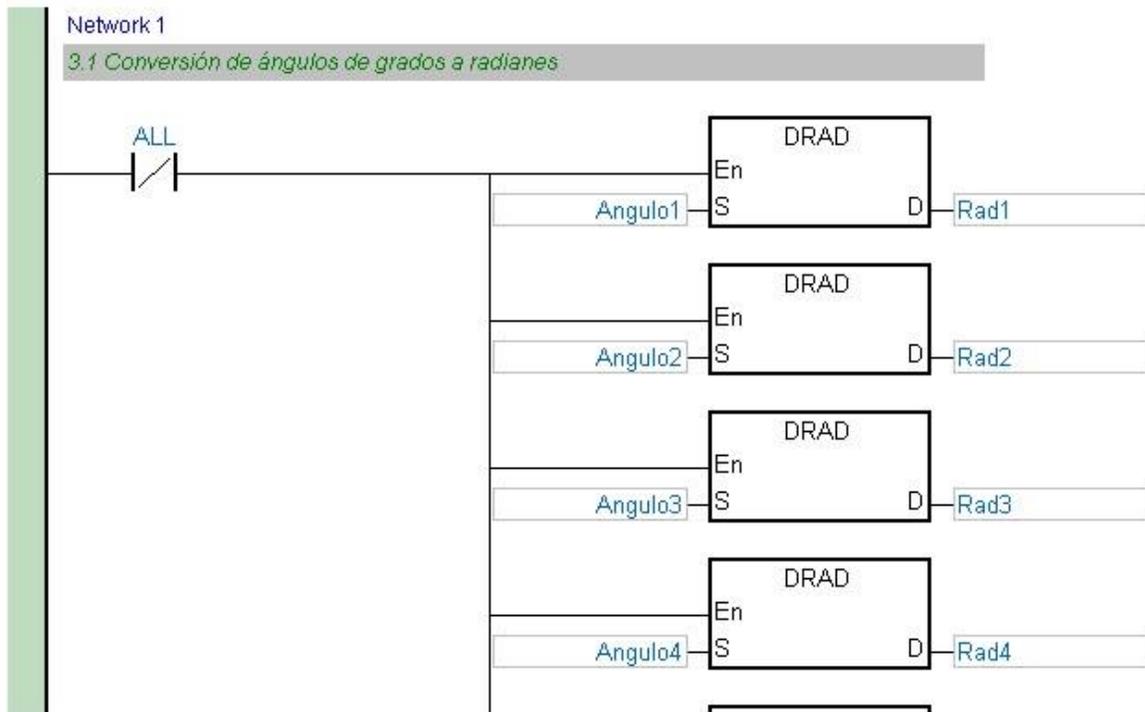


3CinematicalInversa

Declaration Type	Identifiers	Address	Type	Initial Value
VAR	Angulo1	D20	REAL	0.0
VAR	Angulo2	D22	REAL	0.0
VAR	Angulo3	D24	REAL	0.0
VAR	Angulo4	D26	REAL	0.0
VAR	Angulo5	D28	REAL	0.0
VAR	Rad1	D40	REAL	0.0
VAR	Rad2	D42	REAL	0.0
VAR	Rad3	D44	REAL	0.0
VAR	Rad4	D46	REAL	0.0
VAR	Rad5	D48	REAL	0.0
VAR	SinT1	D60	REAL	0.0
VAR	SinT2	D62	REAL	0.0
VAR	SinT3	D64	REAL	0.0
VAR	SinT4	D66	REAL	0.0
VAR	SinT5	D68	REAL	0.0
VAR	CosT1	D80	REAL	0.0
VAR	CosT2	D82	REAL	0.0
VAR	CosT3	D84	REAL	0.0
VAR	CosT4	D86	REAL	0.0
VAR	CosT5	D88	REAL	0.0
VAR	Px1	D100	REAL	0.0
VAR	Px2	D102	REAL	0.0
VAR	Px3	D104	REAL	0.0
VAR	Px4	D106	REAL	0.0
VAR	Px5	D108	REAL	0.0
VAR	Py1	D120	REAL	0.0
VAR	Py2	D122	REAL	0.0
VAR	Py3	D124	REAL	0.0
VAR	Py4	D126	REAL	0.0
VAR	Py5	D128	REAL	0.0
VAR	Term1_a	D140	REAL	0.0
VAR	Term2_a	D142	REAL	0.0
VAR	Term3_a	D144	REAL	0.0
VAR	Term4_a	D146	REAL	0.0
VAR	Term1_b	D148	REAL	0.0
VAR	Term2_b	D150	REAL	0.0
VAR	Term3_b	D152	REAL	0.0
VAR	Term4_b	D154	REAL	0.0
VAR	Term1_c	D156	REAL	0.0
VAR	Term2_c	D158	REAL	0.0
VAR	Term3_c	D160	REAL	0.0
VAR	Term4_c	D162	REAL	0.0
VAR	Term1_d	D164	REAL	0.0
VAR	Term2_d	D166	REAL	0.0
VAR	Term3_d	D168	REAL	0.0
VAR	Term4_d	D170	REAL	0.0
VAR	Term1_e	D172	REAL	0.0
VAR	Term2_e	D174	REAL	0.0
VAR	Term3_e	D176	REAL	0.0

3CinematicaInversa

VAR	Term4 e	D178	REAL	0.0
VAR	Cuadra1a	D228	REAL	0.0
VAR	Cuadra2a	D238	REAL	0.0
VAR	Cuadra3a	D250	REAL	0.0
VAR	Cuadra4a	D260	REAL	0.0
VAR	Cuadra1b	D272	REAL	0.0
VAR	Cuadra2b	D282	REAL	0.0
VAR	Cuadra3b	D294	REAL	0.0
VAR	Cuadra4b	D304	REAL	0.0
VAR	Cuadra1c	D316	REAL	0.0
VAR	Cuadra2c	D326	REAL	0.0
VAR	Cuadra3c	D338	REAL	0.0
VAR	Cuadra4c	D348	REAL	0.0
VAR	Cuadra1d	D360	REAL	0.0
VAR	Cuadra2d	D370	REAL	0.0
VAR	Cuadra3d	D382	REAL	0.0
VAR	Cuadra4d	D392	REAL	0.0
VAR	Cuadra1e	D404	REAL	0.0
VAR	Cuadra2e	D414	REAL	0.0
VAR	Cuadra3e	D426	REAL	0.0
VAR	Cuadra4e	D436	REAL	0.0

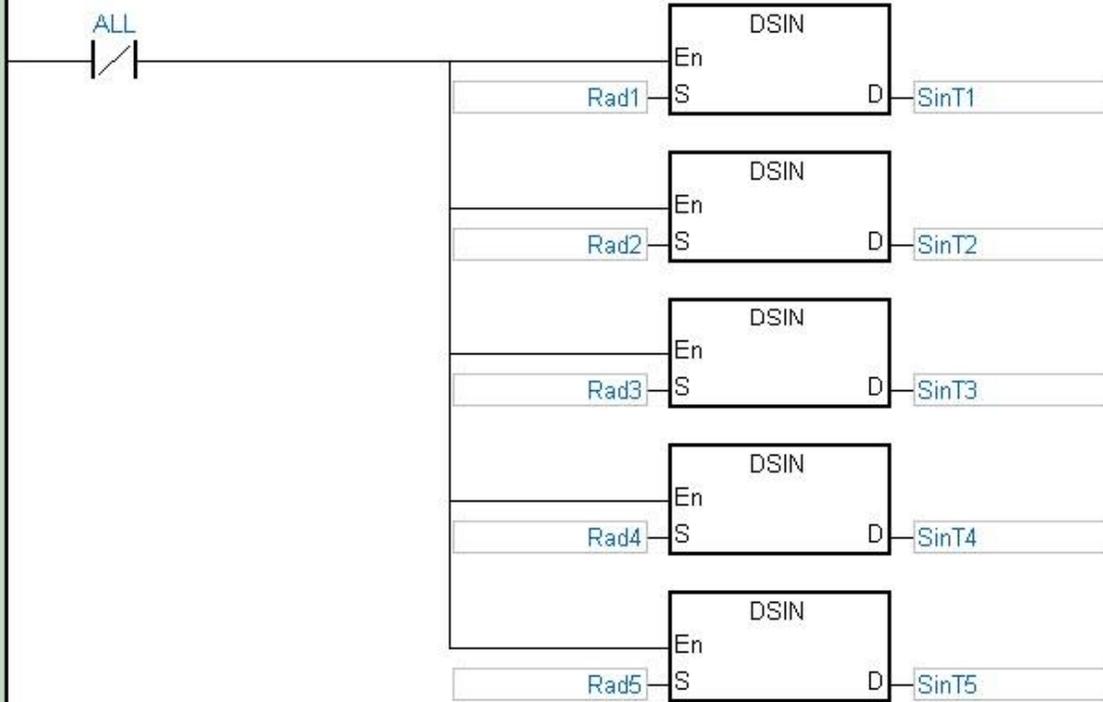


3CinematicaInversa



Network 2

3.2 Obtención de seno de todos los ángulos (radianes)

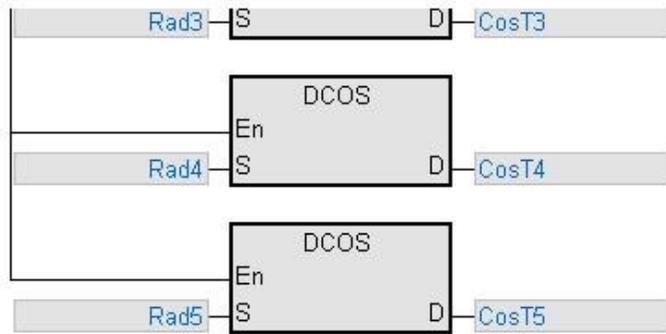


Network 3

3.3 Obtención de coseno de todos los ángulos (radianes)

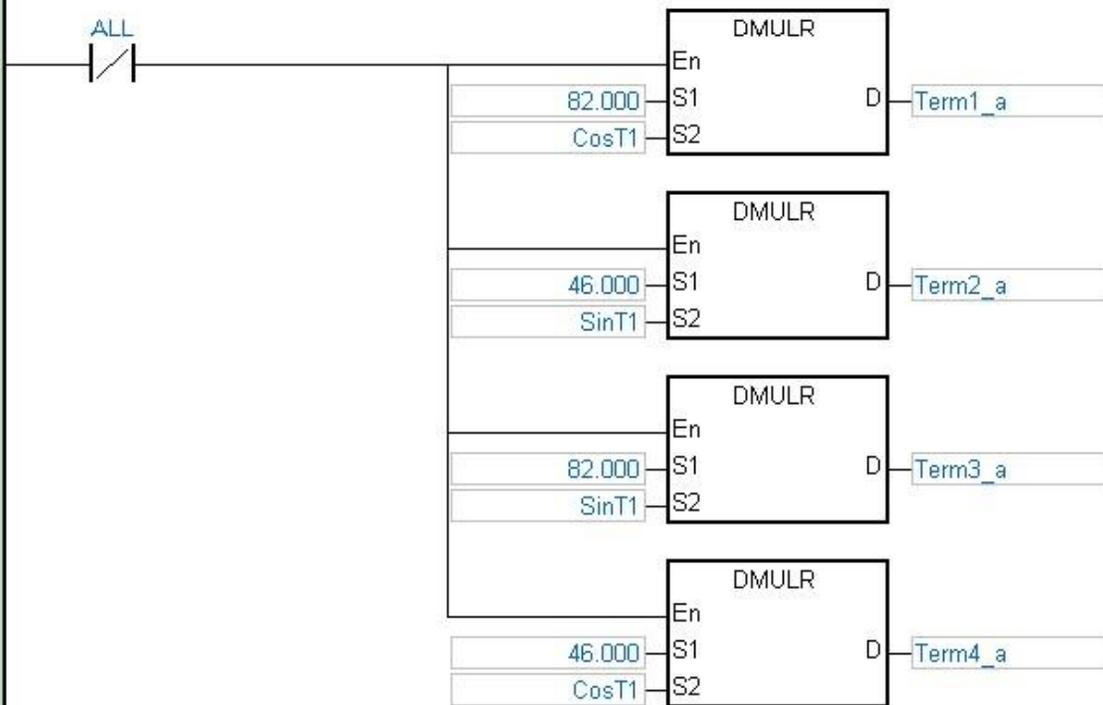


3CinematicaInversa



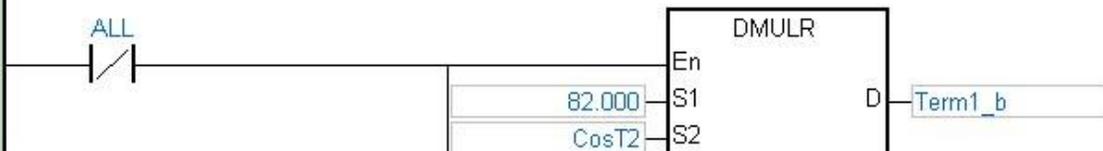
Network 4

3.4 Obtención de términos para cables l1_a y l2_a

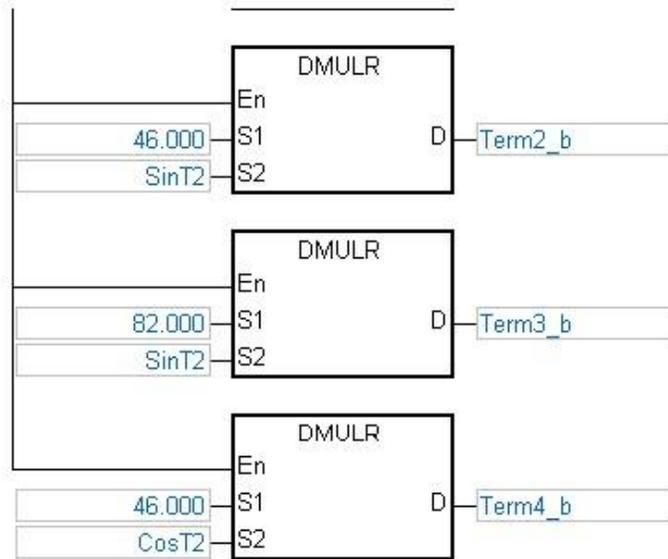


Network 5

3.5 Obtención de términos para cables l1_b y l2_b

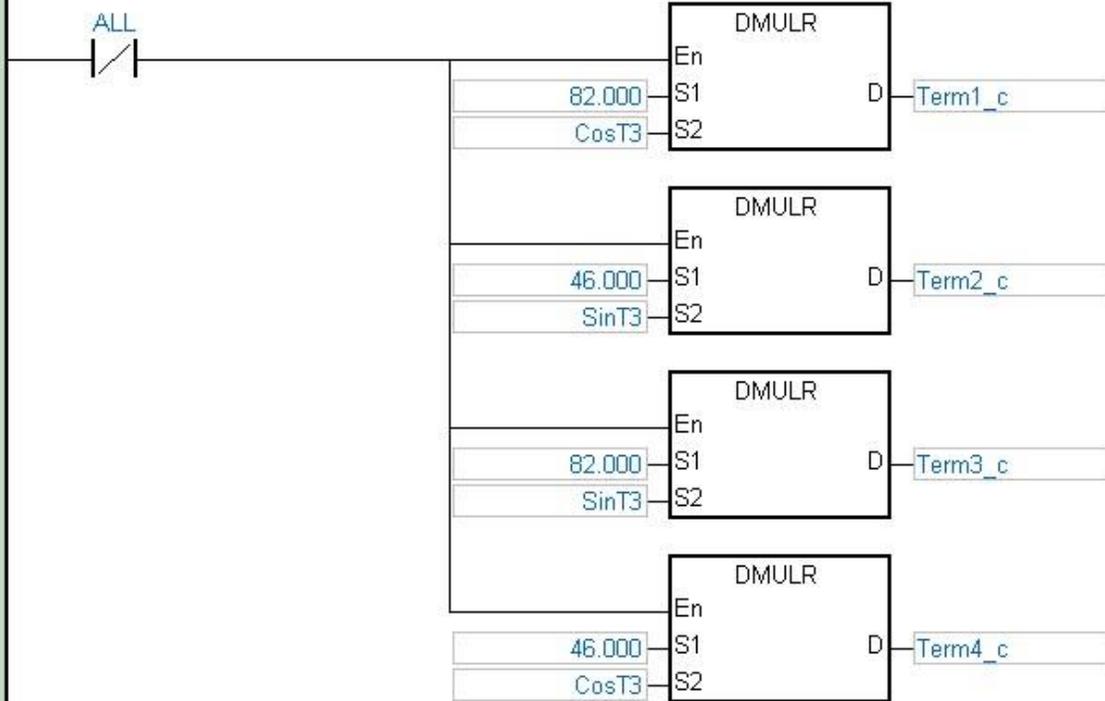


3CinematicaInversa



Network 6

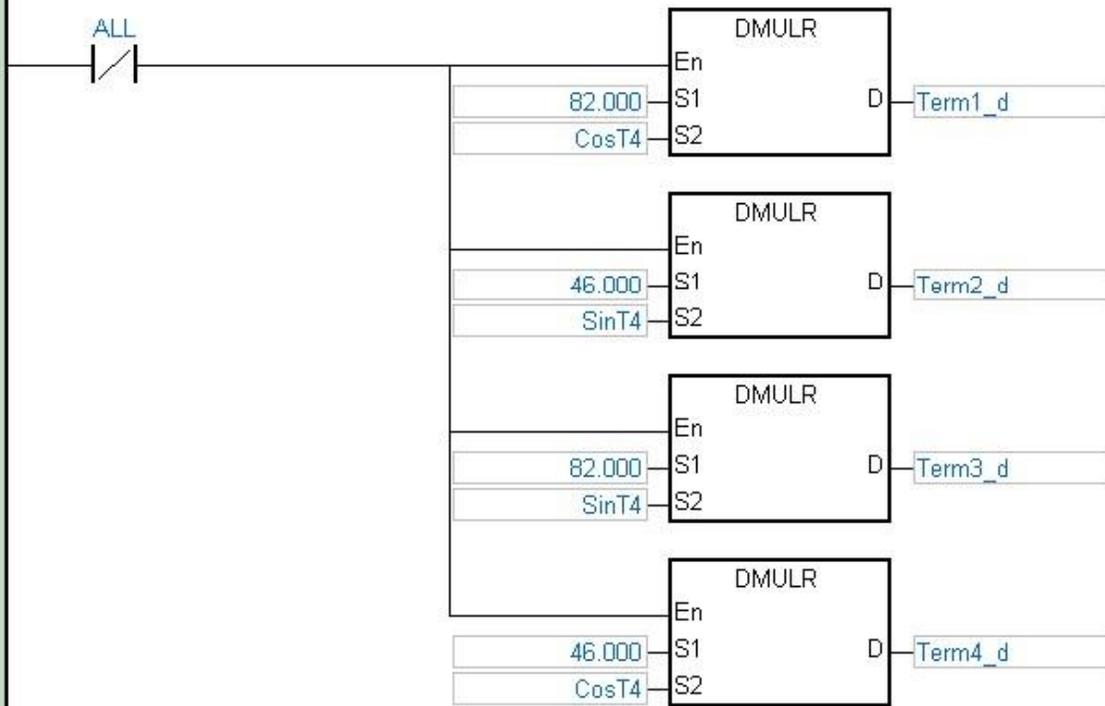
3.6 Obtención de términos para cables I1_c y I2_c



Network 7

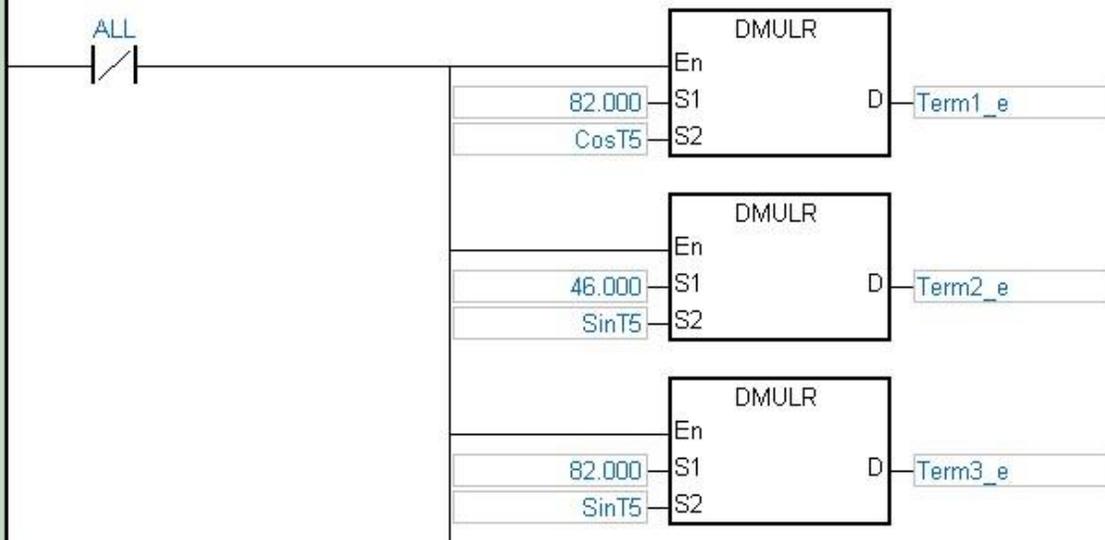
3 Cinematica Inversa

3.7 Obtención de términos para cables $I1_d$ y $I2_d$

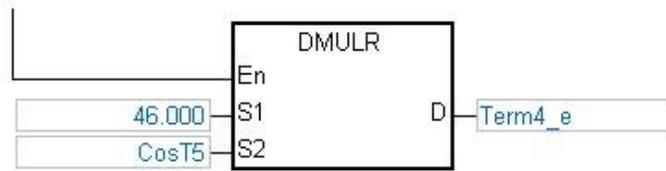


Network 8

3.8 Obtención de términos para cables $I1_e$ y $I2_e$

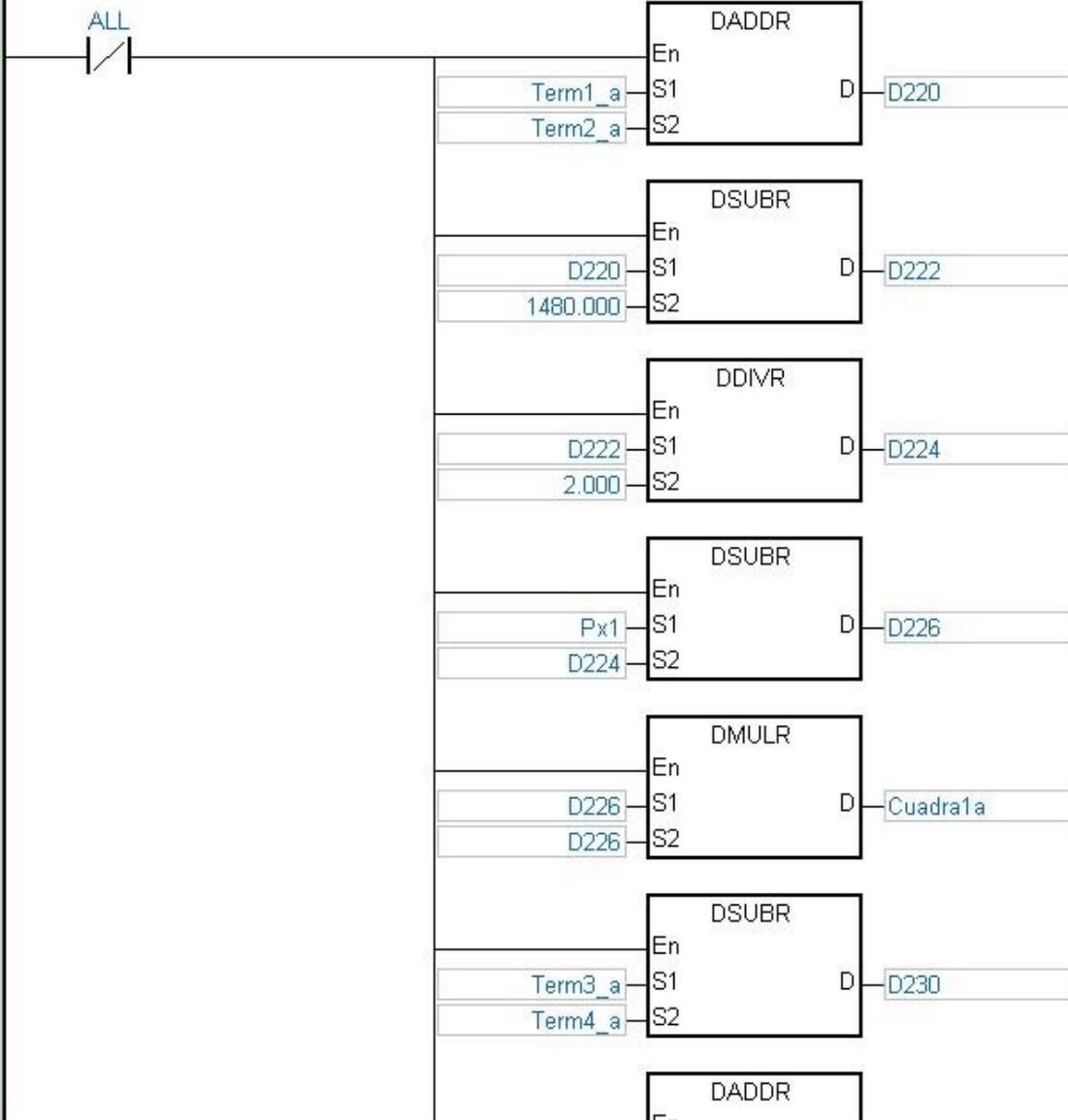


3CinematicalInversa

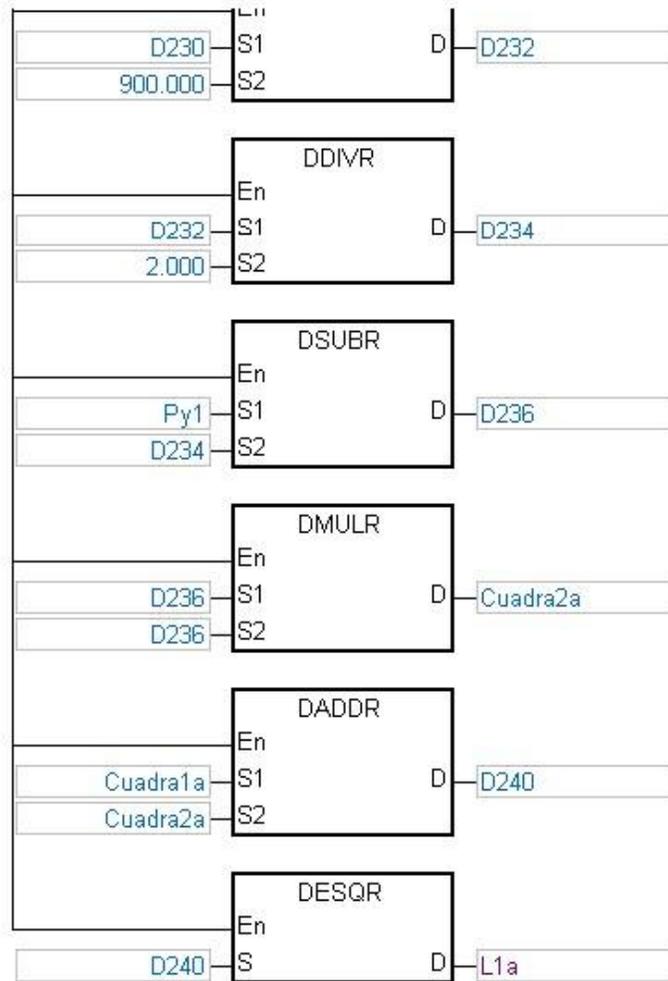


Network 9

3.9 Valor de cable l1_a = sqrt ((Px1 - (82*term1 + 46*term2 - 1480)/2)^2 + (Py1 - (82*term1

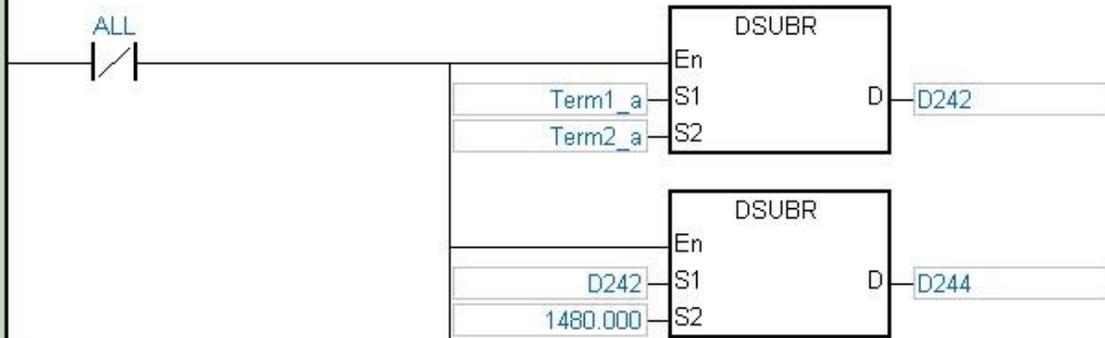


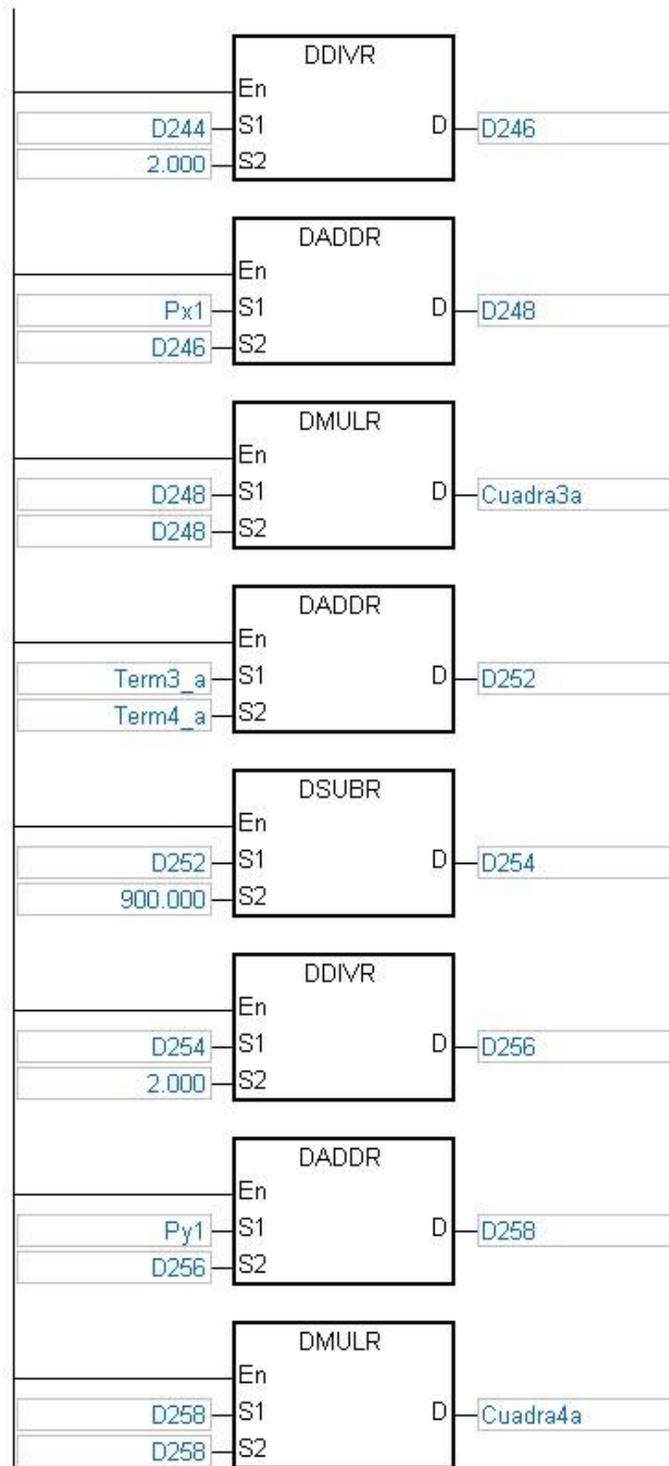
3CinematicaInversa



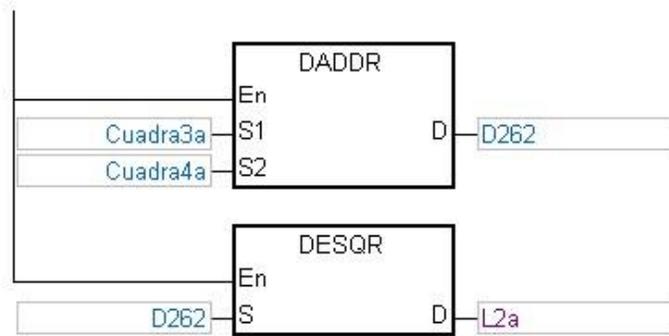
Network 10

$3.10 \text{ Valor de cable } I2_a = \text{sqrt} ((Px1 + (82*term1 - 46*term2 - 1480)/2)^2 + (Py1 + (82*term1 - 46*term2 - 1480)/2)^2)$



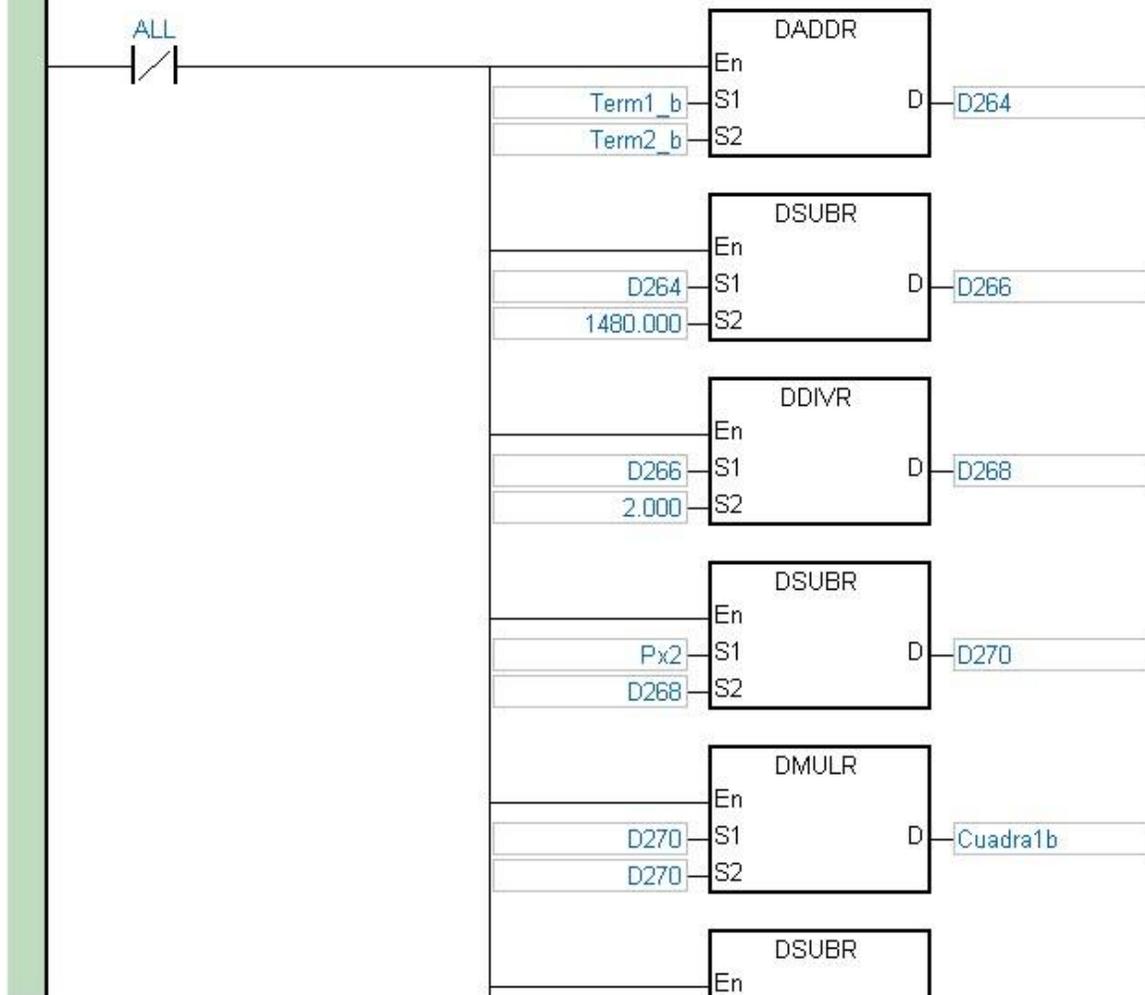


3CinematicalInversa

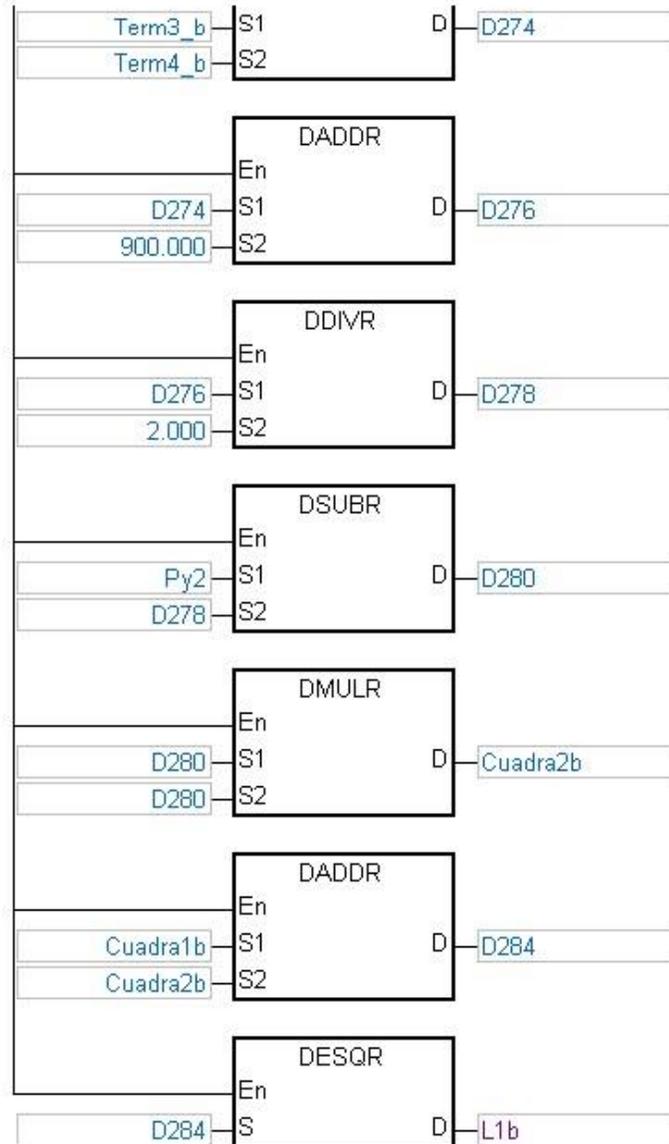


Network 11

3.11 Valor de cable l1_b = sqrt((Px2 - (82*term1 + 46*term2 - 1480)/2)^2 + (Py2 - (82*term1

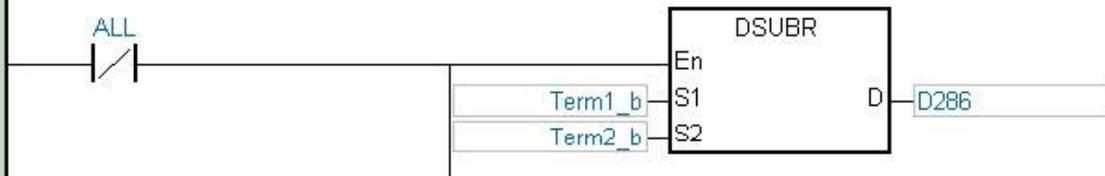


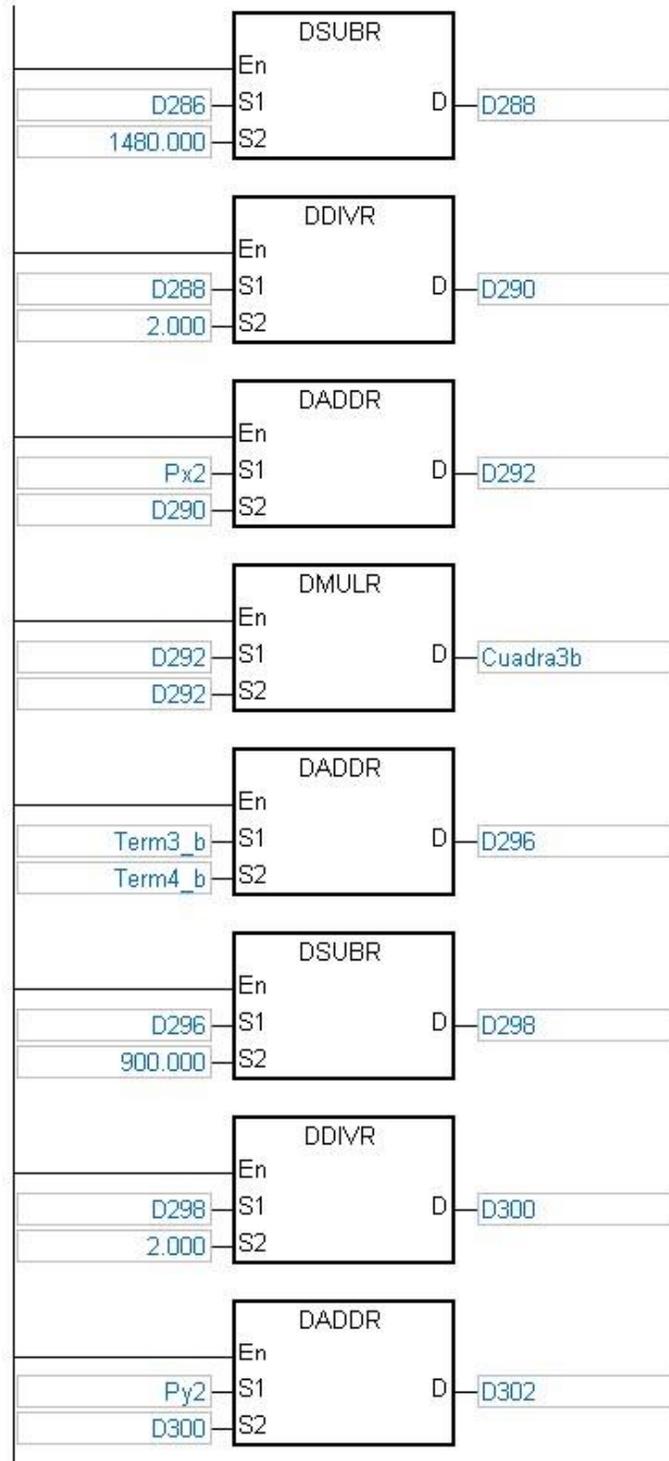
3CinematicalInversa



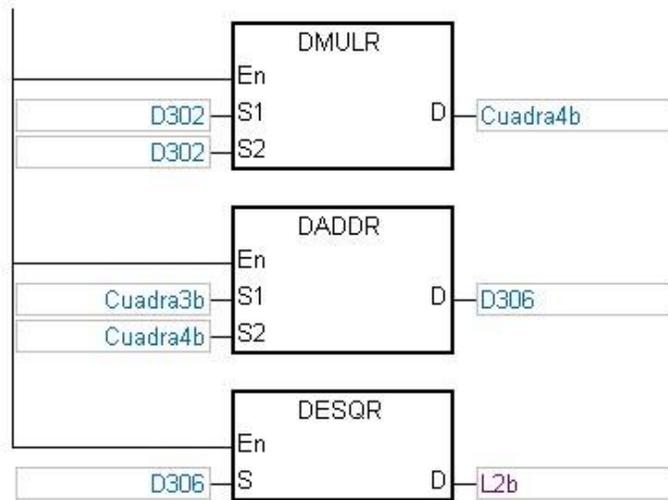
Network 12

3.12 Valor de cable $l2_b = \text{sqrt} ((Px2 + (82*term1 - 46*term2 - 1480)/2)^2 + (Py2 + (82*term1 - 46*term2 - 1480)/2)^2)$



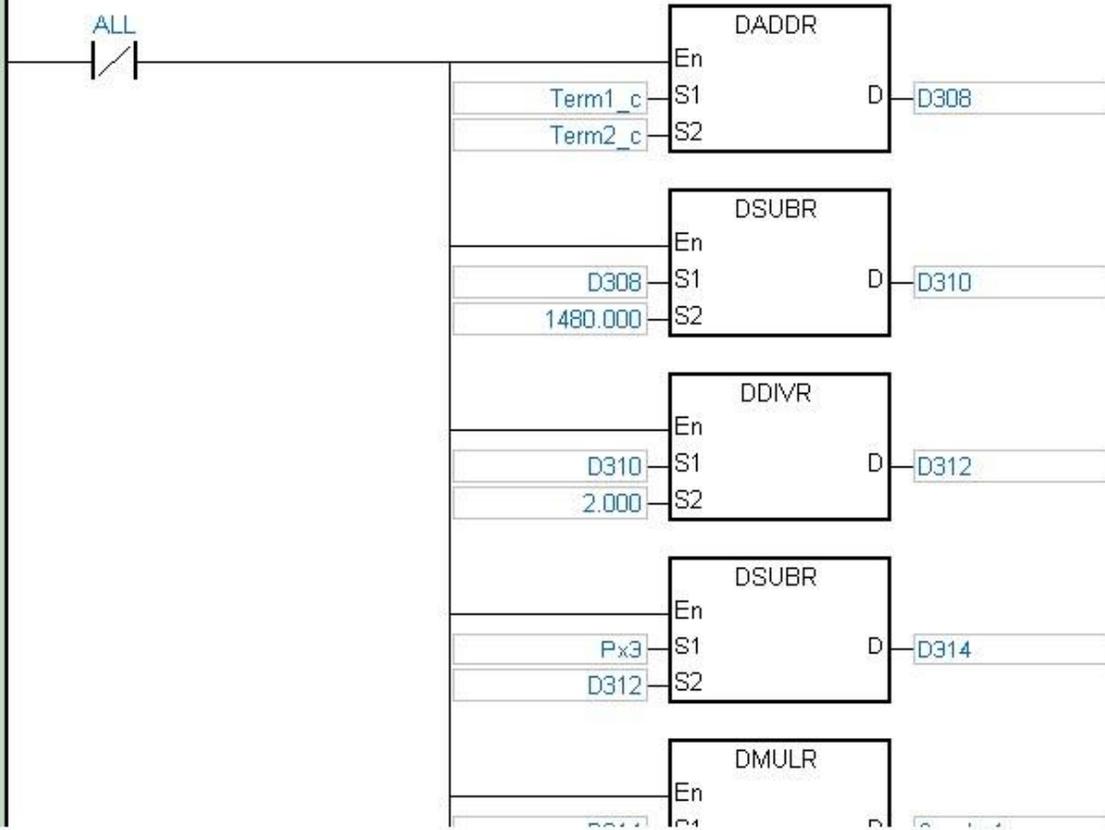


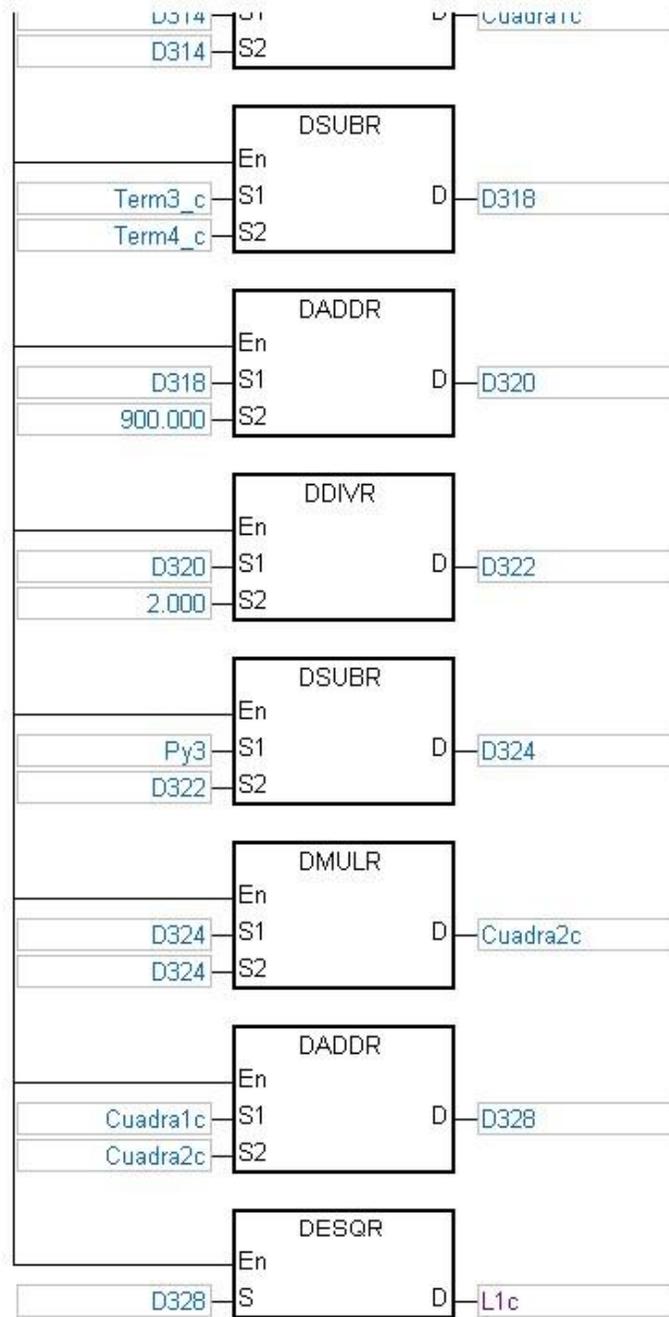
3CinematicalInversa



Network 13

3.13 Valor de cable l1_c = sqrt ((Px3 - (82*term1 + 46*term2 - 1480)/2)^2 + (Py3 - (82*term1 + 46*term2 - 1480)/2)^2)

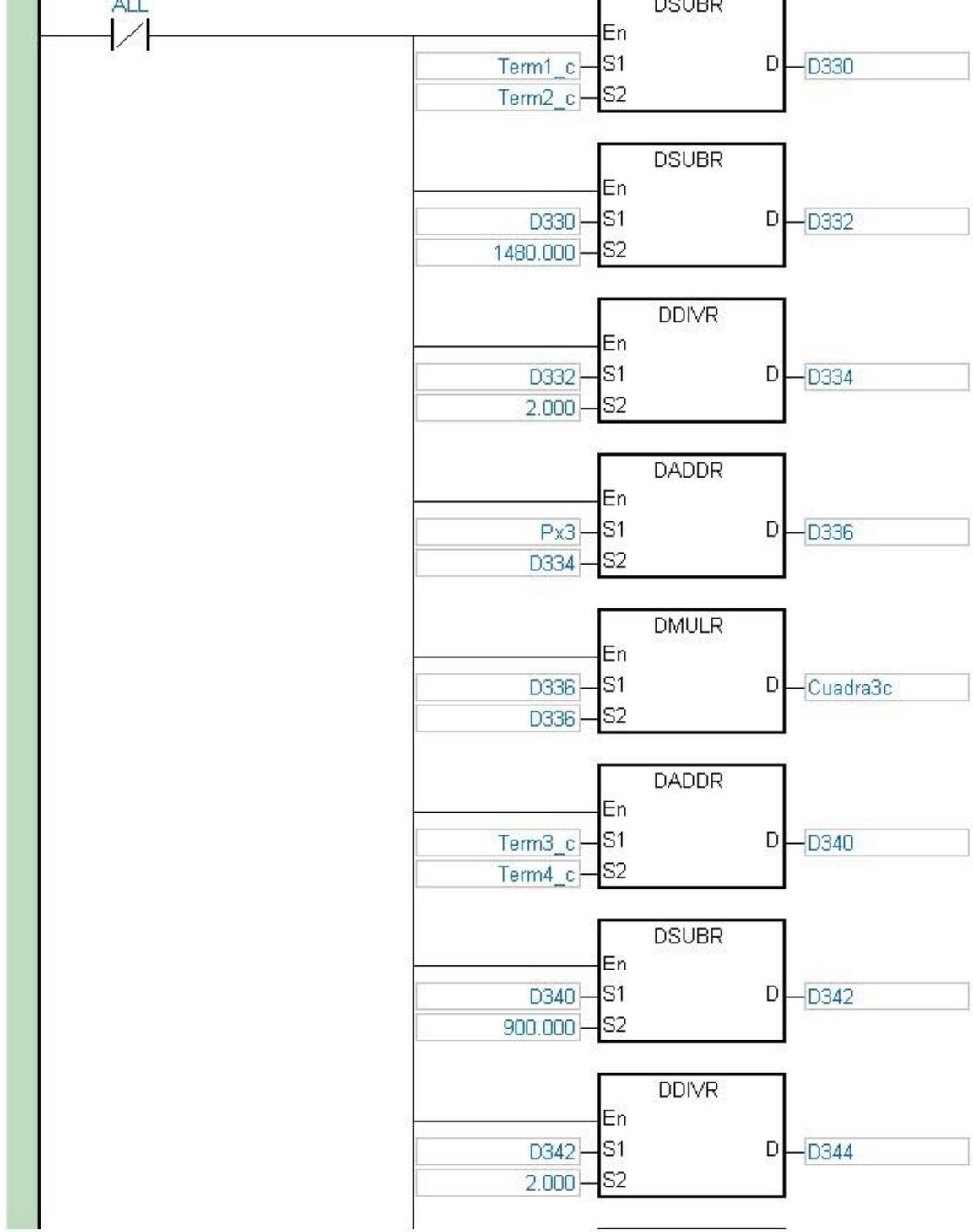




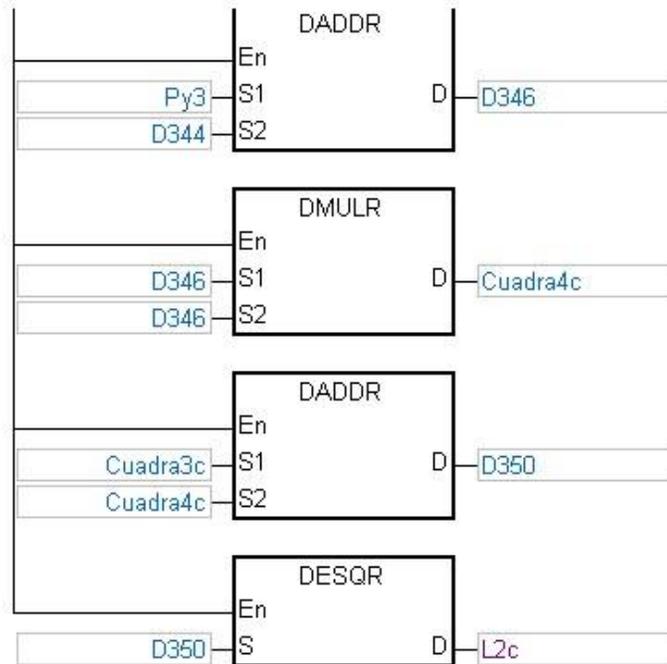
Network 14

3.14 Valor de cable l2_c = sqrt ((Px3 + (82*term1 - 46*term2 - 1480)/2)^2 + (Py3 + (82*term

3CinematicaInversa

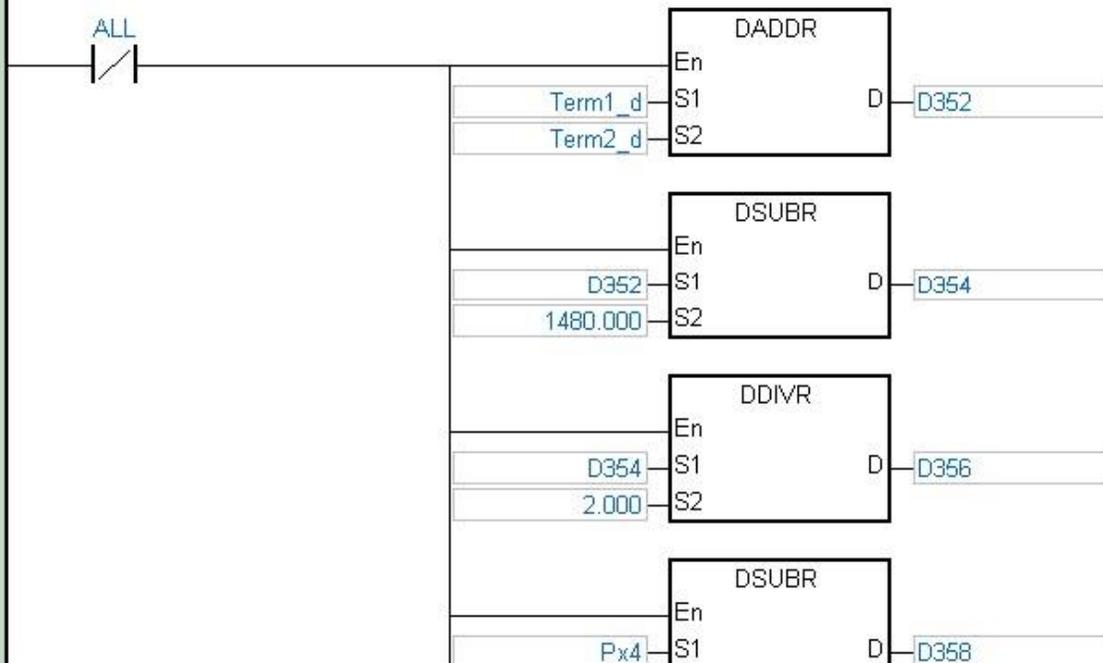


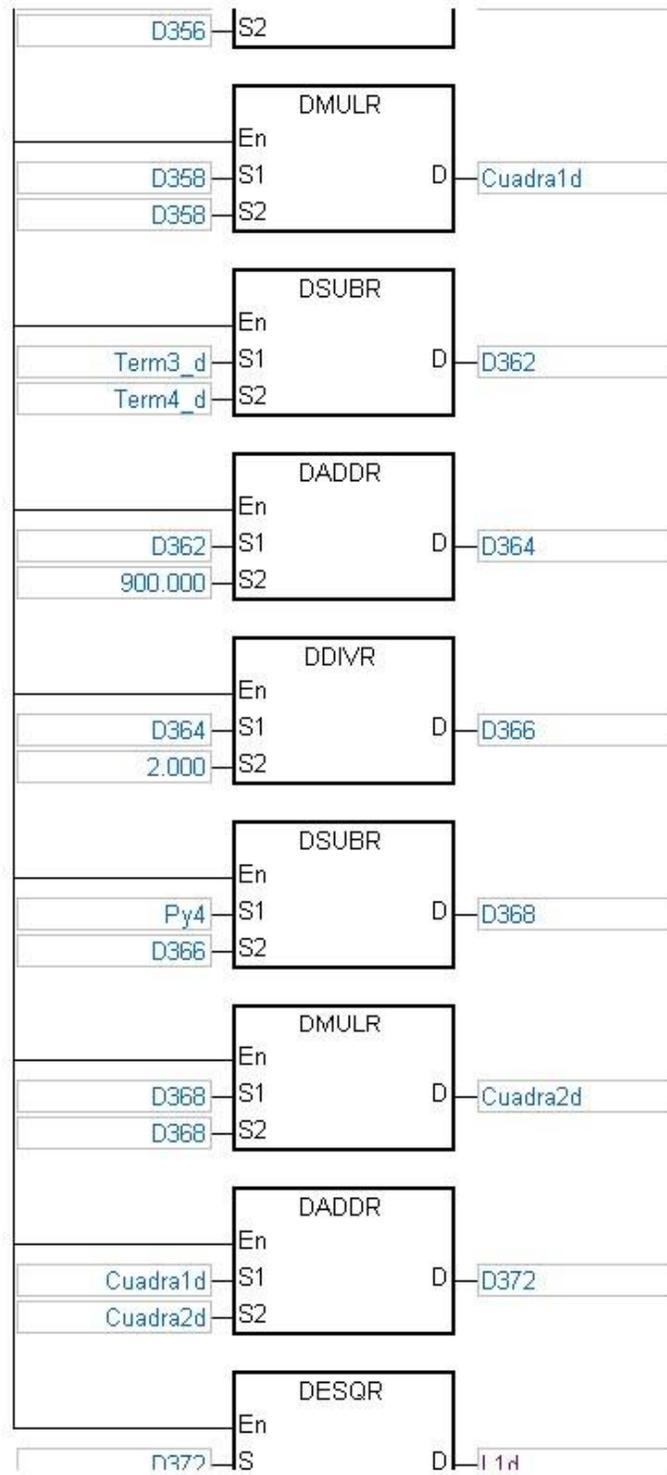
3CinematicalInversa



Network 15

3.15 Valor de cable l1_d = sqrt ((Px4 - (82*term1 + 46*term2 - 1480)/2)^2 + (Py4 - (82*term

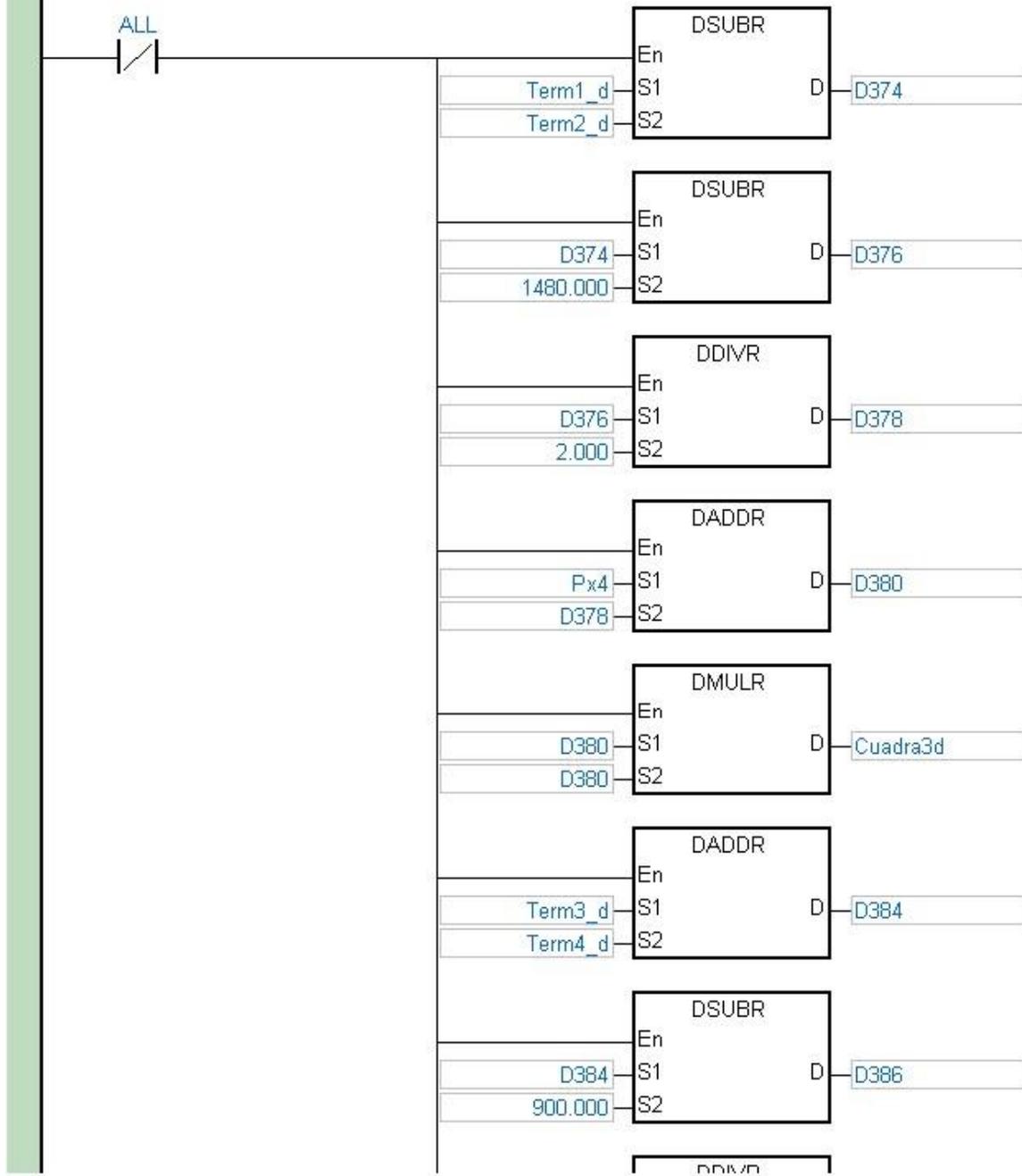




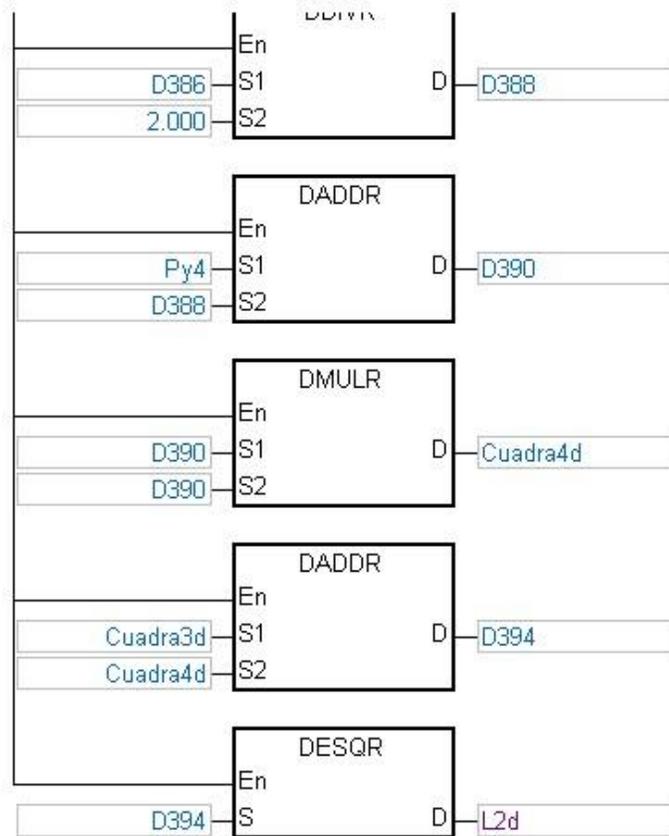
3CinematicaInversa

Network 16

3.16 Valor de cable l2_d = sqrt ((Px4 + (82*term1 - 46*term2 - 1480)/2)^2 + (Py4 + (82*ter

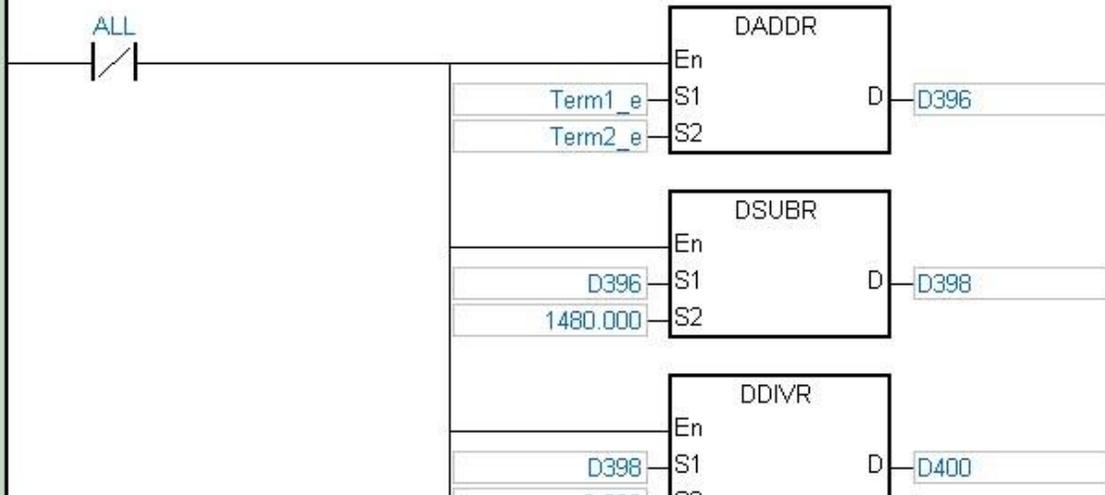


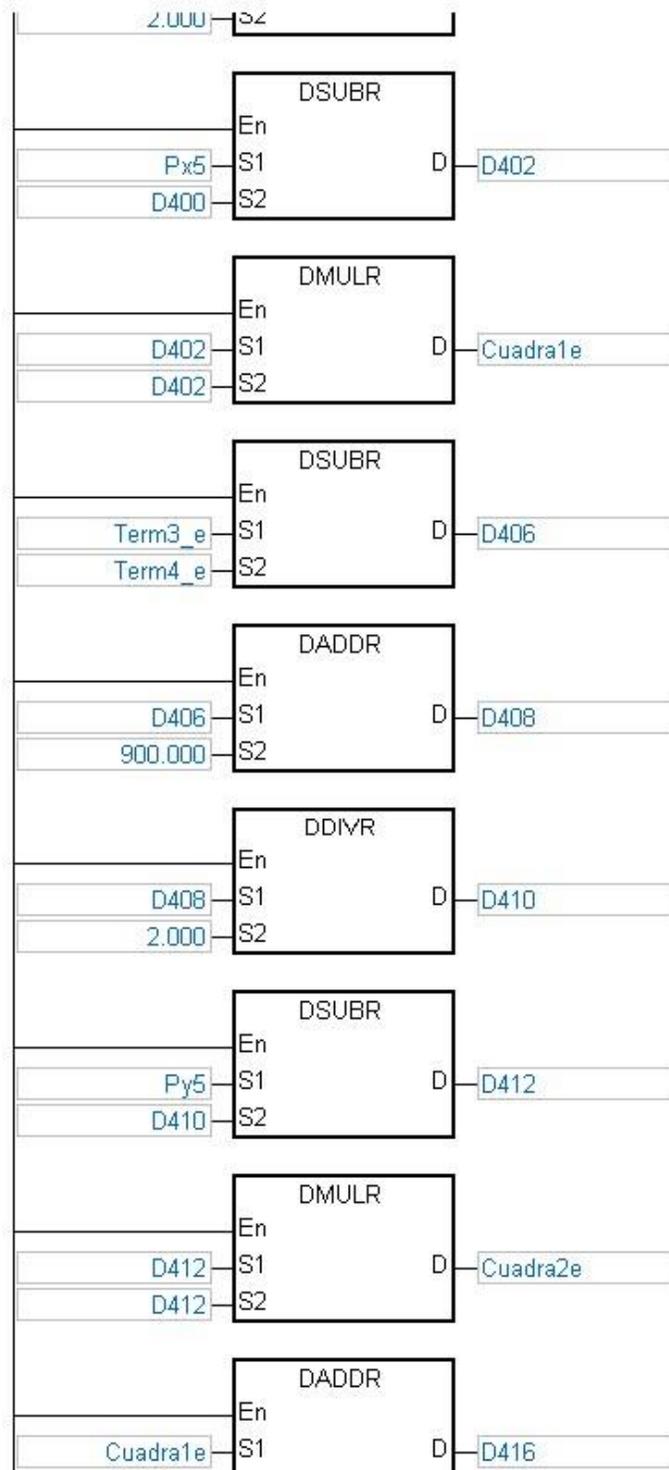
3CinematicaInversa



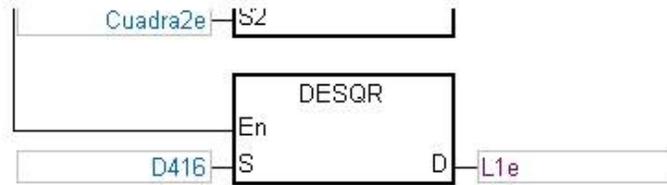
Network 17

3.17 Valor de cable l1_e = sqrt((Px5 - (82*term1 + 46*term2 - 1480)/2)^2 + (Py5 - (82*term



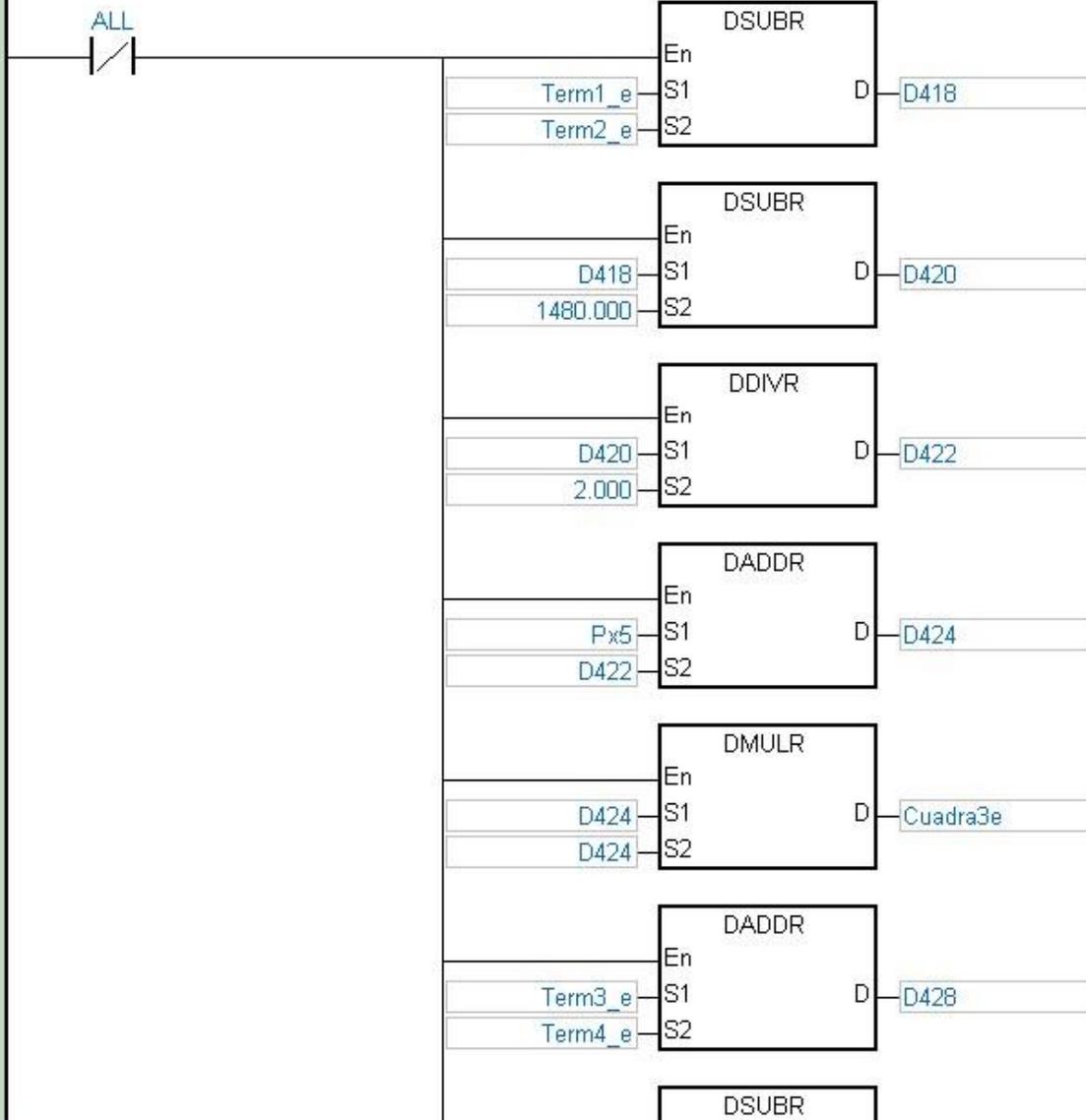


3CinematicalInversa

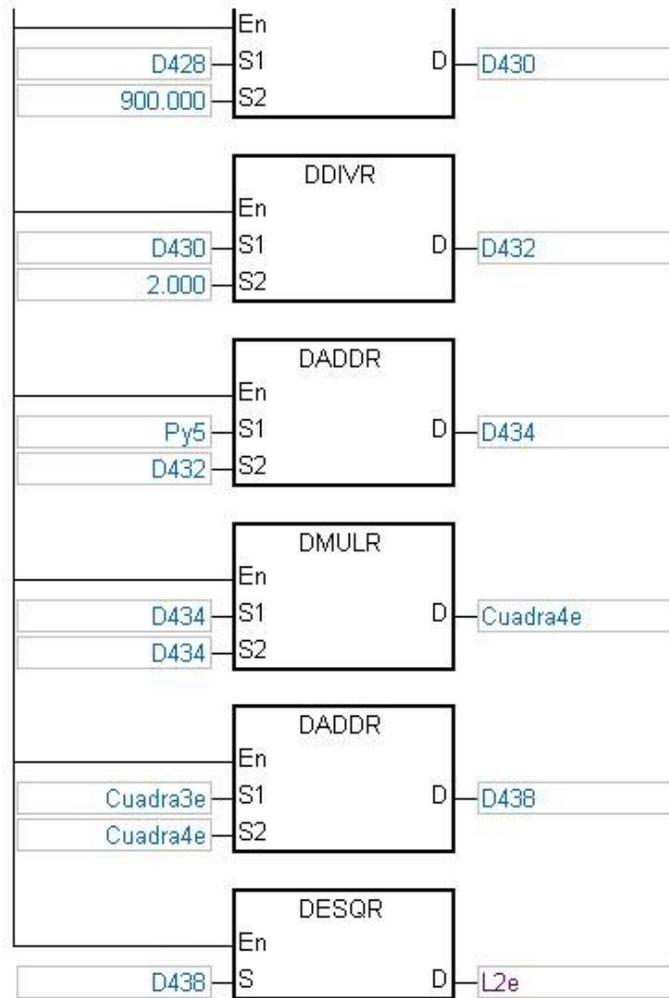


Network 18

$$3.18 \text{ Valor de cable } l2_e = \text{sqrt} ((Px5 + (82*term1 - 46*term2 - 1480)/2)^2 + (Py5 + (82*term$$

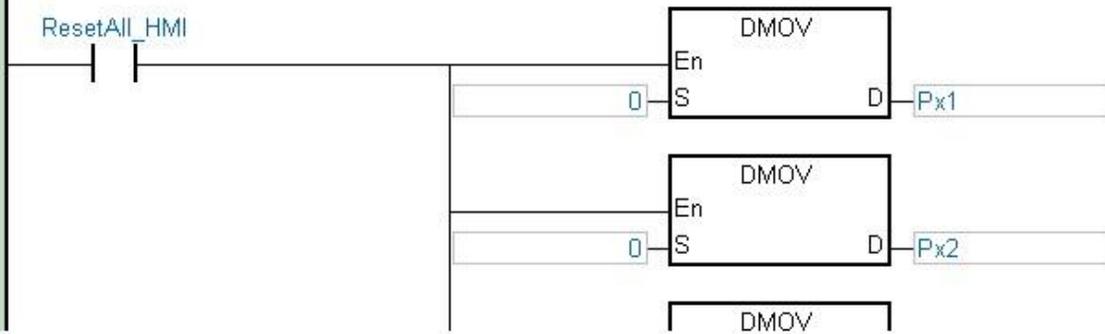


3CinematicalInversa

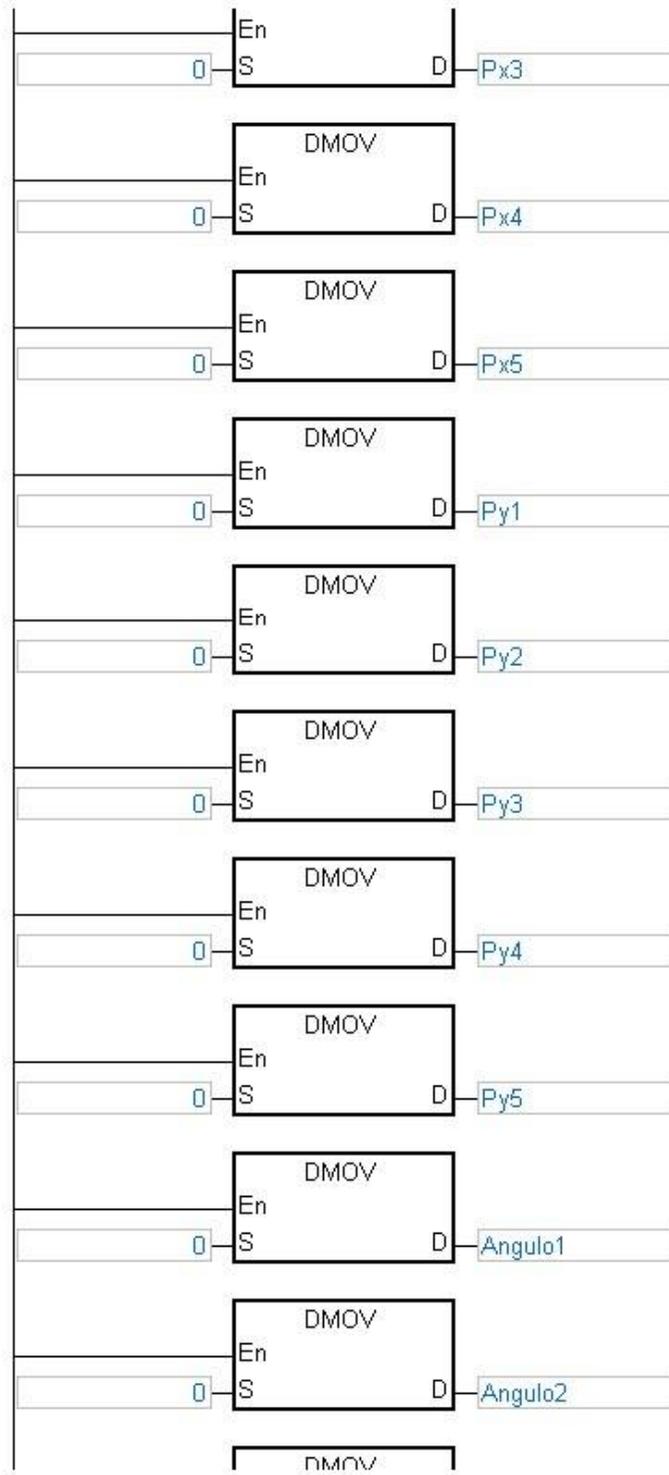


Network 19

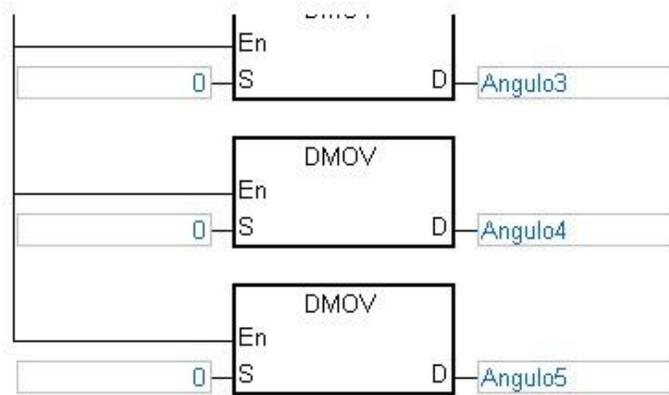
3.19 Al presionar botón "Reset all" retoman el valor de 0 todas las coordenadas y ángulos



3CinematicalInversa

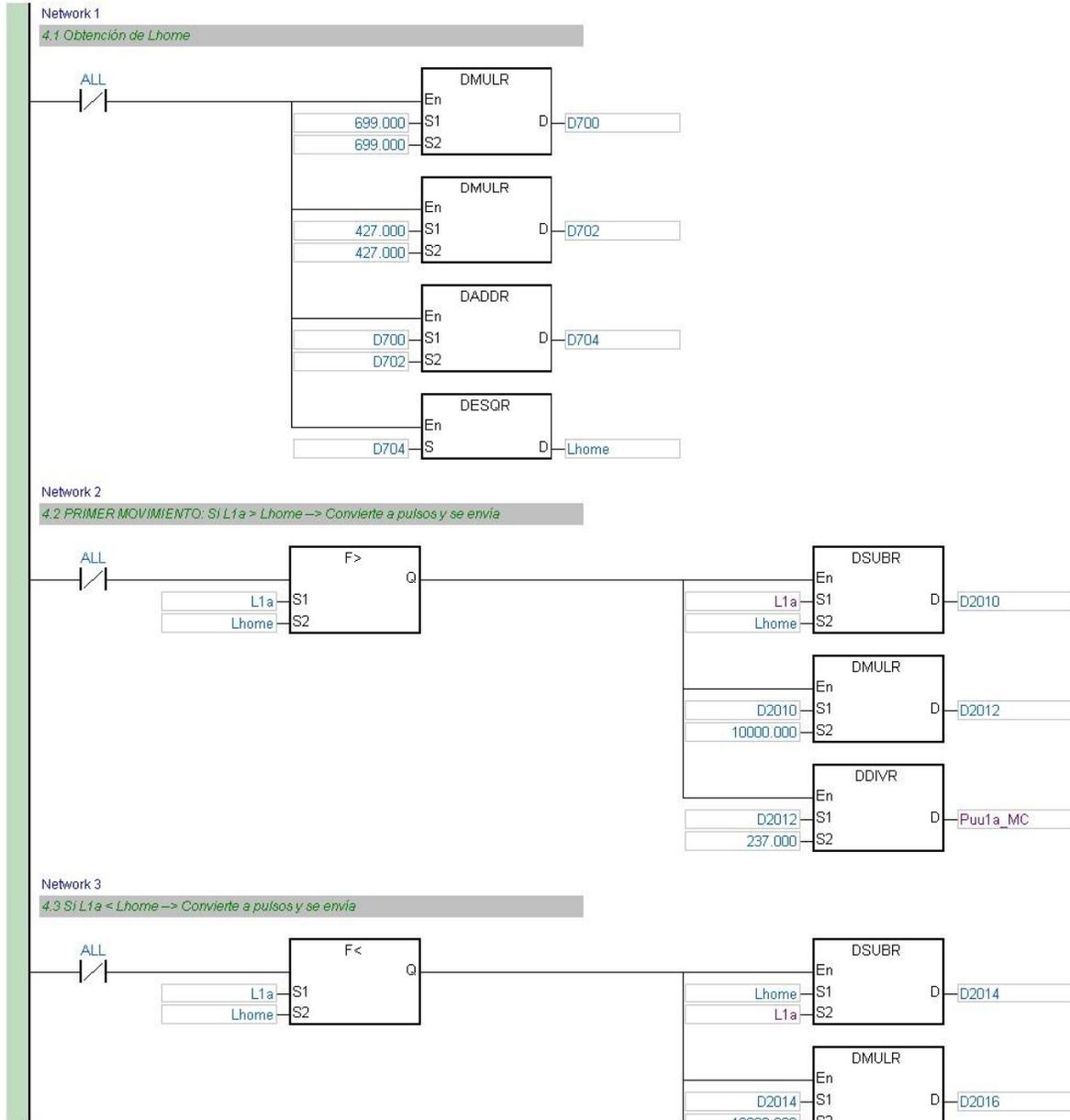


3CinematicalInversa

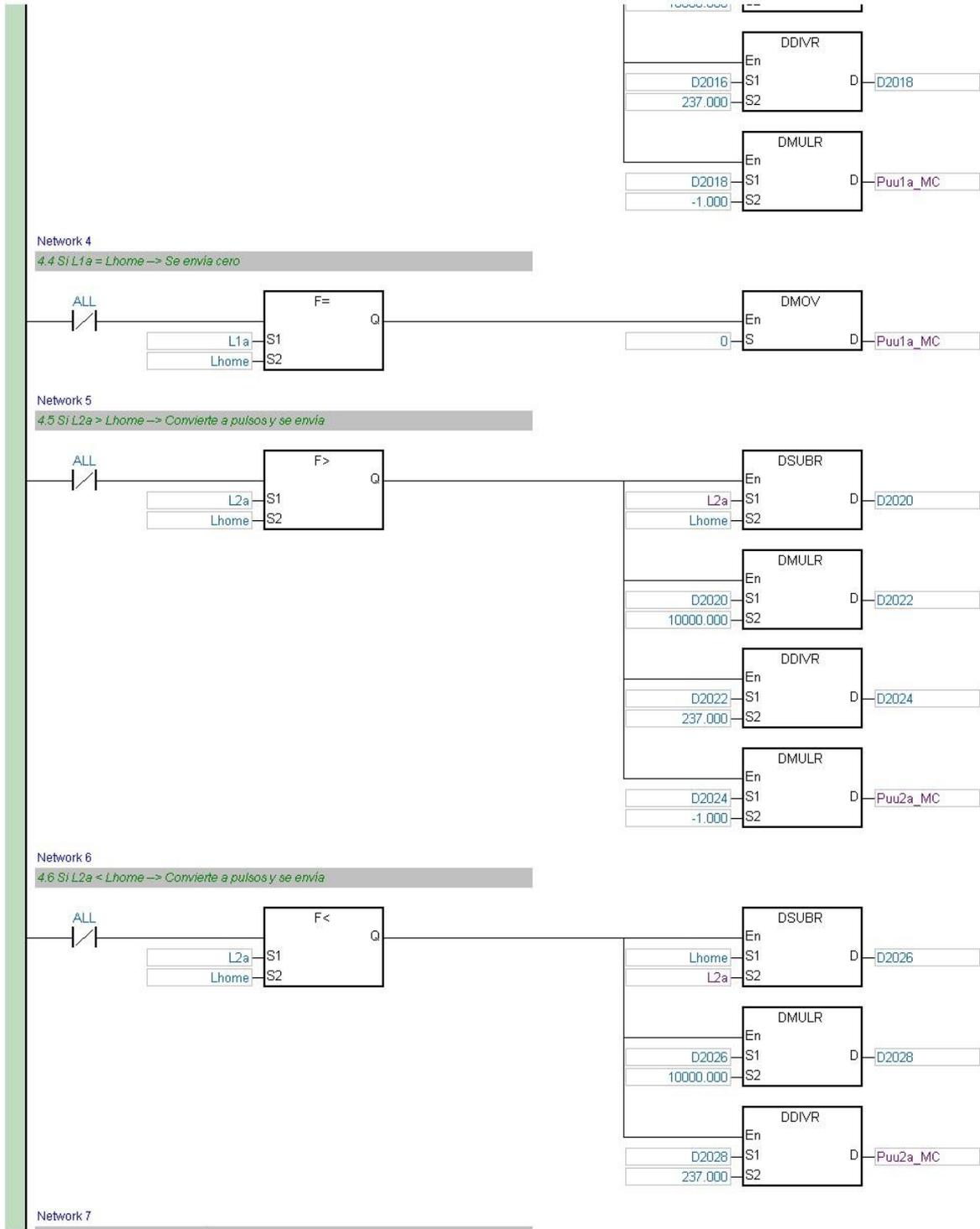


4DireccionMotores

Declaration Type	Identifiers	Address	Type	Initial Value	Identifier Comment
VAR	Lhome	D706	REAL	0.0	
VAR	Puu1b	D6320	REAL	0.0	
VAR	Puu2b	D6322	REAL	0.0	
VAR	Puu1c	D6324	REAL	0.0	
VAR	Puu2c	D6326	REAL	0.0	
VAR	Puu1d	D6328	REAL	0.0	
VAR	Puu2d	D6330	REAL	0.0	
VAR	Puu1e	D6332	REAL	0.0	
VAR	Puu2e	D6334	REAL	0.0	
VAR	Puu1Home	D6336	REAL	0.0	
VAR	Puu2Home	D6338	REAL	0.0	



4DireccionMotores



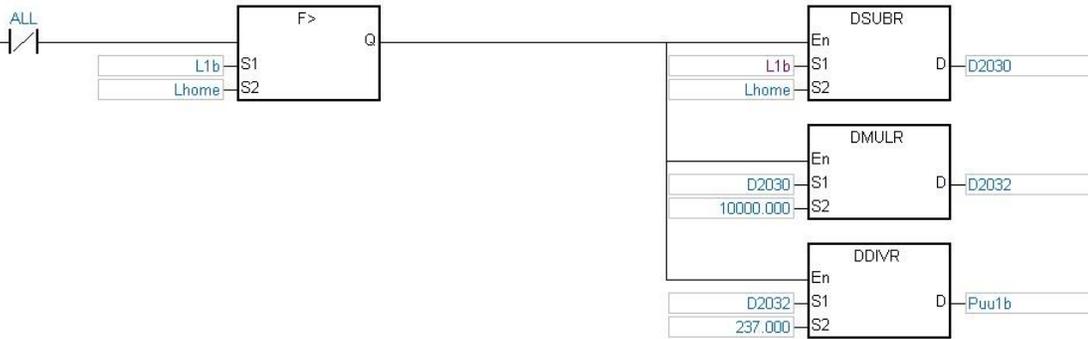
4DireccionMotores

4.7 Si L2a = Lhome -> Se envia cero



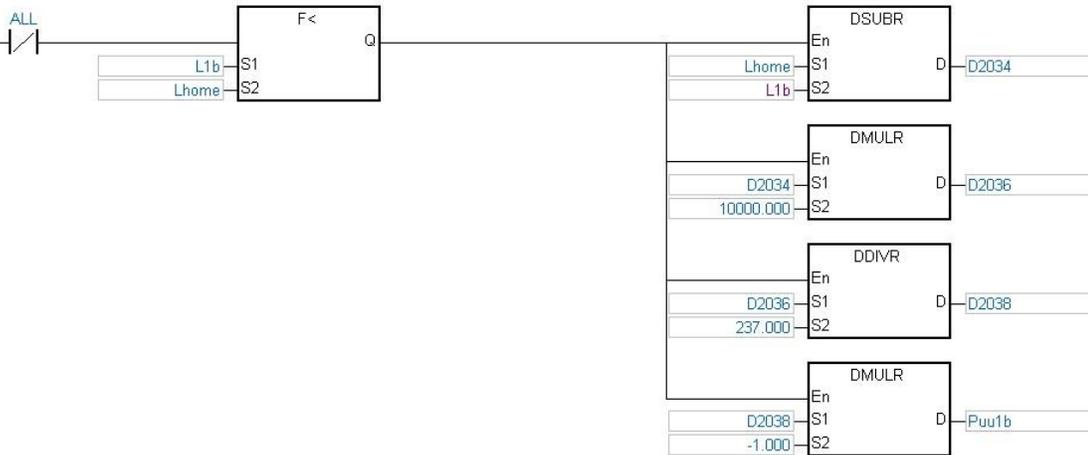
Network 8

4.8 SEGUNDO MOVIMIENTO: Si L1b > Lhome -> Convierte a pulsos y se envia



Network 9

4.9 Si L1b < Lhome -> Convierte a pulsos y se envia



Network 10

4.10 Si L1b = Lhome -> Se envia cero

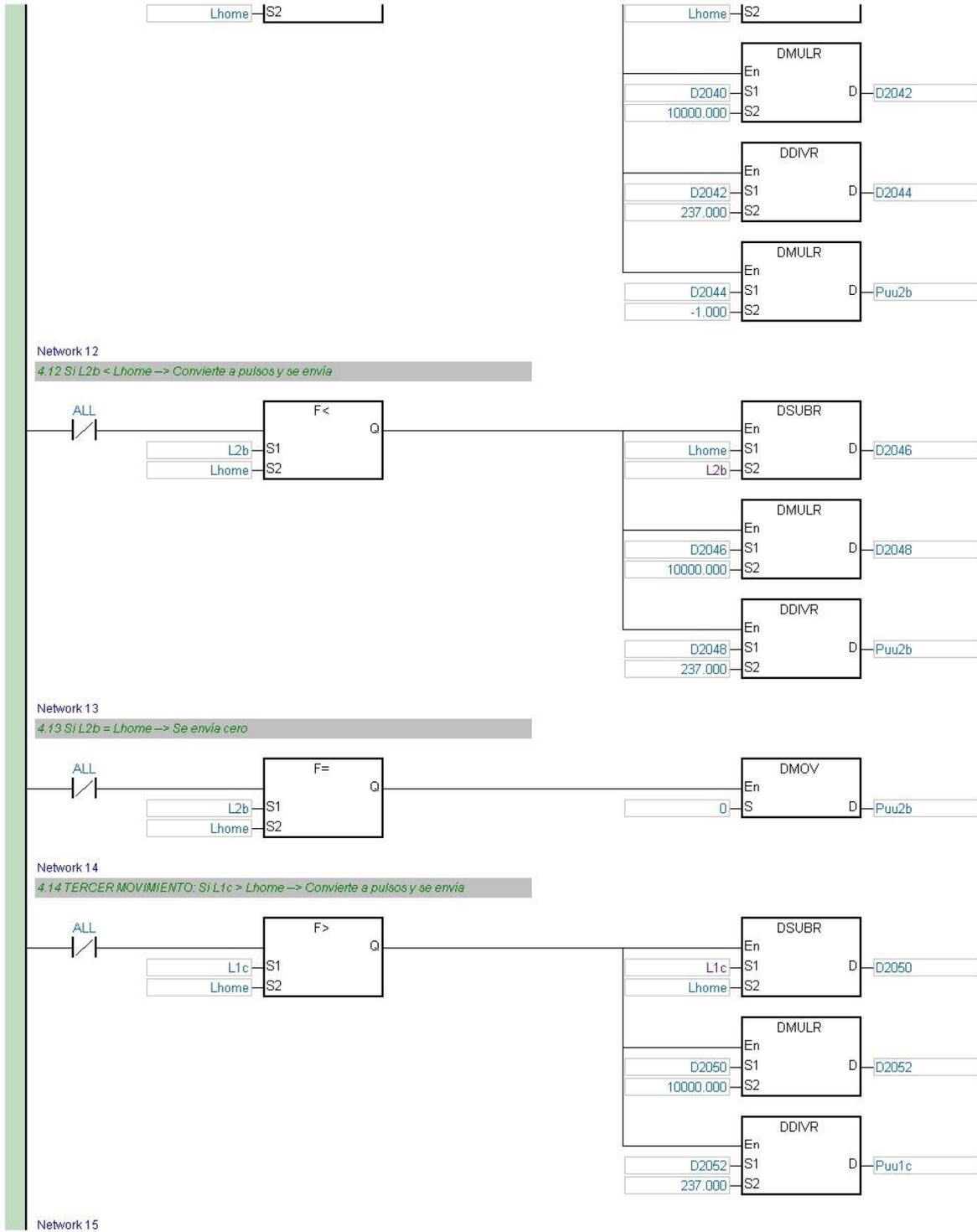


Network 11

4.11 Si L2b > Lhome -> Convierte a pulsos y se envia

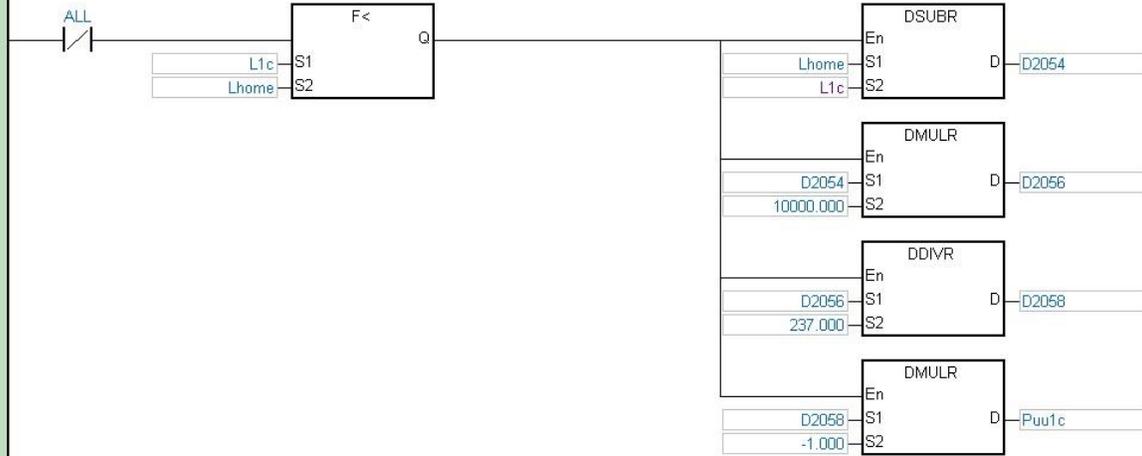


4DireccionMotores



4DireccionMotores

4.15 Si L1c < Lhome -> Convierte a pulsos y se envia



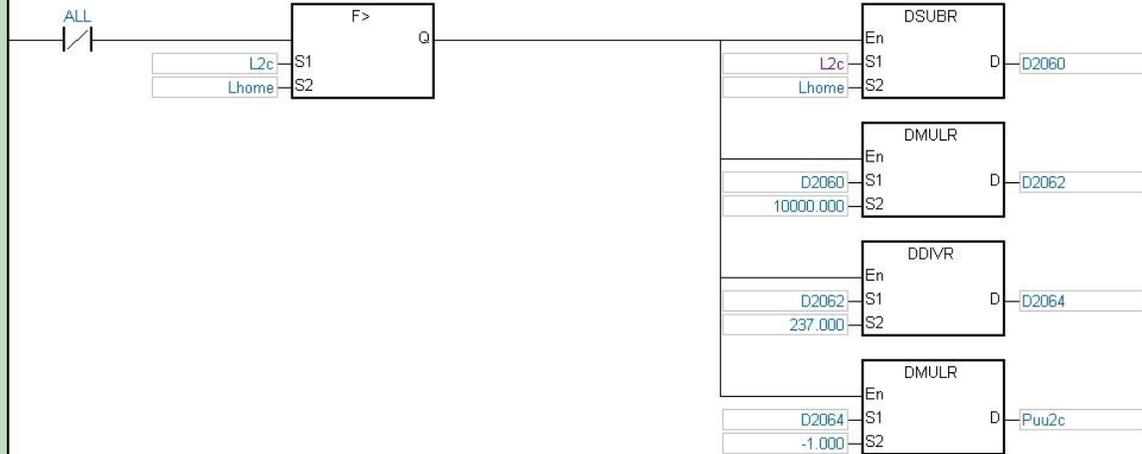
Network 16

4.16 Si L1c = Lhome -> Se envia cero



Network 17

4.17 Si L2c > Lhome -> Convierte a pulsos y se envia



Network 18

4.18 Si L2c < Lhome -> Convierte a pulsos y se envia



4DireccionMotores

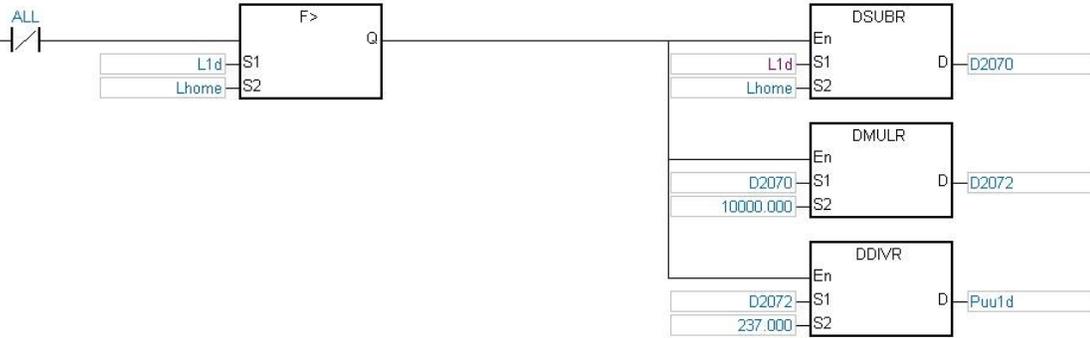
Network 19

4.19 Si L2c = Lhome -> Se envia cero



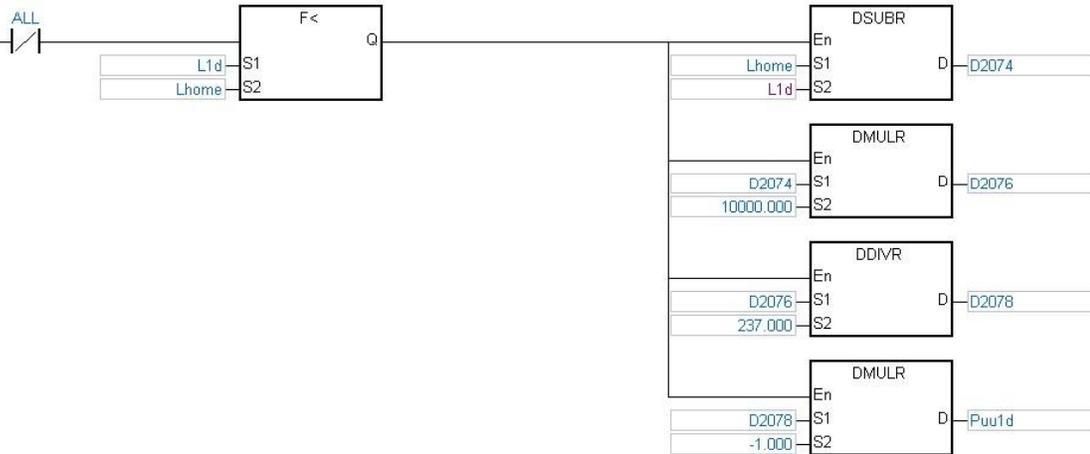
Network 20

4.20 CUARTO MOVIMIENTO: Si L1d > Lhome -> Convierte a pulsos y se envia



Network 21

4.21 Si L1d < Lhome -> Convierte a pulsos y se envia



Network 22

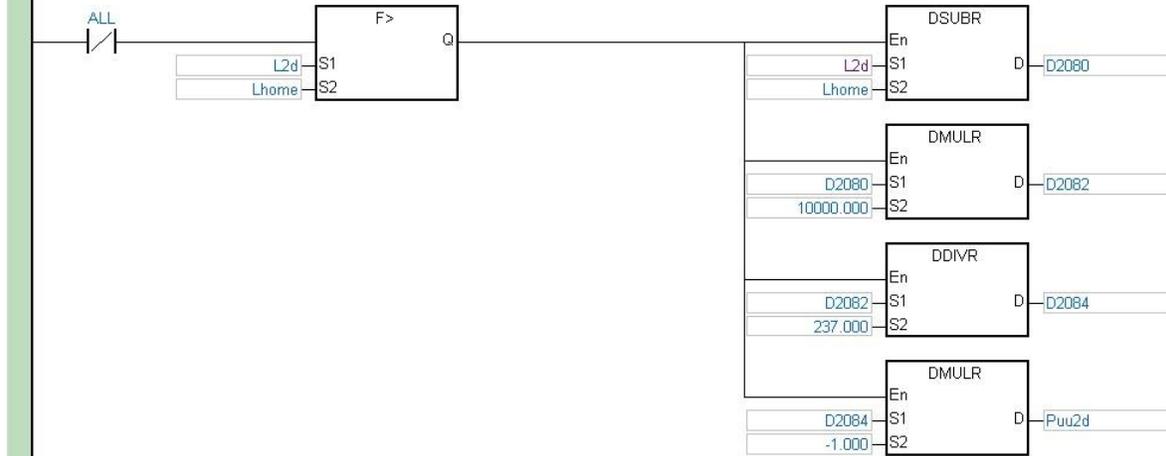
4.22 Si L1d = Lhome -> Se envia cero

4DireccionMotores



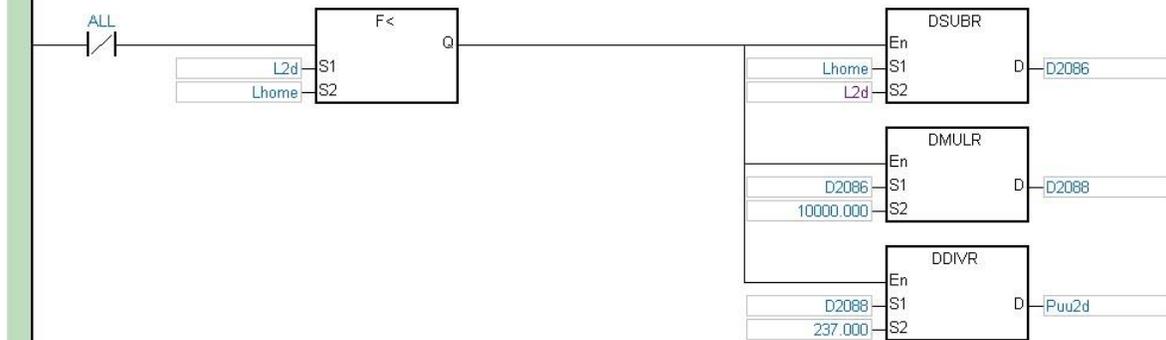
Network 23

4.23 Si L2d > Lhome -> Convierte a pulsos y se envía



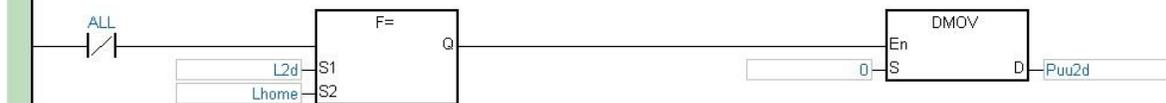
Network 24

4.24 Si L2d < Lhome -> Convierte a pulsos y se envía



Network 25

4.25 Si L2d = Lhome -> Se envía cero



Network 26

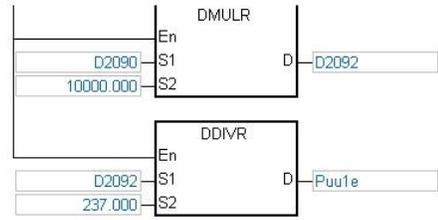
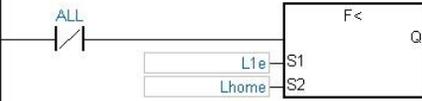
4.26 QUINTO MOVIMIENTO: Si L1e > Lhome -> Convierte a pulsos y se envía



4DireccionMotores

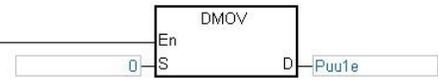
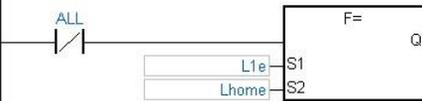
Network 27

4.27 Si L1e < Lhome -> Convierte a pulsos y se envia



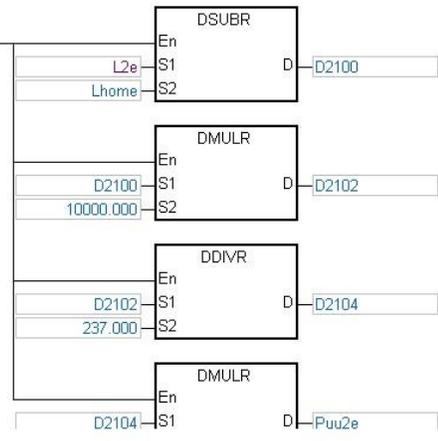
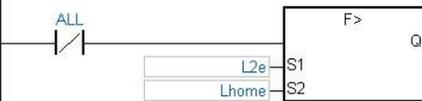
Network 28

4.28 Si L1e = Lhome -> Se envia cero



Network 29

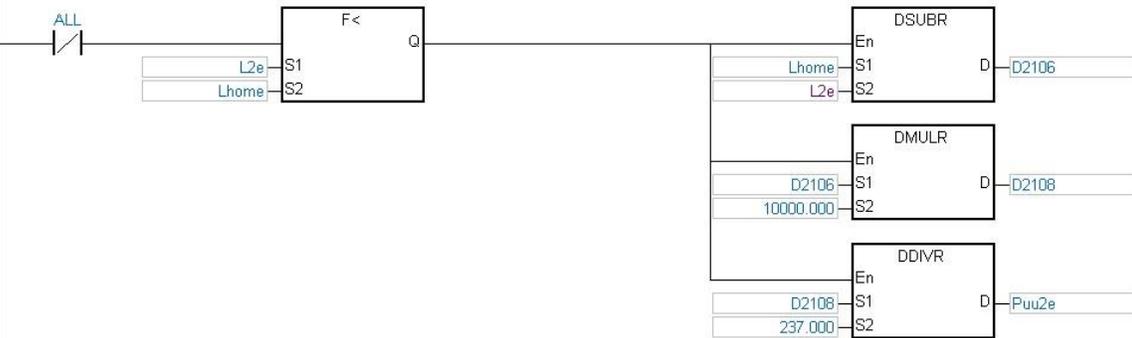
4.29 Si L2e > Lhome -> Convierte a pulsos y se envia



4DireccionMotores

Network 30

4.30 Si L2e < Lhome -> Convierte a pulsos y se envía



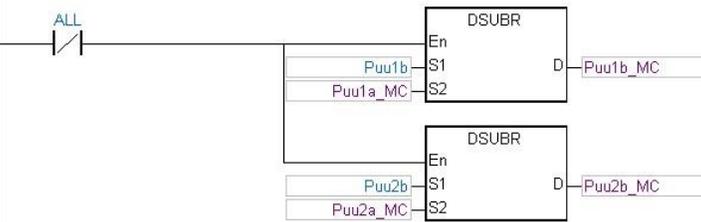
Network 31

4.31 Si L2e = Lhome -> Se envía cero



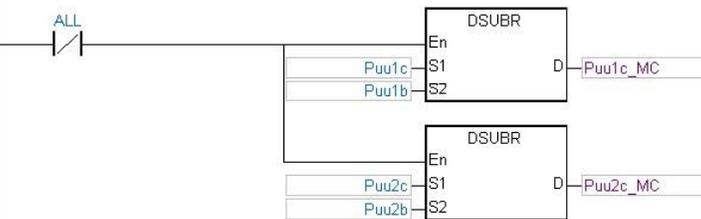
Network 32

4.32 Para el segundo movimiento de los servomotores 1 y 2



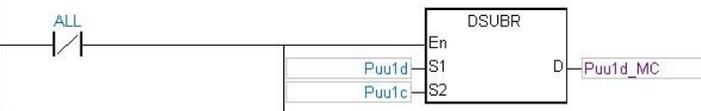
Network 33

4.33 Para el tercer movimiento de los servomotores 1 y 2

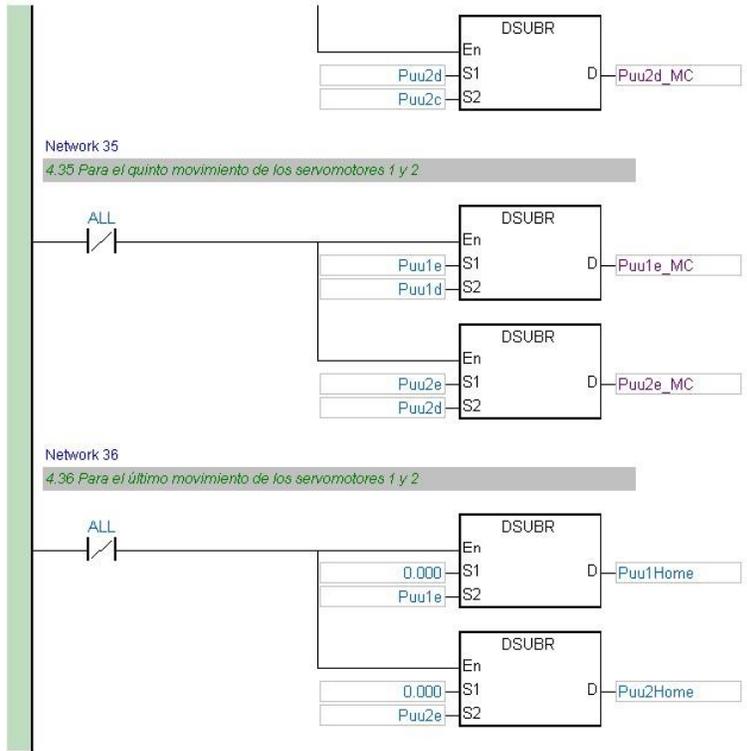


Network 34

4.34 Para el cuarto movimiento de los servomotores 1 y 2

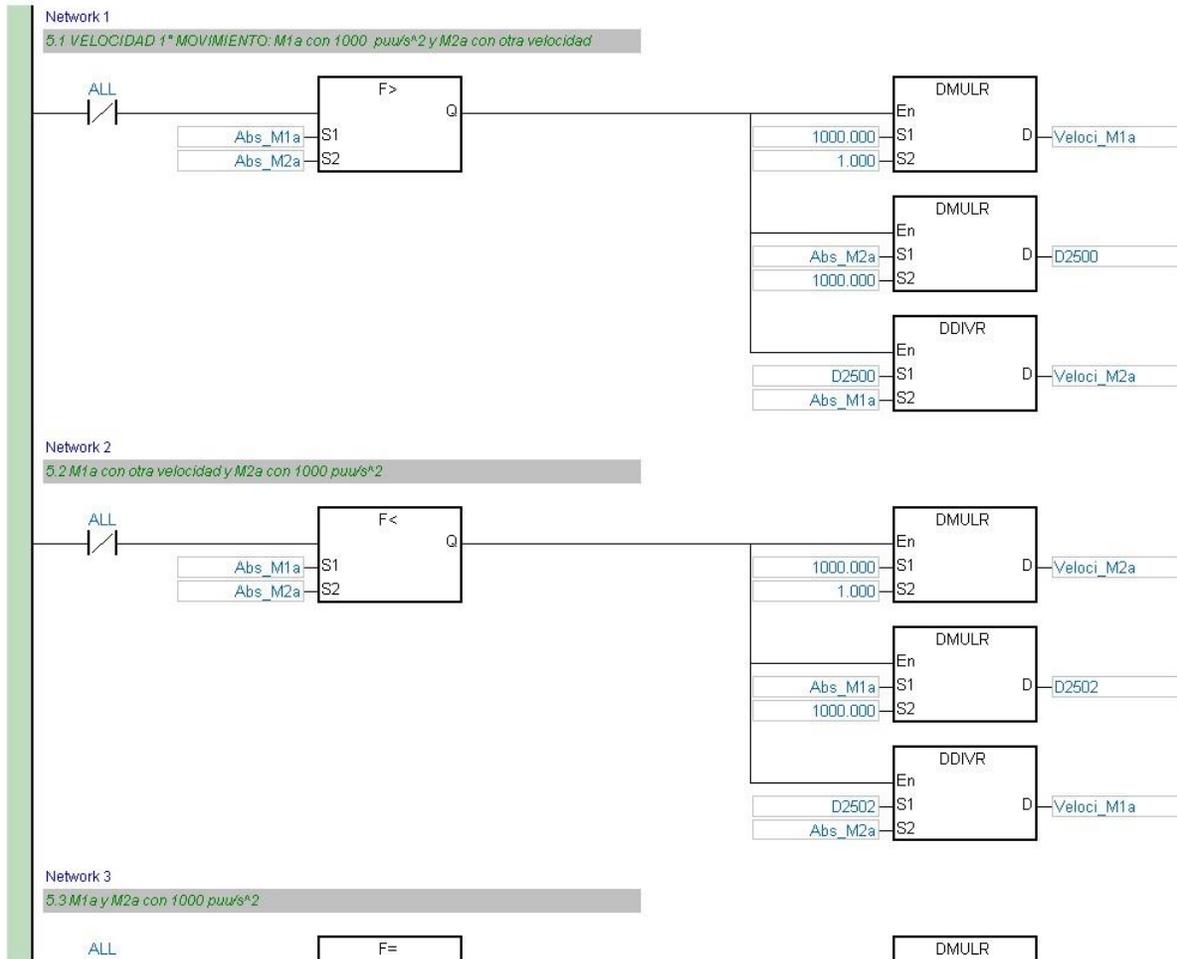


4DireccionMotores



5VelocidadMotores

Declaration	Type	Identifiers	Address	Type	Initial Value	Identifier Comment
VAR		Abs_M1a	D6010	REAL	0.0	
VAR		Abs_M2a	D6012	REAL	0.0	
VAR		Abs_M1b	D6014	REAL	0.0	
VAR		Abs_M2b	D6016	REAL	0.0	
VAR		Abs_M1c	D6018	REAL	0.0	
VAR		Abs_M2c	D6020	REAL	0.0	
VAR		Abs_M1d	D6022	REAL	0.0	
VAR		Abs_M2d	D6024	REAL	0.0	
VAR		Abs_M1e	D6026	REAL	0.0	
VAR		Abs_M2e	D6028	REAL	0.0	
VAR		Abs_M1home	D6030	REAL	0.0	
VAR		Abs_M2home	D6032	REAL	0.0	
VAR		Veloci_M1a	D6350	REAL	0.0	
VAR		Veloci_M2a	D6352	REAL	0.0	
VAR		Veloci_M1b	D6354	REAL	0.0	
VAR		Veloci_M2b	D6356	REAL	0.0	
VAR		Veloci_M1c	D6358	REAL	0.0	
VAR		Veloci_M2c	D6360	REAL	0.0	
VAR		Veloci_M1d	D6362	REAL	0.0	
VAR		Veloci_M2d	D6364	REAL	0.0	
VAR		Veloci_M1e	D6366	REAL	0.0	
VAR		Veloci_M2e	D6368	REAL	0.0	
VAR		Veloci_M1home	D6370	REAL	0.0	
VAR		Veloci_M2home	D6372	REAL	0.0	

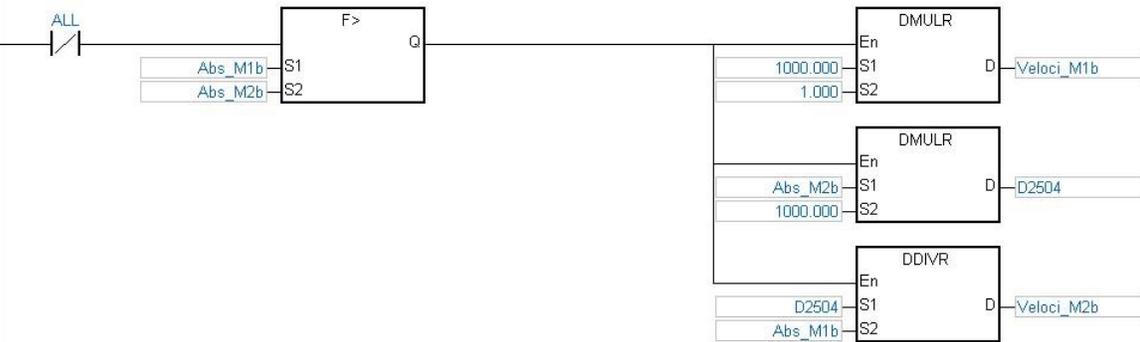


5VelocidadMotores



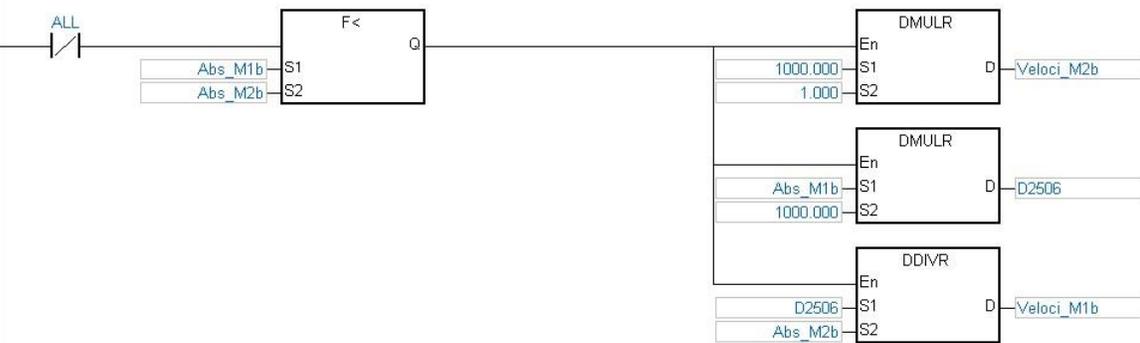
Network 4

5.4 VELOCIDAD 2ª MOVIMIENTO: M1b con 1000 puu/s² y M2b con otra velocidad



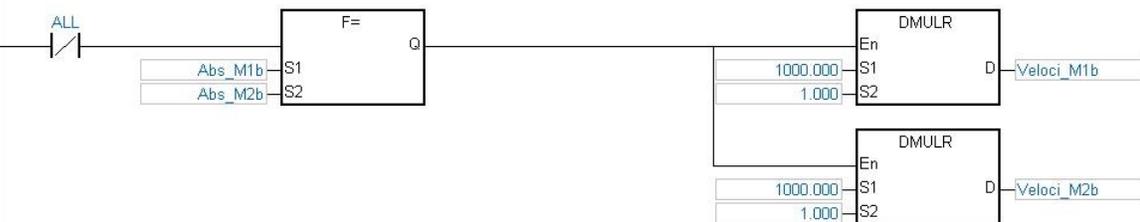
Network 5

5.5 M1b con otra velocidad y M2b con 1000 puu/s²



Network 6

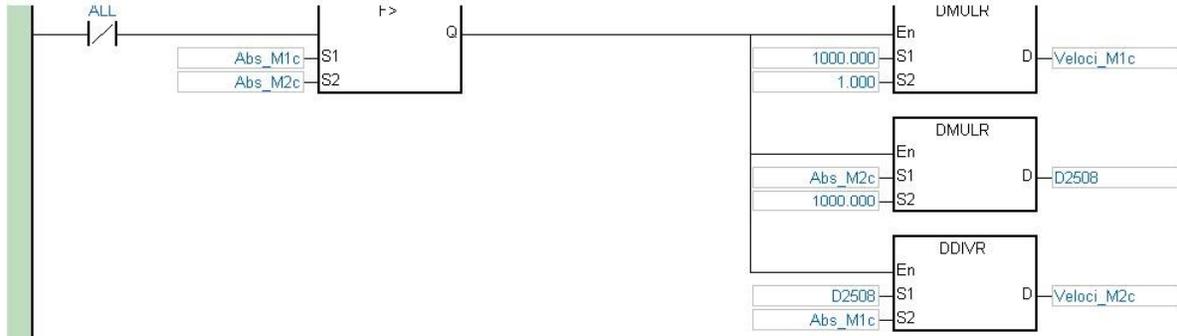
5.6 M1b y M2b con 1000 puu/s²



Network 7

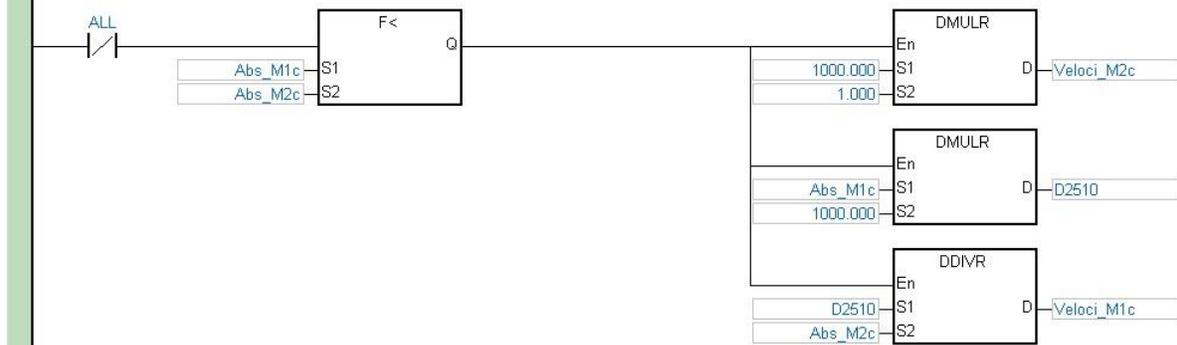
5.7 VELOCIDAD 3ª MOVIMIENTO: M1c con 1000 puu/s² y M2c con otra velocidad

5 Velocidad Motores



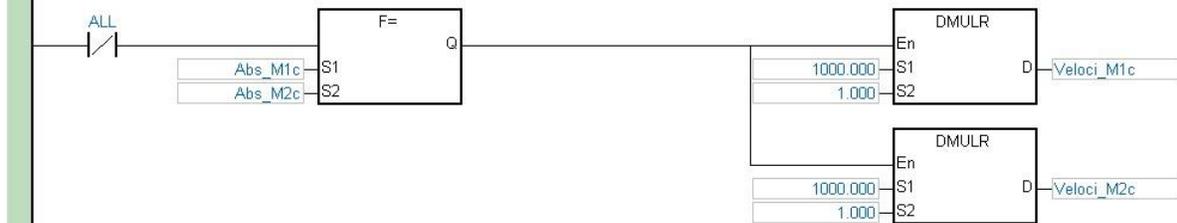
Network 8

5.8 M1c con otra velocidad y M2c con 1000 puu/s^2



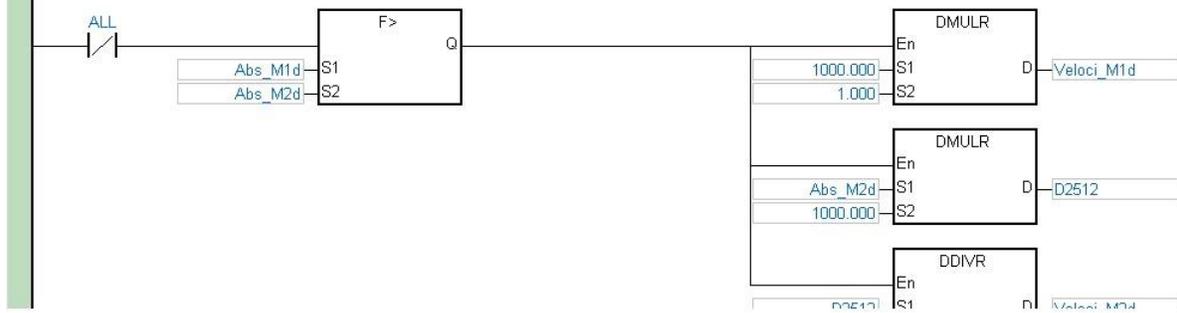
Network 9

5.9 M1c y M2c con 1000 puu/s^2



Network 10

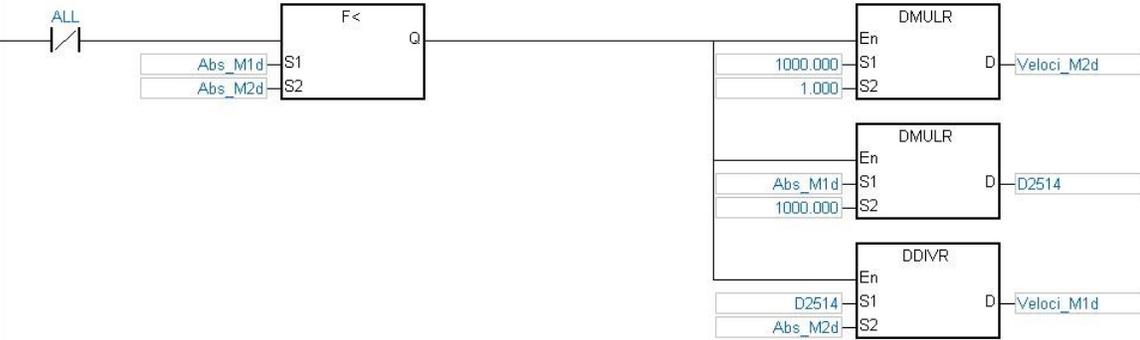
5.10 VELOCIDAD 4° MOVIMIENTO: M1d con 1000 puu/s^2 y M2d con otra velocidad



5VelocidadMotores

Network 11

5.11 M1d con otra velocidad y M2d con 1000 puu/s^2



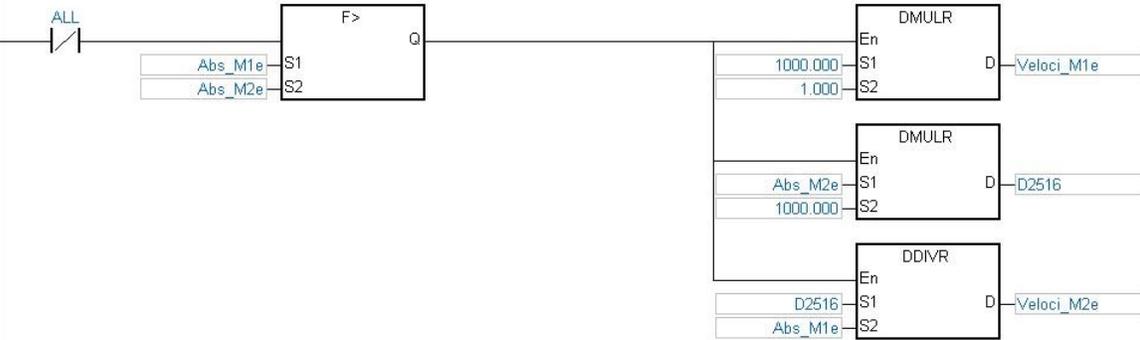
Network 12

5.12 M1d y M2d con 1000 puu/s^2



Network 13

5.13 VELOCIDAD 5° MOVIMIENTO: M1e con 1000 puu/s^2 y M2e con otra velocidad

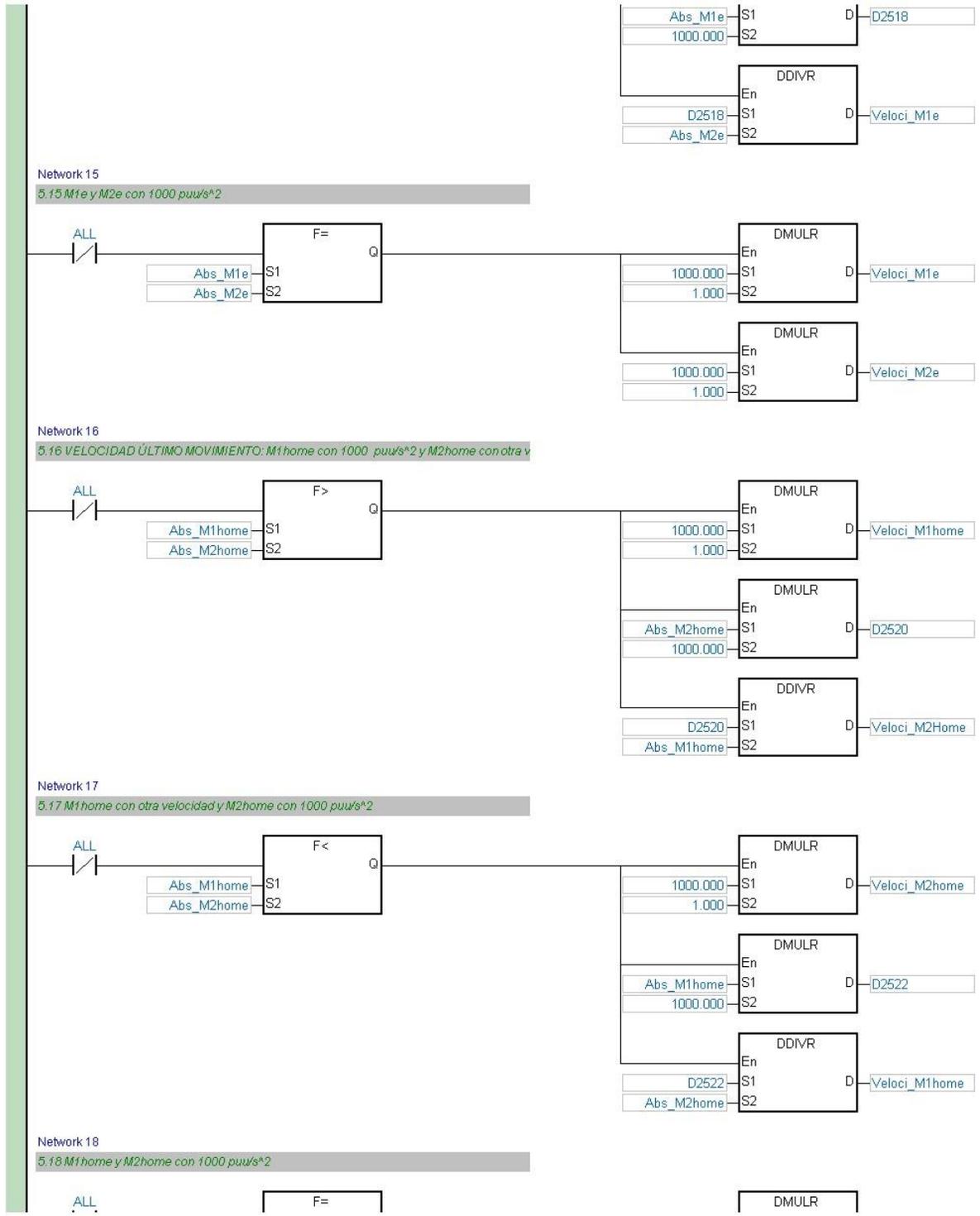


Network 14

5.14 M1e con otra velocidad y M2e con 1000 puu/s^2



5VelocidadMotores

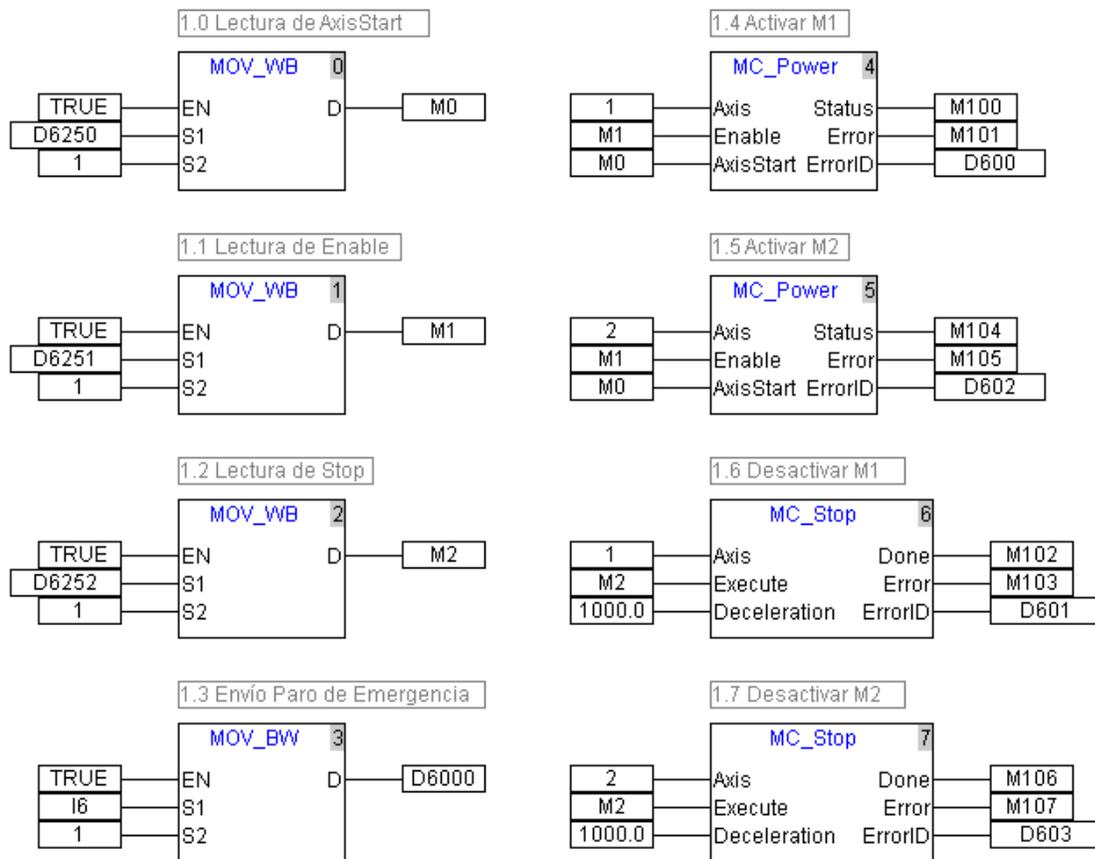




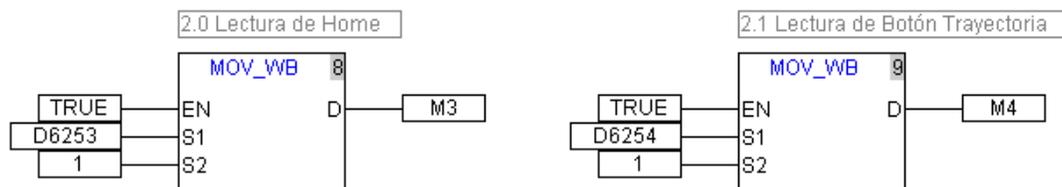
ANEXO E

CÓDIGO DEL PROGRAMA DE MOVIMIENTO EN CANOPEN BUILDER

1 BLOQUE ACTIVACIÓN DE SERVOS



2 BLOQUE MARCAS DE ACTIVACIÓN



3 BLOQUE MOVIMIENTO Y LECTURA DE DATOS

PRIMER MOVIMIENTO

3.0 Posición para M1 (ppu1a)



3.1 Posición para M2 (ppu2a)



SEGUNDO MOVIMIENTO

3.2 Posición para M1 (ppu1b)



3.3 Posición para M2 (ppu2b)



TERCER MOVIMIENTO

3.4 Posición para M1 (ppu1c)



3.5 Posición para M2 (ppu2c)



CUARTO MOVIMIENTO

3.6 Posición para M1 (ppu1d)



3.7 Posición para M2 (ppu2d)



QUINTO MOVIMIENTO

3.8 Posición para M1 (ppu1e)

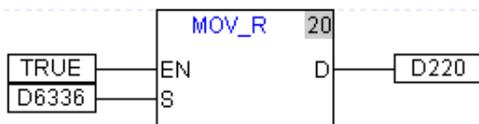


3.9 Posición para M2 (ppu2e)



ÚLTIMO MOVIMIENTO

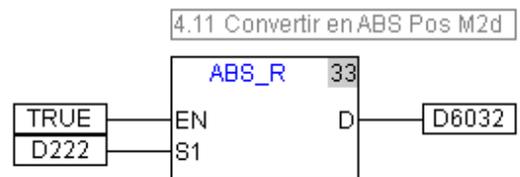
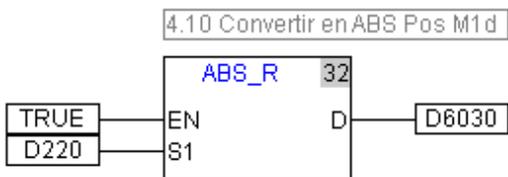
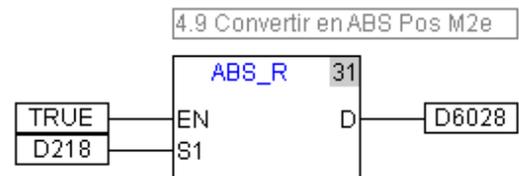
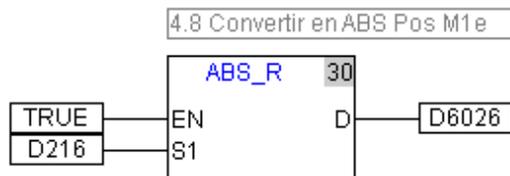
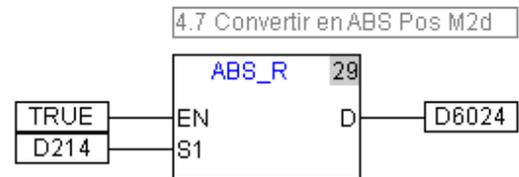
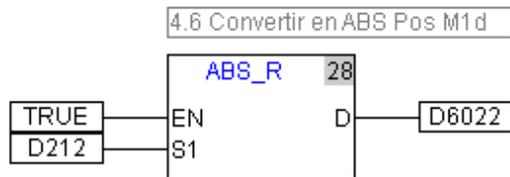
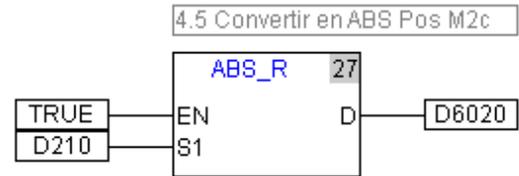
3.10 Posición para M1 (ppu1Home)



3.12 Posición para M2 (ppu2Home)



4 BLOQUE VELOCIDADES



4.12 Primera velocidad SM1



4.13 Primera velocidad SM2



4.14 Segunda velocidad SM1



4.13 Segunda velocidad SM2



4.14 Tercera velocidad SM1



4.15 Tercera velocidad SM2



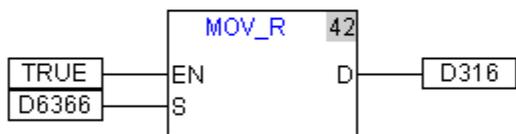
4.16 Cuarta velocidad SM1



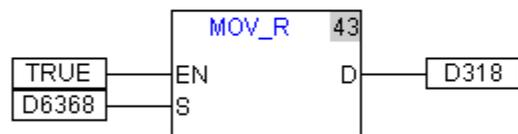
4.17 Cuarta velocidad SM2



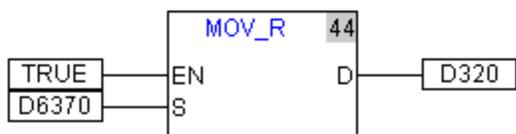
4.18 Quinta velocidad SM1



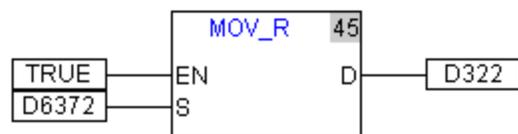
4.19 Quinta velocidad SM2



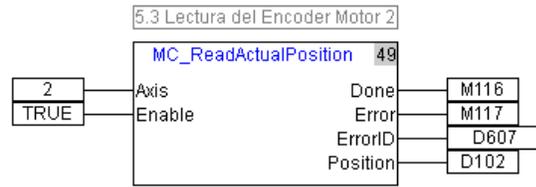
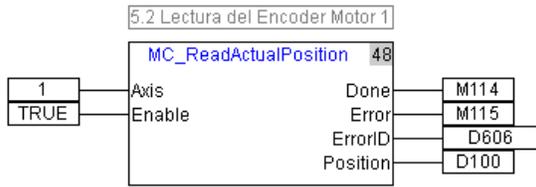
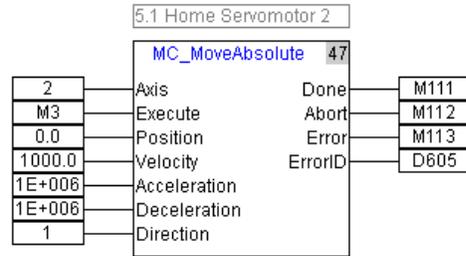
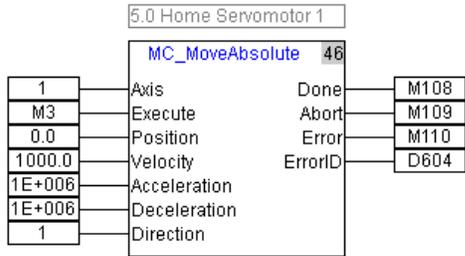
4.20 Última velocidad SM1



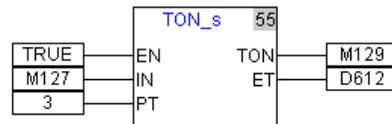
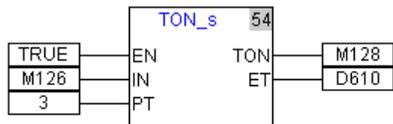
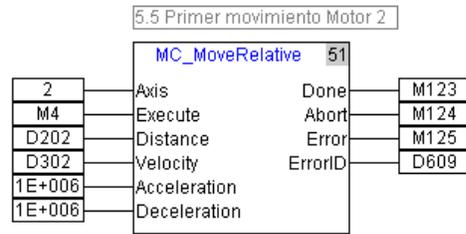
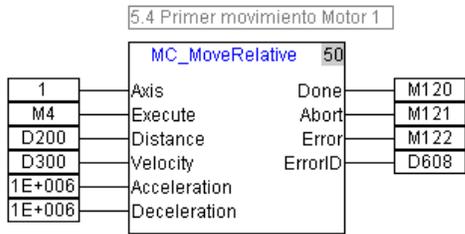
4.21 Última velocidad SM2



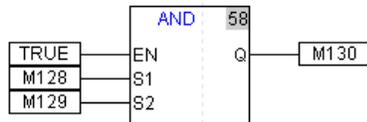
5 BLOQUE MOVIMIENTO DE POSICIÓN



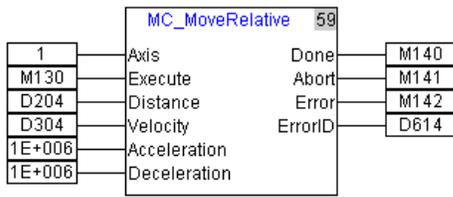
COMIENZO DE MOVIMIENTO



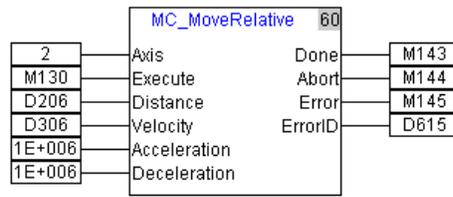
5.8 Cuando enciendan los dos temp. activar segundo movimiento



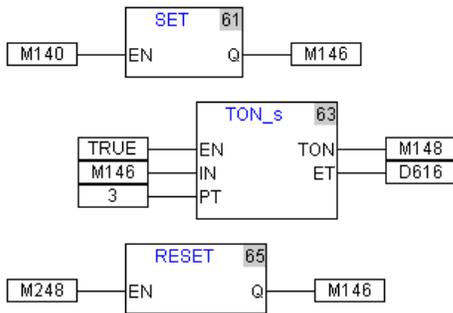
5.9 Segundo movimiento Motor 1



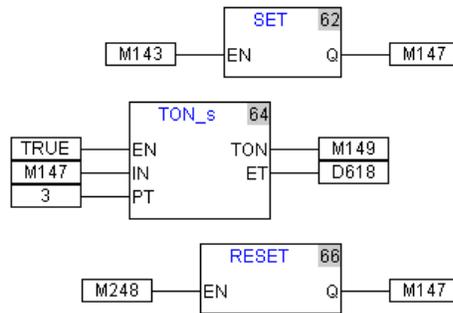
5.10 Segundo movimiento Motor 2



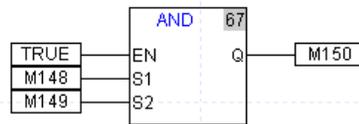
5.11 Tiempo de espera entre 2° y 3° movimiento M1



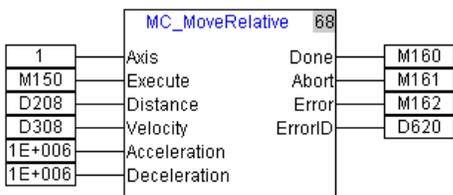
5.12 Tiempo de espera entre 2° y 3° movimiento M2



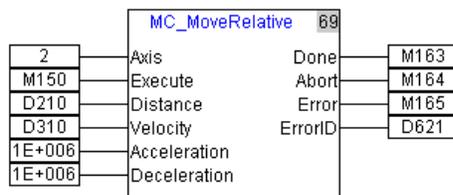
5.13 Cuando enciendan los dos temp. activar tercer movimiento



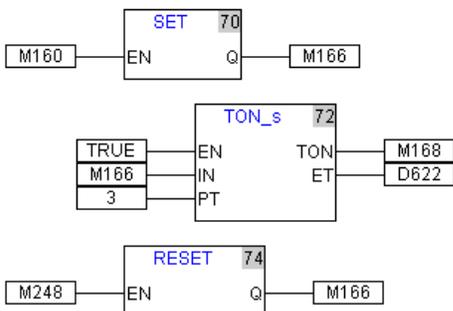
5.14 Tercer movimiento Motor 1



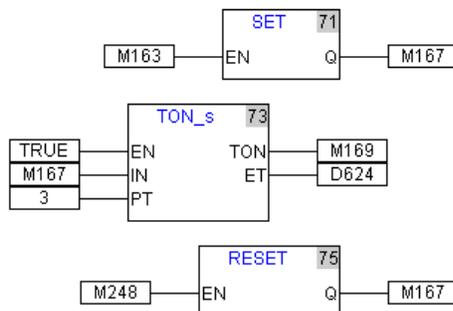
5.15 Tercer movimiento Motor 2



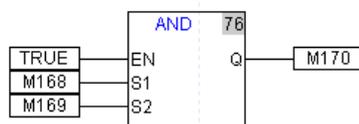
5.16 Tiempo de espera entre 3° y último movimiento M1



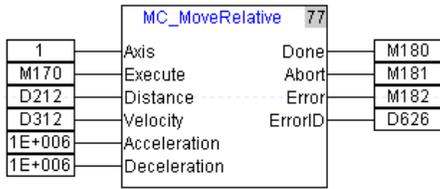
5.17 Tiempo de espera entre 3° y último movimiento M2



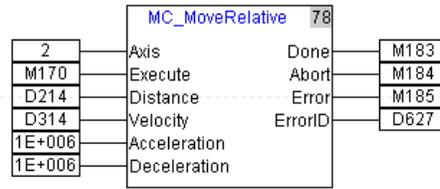
5.18 Cuando enciendan los dos temp. activar último movimiento



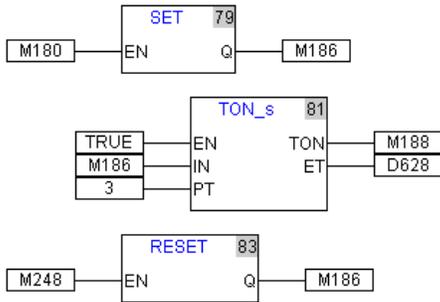
5.19 Cuarto movimiento Motor 1



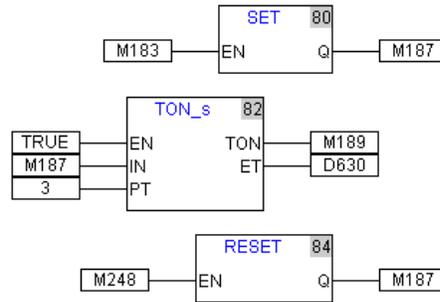
5.20 Cuarto movimiento Motor 2



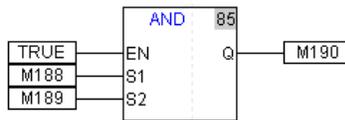
5.21 Tiempo de espera entre 4ª y 5ª movimiento M1



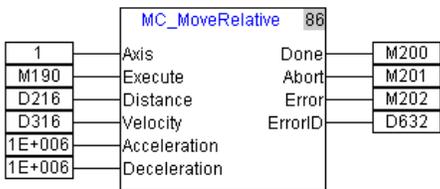
5.22 Tiempo de espera entre 4ª y 5ª movimiento M2



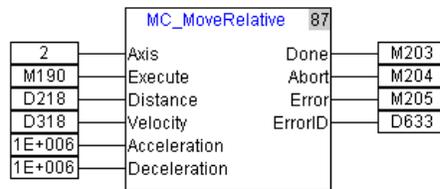
5.23 Cuando enciendan los dos temp. activar quinto movimiento



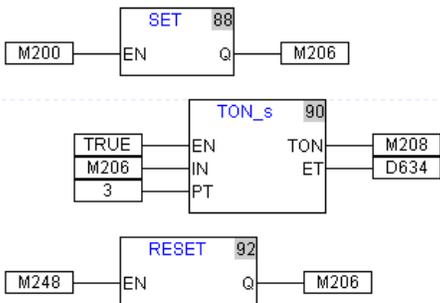
5.24 Quinto movimiento Motor 1



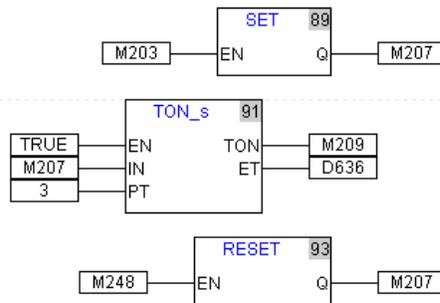
5.25 Quinto movimiento Motor 2



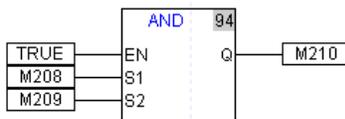
5.21 Tiempo de espera entre 5ª y último movimiento M1



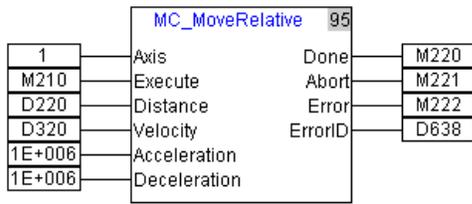
5.22 Tiempo de espera entre 5ª y último movimiento M2



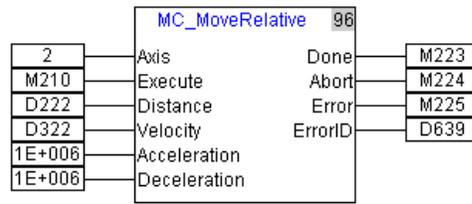
5.23 Cuando enciendan los dos temp. activar quinto movimiento



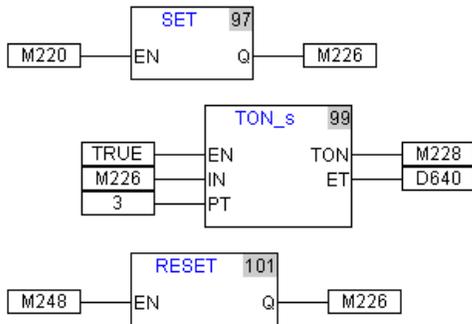
5.24 Último movimiento Motor 1



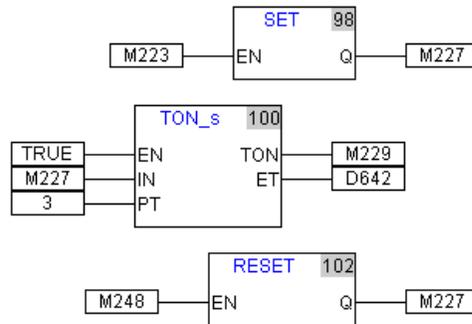
5.25 Último movimiento Motor 2



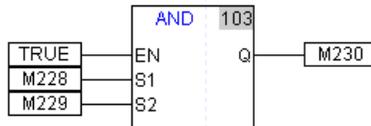
5.21 Tiempo de espera entre último movy Home M1



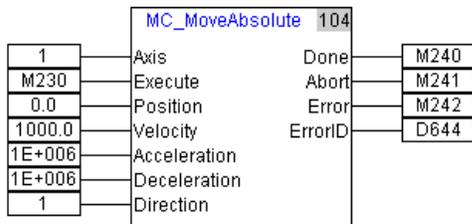
5.21 Tiempo de espera entre último movy Home M2



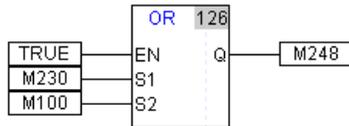
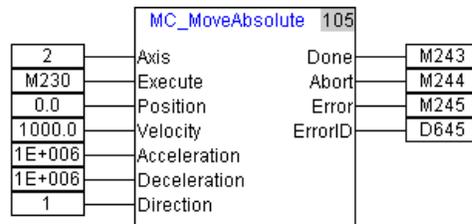
5.23 Cuando enciendan los dos temp. activar quinto movimiento



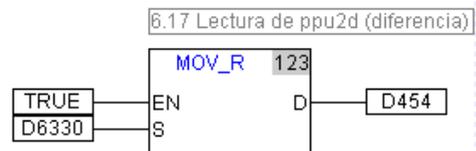
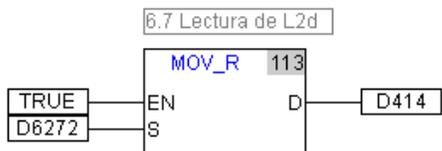
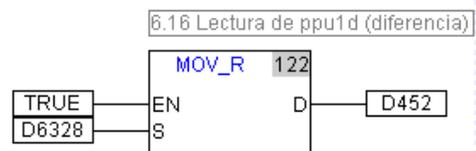
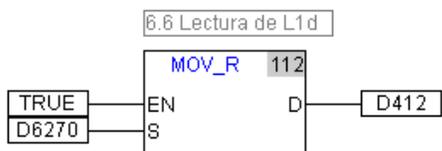
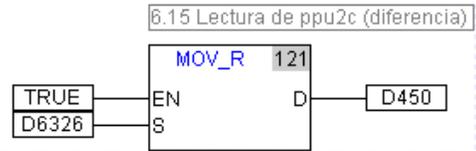
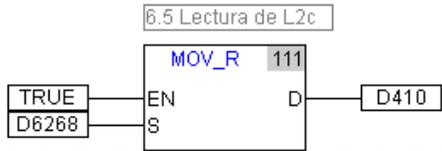
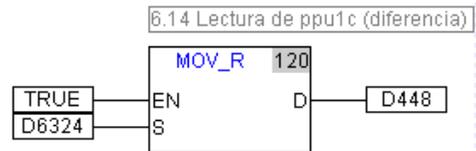
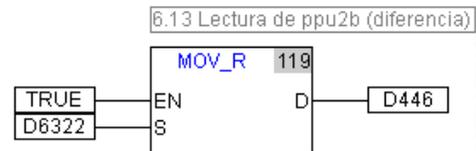
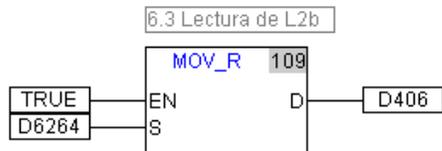
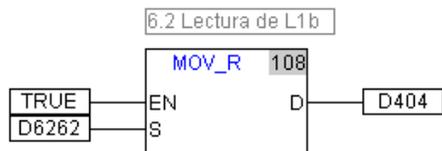
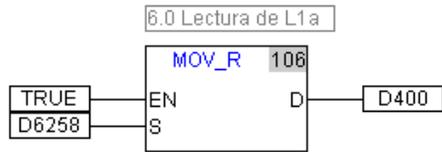
5.24 Home Servomotor 1



5.25 Home Servomotor 2



6 BLOQUE MOVIMIENTO Y LECTURA DE DATOS



6.8 Lectura de L1e



6.9 Lectura de L2e



6.18 Lectura de ppu1e (diferencia)

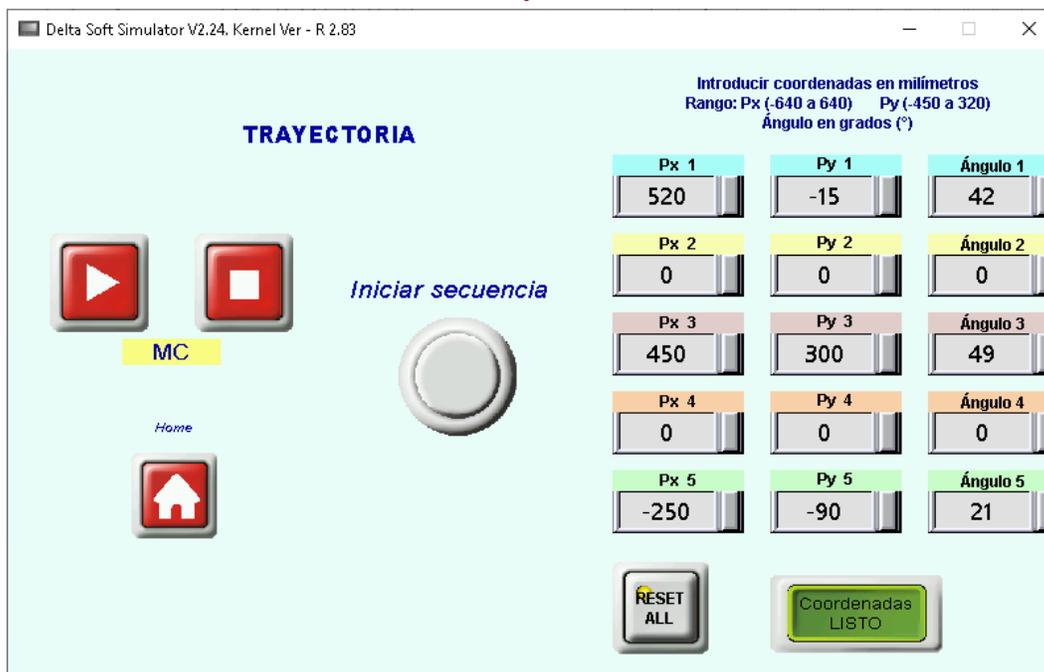


6.19 Lectura de ppu2e (diferencia)



ANEXO F

HMI EN DOPSOFT



ANEXO G

CÓDIGO DE PROGRAMA PARA LA OBTENCIÓN DE LAS COORDENADAS DE REFERENCIA CONTRA LAS REALES EN MATLAB

```
% Instituto Tecnológico de Puebla
% Alumna: Mena Huerta Claudia
% Gráficas finales de tres posiciones del sistema de Control
en el C DPR
% 14-sep-19

clc;
clear all;

C3fDes = [627; -15];
C3fRea = [626; -15];
figure(1)
plot(627, -15, 'g*')
axis([-640 640 -450 450])
hold on
plot(626, -15, 'm*')
plotv(C3fDes, 'g-')
plotv(C3fRea, 'm-')
title('Gráfica final de tercera coordenada con ángulo de
inclinación: referencia vs real');
xlabel('Eje X (mm)')
ylabel('Eje Y (mm)')
```

```

legend({'Coord. deseada (627,-15)', 'Coord. real (626,-15)',
'\surd(Xd^2+Yd^2)', '\surd(Xr^2+Yr^2)'}, 'Location', 'northeast'
)

```

```

C6fDes = [-603; 295];
C6fRea = [-603; 295];
figure(2)
plot(-603, 295, 'g*')
axis([-640 640 -450 450])
hold on
plot(-603, 295, 'm*')
plotv(C6fDes, 'g-')
plotv(C6fRea, 'm-')
title('Gráfica final de sexta coordenada con ángulo de
inclinación: referencia vs real')
xlabel('Eje X (mm)')
ylabel('Eje Y (mm)')
legend({'Coordenada deseada (-603,295)', 'Coordenada real (-
603,295)', '\surd(Xd^2+Yd^2)', '\surd(Xr^2+Yr^2)'}, 'Location', '
southwest')

```

```

C7fDes = [615; -465];
C7fRea = [615; -465];
figure(3)
plot(615,-465, 'g*')
axis([-640 640 -470 470])
hold on
plot(615,-465, 'm*')
plotv(C7fDes, 'g-')
plotv(C7fRea, 'm-')

```

```
title('Gráfica final de séptima coordenada con ángulo de
inclinación: referencia vs real')
xlabel('Eje X (mm)')
ylabel('Eje Y (mm)')
legend({'Coordenada deseada (615,-465)', 'Coordenada real (615,
-465)', '\surd(Xd^2+Yd^2)', '\surd(Xr^2+Yr^2)'}, 'Location',
'southwest')
```

ANEXO H

CÓDIGO DE PROGRAMA PARA LA OBTENCIÓN DE GRÁFICAS DE ERROR DE POSICIÓN EN MATLAB

```
% Instituto Tecnológico de Puebla
% Alumna: Mena Huerta Claudia
% Gráfica final de error de posición del sistema de Control
en el C DPR
% 14-sep-19

clc;
clear all;

C1fd = [0 290];      C1fr = [0 280];
Mod1fd = norm(C1fd);  Mod1fr = norm(C1fr);
C2fd = [0 -450];     C2fr = [0 -450];
Mod2fd = norm(C2fd);  Mod2fr = norm(C2fr);
C3fd = [627 -15];    C3fr = [626 -15];
Mod3fd = norm(C3fd);  Mod3fr = norm(C3fr);
C4fd = [-627 -15];   C4fr = [-627 -15];
Mod4fd = norm(C4fd);  Mod4fr = norm(C4fr);
C5fd = [603 295];    C5fr = [603 294];
Mod5fd = norm(C5fd);  Mod5fr = norm(C5fr);
C6fd = [-603 295];   C6fr = [-603 295];
Mod6fd = norm(C6fd);  Mod6fr = norm(C6fr);
C7fd = [615 -465];   C7fr = [616 -464];
Mod7fd = norm(C7fd);  Mod7fr = norm(C7fr);
C8fd = [-615 -465];  C8fr = [-614.5 -465];
Mod8fd = norm(C8fd);  Mod8fr = norm(C8fr);
```

```

V = [1,2,3,4,5,6,7,8];
ModDes = [Mod1fd, Mod2fd, Mod3fd, Mod4fd, Mod5fd, Mod6fd,
Mod7fd, Mod8fd];
ModReal = [Mod1fr, Mod2fr, Mod3fr, Mod4fr, Mod5fr, Mod6fr,
Mod7fr, Mod8fr];
figure(1)
plot(V, ModDes, 'b*')
axis([0 9 0 800])
hold on
plot(V, ModReal, 'r*')
title({'Gráfica de módulo de cada coordenada: '; 'deseada vs
real'; 'PRUEBAS FINALES'})
xlabel('N° de coordenada')
ylabel('Módulo de cada coordenada')
legend({'Norma L2 de Coordenadas deseadas', 'Norma L2 de
Coordenadas reales'}, ...
'Location', 'southwest')

Error1 = 100 - (Mod1fr*100/Mod1fd);
Error2 = 100 - (Mod2fr*100/Mod2fd);
Error3 = 100 - (Mod3fr*100/Mod3fd);
Error4 = 100 - (Mod4fr*100/Mod4fd);
Error5 = 100 - (Mod5fr*100/Mod5fd);
Error6 = 100 - (Mod6fr*100/Mod6fd);
Error7 = 100 - (Mod7fr*100/Mod7fd);
Error8 = 100 - (Mod8fr*100/Mod8fd);
ErrorFinal = [Error1, Error2, Error3, Error4, Error5, Error6,
Error7, Error8];
figure(2)
bar(ErrorFinal, 'b')

```

```

axis([0 9 -1 10])
title({'Gráfica porcentaje de error de posición según su
norma'; 'PRUEBAS FINALES'})
xlabel('N° de coordenada')
ylabel('Porcentaje de error')
text(1, Error1, num2str(Error1), 'HorizontalAlignment',
'center',...
'VerticalAlignment', 'bottom')
text(2, Error2, num2str(Error2), 'HorizontalAlignment',
'center',...
'VerticalAlignment', 'bottom')
text(3, Error3, num2str(Error3), 'HorizontalAlignment',
'center',...
'VerticalAlignment', 'bottom')
text(4, Error4, num2str(Error4), 'HorizontalAlignment',
'center',...
'VerticalAlignment', 'bottom')
text(5, Error5, num2str(Error5), 'HorizontalAlignment',
'center',...
'VerticalAlignment', 'bottom')
text(6, Error6, num2str(Error6), 'HorizontalAlignment',
'center',...
'VerticalAlignment', 'bottom')
text(7, 25*Error7, num2str(Error7), 'HorizontalAlignment',
'center',...
'VerticalAlignment', 'bottom')
text(8, Error8, num2str(Error8), 'HorizontalAlignment',
'center',...
'VerticalAlignment', 'bottom')

Errorx1 = 100 - (C1fr(1)*100/C1fd(1));

```

```

Errorx2 = 100 - (C2fr(1)*100/C2fd(1));
Errorx3 = 100 - (C3fr(1)*100/C3fd(1));
Errorx4 = 100 - (C4fr(1)*100/C4fd(1));
Errorx5 = 100 - (C5fr(1)*100/C5fd(1));
Errorx6 = 100 - (C6fr(1)*100/C6fd(1));
Errorx7 = 100 - (C7fr(1)*100/C7fd(1));
Errorx8 = 100 - (C8fr(1)*100/C8fd(1));
Errorx = [Errorx1, Errorx2, Errorx3, Errorx4, Errorx5,
Errorx6, Errorx7, Errorx8];
figure(3)
bar(Errorx, 'b')
axis([0 9 -1 10])
title({'Gráfica porcentaje de error de posición sobre eje
"x"; 'PRUEBAS FINALES'})
xlabel('N° de coordenada')
ylabel('Porcentaje de error en "x"')
text(1, 0, num2str(0), 'HorizontalAlignment', 'center',...
'VerticalAlignment', 'bottom')
text(2, 0, num2str(0), 'HorizontalAlignment', 'center',...
'VerticalAlignment', 'bottom')
text(3, Errorx3, num2str(Errorx3), 'HorizontalAlignment',
'center',...
'VerticalAlignment', 'bottom')
text(4, Errorx4, num2str(Errorx4), 'HorizontalAlignment',
'center',...
'VerticalAlignment', 'bottom')
text(5, 0, num2str(Errorx5), 'HorizontalAlignment',
'center',...
'VerticalAlignment', 'bottom')
text(6, 0, num2str(Errorx6), 'HorizontalAlignment',
'center',...

```

```

        'VerticalAlignment', 'bottom')
text(7, 25*Errorx7, num2str(Errorx7), 'HorizontalAlignment',
'center',...
        'VerticalAlignment', 'bottom')
text(8, Errorx8, num2str(Errorx8), 'HorizontalAlignment',
'center',...
        'VerticalAlignment', 'bottom')

Errory1 = 100 - (C1fr(2)*100/C1fd(2));
Errory2 = 100 - (C2fr(2)*100/C2fd(2));
Errory3 = 100 - (C3fr(2)*100/C3fd(2));
Errory4 = 100 - (C4fr(2)*100/C4fd(2));
Errory5 = 100 - (C5fr(2)*100/C5fd(2));
Errory6 = 100 - (C6fr(2)*100/C6fd(2));
Errory7 = 100 - (C7fr(2)*100/C7fd(2));
Errory8 = 100 - (C8fr(2)*100/C8fd(2));
Errory = [Errory1, Errory2, Errory3, Errory4, Errory5,
Errory6, Errory7, Errory8];
figure(4)
bar(Errory, 'b')
axis([0 9 -4 10])
title({'Gráfica porcentaje de error de posición sobre eje
"y"; 'PRUEBAS FINALES'})
xlabel('N° de coordenada')
ylabel('Porcentaje de error en "y"')
text(1, Errory1, num2str(Errory1), 'HorizontalAlignment',
'center',...
        'VerticalAlignment', 'bottom')
text(2, Errory2, num2str(Errory2), 'HorizontalAlignment',
'center',...

```

```
    'VerticalAlignment', 'bottom')
text(3, Errory3, num2str(Error3), 'HorizontalAlignment',
'center',...
    'VerticalAlignment', 'bottom')
text(4, Errory4, num2str(Error4), 'HorizontalAlignment',
'center',...
    'VerticalAlignment', 'bottom')
text(5, Errory5, num2str(Error5), 'HorizontalAlignment',
'center',...
    'VerticalAlignment', 'bottom')
text(6, Errory6, num2str(Error6), 'HorizontalAlignment',
'center',...
    'VerticalAlignment', 'bottom')
text(7, Errory7, num2str(Error7), 'HorizontalAlignment',
'center',...
    'VerticalAlignment', 'bottom')
text(8, 0, num2str(Error8), 'HorizontalAlignment',
'center',...
    'VerticalAlignment', 'bottom')
```

ANEXO I

ARTÍCULO PUBLICADO N° 1



DESARROLLO DE UNA ARQUITECTURA DEL SISTEMA DE ACTUACIÓN DE UN ROBOT PARALELO ACTIVADO POR CABLES

Renata Cebada Posadas¹, Claudia Mena Huerta², Vicente Ramírez Palacios³, Sergio J. Torres Méndez⁴ y José R. Mendoza Vázquez⁵

^{1,2,3,4,5} Tecnológico Nacional de México
Instituto Tecnológico de Puebla
Av. Tecnológico 420 Col. Maravillas
Puebla, Puebla, México, C.P. 72220.



renata12363@hotmail.com

Recibido: Mayo 7, 2018.
Recibido en forma revisada: Julio 10, 2018.
Aceptado: Octubre 30, 2018.

Resumen: En este documento se presenta el desarrollo de una arquitectura del sistema de actuación de un robot paralelo activado por cables. La función del robot es almacenar objetos, posicionándolos en compartimientos con alta precisión y rapidez. La estructura del robot cuenta con una plataforma móvil como actuador final, que se encuentra suspendida mediante cables sujetos a la base del robot. La estructura del sistema de actuación consistirá en dos servomotores cuyo funcionamiento estará gobernado por un módulo de control de movimiento (MC) y un controlador lógico programable (PLC). El PLC será retroalimentado mediante una cámara y/o encoder y controlado por una interfaz HMI, además se describen los diversos protocolos de comunicación que se utilizan entre los dispositivos del sistema, la metodología de desarrollo empleada, así como la descripción del sistema logrado y sus componentes. Finalmente se presentan los resultados experimentales del sistema propuesto.

Palabras clave: Robots activados por cables, arquitectura, servomotor, servo drive, PLC.

Abstract: In this article is presented the design and the construction of an actuation system to a parallel cable-based robot. The robot's function is to store objects, placing them in boxes with high accuracy and velocity. The robot's structure has a mobile platform as a final actuator, the mobile platform is suspended by cables attached to the base of the robot. The

actuation system's structure consists in two servomotors whose behavior will be governed by a movement control module (MC) and a programmable logic controller (PLC). The PLC will be provided with feedback by a camera y/o encoder and controlled by a HMI interface, in addition, it is described the diverse communication protocols used among the system's devices as well as the methodology of the system used and the description of the achieved system and their components. Finally, it is presented the experimental results of the proposed design.

✦ **Keywords:** *Cable-based robots, architecture, servomotor, servo drive, PLC.*

Introducción

Los robots paralelos son aquellos sistemas que poseen estructuras de lazo cerrado, con varias cadenas cinemáticas conectando la base y una plataforma móvil, los cuales presentan muy buen rendimiento en términos de exactitud. Han sido usados en diversas aplicaciones como por ejemplo en el campo de la astronomía, así como también están tomando gran popularidad en la industria y para simuladores de vuelo (Merlet, 2012). La característica de la mayoría de los robots paralelos existentes es que su diseño incluye eslabones rígidos, los cuales no pueden modificar su longitud ya definida.

Un tipo particular de los robots paralelos son los robots paralelos activados por cables. Las estructuras robóticas accionadas por cables se componen normalmente de una parte estructural fija y una plataforma móvil unidas mediante un número determinado de cables. El control en posición y orientación de la plataforma móvil se consigue mediante varios servomotores situados a lo largo de la estructura y/o la plataforma móvil que permiten controlar en tensión y longitud los cables.

Debido a que se puede modificar la longitud de los cables, es notoria la diferencia respecto a los robots paralelos que tienen eslabones rígidos, es decir, los robots activados por cables poseen escalabilidad, ya que cables de gran longitud pueden ser almacenados en carretes o tambores permitiendo que el espacio de trabajo del robot varíe de unos pocos centímetros a metros (Nicolas Riehl, 2010).

Algunas aplicaciones de los robots paralelos activados por cables son: robots tipo grúa, sistemas de construcción automatizados (Robert L. Williams, 2008), en (Stefano Masiero, 2009) se diseña e implementa un robot activado por cables (NeReBot) para rehabilitación en personas con apoplejía y en (Tobias Bruckmann, 2013) nos muestran una familia de tres tipos de robots paralelos activados por cables utilizado para aplicaciones industriales.

Algunos estudios sobre este tipo de robots aparecen en (Tej Dallel, 2011), donde se propone un control basado en visión del diseño de un robot paralelo cableado, en (Mitchell Rushton, 2018) se presenta un análisis de la posibilidad de usar cables para eliminar vibraciones y en (B. Zi, 2008) un modelo dinámico junto con un control activo para este tipo de robots, también en (Mohammad A. Khosravi, 2014) se propone un control PID robusto para robots cableados totalmente restringidos y en (Samir Lahouar, 2009) una planificación de rutas libre de colisiones para robots paralelos activados por cables.

Metodología

La metodología del sistema consta de seis etapas (Figura 1), comenzando por los requerimientos de diseño y selección de componentes. En esta etapa se establecen los elementos necesarios para la realización del proyecto de acuerdo al diseño requerido. Posteriormente se presenta la determinación de los protocolos de comunicación, elegidos de acuerdo a la velocidad demandada por la aplicación del sistema, para especificar la arquitectura y descripción de los componentes, los cuales son elegidos de acuerdo a su compatibilidad, disponibilidad y presupuesto. En desarrollo del prototipo es diseñado el layout de la distribución de los componentes en el módulo de automatización y control. Finalmente se muestran las pruebas de comunicación e integración para obtener un funcionamiento que corresponda a los resultados deseados.



Figura 1. Metodología del sistema

Determinación de requerimientos de diseño

Módulo de Ethernet: Se busca una conexión que utilice un protocolo de comunicación de ethernet para establecer comunicación entre el módulo de control de movimiento (MC), el panel HMI, la cámara y la PC. En la red a trabajar se prefiere utilizar ethernet debido a un criterio de velocidad de transmisión, específicamente por el tiempo de procesamiento de la imagen adquirida para calcular las coordenadas y su retroalimentación con la información hacia el PLC.

PLC: El cual será el encargado de ejecutar el algoritmo del sistema, de la detección de las señales de entrada y/o activación de las salidas a partir de la información recibida por el encoder y/o la cámara, por lo que estará en constante comunicación con el módulo de control de movimiento (MC), el panel HMI y con la PC, informando el estado del proceso.

CPU Motion: Es el responsable de indicar el movimiento de los servomotores gobernados por los servo drives, este módulo de control viene integrado junto con el PLC y funciona basado en CANopen. Posee un puerto de comunicación ethernet compatible con el protocolo Modbus TCP.

Switch: Cuya función será interconectar el panel HMI, la cámara, el módulo de control de movimiento (MC) y la PC, formando una red área local (LAN) cuyas especificaciones técnicas siguen el estándar IEEE 802.3 intercambiando mensajes entre los diferentes dispositivos bajo el protocolo TCP/IP.

Servo drives: Los cuales decodifican la señal que viene del encoder y que además sirve como retroalimentación para un control total en lazo cerrado. Gracias al servo drive se puede posicionar a una velocidad y torque deseado la flecha del motor para posteriormente enviar la información al PLC y al módulo de control de movimiento (MC).

Cámara: Para determinar la posición del efector final del robot por medio de un algoritmo de procesamiento de imágenes como sensor de retroalimentación al sistema de lazo cerrado.

Servomotores: Conectados a los servo drives, los cuales presentan gran exactitud en el control de velocidad y posición, alta fiabilidad de funcionamiento y pérdidas en el rotor muy bajas.

PC: Si bien el PLC podría tener las funciones grabadas y no requerir la conexión de la computadora, se utilizará como HMI, observar en línea el programa y el control de los drivers.

Panel HMI: Es donde se desarrollará la interfaz hombre-máquina de control. En esta interfaz el usuario podrá supervisar y/o indicar instrucciones para el manejo del equipo de manera remota mediante aplicaciones gráficas que ilustran el comportamiento de las variables de control.

Selección de componentes

PLC y Módulo de Control de Movimiento (MC)

El PLC que se seleccionó para el sistema es el modelo DVP10MC11T de la marca Delta. El MC y el PLC se encuentran dentro del mismo módulo de control, el cual cuenta con dos puertos de comunicación COM (RS-232 y RS-485), un puerto de Ethernet, una interfaz para el encoder, un puerto de comunicación CANopen, 8 puertos de entradas digitales y 4 puertos de salidas digitales a nivel transistor. Su ciclo de trabajo es estructurado y soporta máximo 8 módulos de extensión analógicos y 7 módulos de comunicación.

Módulo de Ethernet

El módulo de ethernet DVPEN01-SL marca Delta autodetecta la velocidad de transmisión 10/100 Mbps, admite el protocolo MODBUS TCP y el intercambio de datos punto a punto (con una longitud máxima de 200 bytes). Posee dos puertos de comunicación (RJ-45 y RS-232).

Switch

El DVS-008100 marca Delta el cual soporta el protocolo IEEE 802.1p basado en QoS (Quality of Service), es compatible con diversos protocolos de comunicación tales como Ethernet/IP, Profinet, CC-Link y posee protección contra tormentas de broadcast. Cuenta con 8 puertos RJ45 y su capacidad de conmutación es de 1.6 Gbps.

Servo drives y Servomotores

Se utilizaron servo drives con modelo ASDA-A2-M y servomotores TAP-CB04, ambos marca Delta, donde los servo drives poseen un encoder con una resolución de 20 bits (1280000 pulsos por revolución) para lograr un control de alta precisión. Se comunican por medio del protocolo CANopen al PLC y pueden conectarse hasta un máximo de 127 servo drives. Los servomotores pueden mantener una rotación estable a baja velocidad y su tiempo de establecimiento es por debajo de 1 ms mientras que su repuesta de frecuencia llega hasta 1kHz.

Panel HMI

Finalmente se optó por el touchpanel DOP-B10E615 marca Delta de 10.1" (1024 x 600 pixeles) LCD, que admite tarjeta SD y posee tres puertos COM (RS232, RS422, RS485). La transmisión y descarga de datos puede ser por RS232, USB y Ethernet.

Determinación de protocolos de comunicación

Ethernet TCP/IP

Se dispuso a usar una red Ethernet para la comunicación entre los dispositivos de comunicación como el PLC, el panel de control y la cámara, por ser un estándar de comunicación abierto (el desarrollo de una conexión a Modbus TCP/IP no requiere ningún componente de marca ni compra de licencia), donde su extendido uso supone una reducción de los gastos de conexión, un mayor rendimiento y la probabilidad de incorporar nuevas funciones.

Su velocidad de trabajo (100 Mb/s) nos permite una rápida comunicación de la cámara al mandar la posición real del efector final, así como también al mandar al sistema de control las coordenadas deseadas. La arquitectura del sistema puede evolucionar sin ninguna dificultad si posteriormente se requiere usar otros dispositivos o bien, ampliar el número de elementos, siempre y cuando se respete el número máximo de dispositivos a conectar en el módulo de Ethernet (16 dispositivos).

CANopen

Por otra parte, se decidió usar una red CANopen para la comunicación del Motion Control con los Servo Drives. Esa determinación radica en que CANopen es una red abierta admitida por más de 400 empresas de todo el mundo. CANopen es un protocolo orientado a mensajes, por lo tanto, no se orienta a direcciones IP, a los cuales se les asigna un identificador único (ID) dentro de la red, dichos mensajes están priorizados, evitando colisiones en la comunicación.

Las ventajas que se pueden apreciar sobre esta protocolo son; la elaboración física del cable CANopen permite economizar el coste monetario debido a su mínima complejidad del cableado (par trenzado), su máxima velocidad es de 1Mb/s, es altamente resistente a EMI, pueden conectarse hasta 110 dispositivos en la red CANopen, existe garantía de tiempos de latencia de manera que la transmisión siempre será en tiempo real, posee modo de desconexión autónoma de nodos defectuosos, además, se pueden añadir nodos al bus sin afectar al resto del sistema, y sin necesidad de reprogramación.

Arquitectura del sistema

La arquitectura de nuestro sistema está representada a detalle en la Figura 2 donde se consideran los siguientes elementos:

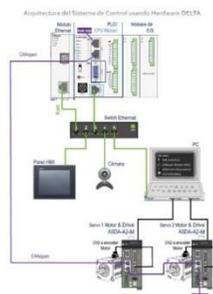


Figura 2. Arquitectura del sistema.

Desarrollo del prototipo

Se colocó el PLC dentro de un panel metálico cerrado de manera que mantuviera el hardware protegido de los problemas ambientales tales como el polvo conductivo, humedad y diversas sustancias corrosivas y dañinas

A partir del layout se definió la ubicación de cada componente a partir de sus características físicas, considerando los efectos del calor, ruido eléctrico, vibración y seguridad; no solamente para satisfacer la aplicación sino también disminuyendo los problemas de mantenimiento y minimizando el recorrido de cables, simplificando la instalación. Posteriormente se colocó un marco de canaletas a manera de protección del cableado y los DIN rail para poder fijar los componentes en el interior del tablero eléctrico como muestra la Figura 3.

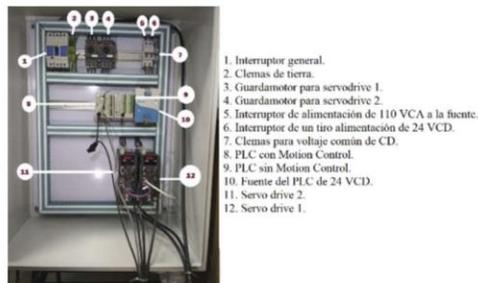


Figura 3. Tablero eléctrico y su distribución.

El robot paralelo activado por cables se puede apreciar en la Figura 4:



Figura 4. Robot paralelo activado por cables.

Pruebas y validación

Se usó el software WPLSoft para la programación del PLC, CANopen Builder para la programación del Motion Control, el software DOPSoft para la interfaz HMI utilizada para la activación de los dos motores del sistema y ASDA Soft para la validación de la lectura del encoder a través de los servo drives. El programa de prueba diseñado se hizo con la finalidad de comprobar la comunicación, instalación e interconexión de los componentes mencionados anteriormente. Tal como podemos ver en la Figura 5. se probó el movimiento del eje del motor 1, a 90 grados.



Figura 5. Prueba en motor 1.

Posteriormente se realizó otra prueba para corroborar la lectura del encoder, al hacer girar los motores diez vueltas y corroborando la lectura de los pulsos leídos por el encoder mediante el servo drive. A pesar de que en el HMI se mandó una posición de 10000, se muestra un resultado de 50000 pulsos por la relación de transmisión de la caja de engranes (5:1) como se puede apreciar en la Figura 6.

Monitor Item	Select Monitor Items	The monitored device: ASDA-A2 Servo
10	Motor feedback pulse number (after the scaling of e-gear ratio)(PUL)	50000
11	Input pulse no. of pulse command (after the scaling of e-gear ratio)(PUL)	30000
12	Deviation between control command pulse and feedback pulse number(PUL)	0
13	The number of motor feedback pulse (Encoder unit, 1,200,000 Pulse/rev)	6310142
14	Input pulse number of pulse command (before the scaling of e-gear)(pulse)	6310142
15	Motor pulse number (after the scaling of e-gear ratio)(pulse)	0
16	The frequency of pulse command input(Kpps)	0
17	Motor speed(r/min)	0.0
18	Speed command input(v/rk)	0.80
19	Speed command input(r/min)	0
110	Torque command input(V/rk)	-0.81
111	Torque command input(%)	4
112	Average load(%)	3
113	Peak load(%)	133
114	Main circuit voltage(V/rk)	207
115	Load/motor inertia ratio(0.11times)	4
116	JWT temperature	26
117	Resonance frequency(Hz)	1250
118	Distance from current position to Z.Rts value is between -5000 and +5000.	2099

Figura 6. Resultados ASDA Soft.

Más adelante, se observó en línea el software CANopen Builder y al mismo tiempo se utilizó el HMI para corroborar la posición actual del servomotor. Se indicó una posición de 26500 a una velocidad de 20000 y como se puede ver en la Figura 7, la lectura del encoder obtuvo exactamente los 26500 pulsos requeridos.

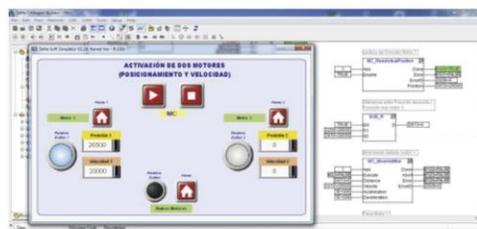


Figura 7. HMI y Motion Control en línea.

Por último se hizo girar los motores a diferentes velocidades y se cronometró el tiempo como lo muestra la Tabla 1.

Velocidad (pulsos/rev)	No. Vueltas	Tiempo (seg)
1000	5	50
10000	5	5
20000	5	2.5
40000	5	1.5
60000	5	1
80000	5	0.5
100000	5	0.2
100000	10	1
100000	10	10

Tabla 1. Datos obtenidos a partir de mediciones.

Conclusiones

Los componentes que se emplearon en el sistema descrito anteriormente son de fácil adquisición ya que se encuentran disponibles en el mercado y además son de costo moderado. El tablero eléctrico diseñado e implementado presenta todas las condiciones funcionales, estéticas y de robustez, además el cableado del tablero eléctrico se hizo en base a la Norma Oficial Mexicana NOM-001-SEDE-2012, que lleva por título «Instalaciones Eléctricas (utilización)».

Las pruebas realizadas fueron de entera satisfacción ya que existe compatibilidad entre los elementos existentes, facilidad en las conexiones y facilidad de movilidad garantizando su versatilidad en su implementación. El sistema propuesto tiene la capacidad de lograr el control de la posición del efector final mediante dos tipos de sensores: utiliza visión por computadora con el uso de una cámara y el procesamiento de imágenes además del encoder que poseen los servo drives.

Agradecimientos

Los autores agradecen al Consejo Nacional de Ciencia y Tecnología, al Tecnológico Nacional de México y al Instituto Tecnológico de Puebla por el apoyo proporcionado para el desarrollo de esta investigación con el financiamiento del proyecto de investigación: "Evaluación del sistema de control a un robot activado por cables en almacenes industriales de gran capacidad" con clave 6790.18-P.

Bibliografía

- + Zi, Duan, Du y Bao (2008). *Dynamic modeling and active control of a cable-suspended parallel robot*. *Mechatronics*, 18, 1-12. ISSN # 0957-4158.
- + Merlet, Jean-Pierre (2006). *Parallel robots*. *Litxemburgo: Springer Science & Business Media*. ISBN # 9781402041334.
- + Rushon y Khajepour (2018). *Transverse Vibration Control in Planar Cable-Driven Robotic Manipulators*. En *Cable-Driven Parallel Robots*, 243-253. *Suiza: Springer*. ISBN # 978-3-319-61430-4.
- + Khosravi y Taghirad (2014). *Robust PID control of fully-constrained cable driven parallel robots*. *Mechatronics*, 24, 87-97. ISSN # 0957-4158.
- + Riehl, Gouttefarde, Baradat y Pierrot (2010). *On the determination of cable characteristics for large dimension cable-driven parallel mechanisms*. *IEEE International Conference on Robotics and Automation*, 4709-4714. ISSN # 1050-4729.
- + Williams, Xin y Bosscher (2008). *Contour-crafting-cartesian-cable robot system concepts: workspace and stiffness comparisons*. En *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2, 31-38. *USA: ASME*. ISBN # 978-0-7918-4326-0
- + Lahouar, Ottaviano, Zeghoul, Romdhane y Ceccarelli (2009). *Collision free path-planning for cable-driven parallel robots*. *Robotics and Autonomous Systems*, 57, 1083-1093. ISSN # 0921-8890.
- + Mastero, Carraro, Ferraro, Gallina, Rossi y Rosati (2009). *Upper Limb Rehabilitation Robotics After Stroke*. *Journal of Rehabilitation Medicine*, 41, 981-985. ISSN # 1651-2081.
- + Dallej, Gouttefarde, Andreff, Michelin y Martinet (2011). *Towards vision-based control of cable-driven parallel robots*. *IEEE International Conference on Intelligent Robots and System*, 2855-2860. ISSN # 2153-0858
- + Bruckmann y Pott (2013). *Cable-Driven Parallel Robots*. *Berlin: Springer*. ISBN # 978-3-642-31987-7

ANEXO J

ARTÍCULO PUBLICADO N° 2

Modelado y simulación de un robot planar paralelo activado por cables

Modelling and simulation of a cable-driven parallel planar robot

Claudia Mena Huerta*, Renata Cebada Posadas, Sergio Javier Torres Méndez, José Rafael Mendoza Vázquez, Vicente Ramírez Palacios.

División de Estudios de Posgrado e Investigación, Tecnológico Nacional de México/Instituto Tecnológico de Puebla

Av. Tecnológico No. 420, Colonia Maravillas. Puebla, Puebla, México. C.P. 72220

*E-mail de autor de correspondencia: claudia.mena.25@hotmail.com, renata12363@hotmail.com, serm7007@yahoo.com.mx, rmendozainaoe@hotmail.com, vramirez07@hotmail.com.

RESUMEN. Los robots de tipo planar activados por cables son sistemas robóticos que poseen cadenas cinemáticas cerradas. En este documento se presenta el modelado y simulación de un sistema compuesto por un robot planar que posee cuatro cables colocados de manera paralela. La metodología desarrollada en este trabajo comienza describiendo de manera general el sistema, posteriormente se presenta la determinación de sus grados de libertad y el diagrama esquemático referente a sus articulaciones, así como también se realiza un estudio de velocidades y un análisis de singularidades. Se desarrollan las ecuaciones cinemáticas directa e inversa del sistema propuesto y, finalmente, se muestran los resultados validados de manera computacional utilizando el software Matlab Simulink.

PALABRAS CLAVE: robot activado por cables, modelado, cinemática, matlab, análisis de singularidades.

ABSTRACT. Cable-driven planar robots are robotic systems which have closed-loop kinematic chains. In this article is presented the modelling and simulation of a planar robot based system which has four cables arranged in a parallel way. The methodology developed at this paper begin by describing as a whole, later it is presented the system's degrees of freedom and the joints schematic diagram are defined. Also it is realized a velocities study and a singularities analysis. Finally the system's kinematic equations of the proposed system are developed, direct kinematics and inverse kinematics and, finally, it is showed the results validated by a computer program by Matlab Simulink.

KEYWORDS: Cable-driven robots, modelling, kinematics, matlab, singularities analysis.

1 INTRODUCCIÓN

Un tipo particular de los robots paralelos son los robots activados por cables. Son estructuras robóticas accionadas por cables que normalmente se componen de una estructura fija y una plataforma móvil, unidas mediante un número determinado de cables. Se usan en diversas aplicaciones, como por ejemplo, en el campo de la astronomía; también están tomando gran popularidad en la industria y en simuladores de vuelo [1].

Su gran aceptación en el campo de la robótica se debe a ciertas cualidades que, por el contrario, no poseen los robots paralelos convencionales y los manipuladores seriales. Esas mejoras relevantes son: bajo peso, escalabilidad, gran agilidad y elevada precisión al manipular grandes cargas a altas aceleraciones, así como bajo costo de construcción [2].

Las singularidades son configuraciones en donde el robot se vuelve incontrolable, son peligrosas y deben ser evitadas. La importancia de la determinación de las posiciones singulares en el control de este tipo de robots se debe a que las fuerzas requeridas sobre los actuadores, en estos puntos, son infinitas y debido a este esfuerzo, se causan daños al sistema de actuación del sistema. En posiciones cercanas a una singularidad, las fuerzas requeridas sobre los actuadores deben tomar valores muy altos para mantener magnitudes razonables de cargas sobre el actuador final [3].

En este documento, se describe la cinemática directa e inversa de un robot planar paralelo de tipo industrial que posee una configuración de cuatro cables colocados de manera paralela entre sí. Además, el estudio de velocidad permite determinar las configuraciones singulares donde los cables pierden tensión.

Estudios sobre la cinemática de robots paralelos activados por cables aparecen en [4], donde se presenta el modelo cinemático inverso de un robot paralelo con un arreglo de tres cables; también se presenta en [5], [6] y [7] métodos que involucran la obtención del modelo cinemático inverso a un robot paralelo con ocho cables.

En [8] y [9] se describe la cinemática de posición y velocidad de un robot, así como un modelado estático para un robot que posee cuatro cables; en [10] se muestran los algoritmos de cinemática inversa y directa utilizados para el robot “Marionet” que posee nueve cables y en [11] para un sistema de seis cables.

Mientras que en [12] se busca evitar las singularidades de un robot planar así como la prevención de colisiones; en [13] y en [14] se considera el problema de detectar singularidades en robots paralelos, en [15] las singularidades de robots de dos y tres cables y, en [16] de ocho cables. Sin embargo los objetos de estudio de las investigaciones mencionadas anteriormente no son parecidos al diseño del robot propuesto en este documento.

2 DESCRIPCIÓN DEL SISTEMA

El sistema analizado corresponde a un robot de estructura cinemática paralela con eslabones elásticos (cables) cuyo movimiento es planar; su base fija posee medidas de 1500 mm de largo x 1000 mm de alto y tiene adaptado un actuador final de tamaño 80 mm x 90 mm. Cada lado posee un tambor acoplado a la flecha del motor; en dicho tambor se irán enrollando y desenrollando cada par de cables dispuestos a cada lado. Su sistema de actuación es de tipo eléctrico y su transmisión es directa (ver Fig. 1.).

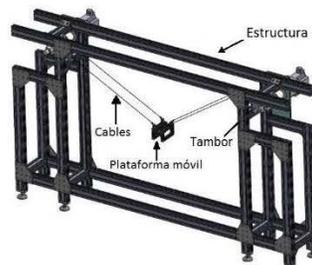


Fig. 1. Robot paralelo planar activado por cables utilizado

3 MODELACIÓN ANALÍTICA

En esta sección se determinan las ecuaciones de movimiento cinemático del robot propuesto.

- a. Determinación de grados de libertad.

A partir del criterio de Grubel-Kutzbach se define la siguiente ecuación:

$$n = \lambda L - \sum_{i=1}^J (\lambda - f_i) \quad (1)$$

Donde:

n = número de grados de libertad

λ = número de grados de libertad máximos

L = número de eslabones móviles

J = número de articulaciones

f_i = número de grados de libertad de la articulación i

El robot espacial mostrado en la Fig. 1 posee cuatro cables y se representa de manera simplificada como un robot planar suspendido por dos cables, debido al paralelismo que existe en cada par de cables que conectan el efector final con los tambores. Esto se puede apreciar en la Fig. 2 donde se observa que el sistema propuesto es del tipo 2-RPR, ya que posee de cada lado cadenas cinemáticas compuestas por eslabones unidos a articulaciones de revolución-prismática-revolución.

Se determinó el número de grados de libertad de cada articulación a partir de la generación del diagrama esquemático que se muestra en la Fig. 2. Cabe mencionar que la articulación de revolución permite solo un movimiento de rotación, mientras que la articulación prismática permite un movimiento de traslación.

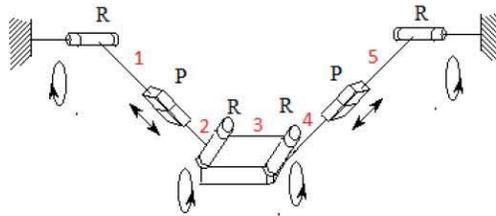


Fig. 2. Diagrama esquemático del sistema

Por lo que:

$\lambda = 3$ (es decir, se encuentra en el plano)

$L = 5$ eslabones

$$n = (3)(5) - [2(3 - 1) + 2(3 - 1) + 2(3 - 1)]$$

$$n = 15 - 12 = 3 \text{ d.o.f.}$$

De acuerdo a los grados de libertad obtenidos, el efector final tiene tres movimientos: una rotación sobre el eje transversal y dos movimientos de traslación en el plano. El sistema que se posee es subactuado ya que se requieren controlar tres coordenadas del efector final con solo dos actuadores; por lo que se asume una orientación fija del efector final para obtener una solución a las otras dos coordenadas de posición.

b. Modelado cinemático inverso

A partir de las coordenadas de la posición del efector final (P_x, P_y, P_z) y utilizando el modelo cinemático inverso, se conoce la longitud de cable que se modifica (recoge o suelta cable). En el modelado cinemático inverso pueden existir los siguientes resultados: no existe solución, existen varias soluciones, existen soluciones infinitas o existe solo una solución.

La Fig. 3 muestra los principales parámetros geométricos del sistema propuesto:

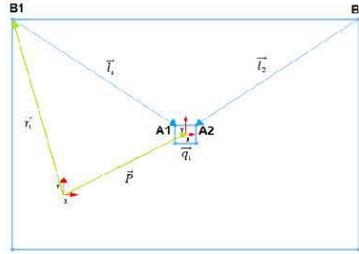


Fig. 3. Parámetros geométricos del robot planar paralelo activado por cables

$$l_1 := \overline{A_1B_1} \quad (2)$$

$$l_2 := \overline{A_2B_2} \quad (3)$$

La ecuación vectorial para cada cable está representada como:

$$\vec{P} + \vec{q}_i = \vec{r}_i + \vec{l}_i \quad \forall i = 1,2 \quad (4)$$

Para la cinemática inversa:

$$\vec{l}_i = \vec{P} + \vec{q}_i - \vec{r}_i \quad (5)$$

Para el caso general donde $\theta \neq 0$; en forma matricial:

$$l_i = P + R_{(z,\theta)} q^i - r_i \quad (6)$$

Donde:

$$q^1 = \begin{bmatrix} -d/2 \\ c/2 \\ 0 \end{bmatrix} \text{ y } r_1 = \begin{bmatrix} -e/2 \\ f/2 \\ 0 \end{bmatrix}$$

$$q^2 = \begin{bmatrix} d/2 \\ c/2 \\ 0 \end{bmatrix} \text{ y } r_2 = \begin{bmatrix} e/2 \\ f/2 \\ 0 \end{bmatrix}$$

Entonces la forma matricial para el cable 1 es:

$$\begin{bmatrix} l_{1x} \\ l_{1y} \\ l_{1z} \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -d/2 \\ c/2 \\ 0 \end{bmatrix} - \begin{bmatrix} -e/2 \\ f/2 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} l_{1x} \\ l_{1y} \\ l_{1z} \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} -\frac{d \cos \theta}{2} - \frac{c \sin \theta}{2} \\ \frac{d \sin \theta}{2} + \frac{c \cos \theta}{2} \\ 0 \end{bmatrix} - \begin{bmatrix} -e/2 \\ f/2 \\ 0 \end{bmatrix} \quad (7)$$

Por lo tanto, la ecuación de la cinemática inversa para l_1 resultó:

$$l_{1x} = P_x - \frac{1}{2}(d \cos \theta + c \sin \theta - e) \quad (8)$$

$$l_{1y} = P_y - \frac{1}{2}(d \sin \theta - c \cos \theta + f) \quad (9)$$

$$l_{1z} = 0$$

Se obtuvo la magnitud de l_1 :

$$l_1 = \|\vec{l}_1\| = \sqrt{l_{1x}^2 + l_{1y}^2} \quad (10)$$

Simplificando, se obtuvo la ecuación general para l_1 :

$$l_1 = \sqrt{P_x^2 + P_y^2 - P_x(d \cos \theta + c \sin \theta - e) - P_y(d \sin \theta - c \cos \theta + f) \dots}$$

$$\sqrt{\dots + \frac{1}{4}(d^2 + c^2 + e^2 + f^2 - 2de \cos \theta - 2ce \sin \theta + 2df \sin \theta - 2cf \cos \theta)} \quad (41)$$

Sin embargo, si se buscan orientaciones constantes y paralelas al piso, entonces $\theta = 0$; por lo tanto:

$$l_1 = \sqrt{P_x^2 + P_y^2 - P_x(d - e) - P_y(-c + f) + \frac{1}{4}(d^2 + c^2 + e^2 + f^2 - 2de - 2cf)} \quad (12)$$

Pasando al caso de l_2 , la forma matricial para el cable 2 es:

$$\begin{bmatrix} l_{2x} \\ l_{2y} \\ l_{2z} \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d/2 \\ c/2 \\ 0 \end{bmatrix} - \begin{bmatrix} e/2 \\ f/2 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} l_{2x} \\ l_{2y} \\ l_{2z} \end{bmatrix} = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} + \begin{bmatrix} \frac{d \cos \theta}{2} - \frac{c \sin \theta}{2} \\ \frac{d \sin \theta}{2} + \frac{c \cos \theta}{2} \\ 0 \end{bmatrix} - \begin{bmatrix} e/2 \\ f/2 \\ 0 \end{bmatrix} \quad (13)$$

Por lo tanto, la ecuación de la cinemática inversa para l_2 resultó:

$$l_{2x} = P_x + \frac{1}{2}(d \cos \theta - c \sin \theta - e) \quad (14)$$

$$l_{2y} = P_y + \frac{1}{2}(d \sin \theta + c \cos \theta - f) \quad (15)$$

$$l_{2z} = 0$$

Se obtuvo la magnitud de l_2 :

$$l_2 = \|\vec{l}_2\| = \sqrt{l_{2x}^2 + l_{2y}^2} \quad (16)$$

Simplificando, se obtuvo la ecuación general para l_2 :

$$l_2 = \sqrt{P_x^2 + P_y^2 + P_x(d \cos \theta - c \sin \theta - e) + P_y(d \sin \theta + c \cos \theta - f) \dots} \\ \sqrt{\dots + \frac{1}{4}(d^2 + c^2 + e^2 + f^2 - 2de \cos \theta + 2ce \sin \theta - 2df \sin \theta - 2cf \cos \theta)} \quad (17)$$

De igual forma, si se buscan orientaciones constantes y paralelas al piso, entonces $\theta = 0$; por lo tanto:

$$l_2 = \sqrt{P_x^2 + P_y^2 + P_x(d - e) + P_y(c - f) + \frac{1}{4}(d^2 + c^2 + e^2 + f^2 - 2de - 2cf)} \quad (18)$$

c. Modelado cinemático directo

Al usar el modelado de cinemática directa, se determinaron las coordenadas de posición del efector final (P_x , P_y , P_z) a partir de la longitud que poseen los cables en un momento dado. En el modelo cinemático directo solo hubo una solución única.

De las ecuaciones de cinemática inversa, se obtuvieron las siguientes ecuaciones que describen la cinemática directa del sistema elevando al cuadrado y sumando (12) y (18):

Se despejó P_x^2 , obteniendo:

$$P_x^2 = l_1^2 - P_y^2 + P_x(d - e) + P_y(-c + f) - \frac{1}{4}(d^2 + c^2 + e^2 + f^2 - 2de - 2cf)$$

Posteriormente:

$$l_2^2 = l_1^2 - P_y^2 + P_x(d - e) + P_y(-c + f) - \frac{1}{4}(d^2 + c^2 + e^2 + f^2 - 2de - 2cf) + P_y^2 + P_x(d - e) \\ - P_y(-c + f) + \frac{1}{4}(d^2 + c^2 + e^2 + f^2 - 2de - 2cf)$$

$$l_2^2 = l_1^2 + 2P_x(d - e)$$

Se despejó y se obtuvo que la coordenada P_x es:

$$P_x = \frac{l_2^2 - l_1^2}{2(d - e)} \quad (19)$$

Se tuvo el siguiente resultado:

$$2P_y^2 - 2P_y(f - c) + \frac{2}{4}(f^2 - 2cf + c^2) = 2 \left[P_y - \frac{(f - c)}{2} \right]^2$$

Y se reescribió como:

$$2 \left[P_y - \frac{(f - c)}{2} \right]^2 = l_1^2 + l_2^2 - 2P_x^2 - \frac{2}{4}(d^2 - 2de + e^2)$$

Se sustituyó el valor de $P_x = \frac{l_2^2 - l_1^2}{2(d - e)}$:

$$2 \left[P_y - \frac{(f - c)}{2} \right]^2 = l_1^2 + l_2^2 - 2 \left[\frac{l_2^2 - l_1^2}{2(d - e)} \right]^2 - \frac{1}{2}(d - e)^2$$

Finalmente se despejó P_y de la última ecuación y se obtuvo:

$$P_y = -\sqrt{\frac{l_1^2 + l_2^2 - 2 \left[\frac{l_2^2 - l_1^2}{2(d - e)} \right]^2 - \frac{1}{2}(d - e)^2}{2}} + \frac{(f - c)}{2} \quad (20)$$

d. Modelado cinemático diferencial

La cinemática diferencial del robot paralelo activado por cables se presenta a continuación:

$$\dot{q} = J_q \dot{p} \quad (21)$$

Donde:

$$\dot{q} = \begin{bmatrix} \dot{l}_1 \\ \dot{l}_2 \end{bmatrix}$$

$$\dot{p} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

$$J = \begin{bmatrix} \frac{\partial l_1}{\partial P_x} & \frac{\partial l_1}{\partial P_y} \\ \frac{\partial l_2}{\partial P_x} & \frac{\partial l_2}{\partial P_y} \end{bmatrix}$$

De las ecuaciones de cinemática inversa del robot paralelo activado por cables planar, se obtuvieron las siguientes ecuaciones de cinemática diferencial inversa:

$$\frac{dl_1}{dt} = \frac{[(2P_x - d + e)v_x + (2P_y + c - f)v_y]}{2\sqrt{Px^2 + Py^2 - Px(d - e) - Py(-c + f) + \frac{1}{4}(d^2 + c^2 + e^2 + f^2 - 2de - 2cf)}} \quad (22)$$

$$\frac{dl_2}{dt} = \frac{[(2P_x + d - e)v_x + (2P_y + c - f)v_y]}{2\sqrt{Px^2 + Py^2 + Px(d - e) + Py(c - f) + \frac{1}{4}(d^2 + c^2 + e^2 + f^2 - 2de - 2cf)}} \quad (23)$$

e. Análisis de singularidades

El análisis de singularidad sobre este sistema se refiere a una configuración particular donde el robot gana o pierde grados de libertad instantáneamente, y se basa en el determinante de la matriz del Jacobiano (se puede indeterminar, por lo que no habría solución).

$$\dot{q} = J^{-1}\dot{p} \quad (24)$$

Donde:

$$\dot{q} = \begin{bmatrix} \dot{l}_1 \\ \dot{l}_2 \end{bmatrix} \quad \dot{p} = \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

$$J = \begin{bmatrix} \frac{\partial l_1}{\partial P_x} & \frac{\partial l_1}{\partial P_y} \\ \frac{\partial l_2}{\partial P_x} & \frac{\partial l_2}{\partial P_y} \end{bmatrix}$$

$$\frac{\partial l_1}{\partial P_x} = \frac{1}{2}[2P_x - (d - e)]r^{-\frac{1}{2}}$$

$$\frac{\partial l_2}{\partial P_x} = \frac{1}{2}[2P_x + (d - e)]s^{-\frac{1}{2}}$$

$$\frac{\partial l_1}{\partial P_y} = \frac{1}{2}[2P_y - (-c + f)]r^{-\frac{1}{2}}$$

$$\frac{\partial l_2}{\partial P_y} = \frac{1}{2}[2P_y + (c - f)]s^{-\frac{1}{2}}$$

$$J^{-1} = \frac{1}{\det(J)} \begin{bmatrix} \frac{\partial l_2}{\partial P_y} & -\frac{\partial l_1}{\partial P_y} \\ -\frac{\partial l_2}{\partial P_x} & \frac{\partial l_1}{\partial P_x} \end{bmatrix}$$

Se buscó el determinante del Jacobiano:

$$\det(J)_1 = \left[\frac{1}{2} [2P_x - (d - e)] r^{-\frac{1}{2}} \right] \left[\frac{1}{2} [2P_y + (c - f)] s^{-\frac{1}{2}} \right]$$

$$\det(J)_2 = \left[\frac{1}{2} [2P_x + (d - e)] s^{-\frac{1}{2}} \right] \left[\frac{1}{2} [2P_y - (-c + f)] r^{-\frac{1}{2}} \right]$$

$$\det(J) = \det(J)_1 - \det(J)_2$$

$$\begin{aligned} \det(J) &= - \left(\frac{2(2P_y + c - f) * (d - e)}{\sqrt{(4Px^2 + (-4d + 4e)Px + 4Py^2 + (4c - 4f)Py + c^2 - 2cf + d^2 - 2de + e^2 + f^2)}} \right) \\ &\quad \left(\frac{1}{\sqrt{(4Px^2 + (4a - 4e)Px + 4Py^2 + (4c - 4f)Py + c^2 - 2cf + d^2 - 2de + e^2 + f^2)}} \right) \end{aligned}$$

$$\begin{aligned} \frac{1}{\det(J)} &= \frac{\sqrt{(4Px^2 + (-4d + 4e)Px + 4Py^2 + (4c - 4f)Py + c^2 - 2cf + d^2 - 2de + e^2 + f^2)}}{2(2P_y + c - f) * (d - e)} \\ &= \frac{\sqrt{(4Px^2 + (4a - 4e)Px + 4Py^2 + (4c - 4f)Py + c^2 - 2cf + d^2 - 2de + e^2 + f^2)}}{1} \end{aligned}$$

$$2Py + c - f = 0$$

Por lo que se obtuvo la siguiente singularidad:

$$P_y = \frac{f - c}{2} \quad (25)$$

4 VALIDACIÓN COMPUTACIONAL

Para la validación del modelo cinemático se utilizó el software Matlab Simulink donde se realizó una simulación del sistema. Para su elaboración, fue necesario tomar en cuenta el diagrama esquemático de la Fig. 2. Se observa en la Fig. 4. las cuatro articulaciones de revolución que posee el robot (recuadro color rojo) así como las articulaciones prismáticas (recuadro color verde).

Se simuló el enrollamiento y desenrollamiento de los cables utilizando dos bloques tipo “Body” de cada lado, por lo que l_1 está representado por “Body 1” y “Body 2” (recuadros color azul); por otra parte l_2 corresponde a “Body 4” y “Body 5” (recuadros color lila). Por último “Body 3” representa al efector final del robot planar paralelo planar activado por cables.

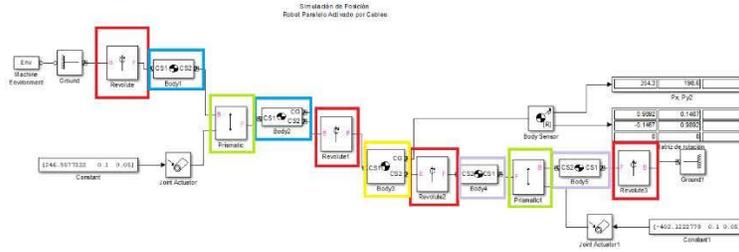


Fig. 4. Modelado de simulación en Matlab Simulink

A partir de las siguientes coordenadas deseadas $P_x = 0$ mm y $P_y = 1200$ mm, se sustituyeron las medidas de la estructura fija ($e = 1500$ mm y $f = 1000$ mm) así como las dimensiones del efector final ($d = 80$ mm y $c = 90$ mm) en las ecuaciones (12) y (18) y, se obtuvo que la longitud final de los cables es:

$$l_{1f} = l_{2f} = 1029.137989 \text{ mm} = 102.9137989 \text{ cm}$$

Inicialmente, ambos cables poseen una longitud de $l_i = 843.282277$ mm, por lo tanto, la longitud que tienen que desenrollar los cables para lograr ubicarse en $(0, 1200)$ mm es:

$$l\Delta_1 = l_{1f} - l_i = 1029.137989 \text{ mm} - 843.282277 \text{ mm} = 185.855712 \text{ mm}$$

$$l\Delta_2 = l_{2f} - l_i = 1029.137989 \text{ mm} - 843.282277 \text{ mm} = 185.855712 \text{ mm}$$

Es decir, l_1 y l_2 aumentaron su tamaño (signo positivo). Para su comprobación, como datos de entrada se colocaron las longitudes obtenidas anteriormente, tal y como lo muestra la Fig. 5.

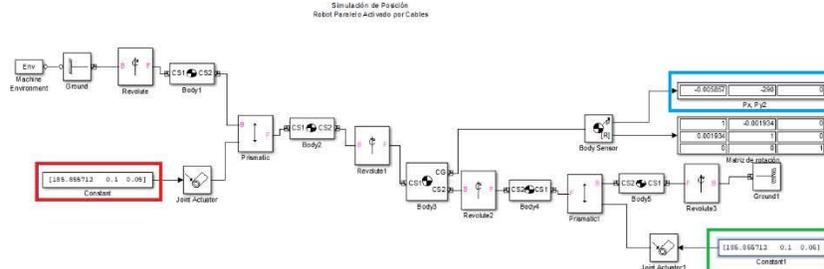


Fig. 5. Simulación de una coordenada en Matlab Simulink

Note que el resultado de $l\Delta_1$ se ingresó como entrada constante en el bloque color rojo en Matlab-Simulink, de igual forma $l\Delta_2$ se ingresó en el bloque verde, y en el bloque azul se obtuvieron las coordenadas resultantes $P_x = -0.005857$, $P_y = -290$ y $P_z = 0$; además, no se presentó rotación. El valor obtenido por Simulink en la coordenada P_y correspondió a la diferencia entre la posición deseada ($P_y = 1200$ mm) y el largo total de la estructura (1500 mm).

La simulación del Robot se observa en la Fig. 6. donde en efecto se demostró que l_1 y l_2 poseen el comportamiento esperado:

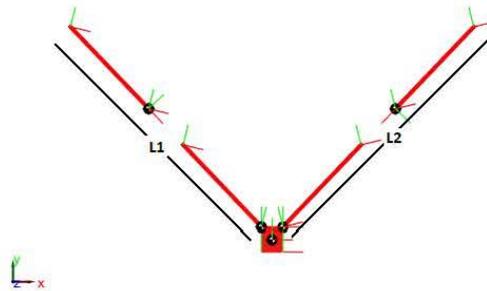


Fig. 6. Simulación de cinemática del sistema

5 CONCLUSIONES

En este documento se presentaron las ecuaciones matemáticas que describen el comportamiento cinemático (directo e inverso) de un robot paralelo planar activado por cables. El análisis de las singularidades del sistema fue un factor importante a considerar en el diseño del robot y para su control.

Las ecuaciones fueron comprobadas computacionalmente mediante el software Matlab Simulink. Entre el resultado numérico obtenido gracias a la simulación y las ecuaciones teóricas existe una pequeña diferencia debido a la aproximación decimal (error de 0.0019 mm medido en simulación).

Las ecuaciones obtenidas en el estudio de la cinemática del sistema, además de ayudar en un futuro a desarrollar un sistema de control óptimo para el robot, servirán para determinar la posición del efector final del robot a partir de la longitud de los cables, o de manera contraria, determinar la cantidad de cable necesario para alcanzar una posición determinada.

6 AGRADECIMIENTOS

Los autores agradecen al Consejo Nacional de Ciencia y Tecnología, al Tecnológico Nacional de México y al Instituto Tecnológico de Puebla por el apoyo proporcionado para el desarrollo de esta investigación.

REFERENCIAS

- [1] Merlet, J. P. *Parallel Robots*. Springer. 2012.
- [2] Salgado, O. *Síntesis, análisis y diseño de manipuladores paralelos de baja movilidad*. University of the Basque Country. 2008.
- [3] Dasgupta, B. Singularity-free path planning for the Stewart platform manipulator. *Mechanism and Machine Theory*. 1998, 711-725.
- [4] Gorman, J. J. The cable array robot: Theory and experiment. *IEEE International Conference*. 2001, 2804-2810.
- [5] Dallej, T. G. Vision-based modeling and control of large-dimension cable-driven parallel robots. *IEEE/RSJ International Conference*. 2012, 1581-1586.
- [6] Roberts, R. G. On the inverse kinematics, statics, and fault tolerance of cable-suspended robots. *Journal of Robotic Systems*. 1998, 581-597.

- [7] Pham, C. B. Force-closure workspace analysis of cable-driven parallel mechanisms. *Mechanism and Machine Theory*. 2006, 53-69.
- [8] Williams Ii, R. L. Translational planar cable-direct-driven robots. *Journal of Intelligent and Robotic systems*. 2003, 69-96.
- [9] Trevisani, A. G. Cable-direct-driven robot (CDDR) with passive SCARA support: theory and simulation. *Journal of Intelligent and Robotic Systems*. 2006, 73-94.
- [10] Merlet, J. P. Kinematics of the wire-driven parallel robot MARIONET using linear actuators. *IEEE International Conference*. 2008, 3857-3862.
- [11] Riehl, N. G. Effects of non-negligible cable mass on the static behavior of large workspace cable-driven parallel mechanisms. *IEEE International Conference*. 2009, 2193-2198.
- [12] Lahouar, S. O. Collision free path-planning for cable-driven parallel robots. *Robotics and Autonomous Systems*. 2009, 1083-1093.
- [13] Angeles, J. Y. Singularity analysis of three-legged, six-DOF platform manipulators with URS legs. *IEEE/ASME transactions on mechatronics*. 2003, 469-475.
- [14] Altuzarra, O. P. A practical procedure to analyze singular configurations in closed kinematic chains. *IEEE Transactions on Robotics*. 2004, 929-940.
- [15] Barrette, G. Determination of the dynamic workspace of cable-driven planar parallel mechanisms. *Journal of mechanical design*. 2005, 242-248.
- [16] Yang, G. Y. Kinematics and singularity analysis of a planar cable-driven parallel manipulator. *Intelligent Robots and Systems*. 2004.

Acerca de los autores



Claudia Mena Huerta, recibió el título de Ingeniera en Electrónica por parte del Tecnológico Nacional de México/Instituto Tecnológico de Puebla en 2015. Actualmente es estudiante de tiempo completo de la Maestría en Ingeniería Electrónica con línea de investigación en Sistemas mecatrónicos interactivos aplicados al control de procesos, en el Instituto Tecnológico de Puebla desde 2017 a través del Programa Nacional de Posgrados de Calidad por el Consejo Nacional de Ciencia y Tecnología (Conacyt); con tema de tesis Control de un robot activado por cables.



Renata Cebada Posadas, es licenciada en ciencias de la electrónica en 2015 por parte de la Benemérita Universidad Autónoma de Puebla (BUAP) y actualmente es estudiante de tiempo completo de la Maestría en Ingeniería en Electrónica con línea de investigación en Sistemas mecatrónicos interactivos aplicados al control de procesos, perteneciente al PNPC del Consejo Nacional de Ciencia y Tecnología, por parte del Instituto Tecnológico de Puebla con tema de tesis aplicado al Control de un robot paralelo activado por cables mediante visión artificial.



Sergio Javier Torres Méndez, recibió el grado de Doctor en Ciencias en Ingeniería Mecánica en 2014 por parte de la Universidad de Waterloo, Ontario, Canadá; el grado de Maestro en Ciencias en Electrónica en 2008 por la Benemérita Universidad Autónoma de Puebla, y el grado de Maestro en Ciencias en Ingeniería Mecánica por parte del Instituto Tecnológico de Veracruz en 1995. Es profesor de tiempo completo en el Instituto Tecnológico de Puebla desde 1995. Es miembro del cuerpo académico Sistemas interactivos y realidad virtual. Sus intereses de investigación se relacionan con el diseño y desarrollo de robots de alta velocidad y de sistemas mecatrónicos interactivos para la asistencia humana.



José Rafael Mendoza Vázquez, recibió el grado de Doctor en Ciencias en Electrónica en 2010 por parte del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE) y el grado de Maestro en Ciencias en Electrónica en 2003 por la misma institución. Es profesor de tiempo completo en el Instituto Tecnológico de Puebla, en el departamento de Ingeniería eléctrica y electrónica. Es miembro del cuerpo académico Sistemas interactivos y realidad virtual. Su investigación se encuentra dentro del área de robótica y control con énfasis en robótica, control de movimiento, modelado, procesamiento de señales y sistemas interactivos.



Vicente Ramírez Palacios, recibió el grado de Maestro en Ciencias en Electrónica en 1997 por parte del Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE). Recibió el título de Ingeniero Industrial en Electrónica en 1996 por parte del Instituto Tecnológico de Puebla. Es profesor de tiempo completo a nivel Licenciatura en el Instituto Tecnológico de Puebla. Trabajando en Automatización y Control de Procesos, utilizando algoritmos PID, Lógica Difusa y Redes Neuronales; comunicando los procesos a través de Redes Industriales como son: Red ASi, Red Profibus y Red Ethernet, teniendo Control y Adquisición de Datos de los procesos por medio de un SCADA. Es miembro del cuerpo académico Sistemas interactivos y realidad virtual.

ANEXO K

PARTICIPACIÓN EN CONGRESO CICOS



Modelado y simulación de un robot planar paralelo activado por cables

Claudia Mena Huerta, Renata Cebada Posadas,
Sergio J. Torres Méndez, José Rafael Mendoza Vázquez,
Vicente Ramírez Palacios.

claudia.mena.25@hotmail.com

INTRODUCCIÓN

Utilizando el modelo de cinemática directa, se pueden determinar las coordenadas de posición (Px, Py, Pz) del efector final de un robot activado por cables a partir de la longitud que poseen sus cables en un momento dado. Por otro lado, utilizando las coordenadas de la posición (Px, Py, Pz) del efector final y a través de su modelo cinemático inverso, se puede conocer la longitud de cable que utilizará.

El estudio de velocidad permite determinar las configuraciones singulares donde los cables pierden tensión.

Las singularidades son configuraciones en donde el robot se vuelve incontrolable, son peligrosas y deben ser evitadas. La importancia de la determinación de las posiciones singulares en el control de este tipo de robots se debe a que las fuerzas requeridas sobre los actuadores, en estos puntos, son infinitas y debido a este esfuerzo, se podrían causar daños al sistema de actuación del sistema.

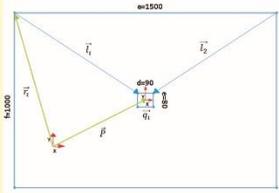
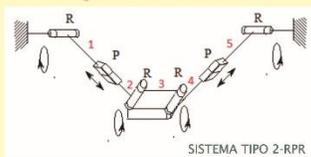


DIAGRAMA ESQUEMÁTICO



MODELADO

MODELADO CINEMÁTICO INVERSO

$$l_1 = \sqrt{l_2^2 + l_3^2 - l_2 l_3 (\alpha - \beta)} - l_2 (\alpha + \beta) + \frac{1}{2} (d^2 + c^2 + a^2 + b^2 - 2dc - 2ef)$$

$$l_2 = \sqrt{l_1^2 + l_3^2 - l_1 l_3 (\alpha - \beta)} - l_1 (\alpha + \beta) + \frac{1}{2} (d^2 + c^2 + a^2 + b^2 - 2dc - 2ef)$$

MODELADO CINEMÁTICO DIRECTO

$$l_1 = \frac{l_2^2 - l_3^2}{2(d - e)}$$

$$l_2 = \sqrt{l_1^2 + l_3^2 - 2 \frac{l_1 l_3 (\alpha - \beta)}{2} - \frac{1}{2} (d - e)^2} \cdot \frac{(d - e)}{2}$$

MODELADO CINEMÁTICO DIFERENCIAL

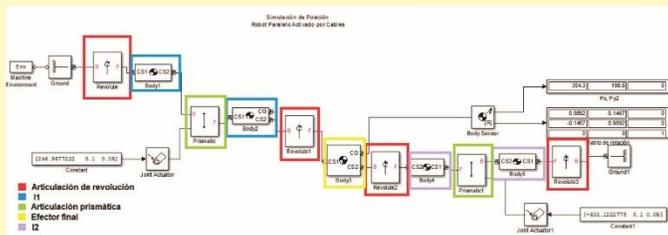
$$\frac{dl_1}{dt} = \frac{[2l_2 \cdot d + 2l_3 \cdot e + 2l_1 \cdot c]}{2 \sqrt{l_1^2 + l_3^2 - 2l_1 l_3 (\alpha - \beta) - \frac{1}{2} (d - e)^2}} \cdot \frac{1}{2} (d - e)$$

$$\frac{dl_2}{dt} = \frac{[2l_1 \cdot d + 2l_3 \cdot e + 2l_2 \cdot c]}{2 \sqrt{l_1^2 + l_3^2 - 2l_1 l_3 (\alpha - \beta) - \frac{1}{2} (d - e)^2}} \cdot \frac{1}{2} (d - e)$$

ANÁLISIS DE SINGULARIDADES

$$P_y = \frac{f - c}{2}$$

SIMULACIÓN



ROBOTS ACTIVADOS POR CABLES

ROBOTS ACTIVADOS POR CABLES

Un tipo particular de los robots paralelos son los robots activados por cables, normalmente se componen de una estructura fija y una plataforma móvil, unidas mediante determinado número de cables.

Algunas de sus características son: bajo peso, escalabilidad, gran agilidad y precisión elevada al manipular grandes cargas a altas aceleraciones, entre otras.

Los robots de tipo planar activados por cables son sistemas robóticos que poseen cadenas cinemáticas cerradas.

CONCLUSIONES

El análisis de las singularidades del sistema es un factor importante a considerar en el diseño del robot y para su control.

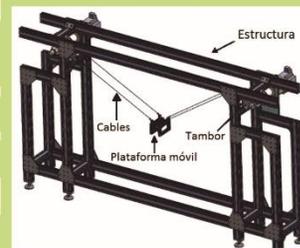
Las ecuaciones obtenidas han sido comprobadas computacionalmente mediante el software Matlab Simulink. Entre el resultado numérico obtenido gracias a la simulación y las ecuaciones teóricas existe una pequeña diferencia debida a la aproximación decimal.

DESCRIPCIÓN DEL ROBOT

El sistema analizado corresponde a un robot de estructura cinemática paralela con cuatro eslabones elásticos (cables), su movimiento es planar, posee medidas de 1500 mm de largo x 1000 mm de alto. Posee un actuador final de tamaño 80 mm x 90 mm.

Cada lado posee un tambor acoplado a la flecha del motor, en dicho tambor se irán enrollando y desenrollando cada par de cables dispuestos a cada lado. Su sistema de actuación es de tipo eléctrico y su transmisión es directa. El efector final tendrá tres movimientos, una rotación sobre el eje transversal y dos movimientos de traslación en el plano.

OBJETO DE ESTUDIO





UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

LA SECRETARÍA ACADÉMICA, LA SECRETARÍA DE INVESTIGACIÓN, EL CHICAP, A TRAVÉS DE LOS CUERPOS ACADÉMICOS “UAEM-OPTIMIZACIÓN SOFTWARE”, “UPEMOR-TECNOLOGÍAS WEB”, Y “UTEZ-REDES CONVERGENTES SEGURAS”

otorga el presente

RECONOCIMIENTO

Ing. Claudia Mena Huerta

Lic. Renata Cebada Posadas

Por su participación con el Póster:

Modelado y simulación de un robot planar paralelo activado por cables

En el 12° Congreso Internacional de Cómputo en Optimización y Software - CICos 2018

Cuernavaca, Mor. del 3 al 5 de octubre, 2018

Por una humanidad culta

Dr. Federico Alonso Pecina
Presidente del Congreso

Dr. Marco Antonio Cruz Chávez
Presidente del Comité Técnico

CARTA CESIÓN DE DERECHOS

En la Ciudad de Puebla, Pue. el día 14 del mes de noviembre del año 2019, la que suscribe **CLAUDIA MENA HUERTA** alumna del Programa de Maestría en Ingeniería Electrónica, con número de registro **MPIEO-2011-13**, adscrito al **Tecnológico Nacional de México (TecNM)**, manifiesta que es la autora intelectual del presente trabajo de Tesis bajo la dirección del M. C. Vicente Ramírez Palacios y codirección del Dr. Sergio Javier Torres Méndez y cede los derechos del trabajo titulado "**Control de un robot activado por cables**", al **Instituto Tecnológico de Puebla** para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso de la autora y/o directores del trabajo. Este puede ser obtenido escribiendo a las siguientes direcciones **cmie@itpuebla.edu.mx**. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

A handwritten signature in black ink, appearing to read 'Claudia', with a date '3.15' and a small asterisk below it.

C. CLAUDIA MENA HUERTA