

SISTEMAS A EVENTOS DISCRETOS, UNA APROXIMACION A LA
TEORIA DE REDES DE PETRI Y GRAFCET

JOHANNA STELLA CASTELLANOS ARIAS

FACULTAD DE INGENIERIA
PROGRAMA DE MECATRONICA
UNIVERSIDAD MILITAR NUEVA GRANADA
2008

SISTEMAS A EVENTOS DISCRETOS, UNA APROXIMACION A LA
TEORIA DE REDES DE PETRI Y GRAFCET

JOHANNA STELLA CASTELLANOS ARIAS

Director Trabajo de Grado
Ing. LEONARDO ENRIQUE SOLAQUE GUZMÁN, PhD

Trabajo de grado para optar por el título de Ingeniero en Mecatrónica

FACULTAD DE INGENIERIA
PROGRAMA DE MECATRONICA
UNIVERSIDAD MILITAR NUEVA GRANADA
2008

DEDICATORIA Y AGRADECIMIENTOS

Quiero dedicar este proyecto a mi familia: Stella, Ricardo, Ykita, Pitu y Dani, quienes con su cariño, paciencia, compañía y consejos, me apoyaron a lo largo, no solo de mi proyecto de grado, sino de mi carrera.

A mi director, Leonardo Solaque, dedico y agradezco la culminación de este trabajo de grado. Gracias por confiar en mí, por brindarme grandes enseñanzas a nivel profesional y a nivel personal, por tenderme una mano y motivarme a continuar por el camino de la investigación y del desarrollo tecnológico, por esto muchas cosas más mil gracias.

Así mismo, quiero agradecer a todas y cada una de las personas que con su apoyo y colaboración, ayudaron a culminar satisfactoriamente este proyecto.

TABLA DE CONTENIDO

LISTA DE FIGURAS.....	6
LISTA DE TABLAS	8
CAPITULO 1: ANTECEDENTES	9
1.1 INTRODUCCIÓN	9
1.2 OBJETIVOS	11
1.2.1 <i>Objetivo General</i>	11
1.2.2 <i>Objetivos Específicos</i>	11
1.3 JUSTIFICACIÓN	11
CAPITULO 2: INTRODUCCIÓN DE LOS SISTEMAS A EVENTOS DISCRETOS (SED).....	12
2.1 DEFINICIÓN:	12
2.2 SISTEMAS DE MANUFACTURA FLEXIBLE (SMF)	14
2.3 EJEMPLOS	14
CAPÍTULO 3: REDES DE PETRI (RDP)	27
3.1 DEFINICIÓN Y CONTEXTO HISTÓRICO	27
3.2 ELEMENTOS DE UNA RDP	28
3.3 MARCADO DE UNA RDP	29
3.4 ÁRBOL DE COBERTURA.....	30
3.5 CARACTERÍSTICAS DE LAS RDP	33
3.6 OTRAS PROPIEDADES DE LAS RDP	33
3.6.1 <i>Recursos compartidos (RC)</i>	33
3.6.2 <i>Conflictos entre Transiciones</i>	35
3.6.3 <i>Concurrencia y Sincronización</i>	35
3.7 MODELADO DE PROCESOS CONCURRENTES CON RECURSOS COMPARTIDOS (<i>PCRC</i>) MEDIANTE RDP..	36
3.8 MATRIZ DE INCIDENCIA	39
3.9 ECUACIÓN FUNDAMENTAL DE LAS RDP	40
3.10 ECUACIONES DE ESTADO PARA TIEMPO REMANENTE EN RDP	41
3.11 ECUACIONES DE ESTADO PARA TIEMPO ACUMULADO EN RDP	43
3.12 EJEMPLOS	45
3.13 CONCLUSIONES	51
CAPITULO 4: GRAFCET (G).....	52
4.1 DEFINICIÓN Y CONTEXTO HISTÓRICO	52
4.2 ESTRUCTURA Y REGLAS DE EVOLUCIÓN DEL GRAFCET	53
4.3 MACRO ETAPAS	55
4.4 ESTRUCTURAS DEL GRAFCET Y OBTENCIÓN DE ECUACIONES DE FUNCIONES LÓGICAS.....	55

4.5 TEMPORIZACIONES	59
4.6 NORMA GEMMA	60
4.7 EJEMPLOS Y APLICACIONES.....	62
4.8 CONCLUSIONES	69
CAPITULO 5: OPTIMIZACIÓN.....	70
5.1 DEFINICIÓN Y JUSTIFICACIÓN	71
5.2 MÉTODOS DE OPTIMIZACIÓN	74
5.2.1 Programación Lineal.....	75
5.2.2 Programación no Lineal	77
5.5 APLICACIONES.....	79
5.6 EJEMPLOS	80
5.7 CONCLUSIONES	85
CAPITULO 6: IMPLEMENTACIÓN DE RDP, GRAFCET, OPTIMIZACIÓN Y SISTEMAS DE MANUFACTURA FLEXIBLE	86
6.1 MODELO Y ANÁLISIS DE UN SISTEMA MEDIANTE RDP	86
6.1.1 Rdp para el Sistema Modelado	87
6.1.2 Caracterización del Sistema	88
6.2 IMPLEMENTACIÓN DEL SISTEMA	95
6.2.1 Simulación Electro - Neumática	96
6.2.2 Montaje Físico Electro - Hidroneumático.....	97
6.2.3 Control mediante GRAFCET – PLC SIEMENS.....	103
EPILOGO.....	104
A. CONCLUSIONES GENERALES	104
B. TRABAJOS FUTUROS	104
ANEXO 1: SOFTWARE DE SIMULACIÓN	105
A1.1 QUEST Y IGRIP:.....	105
A1.2 HILES DESIGNER, PM EDITEUR - PETRI NET - TOOLBOX MATLAB	106
A1.3 SIMATICS – S7 SIEMENS - S7-GRAPH	109
A1.4 FLUIDSIM.....	111
ANEXO 2: SIMULACIÓN NEUMÁTICA CON FLUIDSIM	111
ANEXO 3: GRAFCET DE CONTROL PARA EL SISTEMA FÍSICO IMPLEMENTADO	113
BIBLIOGRAFIA	129

LISTA DE FIGURAS

CAPITULO 2

- Figura 2.1. Representación gráfica de una trayectoria
- Figura 2.2. Representación del procesamiento de un SED
- Figura 2.3. Ejemplo 2.1: Túnel compartido para trenes
- Figura 2.4. Esquema general de la Pasteurización de la leche
- Figura 2.5. Diagrama de bloques del proceso de pasteurización industrial
- Figura 2.6. Diagrama de equipo del proceso de pasteurización industrial
- Figura 2.7. Proceso de tapado y almacenamiento de botellas de agua
- Figura 2.8. Elementos del proceso (Canasta, botellas y tapas)
- Figura 2.9. Simulación del sistema de clasificación y empaquetado

CAPITULO 3

- Figura 3.1. Elementos de una RdP
- Figura 3.2. Ejemplos de evolución de una RdP
- Figura 3.3. Representación y marcado de una RdP
- Figura 3.4. RdP E1
- Figura 3.5. RdP E2
- Figura 3.6.1 Árbol de cobertura de la RdP E1
- Figura 3.6.2 Árbol de cobertura de la RdP E1 generado por Matlab
- Figura 3.7. Árbol de cobertura de la Fig. 3.3
- Figura 3.8.1 Recurso compartido (RC)
- Figura 3.8.2 Evolución de la RdP con RC por la izquierda
- Figura 3.8.3 Evolución de la RdP con RC por la derecha
- Figura 3.9. Conflicto entre transiciones
- Figura 3.10. Solución al conflicto entre transiciones
- Figura 3.11. Concurrencia y sincronización en una RdP
- Figura 3.12. Modelado de PCRC mediante RdP
- Figura 3.13. Modelo de RdP para el ejemplo 3.1
- Figura 3.14. RdP para el proceso de pasteurización de la leche
- Figura 3.15. RdP para el proceso de embotellamiento de agua
- Figura 3.16. RdP para el proceso de clasificado y empaquetado de productos

CAPITULO 4

- Figura 4.1. Etapas representativas del GRAFCET
- Figura 4.2. Representación de un GRAFCET con etapas, transiciones y acciones
- Figura 4.3. Situación de conflicto
- Figura 4.4. Prioridad en transiciones
- Figura 4.5. Representación de una macro etapa con su respectiva secuencia

- Figura 4.6. Secuencia en serie
- Figura 4.7. Secuencia concurrente o en paralelo
- Figura 4.8. Salto
- Figura 4.9. Bucle
- Figura 4.10. Divergencia en OR
- Figura 4.11. Convergencia en OR
- Figura 4.12. Divergencia en AND
- Figura 4.13. Convergencia en AND
- Figura 4.14. Estado inicializado
- Figura 4.15. Temporización en etapas
- Figura 4.16. Temporización en transiciones
- Figura 4.17. Gráfico GEMMA con estados normalizados

CAPITULO 5

- Figura 5.1. Ejemplo de control simplificado

CAPITULO 6

- Figura 6.1. RdP para el sistema de producción de chocolates de barra y de bola
- Figura 6.2. Esquema de funcionamiento de la simulación electro-neumática
- Figura 6.3. Montaje neumático del sistema implementado
- Figura 6.4. Montaje hidráulico del sistema implementado
- Figura 6.5. RdP con los tiempos de operación del sistema implementado

LISTA DE TABLAS

CAPITULO 2

- Tabla 2.1. Eventos del ejemplo 2.1
- Tabla 2.2. Estados del ejemplo 2.1
- Tabla 2.3. Eventos del ejemplo 2.2
- Tabla 2.4. Estados del ejemplo 2.2
- Tabla 2.5. Convenciones utilizadas en el ejemplo 2.3
- Tabla 2.6. Eventos del ejemplo 2.3
- Tabla 2.7. Estados del ejemplo 2.3
- Tabla 2.8. Posiciones implementadas para el ejemplo 2.3
- Tabla 2.9. Eventos del ejemplo 2.4
- Tabla 2.10. Estados del ejemplo 2.4

CAPITULO 3

- Tabla 3.1. Lugares de la RdP de la Fig. 3.12
- Tabla 3.2. Lugares de recurso de la RdP de la Fig. 3.12

CAPITULO 6

- Tabla 6.1.1. Asignación de lugares para la RdP de la Fig. 6.1
- Tabla 6.1.2. Lugares de recurso para la RdP de la Fig. 6.1
- Tabla 6.2. Relación de tiempos (τ)

CAPITULO 1: Antecedentes

1.1 Introducción

Estas memorias presentan una primera aproximación a la teoría de los sistemas a eventos discretos (**SED**), específicamente las teorías relacionadas con Redes de Petri (**RdP**) y GRAFCET (**G**), además incluyen la teoría básica de Optimización aplicada a los sistemas industriales. La temática desarrollada se expone en seis capítulos, en donde se estipula la teoría básica de SED y Optimización, dando un enfoque práctico orientado según las necesidades y recursos con los que cuenta la Universidad Militar Nueva Granada (**UMNG**).

El primer capítulo muestra la introducción al documento plasmando los objetivos del proyecto, así como también su justificación e introducción. A partir del segundo capítulo se introduce al lector en las teorías de los sistemas de eventos discretos (SED), en donde se presenta su definición, contextualización y aplicación. Adicionalmente se exponen cuatro ejemplos trabajados durante el desarrollo de estas memorias, permitiendo la inter relación de los diferentes capítulos, llevando al lector tener una secuencia que facilita la asimilación de las teorías expuestas.

El tercer capítulo hace referencia a todo lo pertinente con las teorías de RdP, desde una definición básica hasta el desarrollo de ecuaciones dinámicas útiles en el proceso de modelado y optimización de procesos. Una RdP es una representación matemática de un sistema distribuido discreto, que generaliza la teoría de autómatas, permitiendo la expresión de eventos concurrentes. También pueden ser consideradas como una herramienta en el modelado de sistemas de información, que son considerados no-determinísticos, concurrentes, paralelos, asíncronos, distribuidos y/o estocásticos.

El cuarto capítulo muestra la teoría del GRAFCET (GRÁfico Funcional de Control de Etapas y Transiciones) que es un método de representación del funcionamiento de sistemas lógicos o SED, independientemente de la materialización tecnológica de los mismos. En el quinto capítulo se estudiará la teoría básica de optimización, la cual se tomó como base para en el desarrollo del modelo matemático planteado para la optimización de procesos industriales.

En el sexto capítulo se integran las teorías estudiadas a lo largo de las memorias, ya que se plantea un sistema hipotético de acuerdo a los recursos existentes en el laboratorio de Automatización del Programa de Mecatrónica de la UMNG. Este sistema fue modelado mediante teorías de RdP y controlado mediante las técnicas de G implementas en el PLC de Siemens S7-300.

El epílogo contempla las conclusiones y los posibles trabajos futuros del trabajo desarrollado. Así mismo, se presenta un anexo que introduce al software de simulación utilizado a lo largo del proyecto.

En síntesis, este documento presenta un aporte a las teorías de control y automatización de la **UMNG**, que complementa los cursos impartidos a lo largo de los contenidos programáticos de la carrera de Ingeniería Mecatrónica. Presenta una aplicación enfocada a la industria, específicamente en la optimización de procesos al maximizar recursos de máquina y minimizando tiempos de producción, lo que se traduce en mayores utilidades para la industria.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un estudio de los sistemas a eventos discretos, mediante RdP y G, implementando a nivel simulador y con aplicación de un modelo físico hipotético.

1.2.2 Objetivos Específicos

Establecer un documento de sistemas a eventos discretos, especificando técnicas de RdP, de G

Manejar software especializado en la programación y simulación de RdP y G.

Desarrollar un sistema hipotético, modelándolo y controlándolo, sujeto a los elementos existentes en la UMNG.

1.3 Justificación

Dentro de la comunidad neogranadina, en especial en el programa de Ingeniería Mecatrónica, no se cuenta con mayor información correspondiente al manejo y utilización de herramientas para el control de eventos discretos, centrados en teorías de RdP y G. Dada la poca profundidad de este tema en la UMNG, y considerando un curso realizado bajo el contexto de un intercambio con la Universidad Politécnica de Valencia, UPV N° 20 de 2006, nace la inquietud de profundizar y aplicar estas temáticas en nuestro ámbito académico Neogranadino. Por tal motivo, se pretendió desarrollar un soporte respecto a las bases teóricas de RdP, G y Optimización, en aras a su exploración e implementación, considerando los equipos existentes en el laboratorio de Automatización de la Universidad.

Adicionalmente, se presentará un modelo matemático que permite evaluar la evolución de estos sistemas. Conjunto con la noción de optimización, se plantea la selección de la mejor secuencia que permita maximizar el uso de los recursos y por ende mejorar la producción.

Es importante mencionar que con la orientación de las memorias aquí presentadas, el lector podrá poner en práctica las técnicas de modelado y control expuestas, ampliando los conceptos impartidos en los cursos de control básico.

CAPITULO 2: Introducción de los Sistemas a Eventos Discretos (SED)

En este capítulo se expondrán los fundamentos de los Sistemas a Eventos Discretos (SED), que son la base de la teoría planteada en este documento. Se introducirá el concepto de Sistemas de Manufactura Flexible (SMF), debido a su gran importancia e impacto dentro de la industria moderna y por ende en el desarrollo de las temáticas aquí presentadas. Adicionalmente se plantearán cuatro ejemplos: acceso a túnel compartido por dos trenes, embotelladora de agua, clasificación y empaquetado y por último, pasteurización de la leche. Estas problemáticas o ejemplos planteados se tratarán en el desarrollo de los siguientes capítulos.

2.1 Definición:

Los problemas de ingeniería contemplan variables físicas continuas que matemáticamente son modelados mediante ecuaciones diferenciales, donde aparece el concepto de "Simulación de Sistemas Continuos". Actualmente existen sistemas híbridos, cuyo funcionamiento requiere la combinación de sistemas continuos u discretos. Sin embargo, resultan inapropiadas las ecuaciones diferenciales en planteamiento y la solución de los sistemas híbridos. Por tal motivo, emergen técnicas de teorías en control digital, que satisfacen sus propias necesidades, las cuales se basan principalmente, en el cálculo de una solución aproximada en determinados instantes discretos de tiempo.

Es así como surgen los Sistemas a Eventos Discretos, que son el resultado de nuevos desarrollos en técnicas de control, introduciendo ventajas favorables con respecto a los métodos clásicos que trabajan la discretización del tiempo. Los SED reemplazan la discretización por la cuantificación de las variables de estado, mejorando la respuesta dinámica de los sistemas de control digital y reduciendo costo computacional y el tráfico de información entre la planta y el controlador.

Los SED fueron desarrollados por Bernard Zeigler a mediados de los setenta, han tenido una gran acogida en las ciencias de la computación debido al amplio desarrollo en aspectos prácticos y teóricos. Los SED permiten representar sistemas en los que el comportamiento de entrada y salida se encuentra descrito mediante secuencias a eventos, sujetos al estado que tenga un número finito de cambios dentro de un intervalo finito de tiempo¹.

¹ KOFNAM, Ernesto. Simulación y Control de Sistemas Continuos por Eventos Discretos. Facultad de Ciencias Exactas, Ingeniería y Agrimensura. Universidad Nacional del Rosario.

Los eventos representan un cambio instantáneo en un espacio determinado del sistema, se encuentran asociados a las posibles entradas del conjunto de estados.

Los estados pueden asociarse a salidas, acciones, niveles e impulsos. Los eventos se relacionan con las entradas del sistema y no tienen duración, es decir, son impulsos.

Una trayectoria está definida por una secuencia de eventos que toman un valor para todo el intervalo de tiempo, exceptuando el instante en el que ocurre un evento, en ese momento se asume el valor correspondiente al evento en sí.

Asumiendo los eventos como " x_n " y los tiempos como " t_n ", se representa una trayectoria mediante la siguiente gráfica.

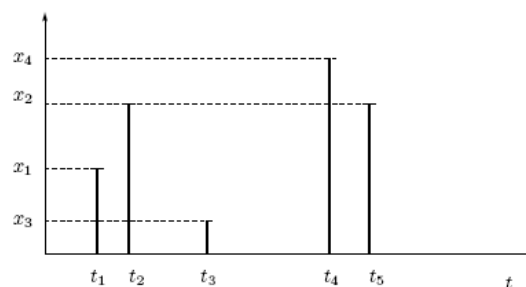


Fig. 2.1. Representación gráfica de una trayectoria

Un modelo de SED utiliza una trayectoria de eventos de entrada, que junto con las condiciones iniciales del mismo, produce una trayectoria de eventos de salida, tal como se puede observar en la Figura 2.2.



Fig. 2.2. Representación del procesamiento en un SED

La representación de sistemas a eventos discretos se utiliza ampliamente en el desarrollo de técnicas de control que reemplazan al control clásico. Está limitado a que representa solo conjuntos finitos de eventos y estados, en tiempo igualmente finito.

Uno de los campos de acción más fuerte, es el control automatizado de procesos, ya que se cuenta con una cantidad determinada de entradas y salidas que caracterizarán el sistema; adicionalmente, los SED se implementan sistemas tales como los de comunicaciones, de tráfico, de computación digital, de circuitos secuenciales, entre otros².

² AGUILAR, José. La Programación Evolutiva en la Identificación de Sistemas Dinámicos a Eventos Discretos. IEEE Latin America Transactions, Vol. 5, No. 5, September 2007.

2.2 Sistemas de Manufactura Flexible (SMF)

Los SMF son sistemas que constan de varias estaciones de trabajo o máquinas controladas comúnmente mediante el computador y conectadas entre sí por un sistema de alimentación y transporte, que permite el flujo de trabajos entre los componentes del sistema global. Esta estructura permite la manufactura de múltiples productos simultáneos brindando gran flexibilidad y adaptabilidad dentro del sistema [15].

Son beneficios de los SMF el incremento de utilización de las máquinas, minimización de máquinas requeridas, gran respuesta ante cambios, minimización de tiempos de entrega, reducción de personal en planta, entre otros. Como se puede observar, el objetivo principal de los SMF es el de optimizar la producción tomando como base los factores que puedan llegar a influir en ella.

Debido a que este tipo de sistemas es más complejo que los convencionales, su modelamiento y programación resulta tener un mayor nivel de complejidad. Por tal motivo, se hace necesario la implementación de las teorías de RdP que permiten modelar satisfactoriamente los SMF, así mismo, para encontrar la secuencia óptima de programación dentro de la producción resulta apropiado utilizar métodos de optimización, métodos heurísticos o técnicas híbridas que combinan las ventajas de diferentes métodos [14].

2.3 Ejemplos

Se presentarán cuatro diferentes situaciones, en primer lugar se evaluará el acceso a un túnel que es compartido por dos trenes que viajan en direcciones opuestas, en segundo lugar se presentará el caso de un proceso dentro de un sistema de embotellamiento de agua, en tercer lugar se analizará un sistema de clasificación y empaquetado de productos, por último se tratará el proceso general de pasteurización de la leche. A cada ejemplo se les deberá encontrar un conjunto de entradas y salidas que satisfagan la descripción de los sistemas propuestos.

Es importante aclarar que la solución planteada no es la única alternativa de desarrollo, el lector podrá encontrar otras soluciones que compensen las necesidades supuestas para cada una de las situaciones. Sin embargo, el lector deberá recordar la solución propuesta en estos enunciados ya que será utilizada

Ejemplo 2.1: Acceso a túnel compartido para trenes

El acceso a un túnel es compartido por dos trenes que van en sentido contrario, en vías independientes. Internamente solo es posible el tránsito de un solo tren. Identificar las entradas o eventos y las acciones o estados que presenta el sistema. El esquema físico, en términos generales, se encuentra en la Fig. 2.3.

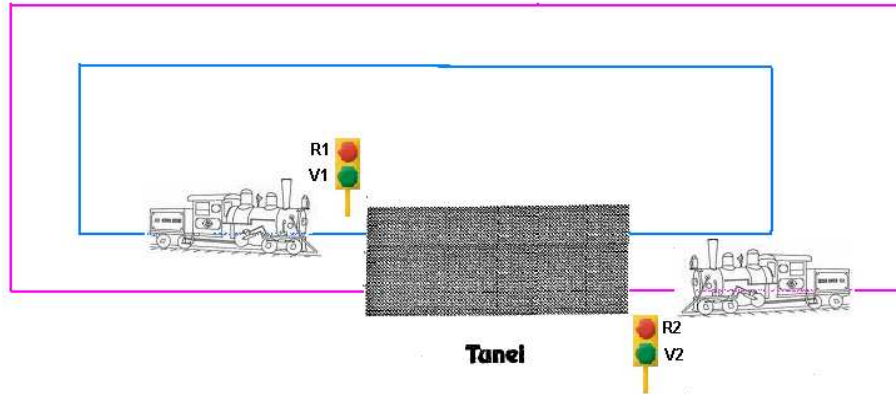


Fig. 2.3. Ejemplo 2.1. Túnel compartido para trenes

El acceso al túnel se encuentra señalizado mediante dos semáforos, uno por cada tren, los cuales tendrán una luz verde y otra roja indicando el acceso o acceso denegado de cada tren respectivamente.

Solución:

Se estimarán las entradas y salidas que sean necesarias; cabe recalcar que la solución planteada no es la única posibilidad de desarrollo.

Como se ha enunciado anteriormente, los eventos están asociados a las entradas del sistema. Se ha considerado entonces, que los eventos plasmados en la tabla, son suficientes para definir el sistema.

Variables Entrada (eventos)	Descripción
SR1	Sensor luz roja Tren 1
SV1	Sensor luz verde Tren 1
SR2	Sensor luz roja Tren 2
SV2	Sensor luz verde Tren 2
ET1	Posición entrada túnel Tren 1
ET2	Posición entrada túnel Tren 2
ST1	Salida del túnel Tren 1
ST2	Salida del túnel Tren 2

Tabla 2.1. Eventos del ejemplo 2.1

Igualmente, se ha estipulado que los estados se encuentran asociados a las acciones que puede efectuar el sistema en un momento determinado, o en otros términos, a las salidas del sistema. La siguiente tabla presenta la relación de estados.

Variables Salida (estados)	Descripción
MT1	Motor On Tren 1
MT2	Motor On Tren 2
R1	Luz acceso denegado túnel Tren 1
V1	Luz acceso túnel Tren 1
R2	Luz acceso denegado túnel Tren 2
V2	Luz acceso túnel Tren 2

Tabla 2.2. Estados del ejemplo 2.1

Ejemplo 2.2: Proceso de pasteurización de la leche

La pasteurización es uno de los métodos más comunes para conservación de la leche, se efectúa mediante un calentamiento que destruye los microorganismos y las enzimas que la deteriora y estropea. Existen diversos métodos de pasteurización, sin embargo, puede diferenciarse según la temperatura y el tiempo de tratamiento [20].

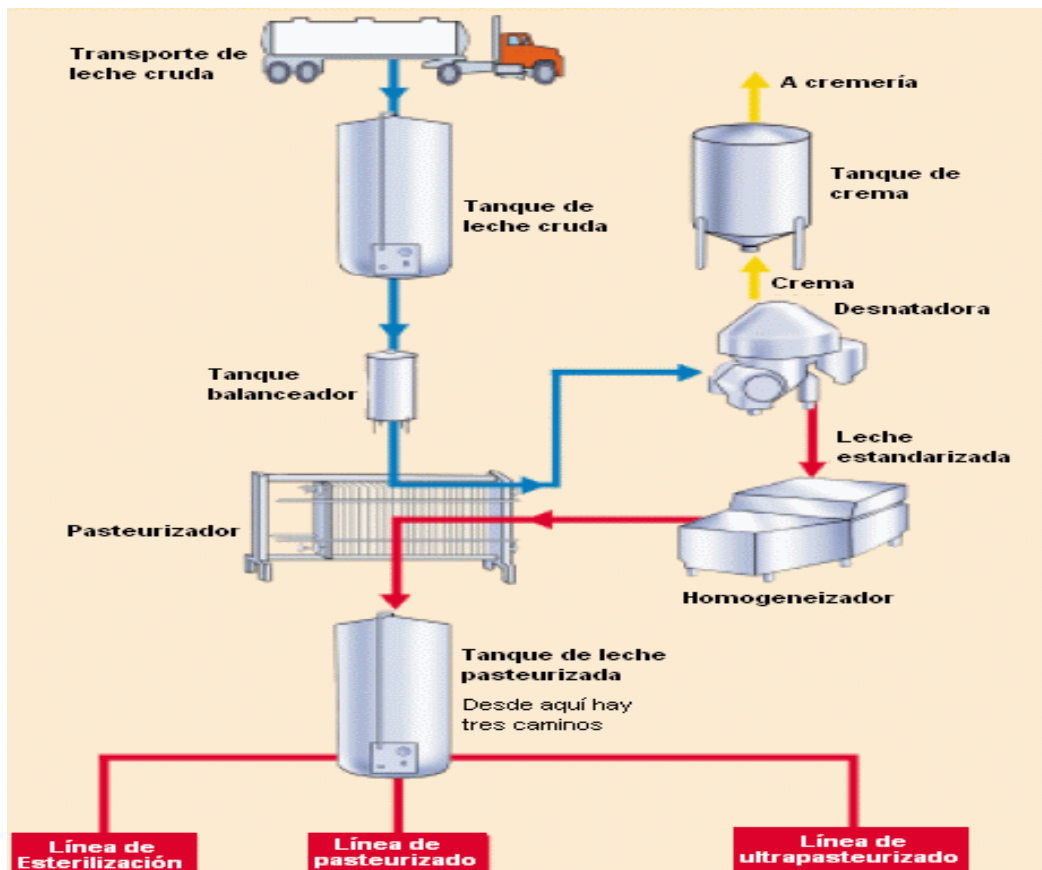


Fig. 2.4. Esquema general de la pasteurización de la leche

Para el ejemplo analizado en esta sección, se tomará como base el procedimiento que se contempla en el siguiente diagrama de bloques de la Fig. 2.5.

El proceso de pasteurización inicia con la recolección de la leche cruda proveniente de los establos, que debe estar a una temperatura de 4°C, mediante los camiones de recepción para posteriormente ser vaciada mediante ductos y ser remitida al proceso de pre enfriado. Durante el proceso de transporte la temperatura de la leche puede aumentar hasta un máximo de 8°C debido al tiempo que se tarda en llevar a cabo esta operación. Se requiere entonces un proceso de filtrado y enfriado, para dejar la leche nuevamente a una temperatura de 4°C para su posterior tratamiento.

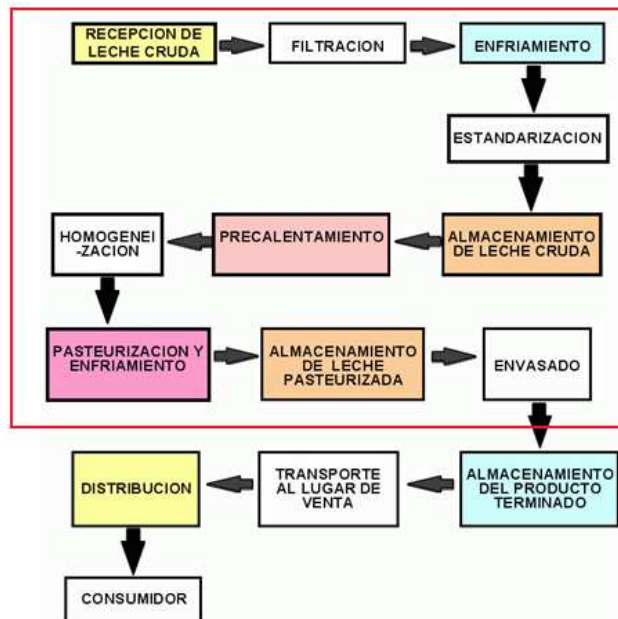


Fig. 2.5. Diagrama de bloques del proceso de pasteurización industrial

El proceso de filtración es realizado con el fin de evitar el ingreso de partículas gruesas al proceso, acto seguido se regula el caudal de entrada de la planta. Una vez filtrada, es llevada a la etapa de enfriamiento, que se efectúa mediante un intercambiador de calor, posteriormente debe realizarse un estandarizado, que consiste en homogenizar la mezcla del líquido por medio de agitadores. Se pasa luego a un almacenamiento de la leche cruda en contenedores de 6.000 L de capacidad, a una temperatura de 4°C, en donde se asegura la inhibición bacteriana.

La leche cruda se hace circular por el pasteurizador, que es un intercambiador de placas de tres niveles de calentamiento, mediante la circulación de agua caliente en contracorriente. Esta es la etapa principal del proceso en la cual se debe controlar la temperatura y el tiempo de permanencia en el pasteurizador. En este caso se eleva la temperatura hasta 75 °C y se regula el flujo para que la leche permanezca por lo menos 12 segundos dentro del pasteurizador.

Posteriormente, la leche pasteurizada es almacenada en contenedores para su posterior envasado, que dependerá del proceso implementado: Esterilizada, pasteurizada, ultra pasteurizada.

En la Fig. 2.6 se muestra el diagrama según las operaciones térmicas y químicas que se realizan durante el proceso. Esto nos servirá para visualizar con mayor facilidad el proceso que debe modelarse.

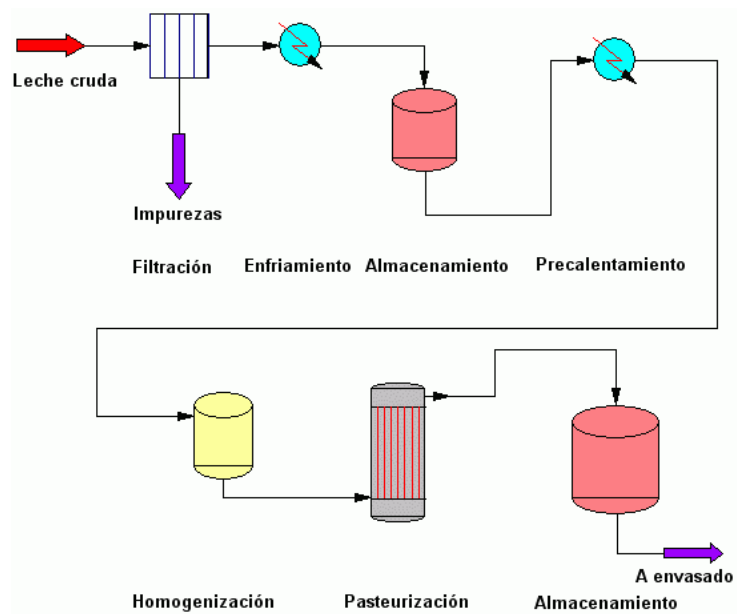


Fig. 2.6. Diagrama de equipo del proceso de pasteurización industrial

Solución:

De acuerdo a la descripción del proceso enunciada anteriormente y según las figuras 2.4, 2.5 y 2.6, que plasman secuencialmente el procedimiento para la pasteurización de la leche, se presenta a continuación una propuesta para desarrollar el problema mediante sistemas de eventos discretos. La tabla 2.3 enuncia las variables de entradas y salidas que enmarcan el proceso total.

Variables Entrada (eventos)	Descripción
START	Start - Inicio del proceso
ELC	Entrada Leche Cruda
FT	Filtrado Total
T4	Tempeperatura 4°
E_LECHE	Leche estandarizada
LL6000	Tanque lleno 6000 L
T40	Temperatura 40°
H_LECHE	Leche Homogenizada
T75	Temperatura 75°C
T_12	Tiempo 12 seg
LE	Leche totalmente fuera para envasado

Tabla 2.3. Eventos del ejemplo 2.2

Variables Salida (estados)	Descripción
VRL	Valvula Recepción Leche
ENF	Enfriamiento Recepción
ME	Mezclador Estandarización
VAP	Válvula Almacenamiento Previo
P	Pre calentamiento
MH	Mezclador Homogeneización
PAST	Calentamiento – Pasteurización
EP	Enfriamiento Pasteurización
VALP	Válvula Almacenamiento Leche Pasteurizada
VE	Valvula a Envasado

Tabla 2.4. Estados del ejemplo 2.2

Ejemplo 2.3: Embotelladora de agua

En la industria de producción de agua, luego de realizar los procesos pertinentes a la purificación y tratamiento, se requiere el envasado del líquido para su posterior almacenamiento y distribución.

En el siguiente ejemplo, se tendrá en cuenta el embotellamiento del agua procesada, no se modelará el llenado del líquido, por lo que se asumirá que las botellas ya se encuentran llenas. La tarea consiste en ubicar las botellas de agua en su respectiva canasta y realizar la operación de tapado para cada una de ellas.

Este proceso consta de tres bandas transportadoras, la primera contiene las botellas de agua, la segunda contiene la canasta de almacenamiento de las botellas y la tercera contiene las tapas de las botellas. El sistema consta de dos manipuladores, uno de estos ubica las botellas dentro de la canasta y el otro tapa herméticamente las botellas. Existe un último manipulador que ubica la canasta llena de botellas tapadas en la zona de transporte para su posterior almacenamiento y distribución.

El sistema propuesto se modeló mediante la herramienta de simulación en 3D denominada IGRIP. En la figura 2.4 se puede observar las condiciones iniciales de la simulación que cumplen las especificaciones descritas para el sistema. En la simulación se realizarán las operaciones de tapado y almacenamiento de 12 botellas de agua para su respectiva canasta de distribución.

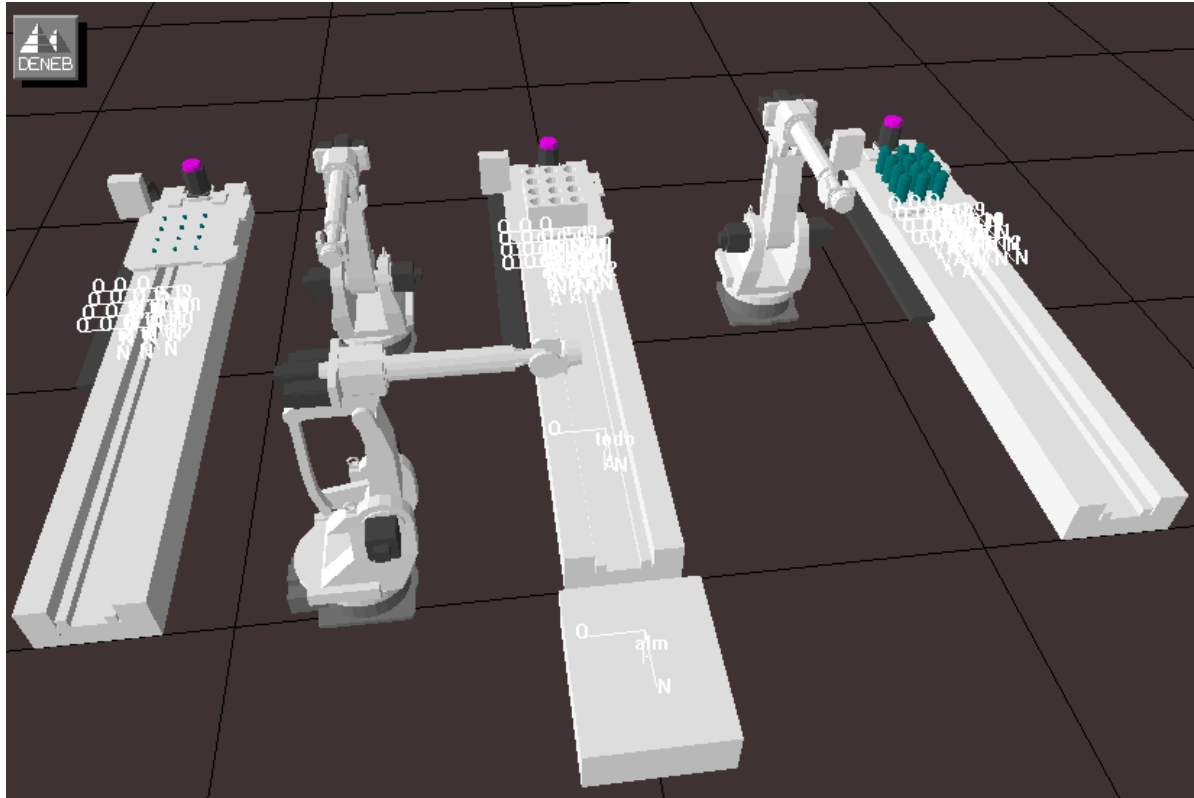


Fig. 2.7. Proceso de tapado y almacenamiento de botellas de agua

El proceso se describe así: una botella es llevada hacia la canasta por el primer manipulador, en ese momento, el segundo manipulador sellará la botella mientras el primero va por otra botella y así sucesivamente hasta llenar por completo la canasta. Una vez la canasta se encuentra llena, la banda transportadora de la canasta avanzará hasta llegar al tercer manipulador, el cual se encargará de trasladar la canasta a la mesa en donde será recogida para su posterior transporte y distribución.

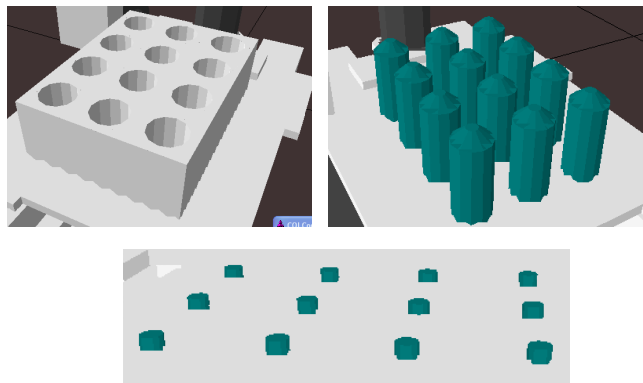


Fig. 2.8. Elementos del proceso (Canasta, botellas y tapas)

Los elementos que intervienen en el proceso, tales como la canasta, las botellas y las tapas, se aprecian en la figura 2.5.

Para el sistema descrito anteriormente, se deben plantear variables de entradas y salidas, que permitan controlar dicho sistema.

Solución:

Para el desarrollo del ejemplo planteado, se establecieron las siguientes variables que incluirán a los estados y a los eventos implícitos en proceso.

A fin de tener clara la nomenclatura utilizada en la solución de este ejemplo, se muestra en la tabla 2.5, algunas convenciones importantes dentro del desarrollo del problema.

MB	Manipulador Botellas
MT	Manipulador Tapas
MC	Manipulador Canasta
i'	i=1,2,...,12

Tabla 2.5. Convenciones utilizadas en el ejemplo 2.3

La tabla 2.6 muestra el marco de eventos que fue diseñado para el sistema de embotellamiento de agua, dentro de la cual, se exponen las señales necesarias para la automatización total del sistema.

Es importante aclararle al lector que esta no es la única solución existente, ya que es posible variar los sensores o señales que interactúen en el sistema, lo importante es que cada planteamiento cumpla con las especificaciones estipuladas.

	Variables Entrada (eventos)	Descripción
	Sbtt	Sensor Banda Transportadora Tapas
	Sbtb	Sensor Banda Transportadora Botellas
	sbtc1	Sensor Banda Transportadora Canastas 1
	sbtc2	Sensor Banda Transportadora Canastas 2
MB	sb(i)	Señal de llegada de MB a la botella 'i'
	sbc(i)	Señal de llegada de botella 'i' a canasta
MT	st(i)	Señal de llegada de MT a la tapa 'i'
	stb(i)	Señal de llegada de tapa 'i' a botella 'i'
MC	Spbc	Señal de llegada de MC a la canasta
	Spmd	Señal de llegada de MC a mesa de distribución

Tabla 2.6. Eventos del ejemplo 2.3

Los estados o variables de salida propuestos para el sistema se describen en la tabla 2.7. Algunas de las variables de movimiento incluyen una posición dinámica, como por ejemplo el manipulador de tapas, que tiene diferentes posiciones finales, para lo cual se plantea una sola variable de salida que cambiará de posición de acuerdo un registro que sea seleccionado según sea el caso deseado. La tabla 2.8 plasma cada una de las posiciones implantadas en el sistema de embotellamiento de agua.

Variables Salida (estados)	Descripción
MBT	Motor Banda de Tapas
MBC	Motor Banda Canastas
MBB	Motor Banda Botellas
PMB(i)	Posición Manipulador Botellas
PMT(i)	Posición Manipulador Tapas
PMC(i)	Posición Manipulador Canasta
GMB	Griper Manipulador Banda
GMT	Griper Manipulador Tapas
GMC	Griper Manipulador Canastas

Tabla 2.7. Estados del ejemplo 2.3

Las variables que incluyen posición se representan de la siguiente forma: VAR(i), donde 'VAR' equivale al nombre de la variable e 'i' a las posibles posiciones que puede llegar a tomar dicha variable.

	Posiciones	Descripción
MB	Hmb	Home MB
	b(i)	botella 'i'
	bc(i)	botella 'i' en canasta
MT	Hmt	Home MT
	t(i)	tapa 'i'
	tb(i)	tapa 'i' en botella 'i'
MC	Hmc	Home MC
	Pbc	Posición Banda Canasta
	Pmd	Posición Mesa Distribución

Tabla 2.8. Posiciones implementadas para el ejemplo 2.3

Cabe recalcar que tanto las variables de entrada como las de salida, se implementarán en posteriores capítulos cuando se estudien las técnicas de RdP y de GRAFCET.

Ejemplo 2.4: Proceso de clasificación y empaquetado

En diversos procesos de la industria se ve la necesidad de clasificar, ya sea productos terminados para su empaquetado y distribución, materias primas para el desarrollo de un producto en especial, o en etapas intermedias de los procesos. Por tal motivo se presenta a continuación un ejemplo que involucra la clasificación de tres diferentes productos y el empaquetado de dos de estos.

El sistema consta de una banda transportadora B_0 que lleva los productos A, B y C, en el momento en que finalizar el recorrido de la banda, un manipulador identificará individualmente el tipo de producto seleccionado y lo trasladará a otras bandas según corresponda, los productos A serán dirigidos la banda B_1 , los B a la banda B_2 y los C a la banda B_3 . Los productos C serán trasladados mediante la banda B_3 al almacén 1.

En la figura 2.6 se muestra la simulación del sistema mediante la herramienta de simulación en 3D llamada QUEST.

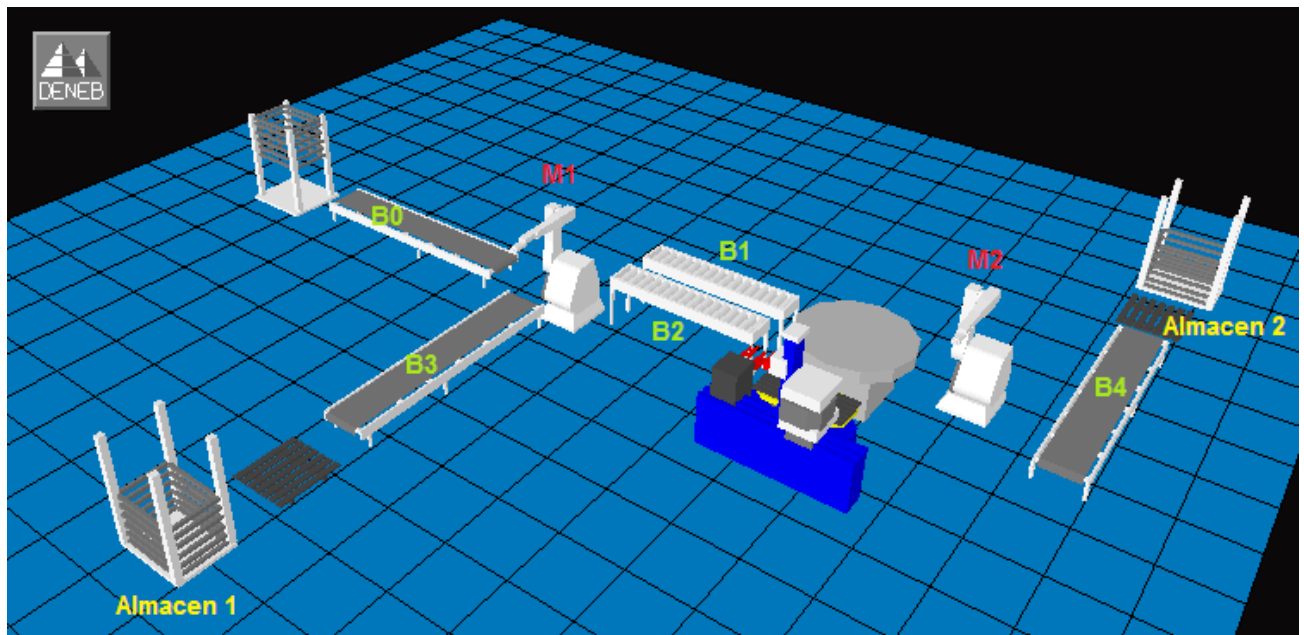


Fig. 2.9. Simulación del sistema de clasificación y empaquetado

Las bandas B_1 y B_2 estarán ubicadas paralelamente, al finalizar dichas bandas se encontrará una mesa giratoria que rotará 90° al momento en que le llegue un producto A y un producto B. Una vez ha girado la mesa, se encontrará frente a una máquina empacadora que unirá los productos A y B en un solo paquete, finalizada esta operación, la mesa girará nuevamente 90° en el mismo sentido, en ese momento un segundo manipulador tomará el paquete llevándolo de la mesa a la banda B_4 , en donde, el nuevo producto será dirigido hacia el almacén 2.

Se debe plantear un sistema de eventos y estados discretos para la automatización del sistema planteado.

Solución:

Para el desarrollo de este ejemplo, se plantearon las variables de entrada estipuladas en la tabla 2.9, las cuales están ligadas directamente con los diferentes eventos presentes dentro del sistema.

Nuevamente se hace hincapié al lector en que la solución planteada no es la única solución posible, existen múltiples alternativas de solución, que dependerán de las señales requeridas por el sistema y de las variables que intervengan dentro del mismo, las cuales serán implementadas libremente por el diseñador.

Variables Entrada (eventos)	Descripción
sb0	Sensor banda 0
sb1	Sensor banda 1
sb2	Sensor banda 2
sb3	Sensor banda 3
sb4	Sensor banda 4
m1b0	M1 en B0
m1b1	M1 en B1
m1b2	M1 en B2
m1b3	M1 en B3
m1home	M1 en posición home
m2mg	M2 en mesa giratoria
m2b4	M2 en B4
m2home	M2 en posición de home
Smg	Sensor mesa giratoria
Sme	Sensor terminao ME

Tabla 2.9. Eventos del ejemplo 2.4

Las variables de salida que representan los estados del sistema se encuentran determinados en la tabla 2.10. En esta representación 'i' hace alusión a cada una de las posiciones que puede tomar tanto el manipulador M1 como el manipulador M2.

Variables Salida (estados)	Descripción
B0	Motor banda 0
B1	Motor banda 1
B2	Motor banda 2
B3	Motor banda 3
B4	Motor banda 4
PM1(i)	Posición Manipulador 1
PM2(i)	Posición Manipulador 2
MGM	Motor giro mesa
ME	Maquina empaquetado

Tabla 2.10. Estados del ejemplo 2.4

2.5 Conclusiones

Los Sistemas a Eventos Discretos permiten el desarrollo de herramientas de automatización que satisfacen las exigencias de la nueva industria, que requiere mayor calidad y demanda de producción, tal es el caso de los Sistemas de Manufactura Flexible, los cuales pretenden abordar a nivel superior diferentes procesos industriales.

En el transcurso de este capítulo se dieron las bases teóricas de los SED y de los SMF, con el fin de centrar al lector en la temática central del proyecto, introduciendo conceptos básicos que serán utilizados a lo largo de este documento.

Adicionalmente, se trataron 4 diferentes problemas a los cuales se les planteo una alternativa de solución para un posterior modelado mediante herramientas de control automático, por tal motivo el lector deberá tener en cuenta el desarrollo de dichos ejemplos en el momento de revisar la solución de los ejemplos propuestos para los siguientes capítulos.

CAPÍTULO 3: Redes de Petri (RdP)

Las Redes de Petri (**RdP**) son una técnica para trabajar sistemas a eventos discretos, utilizando esta técnica es posible modelar sistemas híbridos complejos, como lo puede llegar a ser los procesos concurrentes con recursos compartidos.

En el desarrollo de este capítulo, se mostrará al lector una introducción del funcionamiento de las RdP, una vez se tengan los fundamentos básicos, se revisarán las características y propiedades de las RdP con el fin de entrar al estudio de las ecuaciones dinámicas que pueden representar un sistema modelado mediante RdP. Adicionalmente se mostrará un modelo matemático planteado para el cálculo de tiempos de proceso total para efectos de optimización de secuencias de producción en sistemas industriales. Se retomarán los ejemplos vistos en el capítulo anterior con el fin de acoplar la teoría asimilada a lo largo del capítulo de Redes de Petri.

3.1 Definición y Contexto Histórico

Las Redes de Petri (RdP) son introducidas por Carl Adam Petri sobre los años 60, fueron desarrolladas como una herramienta para simular propiedades dinámicas de los sistemas complejos mediante modelos gráficos de procesos concurrentes y distribuidos³. Una RdP es un método de modelización de Sistemas a Eventos Discretos mediante el cual es posible controlar, evaluar y optimizar diferentes procesos.

La estructura de una **Red** se define:

$$R = \{P, T, I, O\} \quad (3.1)$$

Donde: P: Conjunto de lugares (n).
T: Conjunto de transiciones (m).
I: Conjunto de entradas de arco (PxT) → Post
O: Conjunto de entradas de arco (TxP) → Pre

Se define una **RdP** como:

$$RP = \{R, Mo\} \quad (3.2)$$

$$RP = \{P, T, I, O, Mo\} \quad (3.3)$$

³ PEÑA P. Alirio J. Sobre las Redes de Petri R-difusas. On R-fuzzy Petri Nets. Divulgaciones Matemáticas Vol. 7. No.1. 1999, pp. 87-99. Laboratorio de Algebra teórica y computacional (LATyC). Facultad de Ciencias, Universidad de Zulia, Venezuela.

Donde Mo es el vector $n \times 1$ que representa el marcado inicial de la red.

Con el fin de evaluar el rendimiento de un sistema y de analizar procesos dinámicos, se introducen las Redes de Petri con Tiempo (**RDT**), que se definen como la séxtupla:

$$RPT = \{RP, \tau\} \quad (3.4)$$

$$RP = \{P, T, I, O, Mo, \tau\} \quad (3.5)$$

Donde τ es el conjunto de tiempos de espera (time delay) que necesitan los lugares o las transiciones para ser habilitados(as). Si dicho tiempo corresponde a los lugares, se denomina Red de Petri con Tiempo de Lugar (**RPTL** o TPPN por sus siglas en inglés Time Place Petri Net) y si por el contrario, es asociado a las transiciones, se denomina Red de Petri con Tiempo de Transición (**RPTT** o TTPN por sus siglas en inglés Transition Time Petri Net)

3.2 Elementos de una RdP

Las RdP son grafos orientados compuestos por lugares y transiciones que se encuentran unidos por arcos, tal como se aprecia en la figura 3.1.

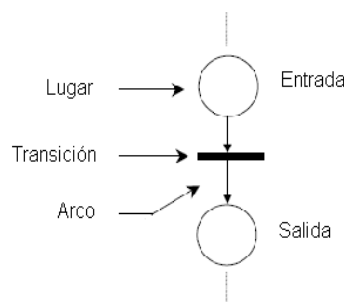


Fig. 3.1. Elementos de una RdP

Los lugares representan los estados del modelo, debido a que las RdP son dirigidas, se tiene un lugar tanto de entrada como de salida para cada transición. Los arcos indican la dirección evolución de la RdP.

Un token (marca) representa aquel estado que se encuentra activo, se representa mediante un punto dentro del lugar, con lo cual, cada lugar tiene un número positivo de tokens o en su defecto ninguno.

Las transiciones son las entradas del sistema, de su activación o desactivación dependerá el funcionamiento de la red. Una transición puede ser habilitada y/o disparada; se dice que está habilitada si cada uno de los lugares de entrada tiene el

número necesario de marcas indicado; y se producirá el disparo en el momento en que se remuevan las marcas de los lugares de entradas y se añadan en los de salida.

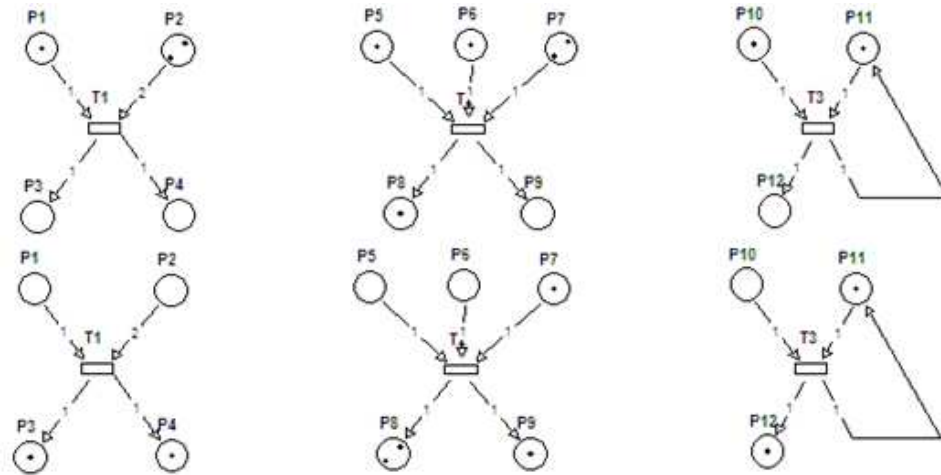


Fig. 3.2. Ejemplos de evolución de una red de Petri

La evolución de una RdP se contempla en la Fig. 3.2. En el momento en que la transición se habilite y active, la red evolucionará, pasando el token del lugar de entrada al lugar de la salida de la transición.

El número de tokens que pasarán al siguiente estado evolucionado, será el indicado en los pesos de los arcos, si no lleva ningún número se asume que es uno.

3.3 Marcado de una RdP

Se le llama Marcado al conjunto de marcas o tokens que se encuentran en toda una RdP. Se representa mediante un vector, en el cual, el término i -ésimo es el número de marcas del lugar P_i .

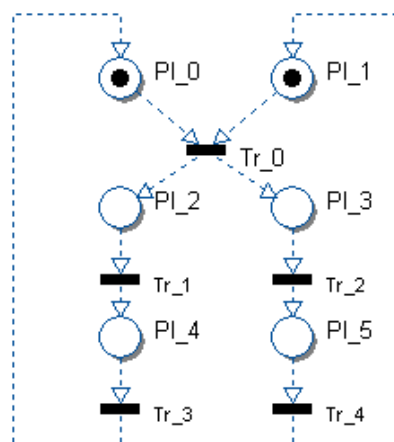


Fig. 3.3. Representación y marcado de una RdP

Una red marcada es el conjunto formado por $\{R, M\}$

Donde: R es una red (definida en 3.1)
M es una aplicación denominada marcado.

Para definir el vector de marcado se debe estipular el orden de los lugares, pues de éste dependerá el resultado del modelamiento y análisis de la red, una vez asignado dicho orden no podrá ser modificado. Se pondrá 0 en el lugar que no tenga marcas, 1 en el lugar que tenga una marca, 2 en el lugar en que tenga dos marcas, así sucesivamente.

$$M_0 = [P_1, P_2, P_3, P_4, P_5, P_6]^T \quad (3.6)$$

De esta manera, el marcado de la RdP expuesta anteriormente es el siguiente.

$$M_0 = [1, 0, 0, 1, 0, 0]^T \quad (3.7)$$

3.4 Árbol de Cobertura

El árbol de cobertura es la representación esquemática de la evolución de una RdP, evalúa todos los estados posibles de la red determinando el comportamiento de la misma.

Para realizar el árbol de cobertura se utilizar el vector de marcado, que representa el estado de la red en un instante determinado. Se inicia con el marcado inicial (M_0) y posteriormente se evalúa disparo a disparo los posibles marcados alcanzables. En la figura 3.4 y 3.5 se exponen dos ejemplos de RdP a los cuales se desarrollará el respectivo árbol de cobertura.

El árbol de cobertura para la RdP E1 es representado en la Fig. 3.6. Como se mencionó anteriormente, el árbol de cobertura permite demostrar las características de una RdP. Analizando la estructura de la Fig. 3.6, se determina que la red alcanza tres estados diferentes y durante su evolución atraviesa por todos los estados sin llegar a bloquearse en ningún punto, con lo cual se afirma que la red es viva, es binaria y limitada debido a que en cada lugar tendrá como máximo un token y es conforme puesto que es viva y binaria.

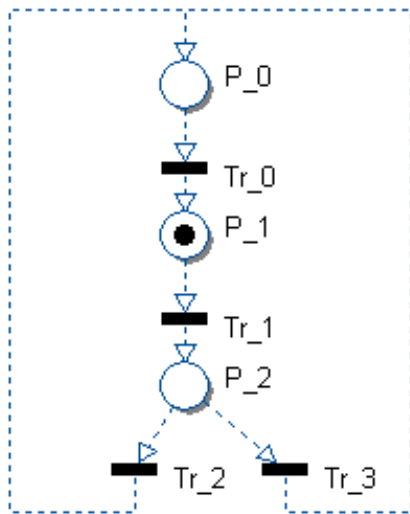


Fig. 3.4. RdP E1

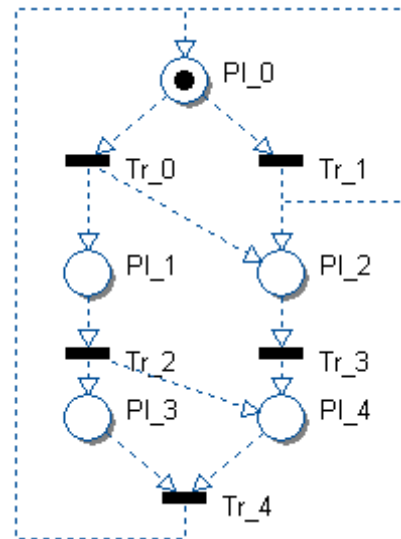


Fig. 3.5. RdP E2

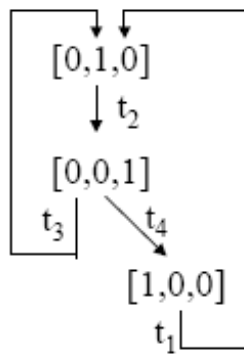


Fig. 3.6.1. Árbol de cobertura de la RdP E1

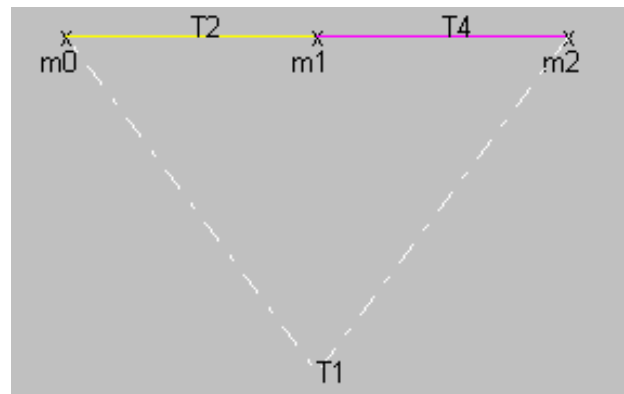


Fig. 3.6.2. Árbol de cobertura de la RdP E1 generado por Matlab

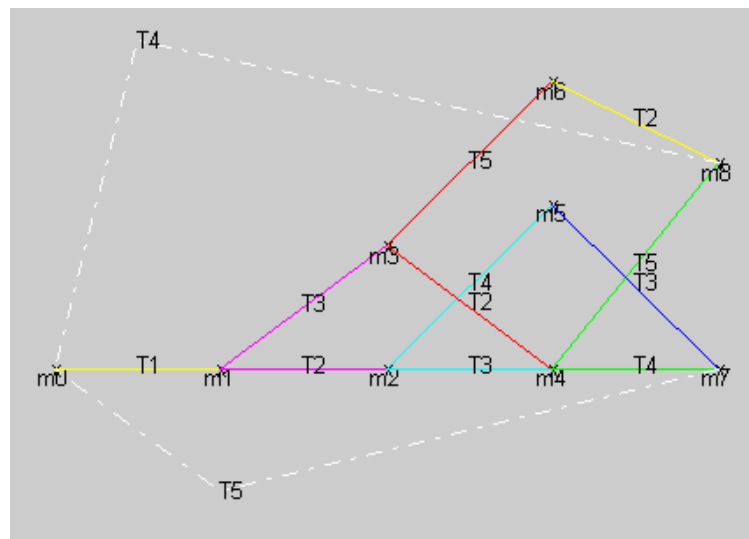


Fig. 3.7. Árbol de cobertura de la Fig. 3.3.

Para la red expuesta en la figura 3.5, no es posible obtener una representación finita de su árbol de cobertura ya que los lugares PI_2 y PI_4 crecerán ilimitadamente. Manualmente es posible llegar a un árbol con estados finitos si se delimita el número y la secuencia de disparos, pero nunca se obtendrá el árbol global y general de la red. Para esta red se cumple la característica de vivacidad, pero según lo argumentado anteriormente, no es ni limitada ni conforme.

Existen simuladores capaces de analizar las características de las RdP. En este proyecto se trabajó específicamente con tres de ellos. En primer lugar se utilizó HiLeS DESIGNER desarrollado por el LAAS en Toulouse Francia [10], que junto con el paquete TINA (Time Petri Nets Analyzer) [11], desarrollado por el mismo instituto, es una poderosa herramienta en el modelamiento de sistemas físicos mediante RdP.

En segundo lugar, se trabajó con el simulador PM Editor 3.1 (Petri Maker Package) [12] desarrollado por el departamento de informática de la Universidad de Hamburgo – Alemania. Este editor permite exportar las redes modeladas a Matlab mediante archivos '.rdp'.

Por último se trabajó con el toolbox de RdP para Matlab [13], desarrollado por el Departamento de Control Automático e Informática Industrial de la Universidad Técnica Gheorghe Asachi – Rumania. Mediante este Toolbox matlab recibe los archivos .rdp y analiza las características de la RdP modelada en el PM Editor. Es así como se obtuvo las representaciones correspondientes de las figuras 3.6 y 3.7.

En la figura 3.6.2 se expone el árbol de cobertura generado por el toolbox de Matlab, como se aprecia, los resultados concuerdan con el árbol que presenta la figura 2.6.1, en los cuales se tienen tres posibles estados para la red con su respectiva secuencia de disparo de activación.

Del mismo modo se presenta en la figura 3.7 el árbol de cobertura generado para el ejemplo presentado en la figura 3.3, en donde se alcanza un total de 9 estados siguiendo cada una de las posibles secuencias de disparo.

Se concluye entonces que los simuladores de RdP son una poderosa herramienta que facilita y agiliza el análisis de las redes de petri.

3.5 Características de las RdP

Dentro de los parámetros primordiales en la caracterización de las RdP, se contempla el que una RdP pueda ser viva, limitada, binaria y/o conforme.

Una RdP es viva si durante toda su evolución no tiene bloqueos en ningún punto, si cada una de las transiciones es disparada en algún instante de tiempo y si cada uno de los lugares es activado por lo menos una vez. En otras palabras, la evolución de la RdP debe cubrir todos y cada uno de los elementos de la red sin llegar a bloquearse.

Es limitada cuando cada uno de los lugares crece de forma definida. Una de las formas limitadas es la binaria, se dice entonces que una RdP es binaria cuando los lugares de la red se encuentran restringidos a tener máximo 1 token, es decir que en los valores de marcado solo se tendrán ceros y unos. La característica de conformidad se da cuando la RdP es viva y al mismo tiempo binaria.

Se debe tener en cuenta que las características de una RdP dependen única y exclusivamente de su marcado inicial (M_0). Es posible tener un esquema de RdP con diferentes marcados iniciales, lo cual generará la variación de las características de la red. Las características de una RdP se demuestran mediante el árbol de cobertura.

3.6 Otras Propiedades de las RdP

3.6.1 Recursos compartidos (RC)

Existen procesos paralelos o concurrentes en los cuales puede existir algún tipo de elemento, al que lo llamaremos “recurso”, que es compartido. Cuando esto sucede, únicamente el estado o lugar posea el recurso podrá ejecutar la secuencia de acciones predeterminadas para llevar a cabalidad el proceso.

En las figuras 3.8 se aprecia dos procesos concurrentes con recurso compartido. Para este caso, solo evolucionará el lugar 3 (PI_3) si el recurso compartido (PI_6) de la red se encuentra disponible, ver figura 3.8.2, así mismo, hasta el instante en que el recurso se encuentre nuevamente disponible, podrá evolucionar la red del lugar 7 (PI_7) al lugar 8 (PI_8), ver figura 3.8.3.

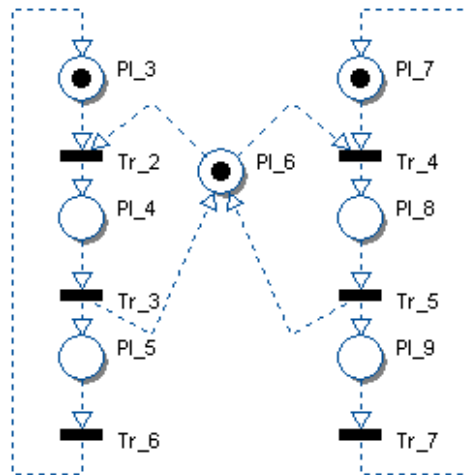


Fig. 3.8.1. Recurso Compartido

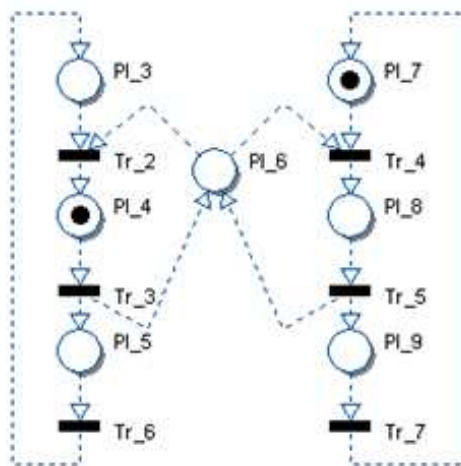


Fig. 3.8.2. Evolución de la red con RC por la izquierda

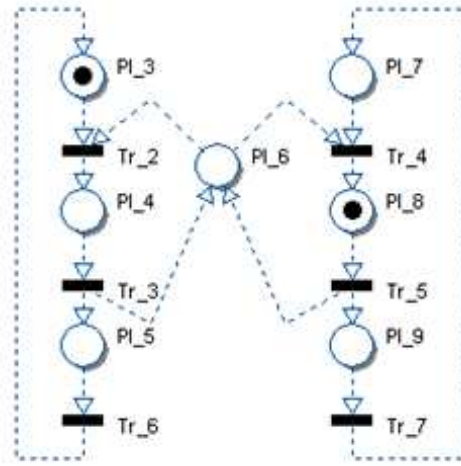


Fig. 3.8.3. Evolución de la red con RC por la derecha

3.6.2 Conflictos entre Transiciones

Si se tienen dos transiciones simultáneamente habilitadas compartiendo el mismo lugar de entrada, tal como se ve en la figura 3.9, se generará un conflicto. Esto es debido a que computacionalmente no es posible realizar dos procesamientos exactamente en el mismo instante de tiempo.

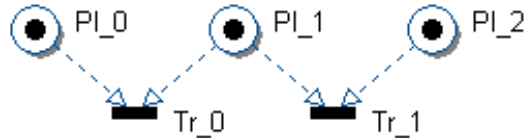


Fig. 3.9. Conflicto entre transiciones

En aras de solucionar el conflicto generado, se hace necesario garantizar que el disparo de las transiciones no se realice de forma sincrónica, es decir, en el mismo instante de tiempo, por tal motivo una solución que resulta conveniente es darle prioridad a las transiciones, tal como se muestra a continuación:

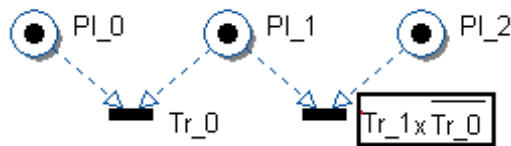


Fig. 3.10. Solución al conflicto entre transiciones

En la figura 3.10 se aprecia cómo se prioriza la transición 0 (Tr_0) en caso de activación simultánea.

3.6.3 Concurrencia y Sincronización

Una concurrencia se da en el momento en que una transición activa procesos independientes para un mismo instante de tiempo, o en otras palabras, la concurrencia en una RdP sucede cuando una transición habilita la ejecución de procesos paralelos.

Para el caso de la figura 3.11, la transición Tr_8 habilita dos procesos paralelos. Un proceso indicado por los lugares PI_10, PI_12 y PI_14 y el otro proceso por los lugares PI_11, PI_13 y PI_15.

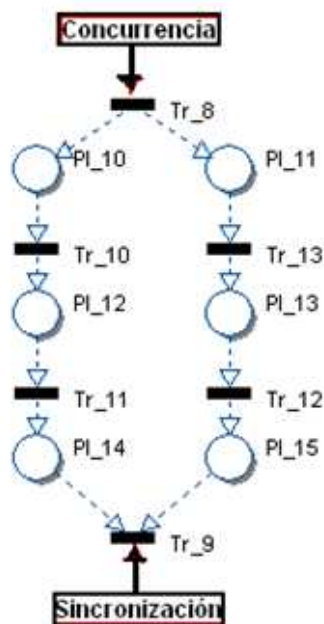


Fig. 3.11. Concurrencia y sincronización en una RdP

La sincronización de una RdP, ocurre en el momento en que una transición espera a que todos los procesos paralelos anteriores a ella hayan concluido, continuando con la evolución de la red.

Para el ejemplo de la figura 3.11, la transición Tr_9 solo se activará en el momento en que exista un token en los lugares PI_14 y PI_15, indicando que los procesos paralelos han finalizado. Para la figura 3.3, la transición Tr_0 representa tanto una concurrencia como una sincronización.

3.7 Modelado de Procesos Concurrentes con Recursos Compartidos (PCRC) mediante RdP

Como se mencionó en el capítulo 2, los Sistemas de Manufactura Flexible (SMF) son capaces de ejecutar procesos paralelos o concurrentes debido a la flexibilidad que estos sistemas admiten. Para modelar este tipo de sistemas se cuenta con herramientas como lo son las RdP, capaces de involucrar además de los procesos paralelos, las máquinas o recursos compartidos con los que cuente el sistema.

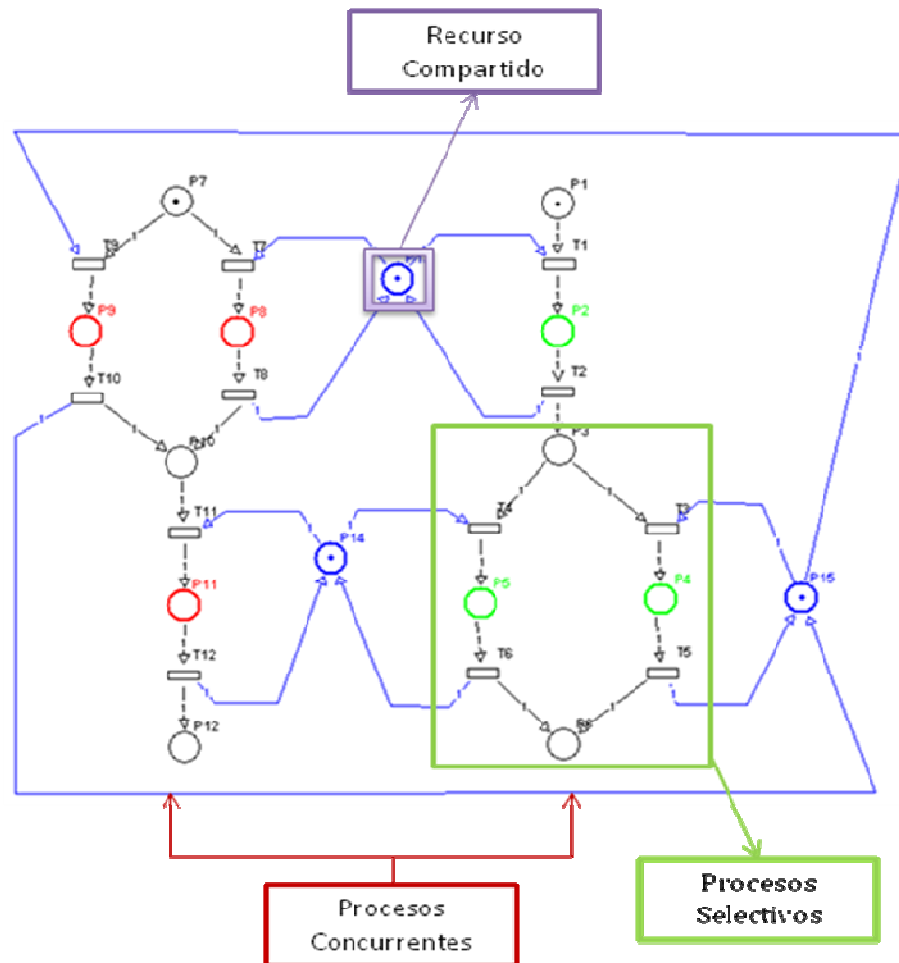


Figura 3.12. Modelado de PCRC mediante RdP

La figura 3.12 presenta un ejemplo de RdP de Procesos Concurrentes con Recursos Compartidos (PCRC).

Lugar	PROCESO 1	Lugar	PROCESO 2
P1	Inicialización Proceso 1	P7	Inicialización Proceso 2
P2	Operación 1: trabajo 1 procesado por maquina 1	P8	Operación 1: trabajo 2 procesado por maquina 1
P3	Operación 1 de proceso 1 concluida	P9	Operación 1: trabajo 2 procesado por maquina 32
P4	Operación 2: trabajo 1 procesado por máquina 3	P10	Operación 1 de proceso 2 concluida
P5	Operación 2: trabajo 1 procesado por máquina 2	P11	Operación 2: trabajo 2 procesado por máquina 2

P6	Operación 2 de proceso 1 concluida	P12	Operación 2 de proceso 2 concluida
----	------------------------------------	-----	------------------------------------

Tabla 3.1. Lugares de la RdP de la Fig. 3.12

Lugar	MAQUINAS
P13	Máquina 1
P14	Máquina 2
P15	Máquina 3

Tabla 3.2. Lugares de Recurso de la RdP de la Fig. 3.12

Se hace necesario el planteamiento de algunos conceptos adicionales a los ya estudiados, los cuales permitirán el modelamiento de sistemas mediante RdP, tales como: lugar de inicialización y finalización, lugar de operación, lugar de recurso, sistema y secuencia de actividades [16] [19].

Un lugar de inicialización es aquel que no tiene arcos de entrada y representa el inicio de un proceso; un lugar de finalización es aquel que no posee arcos de salida y representa la culminación de un proceso.

El lugar de operación es aquel que representa una acción dentro de un proceso (operación), mientras que el lugar de recurso representa la(s) máquina(s) involucradas en el sistema. Los lugares que no tienen asociadas acciones o recursos, indican la culminación de operaciones. En la tabla 3.1 y 3.2 se plasma la representación de cada uno de los lugares asociados a la RdP de la Fig. 3.12

Una Secuencia de Actividades (SA) es una sub RdP asociada que modela un proceso asociado con un trabajo. Las SA llevan implícitas restricciones, rutas o caminos alternativos y tiempos de operación. Cada SA tiene un único lugar de inicialización y de finalización, se puede inferir entonces que una SA no contiene ciclos.

Un sistema se define como el conjunto de SA. La RdP de la Fig. 3.12 es unos sistemas conformados por dos procesos o SA concurrentes, el proceso 1 y el proceso 2, con tres máquinas o recursos compartidos. Para esta misma red, los lugares en verde indican los estados de operación para la SA 1, los lugares rojos muestran los lugares de operación para SA 2 y los lugares en azul representan las máquinas o recursos compartidos del sistema.

3.8 Matriz de Incidencia

La matriz de incidencia de una RdP es la expresión matemática que contiene la información del esquema de la red, relacionando en una sola matriz las entradas y salidas de una RdP. Resulta de la resta de la matriz de incidencia de salida con la matriz de incidencia de entrada.

$$A = A^+ - A^- \quad (3.8)$$

La matriz de incidencia es de dimensiones $n \times m$, donde n es el número de lugares y m es el número de transiciones.

$$[A] = \begin{bmatrix} P_1, T_1 & P_1, T_2 & P_1, T_3 & \dots & P_1, T_m \\ P_2, T_1 & P_2, T_2 & P_2, T_3 & \dots & P_2, T_m \\ P_3, T_1 & P_3, T_2 & P_3, T_3 & \dots & P_3, T_m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_n, T_1 & P_n, T_2 & P_n, T_3 & \dots & P_n, T_m \end{bmatrix} \quad (3.9)$$

La matriz de incidencia de entrada o matriz negativa es una matriz binaria que representa los lugares de entrada de cada transición mediante unos (1).

$$A^- = [A_{ij}^-] \quad \text{donde } A_{ij}^- = \text{Pre}(P_i, T_j) \quad (3.10)$$

De este modo, $[P_1, T_1] = 1$ si P_1 es un lugar de entrada de T_1

$[P_1, T_1] = 0$ si P_1 es no es un lugar de entrada de T_1

Para la RdP expuesta en la figura 3.3, la matriz de incidencia de entrada es:

$$A^- = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.11)$$

La matriz de incidencia de salida o matriz positiva, es una matriz binaria que representa los lugares de salida de cada transición de la RdP mediante unos (1).

$$A^+ = [A_{ij}^+] \quad \text{donde } A_{ij}^+ = \text{Post}(P_i, T_j) \quad (3.12)$$

La matriz de incidencia de salida para la RdP de la figura 3.3 es:

$$A^+ = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix} \quad (3.13)$$

Según la ecuación 3.8 se tiene que la matriz de incidencia para la red de la figura 3.3 se la siguiente:

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ 1 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 1 & -1 \end{bmatrix} \quad (3.14)$$

El término A_{12} indica la relación existente entre P_1 y T_2 . Aparentemente P_1 no es lugar de entrada ni de salida para T_2 , sin embargo, al observar la figura 3.3 se puede determinar que este lugar es tanto de entrada como de salida, con lo cual se llega a la conclusión de que en ocasiones no es suficiente la matriz de incidencia global para determinar la estructura de una RdP, serán necesarias las matrices de entrada y de salida que soportarán la estructura total de la red.

3.9 Ecuación Fundamental de las RdP

La ecuación fundamental de las RdP permite representar el estado de la red en cualquier instante de tiempo k , tanto dinámica como estáticamente.

Mediante la expresión 3.15 se obtiene el marcado siguiente al instante k .

$$M_{k+1} = M_k + A \cdot u_k \quad k = 0,1,2 \dots \quad (3.15)$$

- Donde
- M_k es el vector columna $[n \times 1]$ de marcado para el instante, indica el número de tokens ubicados en los n lugares de la red.
 - M_{k+1} es el vector columna $[n \times 1]$ de marcado siguiente, representa el estado de la red después de haber disparado la transición denotada por el vector de control.
 - A es la matriz de incidencia de la red $[n \times m]$
 - u_k es el vector columna $[m \times 1]$ de control, indica que transición ha sido disparada en el instante k
 - $u_{j,1} = 1$ si la transición j ha sido disparada
 - $u_{j,1} = 0$ en caso contrario

Dinámicamente el comportamiento de la red se determina mediante la siguiente ecuación:

$$M_x = M_k + A \cdot \sum_{k=0}^N u_k \quad (3.16)$$

En donde es posible hallar un marcado x, teniendo en cuenta la sumatorias de los vectores de control estipulados para una secuencia de disparos.

La ecuación 3.17 representa el marcado final que puede alcanzar una RdP posterior a una secuencia de disparos aplicada. La diferencia con la expresión 3.16 radica en que esta relación es estática, debido a que no se encuentra explícita la evolución puntual de cada uno de los estados sino la evolución global de los mismos.

$$M_f = M_o + A \cdot S \quad (3.17)$$

Donde

M _f	es el marcado final
M _o	es el marcado inicial
S	secuencia de disparo, equivalente a la sumatoria del vector de control u_k

3.10 Ecuaciones de Estado para Tiempo Remanente en RdP

La ecuación de estado que se muestra a continuación es una representación del tiempo remanente en un sistema modelado mediante RdP, evaluando dinámicamente la evolución de cada transición en la red y calculando paso a paso el tiempo restante para la culminación de cada uno de los procesos [16].

$$X_r(k+1) = A(k)X_r(k) + B(k)u(k) \quad (3.18)$$

- $X(k)$ es el vector de estado.

$$X_r(k) = \begin{bmatrix} M(k) \\ M_r(k) \end{bmatrix} \quad (3.19)$$

$M(k)$ representa el vector de marcado, que indica el estado de la red en el instante anterior. $M_r(k)$ es el vector de tiempo remanente, que contiene en su i -ésima posición el tiempo remanente o restante para la culminación del proceso según la transición de salida del i -ésimo lugar. Tanto el vector de marcado como el de tiempo remanente, en el vector de estado $X(k)$, están considerados justo antes del disparo de la k -ésima transición.

- $A(k)$ es la matriz del sistema y es representada según la expresión 3.20.

$$A(k) = \begin{bmatrix} I_n & O_n \\ -\delta(k)P & I_n \end{bmatrix} \quad (3.20)$$

En donde I_n es la matriz identidad de dimensiones $[n \times n]$, O_n es la matriz de ceros de dimensiones $[n \times n]$. P es una matriz diagonal de dimensiones $[n \times n]$, que diferencia los lugares de operación del resto de lugares modelados en la RdP, matemáticamente P puede definirse como el conjunto definido por la expresión 3.21. $\delta(k)$ es el tiempo transcurrido entre dos disparos de transición consecutivas, también puede definirse como el tiempo de operación para el lugar analizado en el instante k , o como el valor de tiempo de la k -ésima posición del vector de tiempo remanente.

$$P = \{p_{ij}\}, \quad i = 1 \dots n, \quad j = 1 \dots m \quad (3.21)$$

Donde $p_{ij} = 1$ cuando $i = j \wedge i$ es un lugar de operación
 $p_{ij} = 0$ en cualquier otro caso.

- $B(k)$ se denomina la matriz de distribución, la cual transforma la acción de control $u(k)$, adicionando o removiendo los tokens cuando se dispara una transición representada en el vector $u(k)$.

$$B(k) = \begin{bmatrix} A \\ TA^+ \end{bmatrix} \quad (3.22)$$

Para la expresión 3.22 A representa la matriz de incidencia y A^+ representan la matriz de incidencia de entrada. T simboliza la matriz diagonal de tiempos de proceso para los lugares de operación, es de dimensiones $[n \times n]$ y matemáticamente se evalúa como el conjunto expuesto en la expresión 3.23.

$$T = \{t_{ij}\} \quad (3.23)$$

Donde $t_{ij} = \tau_i$, cuando $i = j$ y cuando t_{ij} sea un lugar de operación
 $t_{ij} = 0$ en cualquier otro caso.

τ_i es la i -ésima posición del vector de tiempo de espera (time delay) asociados con los lugares de operación, o visto desde otra perspectiva se dice que τ_i es el tiempo que tarda en llevarse a cabo la operación asociada al lugar i dentro de un proceso simulado mediante una RdP.

- $u(k)$ es el vector de control de dimensiones $[m \times 1]$, determina que transición es disparada después de k disparos. $u_j(k)$ es la j -ésima posición de u en el tiempo k .

$u_j(k) = 1$ si la transición j es disparada

$u_j(k) = 0$ si la transición j no es disparada

Una vez planteadas las expresiones pertinentes al vector de estado, la matriz del sistema, la matriz de distribución y el vector de estado, se presenta al lector la ecuación de estado tal como se enuncia en la expresión 3.24.

$$X_r(k+1) = \begin{bmatrix} M(k+1) \\ M_r(k+1) \end{bmatrix} = \begin{bmatrix} I_n & O_n \\ -\delta(k) \cdot P & I_n \end{bmatrix} \cdot \begin{bmatrix} M(k) \\ M_r(k) \end{bmatrix} + \begin{bmatrix} A \\ T \cdot A^+ \end{bmatrix} \cdot u(k) \quad (3.24)$$

Esta ecuación dinámica permite calcular el tiempo restante para la finalización del proceso o del sistema modelado en cada instante de tiempo k , resultando apropiada para aplicaciones de optimización mediante técnicas heurísticas.

3.11 Ecuaciones de Estado para Tiempo Acumulado en RdP

La ecuación de estado para el tiempo acumulado calcula para cada secuencia de disparo o vector de control el tiempo total utilizado por cada proceso dentro de un sistema que se ha modelado mediante RdP.

$$X_T(k+1) = X_T(k) + C(k) \quad (3.25)$$

$$Y(k+1) = D \cdot X_T(k+1) \quad (3.26)$$

La expresión 3.25 evalúa dinámicamente la evolución de la red, acumulando en el vector de estado el tiempo de proceso por operación. Para hallar el tiempo acumulado total será necesario realizar una combinación lineal de las entradas, tal como se plantea con la ecuación 3.26, en donde se selecciona los tiempos requeridos para la optimización del sistema.

- $X_T(k)$ es el vector de estado

$$X_T(k) = \begin{bmatrix} M(k) \\ M_T(k) \end{bmatrix} \quad (3.27)$$

Al igual que en la ecuación de estado para tiempo de procesos remanentes, $M(k)$ representa el vector de marcado de la red, que contiene internamente la evolución de la red. El vector $M_T(k)$ representa el tiempo de proceso acumulado, este vector almacena la información de tiempo estipulado para cada operación según sea la evolución de la red.

- $C(k)$ es la matriz dinámica de tiempos, evalúa para el tiempo $k+1$ la evolución de la red y el tiempo utilizado en el proceso para los estados habilitados en el tiempo k .

$$C(k) = \begin{bmatrix} A \cdot u(k) \\ T \cdot M(k) \end{bmatrix} \quad (3.28)$$

Los términos utilizados para la formulación de la expresión 3.28, se encuentran definidos en la sección 3.10 donde se habla del tiempo de proceso remanente, en donde se enunció que A es la matriz de incidencia del sistema modelado, $u(k)$ es el vector de control, T es la matriz de tiempos de operación y $M(k)$ es el vector de marcado para el instante k .

El modelo matemático planteado para el vector de tiempos de procesos acumulados según una secuencia o vector de control, se precisa mediante la siguiente 3.29 y matricialmente, dicho modelo o ecuación dinámica se plantea según la expresión 3.30.

$$M_T(k+1) = M_r(k) + T \cdot M(k) \quad (3.29)$$

$$X_T(k+1) = \begin{bmatrix} M(k+1) \\ M_T(k+1) \end{bmatrix} = \begin{bmatrix} M(k) \\ M_T(k) \end{bmatrix} + \begin{bmatrix} A \cdot u(k) \\ T \cdot M(k) \end{bmatrix} \quad (3.30)$$

La diferencia entre esta ecuación y la planteada en la sección 3.10 radica principalmente en que la ecuación de tiempos acumulados se inicializa en ceros, agregando los tiempos de los lugares de operación dentro del vector de tiempo acumulado, mientras que la ecuación de tiempo remanente toma como base un valor de tiempo, que se reducirá hasta llegar a cero o un valor negativo que indique la finalización de los procesos.

Este modelo fue planteado con el objeto de generar un algoritmo basado en técnicas heurísticas o de optimización, como algoritmos genéticos, colonia de hormigas, entre otros, que sea capaz de hallar la secuencia óptima dentro de cualquier proceso simulado mediante RdP.

3.12 Ejemplos

Las RdP tienen aplicaciones en diferentes áreas, como lo son modelado de redes abstractas, procesamiento paralelo y distribuido, teorema de grafos, problemas de transporte, problemas de decisión, reconocimiento de patrones, entre otros.

Ejemplo 3.1:

Realizar el diseño de una RdP que controle proceso contemplado en el ejemplo 2.1.

Solución:

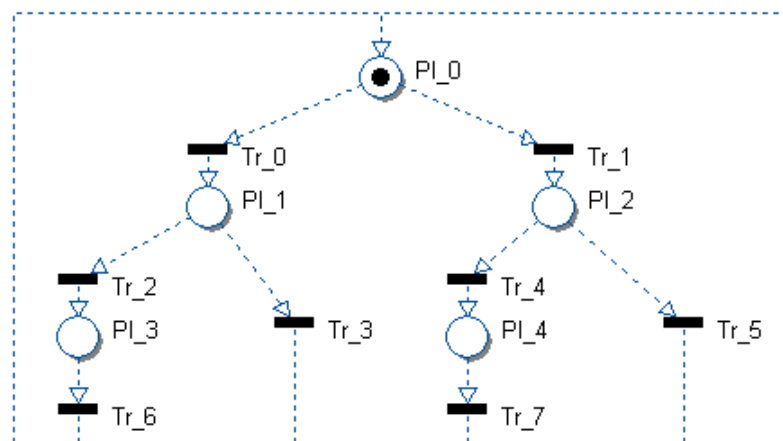
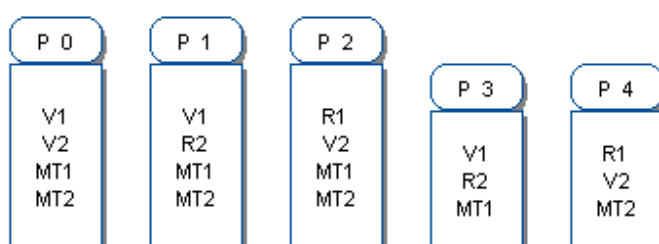


Fig. 3.13. Modelo de RdP para el ejemplo 3.1



$$T0 = ET1$$

$$T1 = ET2. \overline{ET1}$$

Leonardo E. Solaque G.

$$T2 = ET2$$

$$T3 = ST1$$

$$T4 = ET1$$

$$T5 = ST2$$

$$T6 = ST1$$

$$T7 = ST2$$

Ejemplo 3.2:

Hallar la matriz de incidencia de entrada, de salida y la total de la RdP resultante del ejemplo 3.1.

Solución:

Para solucionar este ejemplo, el lector debe remitirse a la sección 3.8 en donde se expone la teoría concerniente a la matriz de incidencia de una RdP.

$$A^- = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad A^+ = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Una vez hallada la matriz de incidencia de entrada y la de salida, se obtiene la matriz de incidencia total como la resta entre ellas.

$$A = A^+ - A^-$$

$$A = \begin{bmatrix} -1 & -1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & -1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}$$

Ejemplo 3.3: Pasteurización leche

Elaborar una RdP que modele el proceso enunciado en el ejemplo 2.2 concerniente al proceso de pasteurización de la leche. Una vez modelada la RdP, se deben encontrar las ecuaciones características que describen la red.

Solución:

El modelo de la RdP implementado para el proceso de pasteurización de la leche se encuentra en la figura 3.14. Dentro del diagrama se encuentra la explicación de cada uno de las operaciones modeladas. Para este modelado no se tuvieron en cuenta ni los lugares de inicialización ni los de finalización, se trabajó directamente con los lugares de operación, para lo cual se asumió que las transiciones de la RdP simularán el inicio o fin de dichas operaciones.

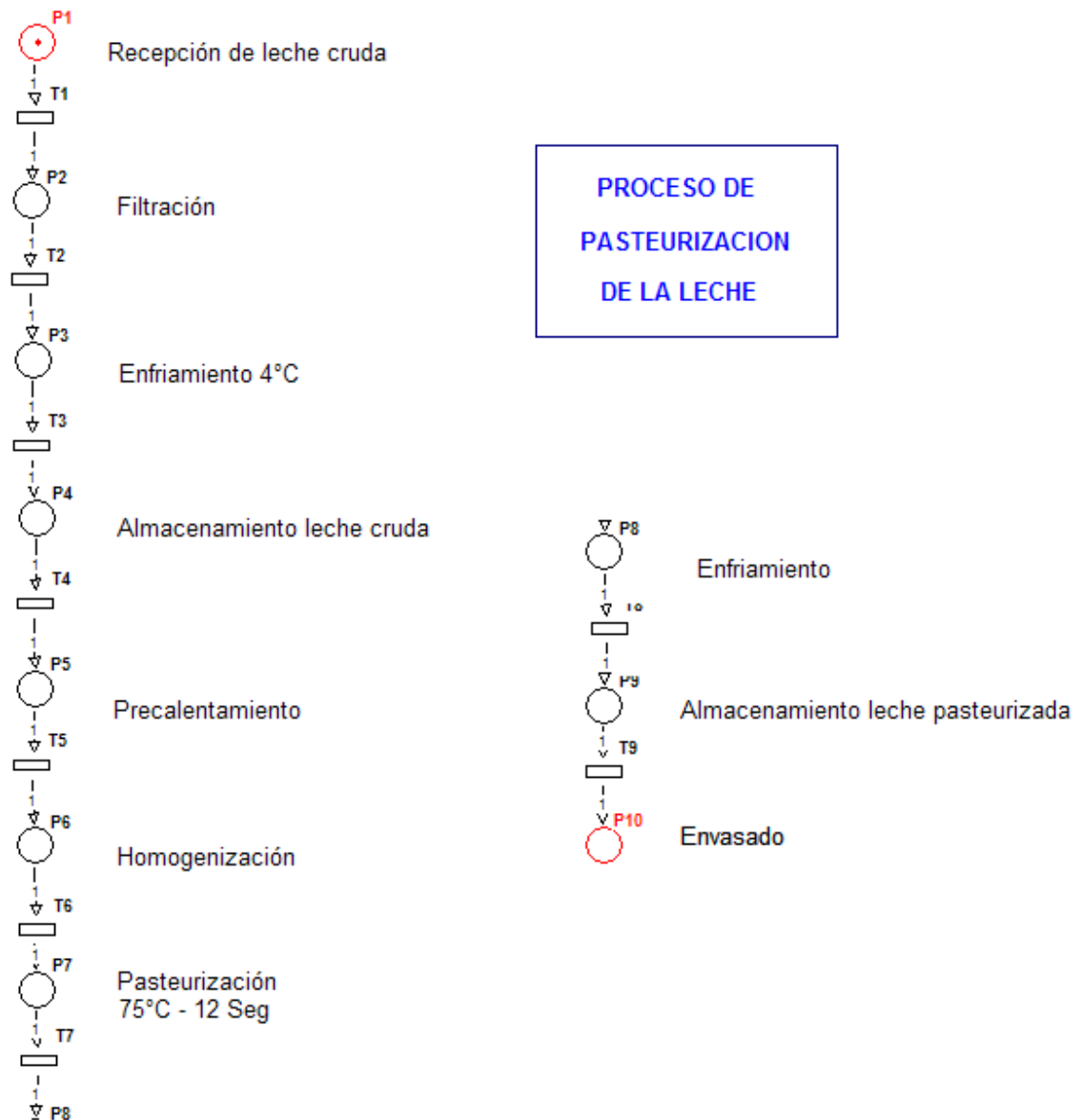


Fig. 3.14. RdP para el proceso de pasteurización de la leche

Ejemplo 3.4:

Establecer una RdP que modele el proceso de embotellamiento de agua descrito en el ejemplo 2.3. Una vez estructurada la red hallar su matriz de incidencia global.

Solución:

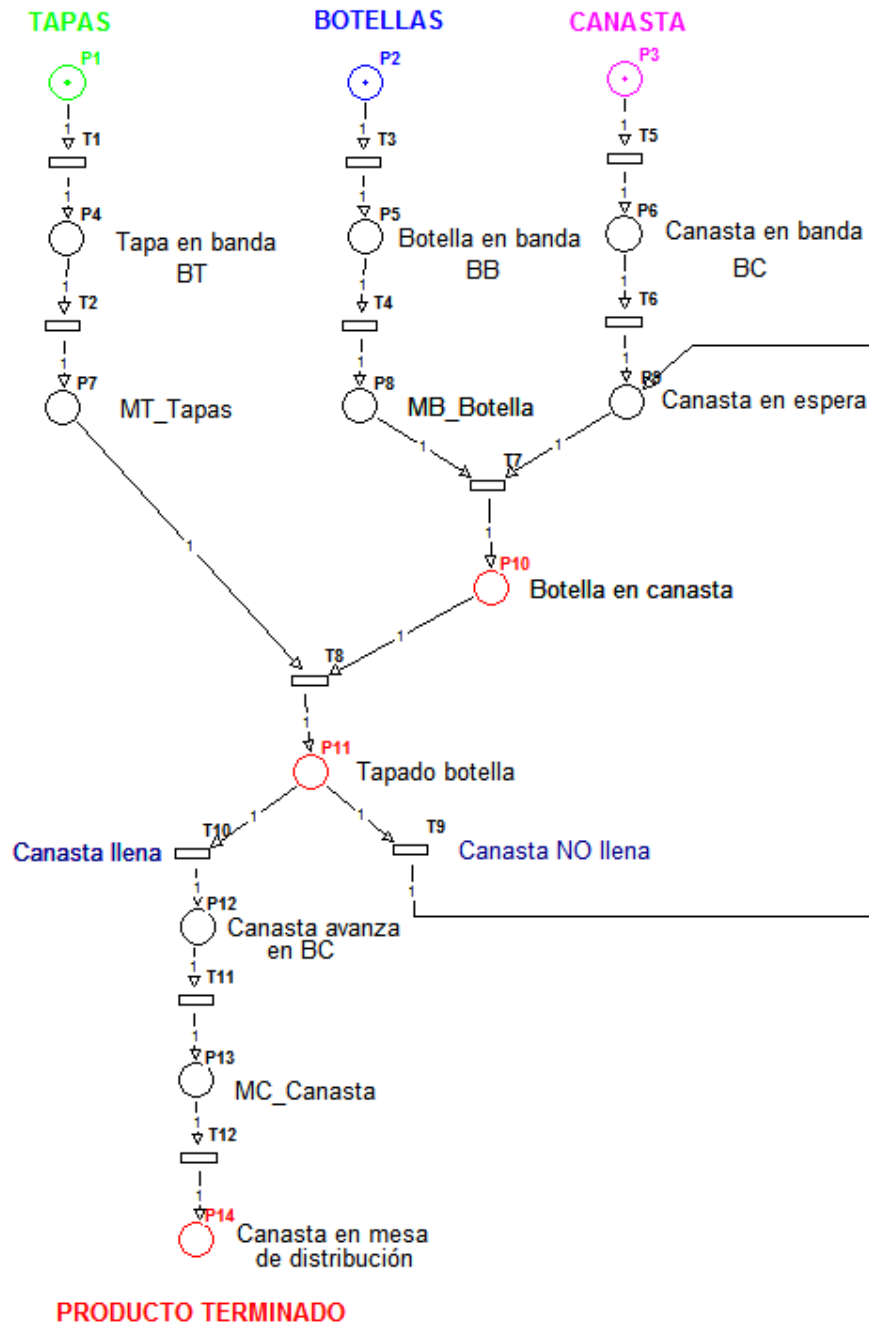


Fig. 3.15. RdP para el proceso de embotellamiento de agua

Ejemplo 3.5:

Diseñar una RdP que satisfaga las condiciones del ejemplo 2.4 enunciado como proceso de clasificado y empaquetado. Hallar la matriz de incidencia de la red diseñada.

Solución:

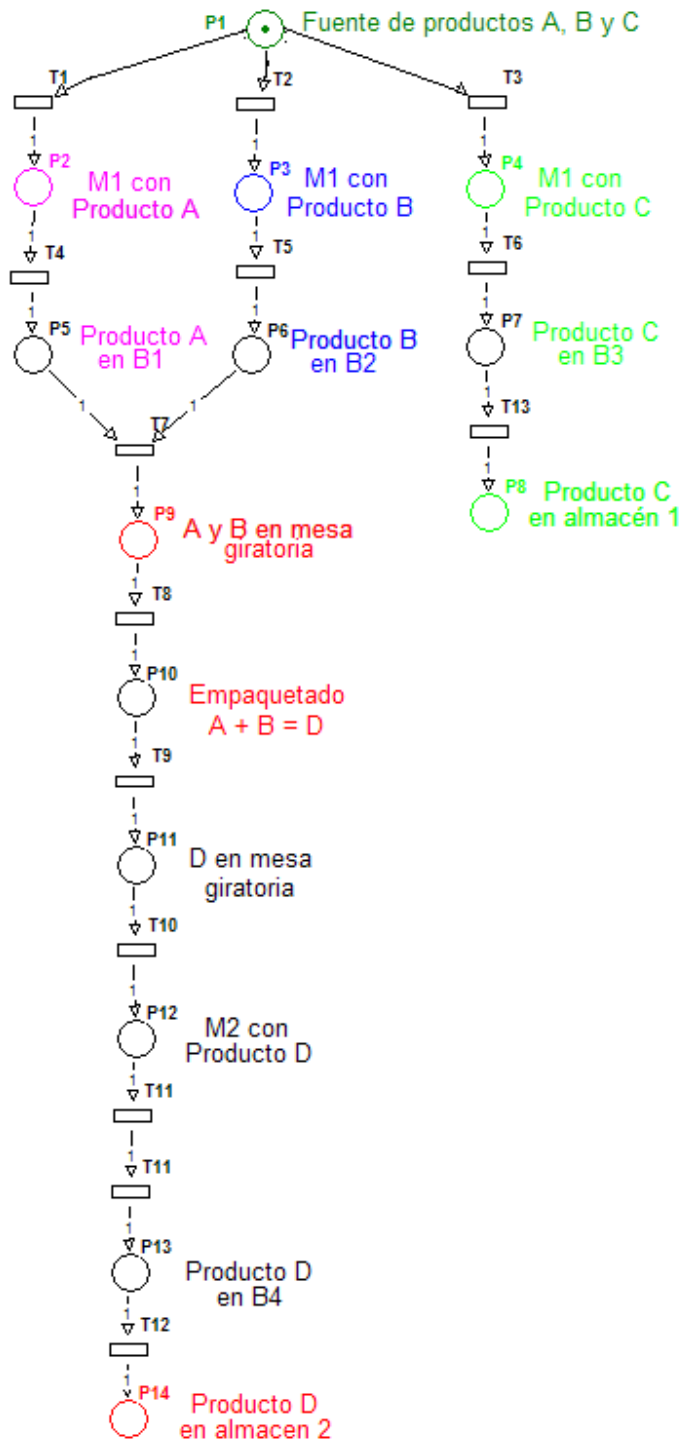


Fig. 3.16. RdP para el proceso de clasificado y empaquetado de productos

3.13 Conclusiones

Las RdP son una potente herramienta de modelado de los Sistemas a Eventos Discretos, mediante la cual es posible obtener un esquema de funcionamiento del proceso, así como también una representación matemática del mismo. Teniendo esta estructura y modelo matemático se llega a un análisis más profundo de acuerdo a las ecuaciones que rigen a las RdP.

Este capítulo contiene la teoría básica que encierra el estudio de las RdP, desde su definición hasta las ecuaciones que modelan su comportamiento. Se mostró algunas ecuaciones que permiten, mediante técnicas heurísticas o de optimización, hallar el menor tiempo de operación dentro de un proceso, teniendo en cuenta el comportamiento total del sistema.

Se planteó un modelo matemático que calcula el tiempo de operaciones en procesos concurrentes con recursos compartidos, cuyo objetivo es encontrar una secuencia que optimice los tiempos de producción del sistema modelado.

Adicionalmente, se plantearon algunos ejemplos basados en los problemas propuestos en el capítulo anterior.

CAPITULO 4: GRAFCET (G)

En este capítulo se presentarán los conceptos básicos necesarios para el diseño mediante la técnica de GRAFCET. Para ello se definirá concretamente lo que es el GRAFCET, se presentará el manejo de las posibles estructuras de configuración del G, así como también su evolución, se obtendrán funciones lógicas a partir del Grafo y se estudiara el comportamiento de los temporizados tanto para etapas como para transiciones dentro del funcionamiento del G.

También se incluirá en el desarrollo del capítulo la normatividad GEMMA, diseñada especialmente para G y para el diseño óptimo de automatismos. Finalmente se plantearán algunos ejemplos y aplicaciones enfocados en la implementación de la técnica del G.

4.1 Definición y Contexto Histórico

La sigla GRAFCET significa GRAfico Funcional de Control de Etapas y Transiciones. Nace de una serie de estudios realizados por una comisión creada por la AFCET⁴, en Francia, conformada por universidades, fabricantes y usuarios, con el objetivo de unificar el lenguaje de los sistemas lógicos y particularmente el funcionamiento secuencial de los mismos; el resultado obtenido es un lenguaje gráfico, que se apoya en métodos algebra lógica estandarizando cada una de las necesidades de los usuarios finales.

El desarrollo del GRAFCET inicia sobre la década de los setentas. En ese entonces, la necesidad era establecer un método que describiera procesos mediante un gráfico funcional, que pudiese ser interpretado por usuarios no especializados en automatización.

GRAFCET es un método gráfico de modelación y diseño de automatismos, que se fundamenta en la teoría de Redes de Petri. Dentro de las ventajas que introduce este método están la metodología y la estructura que lo representa, ya que con esto es posible obtener gran claridad, legibilidad y síntesis a la hora de controlar procesos o sistemas a eventos discretos.

Un GRAFCET es una Red de Petri viva, binaria y por ende conforme; por tal motivo, los elementos que componen al GRAFCET se aproximan en gran medida, por no decir que son los mismos, a los de una RdP.

⁴ AFCET (Association Française pour la Cybernétique Économique et Technique)

4.2 Estructura y Reglas de Evolución del GRAFCET

Al igual que las Redes de Petri, GRAFCET está conformado por nodos que tienen etapas y transiciones. Una etapa se define como un estado o situación del sistema a la que se le asocian acciones determinadas. Existen etapas activas, inactivas y de inicialización.

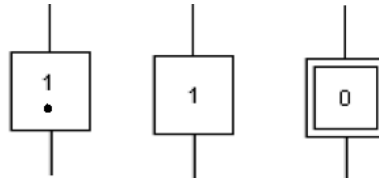


Fig. 4.1. Etapas representativas del GRAFCET

Cada estado tiene una acción asociada que representa un instante de tiempo del proceso modelado. Dichas acciones pueden ser de nivel o de impulso; si son de nivel, se dejará de realizar la acción cuando la etapa se desactive, mientras que si es de impulso, la acción durará el tiempo que tenga programado internamente, sin importar si la etapa esta o no activa.

Las acciones se pueden clasificar por un lado en reales o virtuales, o por otro lado en condicionales e incondicionales. Las reales son aquellas que tiene el sistema a controlar, las virtuales son aquellas que el programador asigna en un momento determinado; las condicionales son, como su nombre lo indica, aquellas que llevan consigo una condición de funcionamiento, de igual manera, las acciones incondicionales son aquellas que se activan sin ningún tipo de requerimiento.

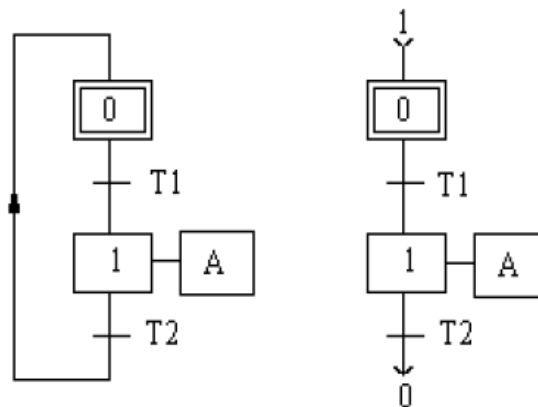


Fig. 4.2. Representación de un GRAFCET con etapas, transiciones y acciones

Luego de tener claro el concepto de estado o etapa, se pasará a definir una transición. Una transición es una barrera que se encuentra entre dos estado, permite la evolución del sistema a medida que las condiciones asociadas a estas se cumplan, activando, o dicho de otro modo, disparando dichas transiciones. Ahora bien, un arco es una línea

recta que une a una etapa con una transición o a una transición con una etapa. Un GRAFCET evoluciona según la activación y desactivación de etapas y transiciones.

Todo GRAFCET debe tener una etapa o un estado inicial, de esta manera, a medida que se activen las transiciones, las acciones se van ejecutando y el sistema va evolucionando. Ocurre el disparo de una transición en el momento en que la entrada asociada a dicha transición se active. Ahora bien, cuando una transición se dispara, la etapa anterior se desactiva y la etapa posterior pasará a estar activa.

Se debe tener en cuenta que no es posible disparar dos transiciones exactamente al mismo tiempo, ya que el procesador realiza las operaciones una por una, así que siempre se ejecutará una orden primero que la otra, aunque la diferencia sea por un lapso muy corto de tiempo; motivo por el cual, debe evitarse este tipo de situaciones.

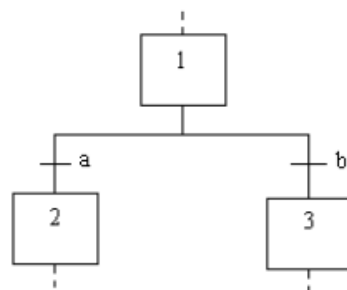


Fig. 4.3. Situación de conflicto

En la figura – se ve el conflicto generado. Para evitar esta situación, se debe condicionar una de las transiciones, de tal manera que, el sistema pueda tomar por un único camino en el momento en que las dos transiciones llegasen a activarse simultáneamente, tal como se ve en la siguiente figura.

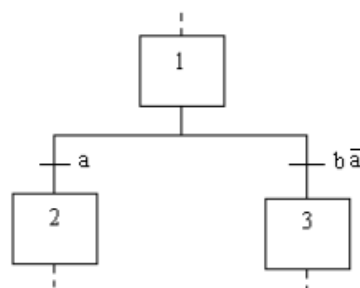


Fig. 4.4. Prioridad en transiciones

De este modo, lo que se hace es establecer una prioridad para una transición determinada, según sea conveniente. Para el caso de la figura 3.4, la prioridad la tiene 'a' sobre 'b'.

4.3 Macro Etapas

En ocasiones, una secuencia se utiliza en diferentes partes de un mismo programa o GRAFCET; para evitar escribir o graficar varias veces la misma secuencia, es posible reunir este conjunto reiterativo de etapas y transiciones en una sola llamada macro etapa. La macro etapa se representa de la siguiente manera:

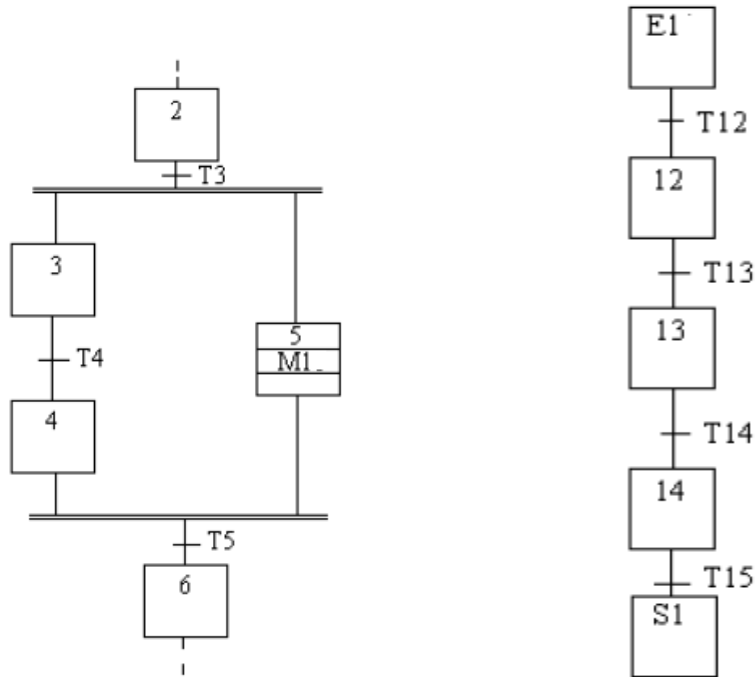


Fig. 4.5. Representación de una macro etapa con su respectiva secuencia

Ahora bien, en el momento de declarar o asignar la secuencia a la macro etapa, esta deberá iniciar con una entrada E y finalizar con una salida S; en el momento en que se active la macro etapa, se inicializará la etapa E y cuando finalice esta evolución en la etapa S, se regresará al funcionamiento del GRAFCET global, en donde se encuentre simbolizada la macro etapa.

4.4 Estructuras del GRAFCET y Obtención de Ecuaciones de Funciones Lógicas

Las estructuras más frecuentes del GRAFCET son las secuencias lineales, paralelas, divergencia y convergencia tanto en AND como en OR, los saltos y los bucles.

Secuencias Lineales y Paralelas:

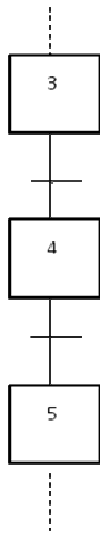


Fig. 4.6. Secuencia en Serie

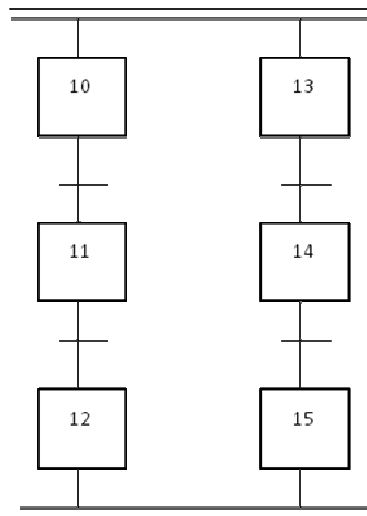


Fig. 4.7. Secuencia concurrente o en Paralelo

Salto y Bucle:

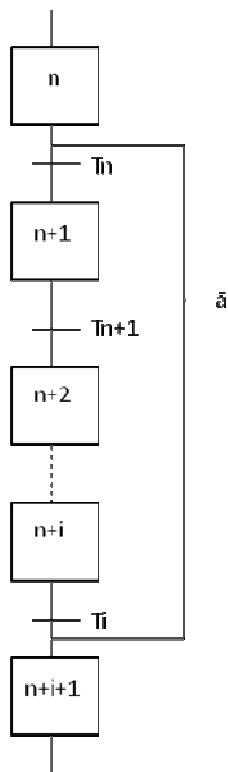


Fig. 4.8. Salto

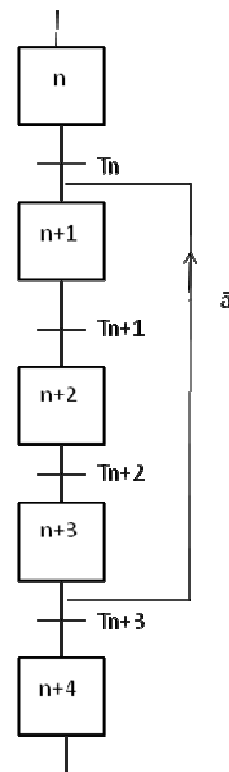


Fig. 4.9. Bucle

Obtención de Funciones Lógicas a partir de GRAFCET:

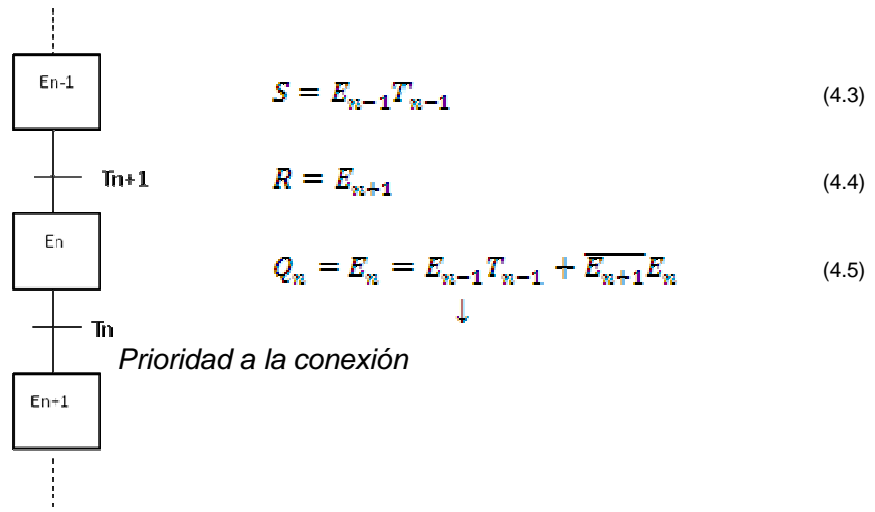
Es posible de pasar del GRAFCET a las ecuaciones lógicas representativas. Se estudiará la ecuación de activación y desactivación de etapas, está orientada con prioridad a la conexión o a la desconexión.

Prioridad a la conexión: $Q_{t+\Delta t} = S + \bar{R}Q_t$ (4.1)

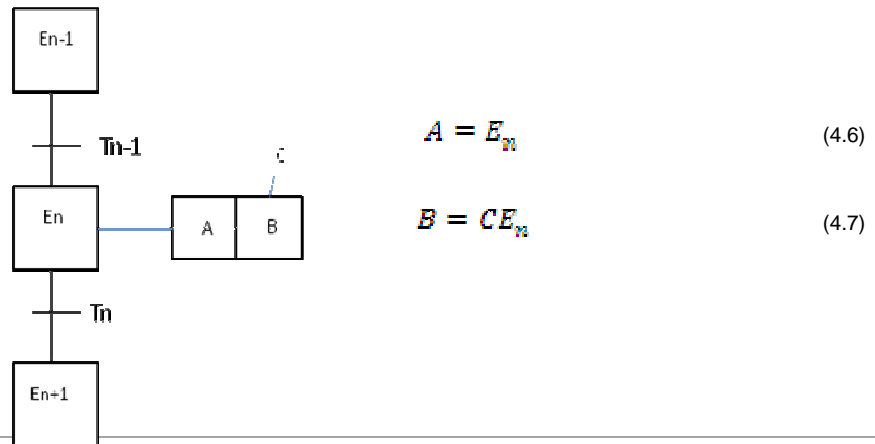
Prioridad a la desconexión: $Q_{t+\Delta t} = \bar{R}(S + Q_t)$ (4.2)

- Donde
- $Q \rightarrow$ Estado de la etapa
 - $S \rightarrow$ Condición de activación de la etapa
 - $R \rightarrow$ Condición de desactivación de la etapa

Tomando como base el siguiente GRAFCET, se puede definir S y R y E_n como se ve a continuación.



Las acciones asociadas a cada estado se pueden caracterizar de la siguiente forma:



Luego de tener los conceptos globales para la formulación de ecuaciones lógicas, se evaluarán dichas representaciones en las estructuras que posee GRAFCET, aplicando la concepción de prioridad a la conexión.

Divergencia y Convergencia en OR:

La situación ideal para la utilización de las estructuras OR es la escogencia una de las opciones del diseño del GRAFCET planteado, es decir, se va por un camino o por otro.

$$S = E_{n-1}T_{n-1}$$

$$R = E_{n+1} + E_{n+2}$$

$$E_n = E_{n-1}T_{n-1} + \overline{E_{n+1}}\overline{E_{n+2}}E_n \tag{4.8}$$

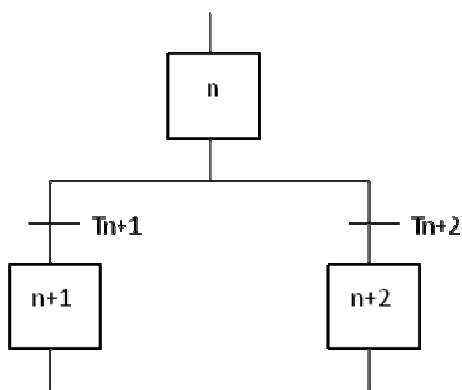


Fig. 4.10. Divergencia en OR

$$S = E_{n-1}T_{n-1} + E_{n-2}T_{n-2}$$

$$R = E_{n+1}$$

$$E_n = E_{n-1}T_{n-1} + E_{n-2}T_{n-2} + \overline{E_{n+1}}E_n \tag{4.9}$$

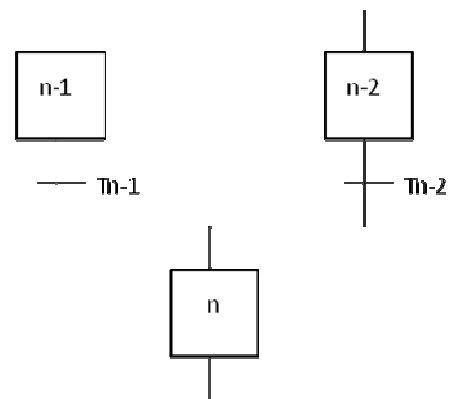


Fig. 4.11. Convergencia en OR

Divergencia y Convergencia en AND:

Las estructuras en AND se implementan cuando se requiera realizar procesos paralelos simultáneos.

$$S = E_{n-1}T_{n-1}$$

$$R = E_{n+1}E_{n+2}$$

$$E_n = E_{n-1}T_{n-1} + (\overline{E_{n+1}} + \overline{E_{n+2}})E_n$$

$$S = E_{n-1}E_{n-2}T_n$$

$$R = E_{n+1}$$

$$E_n = E_{n-1}E_{n-2}T_n + \overline{E_{n+1}}E_n$$

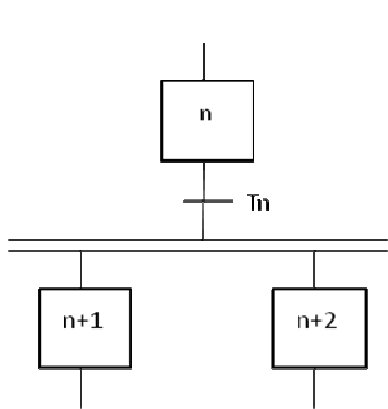


Fig. 4.12. Divergencia en AND

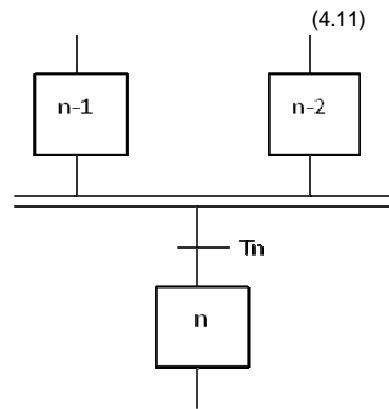
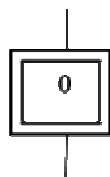


Fig. 4.13. Convergencia en AND

Etapas de Inicialización:

Las condiciones iniciales se representan mediante la siguiente ecuación; del mismo modo, un estado de inicialización se representa mediante el bloque que se presenta a continuación.



$$C_i = \sum_{v_n} E_n \tag{4.12}$$

Fig. 4.14. Estado Inicializado

Por lo que la ecuación total para el estado inicial es la siguiente:

$$E_n = S + \bar{R}E_n + C_i \tag{4.13}$$

4.5 Temporizaciones

En etapas

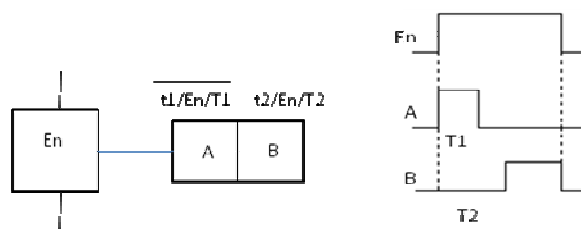


Fig. 4.15. Temporizado en etapas

En transiciones

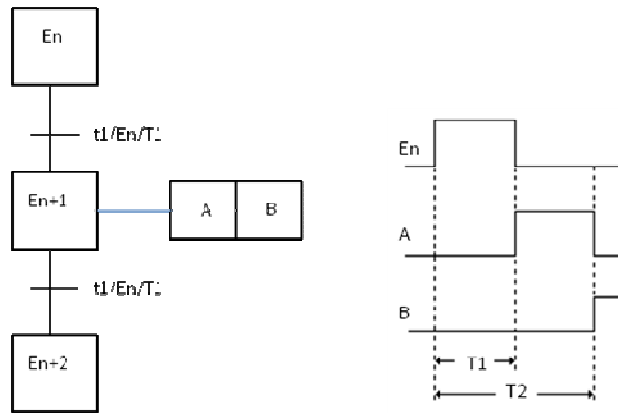


Fig. 4.16. Temporizado en transiciones

4.6 Norma GEMMA

GEMMA (Guide d'Étude des Modes de Marches et d'Arrêts) es un método que estudia las posibles situaciones de la puesta en marcha y parada de los sistemas controlados por GRAFCET [4]. Fue desarrollado en Francia por el ADEPA (Agence pour le Développement de la Productique Appliquée) fundamentando la necesidad de concebir un procedimiento que conciba especificaciones correctas y seguras, logrando que no se generen situaciones imprevistas que puedan llevar a fallo el sistema.

GEMMA se apoya en un gráfico que representa cada uno de los estados (rectángulos) estandarizados de la parte operativa de un proceso, mostrando la evolución de unos estados a otros. Ver figura 4.17.

Generalmente los procesos industriales automatizados no utilizan todos los estados planteados por la norma GEMMA, en estos casos se debe poner una cruz sobre el estado no implementado, anulando dichos estados.

Una vez estipulados los estados aplicables a un proceso, será necesario determinar las evoluciones posibles, para esto se utiliza normalmente las guías punteadas del gráfico GEMMA.

Es importante aludir que la norma GEMMA permite flexibilidad para su implementación y diseño dentro de cada proceso, por lo que es posible el planteamiento de nuevos estados o caminos de evolución según sea pertinente para cada caso en particular.

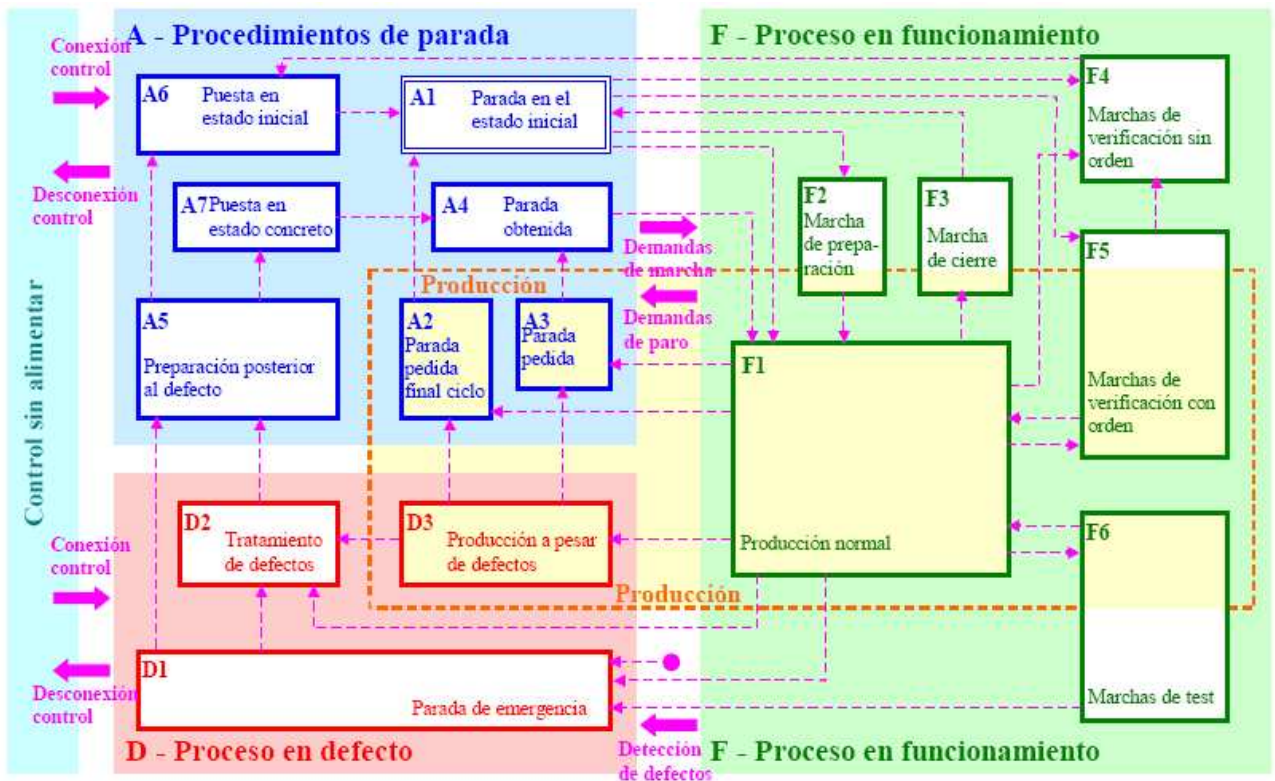


Fig. 4.17. Gráfico GEMMA con los estados normalizados

Existen diferentes casos específicos de la norma GEMMA que son utilizados frecuentemente:

- Marcha por ciclos y parada a fin de ciclo
- Marcha de verificación con orden
- Marcha de verificación sin orden
- Paradas de emergencia
- Parada en un punto

La norma GEMMA plantea una metodología para el diseño de automatismos [4], teniendo en cuenta las siguientes fases:

- Estudio de las acciones del proceso
Definición del ciclo normal de producción.
Formulación del GRAFCET de base.
- Definir la parte operativa (accionamientos y sensores).
GRAFCET operacional, con definición de la parte operativa.
- Definir modos de marcha y paro con ayuda del gráfico GEMMA.

- d) Definir, con apoyo del GEMMA, las condiciones de evolución entre los diferentes estados de funcionamiento, parada y fallo.
- e) Definición de la comunicación hombre-máquina y con otros controladores del proceso.
- f) Formulación del GRAFCET total.
- g) Escoger la tecnología de control (eléctrico, neumático, electrónico, programable o no, etc.)
- h) Concepción del esquema o programa de control.

Para profundizar en el manejo de la norma GEMMA se remite al lector a las referencias [17] [18].

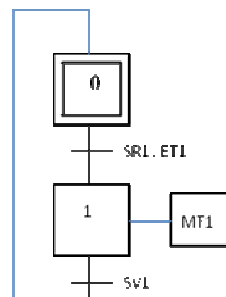
4.7 Ejemplos y Aplicaciones

Ejemplo 4.1:

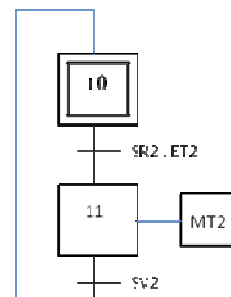
Diseñar el GRAFCET que controle el proceso estipulado en el ejemplo 1.

Solución:

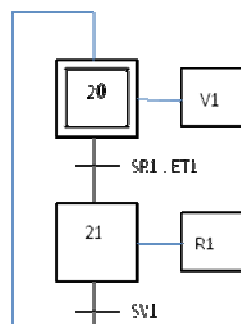
GT1: Grafcet Tren 1



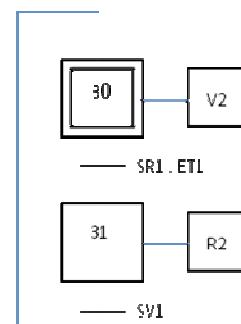
GT2: Grafcet Tren 2



GS1: Grafcet Semáforo 1



GS2: Grafcet Semáforo 2



Ejemplo 4.2:

Se pretende implementar el sistema del ejemplo 2.3, basado en el embotellamiento de agua. Para esto se utilizara el PLC de SIEMENS S7 – 300, además de los sensores y actuadores propuestos en la solución del ejemplo 2.3. El objetivo será entonces diseñar el GRAFCET de control que logre realizar la tarea de embotellamiento de agua.

Solución:

Se diseñaron GRAFCET independientes para cada una de las máquinas que intervienen en el proceso que se encuentran sincronizados entre sí. Para ver con mayor claridad el diseño de GRAFCET implementado, se mostrará en el siguiente diagrama la distribución empleada. Para el desarrollo de este enunciado, se utilizarán las variables estipuladas en la solución del ejemplo 2.3.

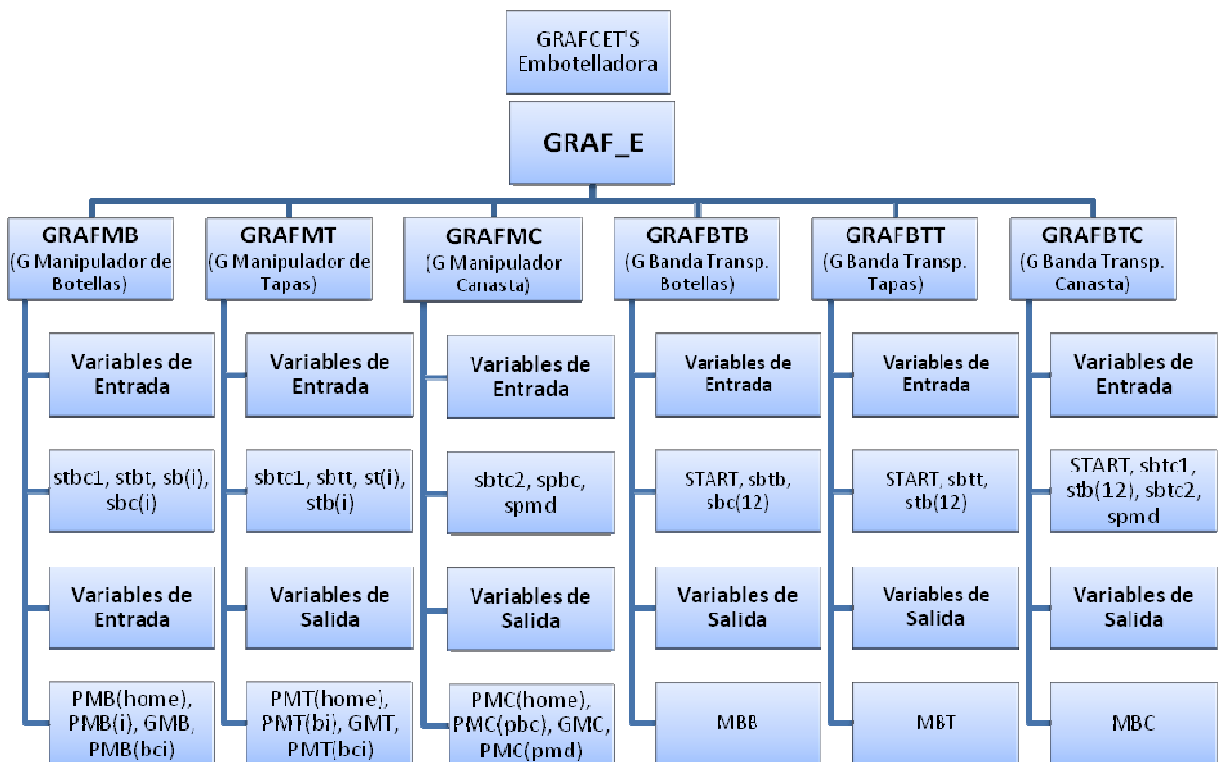
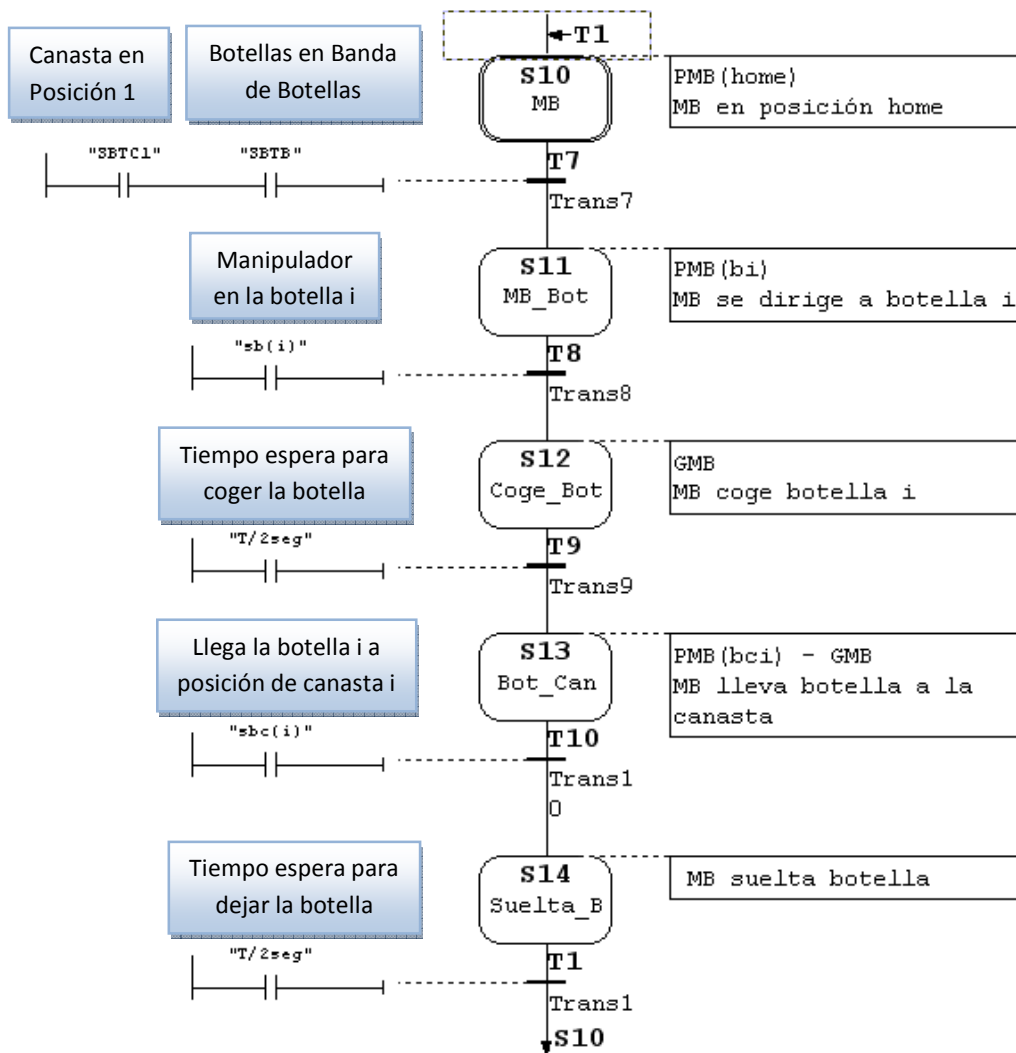


Fig. 4.18. Diagrama distribución GRAFCET Embotelladora

Se hace énfasis en que la solución de GRAFCET planteada al lector, es una solución generalizada para una cantidad de botellas 'i', por lo tanto, en el momento en que se quiera implementar, debe efectuarse un GRAFCET que incluya cada una de las posibilidades contempladas en la variable 'i'.

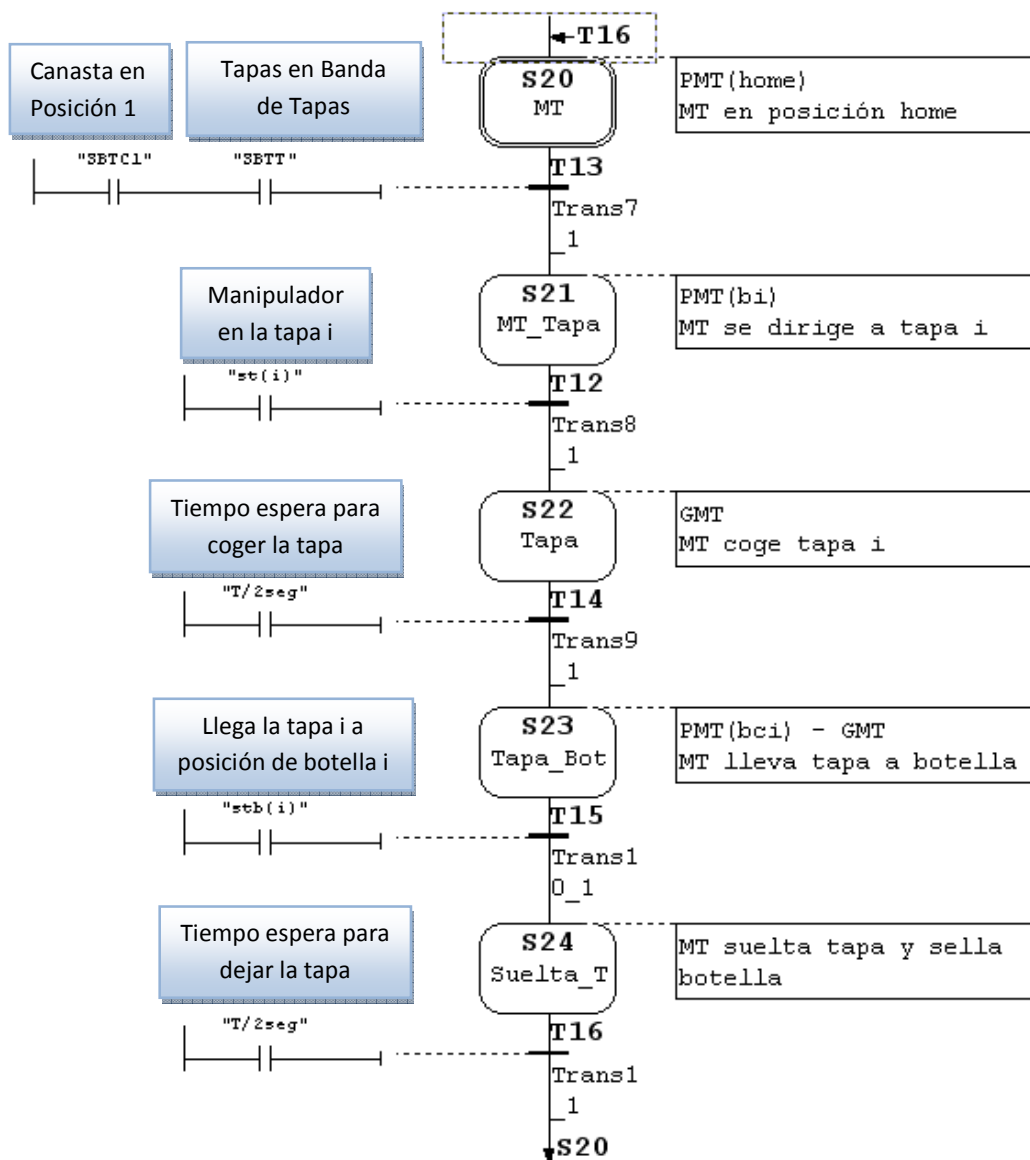
GRAFMB: GRAFCET del manipulador de botellas

Inicialmente, el manipulador de botellas estará en posición de home o reposo. En el momento en que se activan las señales que indican el posicionamiento de la canasta y de las botellas en las bandas transportadoras, el manipulador tomará las botellas acomodándolas una por una dentro de la canasta, de acuerdo a las posiciones que tiene programadas internamente.



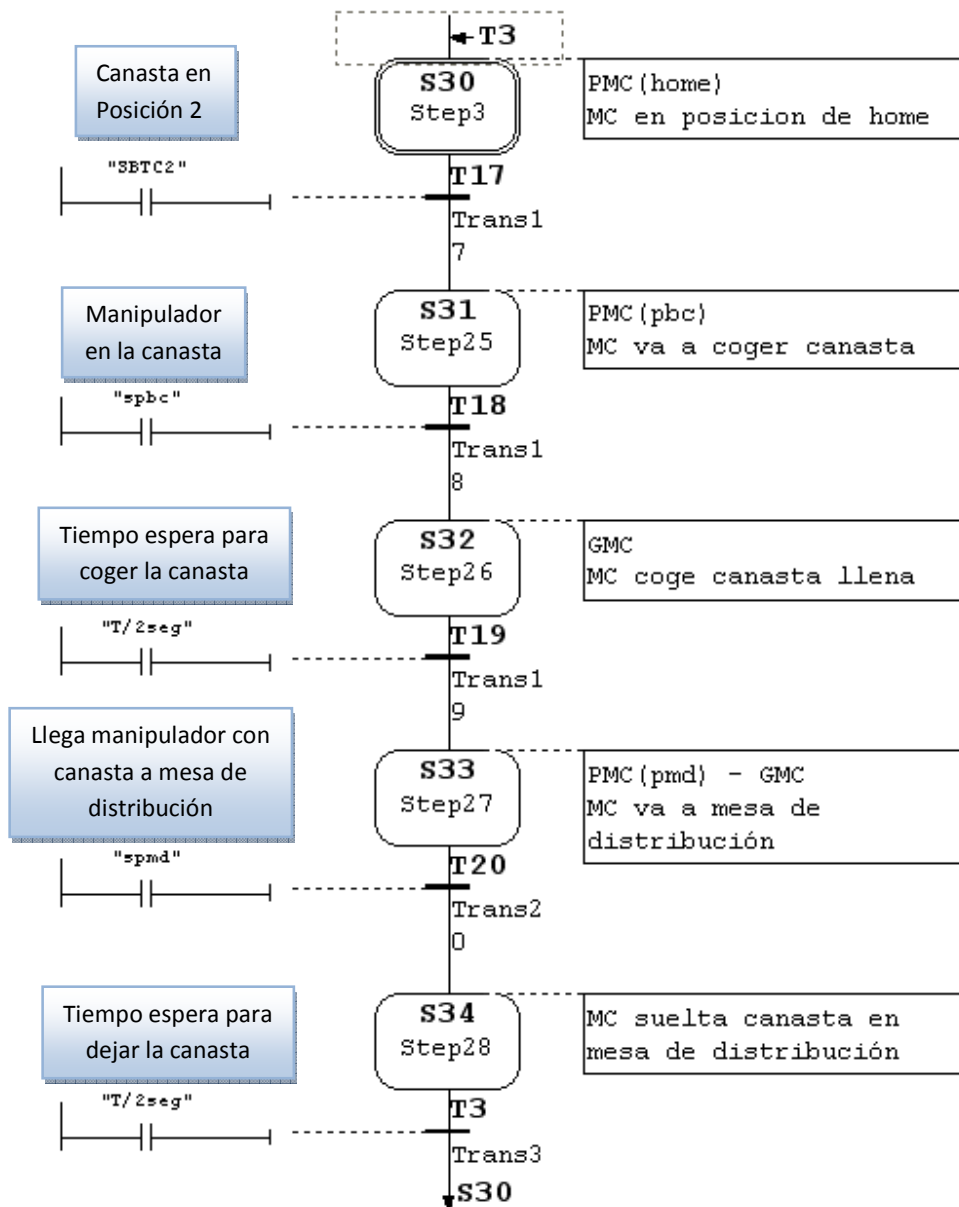
GRAFMT: GRAFCET del Manipulador de Tapas

El funcionamiento del GRAFCET manipulador de tapas es muy similar al del manipulador de botellas. En el momento en que se detecta la presencia de la canasta y de las tapas, el manipulador procederá a tapar cada una de las botellas depositadas en la canasta.



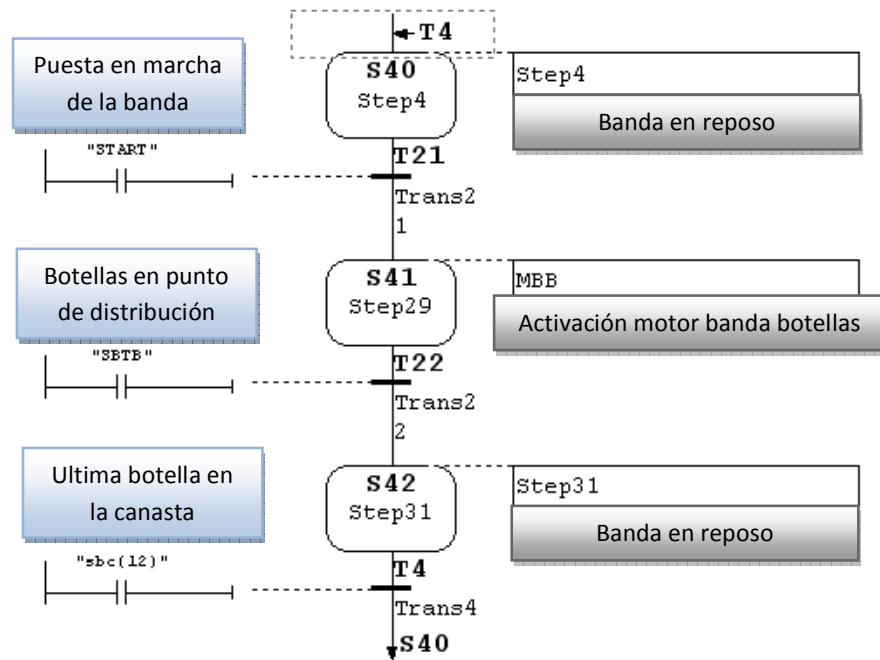
GRAFMC: GRAFCET del Manipulador de la Canasta

El GRAFCET que manipula la canasta interviene en el proceso en el momento en que la canasta esté completamente llena y con sus respectivas botellas selladas, este suceso se presenta cuando la canasta esté en la posición 2 de la banda transportadora de canastas. En ese momento, el manipulador tomará la canasta y la ubicará en la mesa de distribución, en donde posteriormente será llevada a la bodega de almacenamiento.

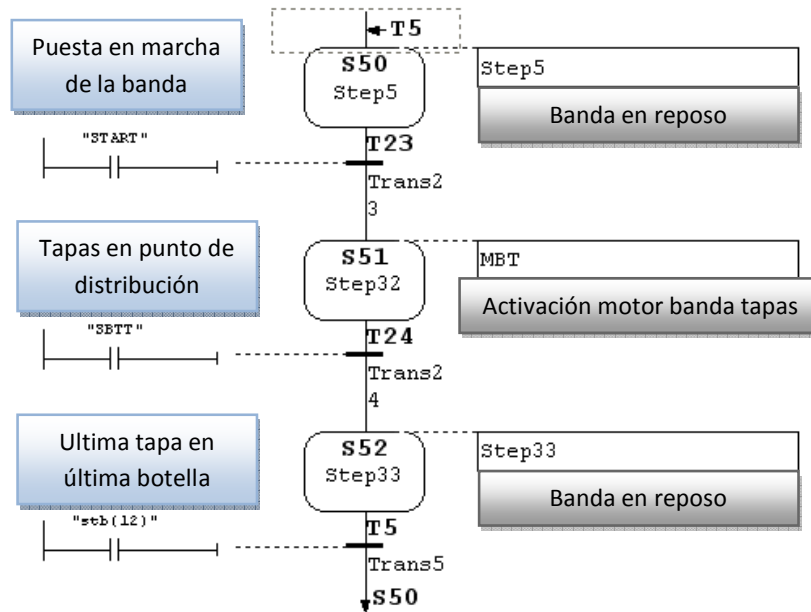


GRAFBTB: GRAFCET Banda Transportadora de Botellas

La banda transportadora de las botellas de agua estará inicialmente en reposo hasta que sea pulsado el START, en ese momento se accionará el motor de la banda hasta que se active la señal de llegada de las botellas al punto de distribución.

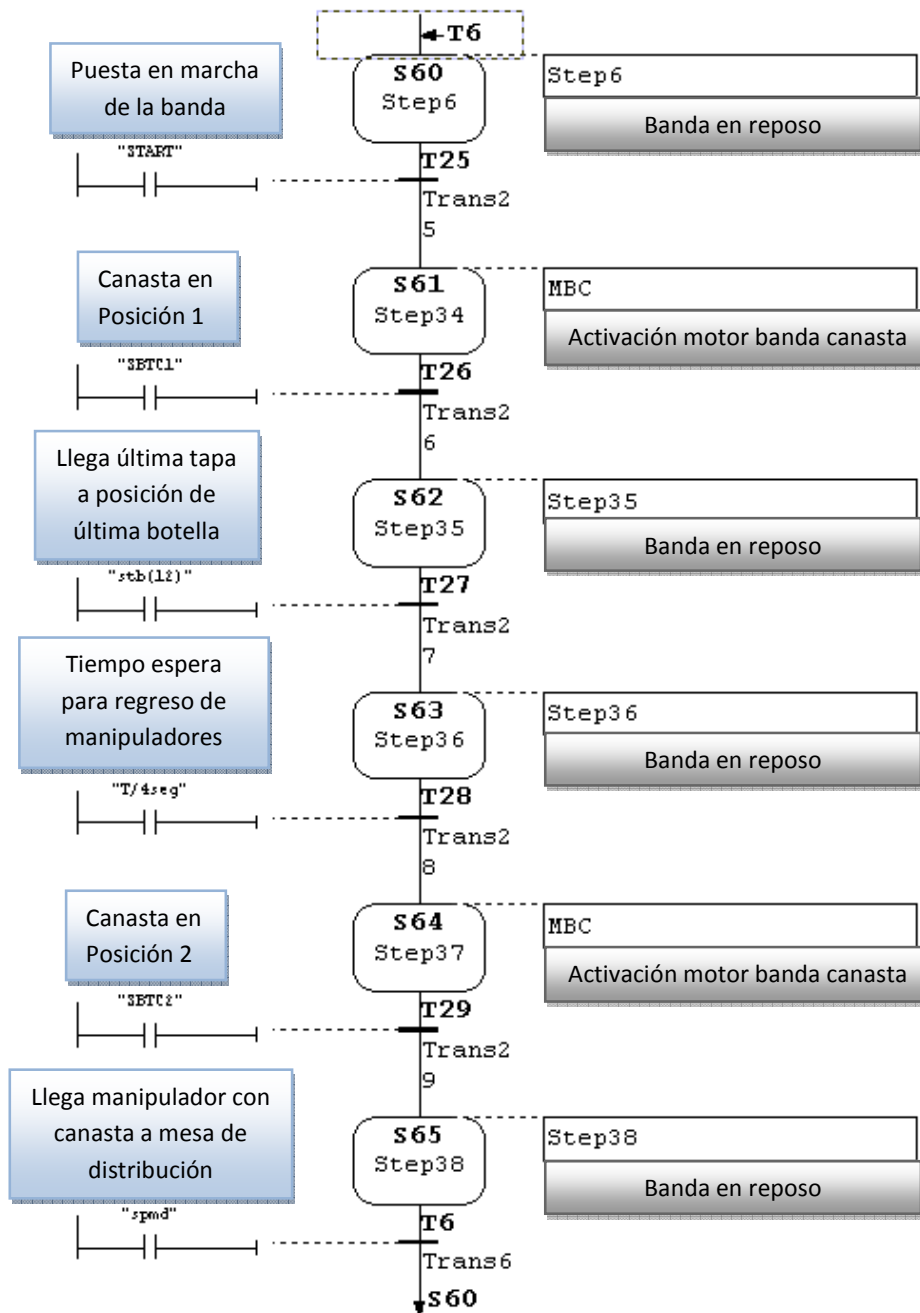


GRAFBTT: GRAFCET Banda Transportadora de Tapas



GRAFBTC: GRAFCET Banda Transportadora de Canasta

La banda transportadora de canastas se pondrá en funcionamiento una vez se haya pulsado START, en ese instante el motor se activará haciendo que la banda avance hasta que se active la señal de posición 1 de la canasta, para lo cual se detendrá la banda. Una vez se haya terminado el proceso de cierre de botellas, la banda avanzará nuevamente hasta llegar a la posición 2 de la canasta en donde se detendrá hasta que la canasta haya sido ubicada en la mesa de distribución.



4.8 Conclusiones

El uso de GRAFCET dentro de la automatización de procesos, resulta ser una importante herramienta de diseños, debido a que permite un esquema visual del proceso, permite un análisis del sistema en general, adicionalmente, me permite utilizar los diseños implementados mediante Rdp debido a que son técnicas que tienen bastantes cosas en común.

Es así como en este capítulo se desarrollaron las bases para el diseño de automatismos mediante la técnica del GRAFCET, planteando su funcionamiento, ecuaciones características, aplicaciones y ejemplos. Se trabajó el ejemplo planteado en el capítulo dos, que trata sobre el embotellamiento de agua.

Para el desarrollo de posteriores capítulos, como lo es el sexto, el lector deberá tener claros los conceptos estudiados a lo largo del documento, sobre todo los capítulos tres y cuatro que hacen alusión a RdP y GRAFCET.

CAPITULO 5: Optimización

En el presente capítulo se introducirá al lector en los principios de la teoría de control optimal u optimización. Al igual que en los capítulos precedentes, el lector podrá conocer sobre las generalidades de la temática expuesta, para posteriormente profundizar las técnicas específicas, que para este caso hacen alusión a la optimización.

Dentro del proceso de optimización, se parte de la generación de modelos matemáticos, que sean capaces de simular el sistema a optimizar. Una vez planteado el problema se evalúan las mejores posibilidades de solución según sean las características intrínsecas del sistema. En el desarrollo de este capítulo, se enunciarán los métodos de optimización para problemas lineales o no lineales, con y sin restricciones. Se aclara al lector que la teoría aquí expuesta es solo el principio de los conceptos de optimización, por lo que no se profundizará en cada uno de los tópicos aquí mencionados. Si el lector está interesado en profundizar las temáticas expuestas, puede remitirse a las referencias [1] y [2].

5.1 Definición y Justificación

El diseño de los sistemas de control clásicos se basa principalmente en prueba y error, en donde se obtienen parámetros “aceptables para cada sistema. Generalmente se define en términos en el dominio del tiempo o de la frecuencia, siguiendo criterios como el tiempo pico, el máximo sobre impulso, entre otros.

Sin embargo, en los problemas complejos de múltiples entradas y múltiples salidas, los sistemas requieren de un desarrollo más profundo que logre satisfacer las demandas de dichos sistemas. Es así como surgen teorías como la del control optimal u optimización que se integra con el desarrollo de la tecnología digital.

El objetivo principal de la teoría de control optimal es el de determinar las señales de control que permitirán satisfacer las restricciones físicas de un proceso e igualmente minimizar o maximizar algunos criterios de funcionamiento [1].

La optimización es un proceso en el cual se maximiza (distancia, ganancia, velocidad, eficiencia, entre otros) o se minimiza (tiempo, error, costos, entre otros) un determinado criterio, teniendo en cuenta restricciones, que indican que no toda decisión puede llegar a ser viable. El objetivo principal es entonces, el de cuantificar el rendimiento y medir la calidad de decisión, maximizando o minimizando procesos, teniendo en cuenta las restricciones puntuales que puedan llegar a surgir en un problema específico [2].

Dentro del proceso de optimización, se hace fundamental la formulación de un modelo matemático lo suficientemente complejo y manipulable que satisfaga las necesidades y requisitos de un caso concreto; razón por la cual se dice que la optimización es un instrumento de conceptualización.

En términos generales, un modelo matemático está compuesto por los siguientes elementos:

- Variables de Estado

$$x_1(t), x_2(t), \dots, x_n(t) \quad (5.1)$$

- Entradas de Control

$$u_1(t), u_2(t), \dots, u_m(t) \quad (5.2)$$

- Trayectoria de Estado: $[x]$: Historial de valores de variables estado en el intervalo de tiempo $[t_0, t_f]$
- Control $[u]$: Historial de valores de entradas de control en el intervalo $[t_0, t_f]$

Tomando esto como base se describe a un sistema como una relación de n -ecuaciones diferenciales, así:

$$\begin{aligned} \dot{x}_1 &= a_1(x_1(t), x_2(t), \dots, x_n(t), u_1(t), u_2(t), \dots, u_m(t)) \\ \dot{x}_2 &= a_2(x_1(t), x_2(t), \dots, x_n(t), u_1(t), u_2(t), \dots, u_m(t)) \\ &\vdots \\ \dot{x}_n &= a_n(x_1(t), x_2(t), \dots, x_n(t), u_1(t), u_2(t), \dots, u_m(t)) \end{aligned} \quad (5.3)$$

Expresando de forma global y en términos funcionales, se puede obtener la siguiente expresión.

$$\begin{aligned} \max(\min) \quad & f(x) \\ & x \in \Omega \subseteq \mathbb{R}^n \end{aligned} \quad (5.4)$$

En donde $x = \{x_1, x_2, \dots, x_n\}$ es un vector y representa las variables de decisión, $f(x)$ es llamada la función objetivo y Ω es el conjunto de decisiones factibles o restricciones del problema.

Las restricciones pueden representarse mediante igualdades o desigualdades, como se ve a continuación:

$$g(x_1, x_2, \dots, x_n) \leq 0 \quad (5.5.1)$$

$$h(x_1, x_2, \dots, x_n) = 0 \quad (5.5.2)$$

Resulta conveniente trabajar con ecuaciones de estado, que se derivan de las siguientes ecuaciones, presentadas en forma estándar.

$$\begin{aligned}
 &\text{minimizar} && c_1x_1 + c_2x_2 + \dots + c_nx_n \\
 &\text{sujeto a} && a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\
 &&& a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\
 &&& \vdots \\
 &&& a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \\
 &\text{y} && x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0
 \end{aligned} \tag{5.6}$$

Donde b_i, c_i y a_{ij} son constantes reales fijas, y las x_i son los valores reales a determinar.

Una forma de expresar lo que se ha enunciado anteriormente de un modo vectorial es como se indica a continuación

$$\text{minimizar } C^T x \tag{5.7.1}$$

$$\text{sujeto a } Ax = b \quad \text{y} \quad x \geq 0 \tag{5.7.2}$$

Donde x es un vector columna n-dimensional
 C^T es un vector fila n-dimensional
 A es una matriz m x n
 b es un vector columna m-dimensional

En la teoría de control resulta más beneficioso trabajar con ecuaciones de estado, es por tal motivo que se definirá a continuación el vector de estado, el vector de control y la ecuación de estado.

$$\text{Vector de Estado:} \quad x(t) \triangleq \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} \tag{5.8}$$

$$\text{Vector de Control:} \quad u(t) \triangleq \begin{bmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_m(t) \end{bmatrix} \tag{5.9}$$

Ecuación de Estado: $\dot{x} = a(x(t), u(t), t)$ (5.10)

Donde x y u se han definido anteriormente como trayectoria de estado e historial de control, respectivamente.

Una vez claros estas nociones, es posible precisar nuevos conceptos. Se le da el nombre de Control Admisible $[U]$, al conjunto de ecuaciones que satisfacen las restricciones de control en el intervalo $[t_0, t_f]$. Ahora bien, una Trayectoria Admisible $[X]$, es la trayectoria de estado que satisface las restricciones de las variables de estado durante el intervalo $[t_0, t_f]$.

$$\begin{aligned} u &\in U \\ x &\in X \end{aligned} \tag{5.11}$$

El concepto de admisibilidad es importante debido a que este reduce el rango de los posibles valores que pueden llegar a asumir las variables tanto de control como de estado.

Por otra parte, también es posible enunciar el problema de control optimal u optimización como se muestra en la siguiente expresión.

$$J = h(x(t_f), t_f) + \int_{t_0}^{t_f} g(x(t), u(t), t) dt \tag{5.12}$$

Donde $t_0 \rightarrow$ Tiempo inicial
 $t_f \rightarrow$ Tiempo final
 h y $g \rightarrow$ Funciones escalares

Ahora bien, es posible establecer el concepto de Control Optimal como la solución de control admisible u^* tal que satisfaga el sistema 4.10; del mismo modo, la Trayectoria Optima x^* será aquella que minimice o maximice el criterio de funcionamiento enunciado por la expresión 5.12.

5.2 Métodos de optimización

Una vez se ha obtenido un modelo matemático representativo a una problemática específica de optimización, se debe establecer la solución a dicho criterio. Para realizar este proceso, se debe determinar un método solución, basado en técnicas especializadas, cuyos objetivos deberán ser los más apropiados para resolver cada situación en particular.

La optimización siempre estará orientada a maximizar o minimizar algunos parámetros determinados, sin embargo, dentro del proceso de encontrar una solución, se hacen dos clasificaciones concretas, la optimización lineal y la no lineal. En la lineal, los problemas representados mediante sus ecuaciones y restricciones son lineales; del mismo modo, en la optimización no lineal, estos parámetros son no lineales. Cabe recalcar, que para cada uno de estos, existen métodos diferentes de solución.

Matemática y analíticamente, es posible llegar a una solución, sin embargo este no es el procedimiento más óptimo ya que las herramientas de software permiten establecer algoritmos iterativos que logan establecer una posible solución optimal.

Se tiene que para los problemas lineales la iteración es finita, obteniendo la solución exactamente después del número finito de pasos; pero para los problemas no lineales, en la mayoría de los casos, la sucesión no alcanza el punto de solución sino que converge hacia él, se determina entonces, que el proceso termina prácticamente cuando un punto se aproxima suficientemente a la solución.

Para la creación de algoritmos, es necesario tener en cuenta tres aspectos fundamentales. En primer lugar, la generación del algoritmo; en segundo lugar, el análisis de convergencia global; y por último, el tiempo de convergencia que se ve reflejado en el número de iteraciones necesarias para llegar a una solución. Vale la pena traer a colación la siguiente frase de Confucio: “Una buena teoría vale más que mil ejecuciones de computador”.

5.2.1 Programación Lineal

Dado un conjunto Q de la expresión 5.7.2, representado por A ($m \times n$), de m ecuaciones lineales simultáneas de n incógnitas; sea B un conjunto o submatriz de m columnas linealmente independientes, no singular, conformado por columnas de A y de dimensiones $m \times m$, se declaran los siguientes conceptos:

$$Bx_n = b \quad (5.13)$$

Solución básica: Si todas la n-m componentes de x no asociadas a columnas de B se igualan a cero (0), la solución del conjunto resultante de ecuaciones recibe el nombre de “solución básica” de Q respecto a la base B.

Variables básicas: Son las componentes de x asociadas a columnas de B.

Es posible que la ecuación de la expresión 5.7.2 no tenga soluciones básicas, motivo por el cual se suele suponer que:

- 1) $n > m$: El número de variables x_i es superior al número de restricciones.
- 2) Las filas de A son linealmente independientes, asegurando así la independencia lineal de las m ecuaciones.

Es importante aclarar que dependencia entre filas de A , harán que no sea posible encontrar la solución de Q .

Se deben considerar otros conceptos en desarrollo de esta teoría. Se dice que se tiene "rango completo" cuando la matriz A de $m \times n$ tiene $m < n$ y las m filas de A son linealmente independientes.

Se define una "solución básica degenerada" a aquella en la que una o más variables básicas están igualadas a cero (0). Una "solución factible", es la que satisfaga la expresión 5.7.2 para las todas sus restricciones.

Una solución "factible básica" es tanto factible como básica y por último se tienen la "solución factible básica degenerada", la cual satisface todas las condiciones que ella misma enuncia.

Para resolver un problema de solución óptima se tendrán en cuenta solamente las soluciones factibles básicas, debido a que con estas se conseguirá el valor óptimo.

Una solución factible óptima es aquella que alcanza una solución para el conjunto de restricciones y el valor mínimo para satisfacer la función objetivo; si es básica, se denomina solución factible básica óptima.

Teorema Fundamental de la Programación Lineal

Dado un programa lineal de la forma estándar (ver expresiones 5.7.1 y 5.7.2), donde A es una matriz de $m \times n$ de rango m :

- 1) Si hay una solución factible, hay una solución factible básica.
- 2) Si hay una solución factible óptima, hay una solución factible básica óptima.

“Al buscar una solución óptima para un programa lineal solo es necesario tener en cuenta las soluciones factibles básicas, ya que en estas, siempre se encontrará el valor óptimo”.⁵

Los métodos de solución óptima mediante programación lineal se enuncian a continuación:

- Método simplex
- Método simplex reducido
- Método simplex y la descomposición LU
- Descomposición
- Programas lineales duales
- Método simplex dual
- Algoritmo primal dual
- Triangularidad de las bases
- Flujo maximal (basado en redes - RdP)
- Flujo de costo mínimo (basado en redes - RdP)

5.2.2 Programación no Lineal

Los métodos o algoritmos de solución para los problemas no lineales se pueden dividir en dos amplios grupos: problemas sin restricciones y problemas con restricciones.

Matemáticamente la programación no lineal con restricciones se plantea según la expresión 4.13.

$$\begin{aligned} & \text{minimizar} && f(x) \\ & \text{sujeto a} && h_i(x) = 0, \quad i = 1, 2, \dots, m \\ & && g_j(x) \leq 0, \quad j = 1, 2, \dots, r \\ & && x \in S \end{aligned}$$

(5.14)

Donde x es un vector de incógnitas n -dimensional

y $x = (x_1, x_2, \dots, x_n)$ y f
 $h_i, i = 1, 2, \dots, m$ y $g_j, j = 1, 2, \dots, r$
son funciones con valores reales de las variables x_1, x_2, \dots, x_n

⁵ Luenberger D. PROGRAMACION LINEAL Y NO LINEAL. Capítulo 2 – Propiedades básicas de los programas lineales. Parte I - Programación lineal. Addison Wesley Longman. ISBN 968-4444-362-5

el conjunto S es un subconjunto de un espacio n -dimensional

Para la representación enunciada en 5.14, $f(x)$ es la función objetivo del problema, mientras que las ecuaciones, desigualdades y el conjunto de restricción son restricciones.

Se han desarrollado técnicas para la solución de problemas de optimización planteados como no lineales y a los cuales se les tiene en cuenta restricciones. Se nombra a continuación algunos de los métodos existentes dentro de la programación no lineal con restricciones:

- Métodos de dirección factible
- Métodos de conjunto activo
- Método de proyección del gradiente
- Método del gradiente reducido
- Métodos de penalización
- Métodos de barrera
- Método de Newton
- Lagrangianos aumentados
- Métodos del plano cortante
- Algoritmo de plano cortante convexo de Kelley
- Programación cuadrática
- Métodos directos
- Métodos de Newton modificados
- Métodos cuasi Newton

En la programación no lineal sin restricciones, la expresión 5.14 queda reducida a minimizar (o maximizar) la función objetivo $f(x)$, donde $f(x)$ es una función no lineal. Así como para la solución de problemas no lineales con restricciones se han desarrollado algunas metodologías, la programación no lineal sin restricciones también cuenta con algunos algoritmos o métodos de solución a estos planteamientos, tal como se expone en la siguiente lista:

- Minimización y maximización de funciones convexas
- Método básico de descenso
- Método de descenso de mayor pendiente
- Método de Newton
- Método de descenso coordinado
- Método de dirección conjugada
- Método de gradiente conjugado

- Método de gradiente conjugad parcial
- Método de Newton modificado
- Método de Davidon-Fletcher-Powell
- Método cuasi Newton sin memoria
- Combinación del método de descenso de mayor pendiente con el método de Newton

5.5 Aplicaciones

La optimización tiene una amplia gama de aplicaciones dentro de diferentes ramas del conocimiento aplicadas a la vida cotidiana.

La primera de las aplicaciones que se tratará, va enfocada a variables físicas que atañen comúnmente a la ingeniería. Dentro de estas, se puede destacar el control de velocidad, distancia, posición, entre otros.

En los negocios por ejemplo, es posible llegar a la maximización de las ganancias, reduciendo el margen de de pérdidas; este tipo de análisis resulta fundamental en el desarrollo y desempeño de cualquier empresa, ya que de estos se derivan las posibles decisiones que se tomen para un funcionamiento óptimo.

En este mismo campo se puede contemplar la aplicación relacionada con el beneficio, haciendo un enfoque en la inversión, se establece que tan arriesgada y que tan viable puede llegar a ser; lo cual resulta esencial en el dinamismo de las compañías.

Ahora bien, también puede surgir la necesidad de optimización en el funcionamiento y toma de decisiones del gobierno; si se contempla un enfoque en el ámbito de bienestar social, es posible analizar diferentes ítems referentes a la calidad de vida de la población y de qué manera se pudiese maximizar la misma, según los aspectos fundamentales que benefician y afectan a la población.

Una de las aplicaciones más comunes es la de reducción de costos, y esta es aplicada tanto a nivel laboral como a nivel de persona; resulta ilustrativa la situación de la reducción máxima de gastos obteniendo el mayor beneficio posible, se ven casos como el rendimiento de una empresa, el manejo del dinero en la casa, una dieta balanceada baja en coste, entre otros.

5.6 Ejemplos

Los ejemplos que se anuncian a continuación incluyen el modelo matemático para optimizar cada una de las problemáticas planteadas. Se tratarán 4 casos, cada uno de ellos enfocado a temáticas diferentes según la teoría vista a lo largo del capítulo, el primero de ellos se enfocará en el modelado de un problema para solucionarlo matemáticamente, el segundo trabaja con un problema de asignación que se soluciona mediante técnicas de programación lineal, el tercero se enfoca en el planteamiento de problemas de control para ser solucionado mediante técnicas de programación no lineal sin restricciones y el cuarto presenta el ejemplo típico de volumen máximo que está ligado a los problemas de programación no lineal con restricciones.

Es importante aclarar al lector que en el desarrollo de los ejemplos no se desarrollará la solución óptima para cada uno de los casos, sino que se dejará el planteamiento del modelo matemático tal y como se estudió durante el capítulo. Si se quiere profundizar en la solución óptima, el lector debe remitirse a la fuente [1] y [2]

Ejemplo 5.1:

El carro mostrado en la Fig. 5.1 va en línea recta y parte del reposo desde el punto O. La distancia del carro desde el punto O en el tiempo t se denota mediante $d(t)$. Para simplificar el modelo, se considerará al carro como una unidad puntual de masa, que podrá acelerarse usando el acelerador y podrá desacelerarse utilizando el freno.

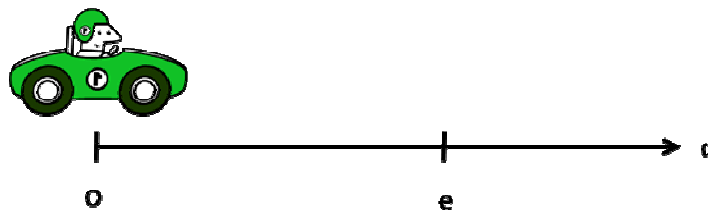


Fig. 5.1 Ejemplo de control simplificado

El objetivo es hallar el modelo matemático que satisfaga las condiciones planteadas mediante ecuaciones diferenciales y ecuaciones de estado. Se debe considerar el problema asumiendo que el automóvil arranca en el punto O y se detiene en el punto e .

Solución:

A nivel general, la ecuación diferencial que caracteriza al sistema planteado es la que se ve en la expresión 5.15.

$$\ddot{d}(t) = \alpha(t) + \beta(t) \quad (5.15)$$

Donde α simboliza el control de aceleración y β el de desaceleración. La posición y la velocidad se plantean como variables de estado, así:

$$x_1(t) \triangleq d(t) \quad y \quad x_2 \triangleq \dot{d}(t) \quad (5.16)$$

Del mismo modo

$$u_1(t) \triangleq \alpha(t) \quad y \quad u_2(t) \triangleq \beta(t) \quad (5.17)$$

Las ecuaciones de estado quedan como se muestran en la expresión 5.18.

$$\begin{aligned} \dot{x}_1 &= x_2(t) \\ \dot{x}_2 &= u_1(t) + u_2(t) \end{aligned} \quad (5.18)$$

Si utilizamos notación matricial, las ecuaciones de estado se representan tal como se ve en la expresión 5.19

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x(t) + \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix} u(t) \quad (5.19)$$

Considerando que t_0 es el tiempo de arranque desde el punto O y que t_f es el tiempo de llegada al punto e, se tiene:

$$\begin{aligned} x_1(t_0) &= 0 \\ x_1(t_f) &= e \end{aligned} \quad (5.20)$$

Adicionalmente, tenemos que la velocidad en el punto inicial y el punto final es cero debido a que el automóvil parte del reposo y termina en reposo, por lo que se cumple:

$$\begin{aligned} x_2(t_0) &= 0 \\ x_2(t_f) &= 0 \end{aligned} \quad (5.21)$$

Matricialmente estas condiciones de frontera se expresan tal como se ve en 5.21.

$$x(x_0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0 \quad y \quad x(t_f) = \begin{bmatrix} e \\ 0 \end{bmatrix} \quad (5.22)$$

Y si se asume que el automóvil no regresa, se pueden considerar las siguientes restricciones:

$$\begin{aligned} 0 \leq x_1(t) \leq e \\ 0 \leq x_2(t) \end{aligned} \quad (5.23)$$

Se tomará la máxima aceleración del automóvil como $M_1 > 0$ la máxima deceleración como $M_2 > 0$, de este modo los controladores pueden satisfacer que:

$$\begin{aligned} 0 \leq u_1(t) \leq M_1 \\ -M_2 \leq u_2(t) \leq 0 \end{aligned} \quad (5.24)$$

Con la expresión 5.24 se ha concluido el modelo matemático propuesto para el problema enunciado, incluyendo las restricciones típicas que pueden llegar a encontrarse dentro de este contexto.

Ejemplo 5.2: Problema de la asignación

Se debe asignar óptimamente n robots a n tareas. Si al robot i se le asigna una tarea j , hay un beneficio de c_{ij} . A cada robot debe asignársele una tarea y cada tarea debe tener asignado un robot. El objetivo es realizar la asignación para maximizar el valor total de la asignación.

Solución:

La formulación general para el problema de asignación es hallar

$$\begin{aligned} x_{ij}, \quad i = 1, 2, \dots, n; \quad j = 1, 2, \dots, n \quad (5.25) \\ \text{minimizar} \quad \sum_{j=1}^n \sum_{i=1}^n c_{ij} x_{ij} \\ \text{sujeto a} \quad \sum_{j=1}^n x_{ij} = 1 \quad \text{para } i = 1, 2, \dots, n \\ \sum_{i=1}^n x_{ij} = 1 \quad \text{para } j = 1, 2, \dots, n \\ x_{ij} \geq 0 \quad \text{para } i = 1, 2, \dots, n \\ \quad \quad \quad j = 1, 2, \dots, n \end{aligned} \quad (5.26)$$

En este planteamiento será necesario que cada variable x_{ij} , tome los valores de 0 o 1, ya que de otro modo, la solución no será significativa, ya que no es posible hacer asignaciones fraccionarias.

Para el desarrollo del problema de asignación se desarrollo un algoritmo muy eficiente que fue trabajado por dos matemáticos húngaros y posteriormente fue generalizado para el desarrollar el método primal – dual de la programación lineal.

Ejemplo 5.3: Problema de Control

Los problemas dinámicos, donde las variables corresponden a acciones tomadas en una sucesión de momentos de tiempo, pueden formularse como problemas de optimización sin restricciones.

Para este ejemplo, se supondrá que la posición de un objeto grande es controlada por una serie de fuerzas de control correctoras, en donde el error de posición está dado por la ecuación 5.27.

$$x_{k+1} = x_k + u_k \quad (5.27)$$

Donde x_k es el error en el tiempo k , y u_k es la fuerza efectiva aplicada al tiempo u_k luego de haberse normalizado para tener en cuenta la masa del objeto y la duración de la fuerza. El valor de x_0 está dado.

Se pretende encontrar la sucesión u_1, u_2, \dots, u_n para minimizar el error de la variable a controlar.

Solución:

El objetivo del problema planteado se expresa en la ecuación 5.18, en donde se desea igualar el error x_k a 0 teniendo en cuenta que la acción de control u_k es costosa, debe entonces encontrarse un equilibrio óptimo.

$$J = \sum_{k=0}^n \{x_k^2 + u_k^2\} \quad (5.28)$$

Es posible convertir este problema a uno sin restricciones eliminando las variables x_k , $k = 1, 2, \dots, n$, de la función objetivo (5.28). Se tienen entonces:

$$x_k = x_0 + u_0 + u_1 + \dots + u_{k-1} \quad (5.29)$$

De este modo la función objetivo puede expresarse nuevamente tal como se muestra en la ecuación 5.30, que es una función cuadrática de las incógnitas u_k

$$J = \sum_{k=0}^n \{(x_0 + u_0 + u_1 + \dots + u_{k-1})^2 + u_k^2\} \quad (5.30)$$

Ejemplo 5.4: Volumen Máximo

Se desea construir una caja de cartón de volumen máximo, dada un área fija de cartón.

Solución:

Representando con x , y , z las dimensiones de la caja, el problema puede expresarse según la relación de ecuaciones expuestas en 5.31.

$$\begin{aligned} &\text{minimizar} && xyz \\ &\text{sujeto a} && (xy + yz + xz) = \frac{c}{2} \end{aligned} \quad (5.31)$$

Donde $c > 0$ es el área dada del cartón. Al introducir un multiplicador de Lagrange, se halla fácilmente que las condiciones necesarias de primer orden son:

$$\begin{aligned} yz + \lambda(y + z) &= 0 \\ xz + \lambda(x + z) &= 0 \\ xy + \lambda(x + y) &= 0 \end{aligned} \quad (5.32)$$

Para resolver las ecuaciones, se multiplica la primera por x y la segunda por y , y después de restan para obtener la expresión 5.33.

$$\lambda(x - y)z = 0 \quad (5.33)$$

En la segunda y tercera se opera de manera similar para obtener

$$\lambda(y - z) = 0 \quad (5.34)$$

Como ninguna variable puede ser cero, resulta que $x = y = z = \sqrt{c/2}$ es la única solución a las condiciones necesarias, la caja debe ser un cubo.

5.7 Conclusiones

A lo largo de este capítulo se estudió la teoría básica del control optimal, haciendo referencia a las diferentes categorías de problemas que pueden llegar a plantearse en el momento de modelar un sistema y llevarlo a su posterior optimización, tales como lo son los problemas lineales, los no lineales con restricciones y los no lineales sin restricciones, para cada uno de estos, se mencionaron algunos de los algoritmos utilizados para establecer un control optimal de acuerdo a las características internas del problema en sí.

Una herramienta indispensable en el desarrollo del control optimal es la programación lineal, la cual se expone en el presente capítulo. Del mismo modo, se plantean algunas problemáticas que pretenden modelar los sistemas propuestos, sin embargo, se aclara al lector que en este documento no se profundiza en los algoritmos de desarrollo, tan solo se deja el planteamiento matemático. Si el autor quisiera profundizar en el tema, deberá consultar las referencias [1] y [2].

CAPITULO 6: Implementación de RdP, Grafcet, Optimización y Sistemas de Manufactura Flexible

Los sistemas de manufactura flexible (SMF) pretenden optimizar la producción mediante herramientas de automatización y control, minimizando costos y tiempos de producción y al mismo tiempo maximizar la vida útil de las máquinas. Resulta de gran importancia dentro del marco industrial la aplicación e integración de las teorías estudiadas en los capítulos 2, 3, 4 y 5, que abarcan las temáticas concernientes a RdP, GRAFCET y Optimización.

En el presente capítulo se aplicarán los conceptos de las teorías planteadas en un sistema físico hipotético, generando un modelo que permite determinar la secuencia óptima dentro de un sistema de producción industrial con procesos concurrentes y recursos compartidos.

6.1 Modelo y Análisis de un Sistema mediante RdP

Para la implementación del modelo mediante RdP, se plantea un sistema físico hipotético relacionado con la producción de chocolates. El sistema está conformado por dos procesos paralelos, P1 y P2. Los productos finales del proceso P1 son barras de chocolate, mientras que para P2 los productos finales son bolas de chocolate.

Para este ejemplo, cada proceso consta de dos operaciones, la primera corresponde a la etapa de llenado para los productos barra y bola de chocolate en sus respectivos moldes y la segunda hace alusión al transporte de dichos productos para su posterior enfriamiento.

PROCESO 2 (P2)



PROCESO 1 (P1)



Acciones

1. Llenado
2. Transporte

Máquinas

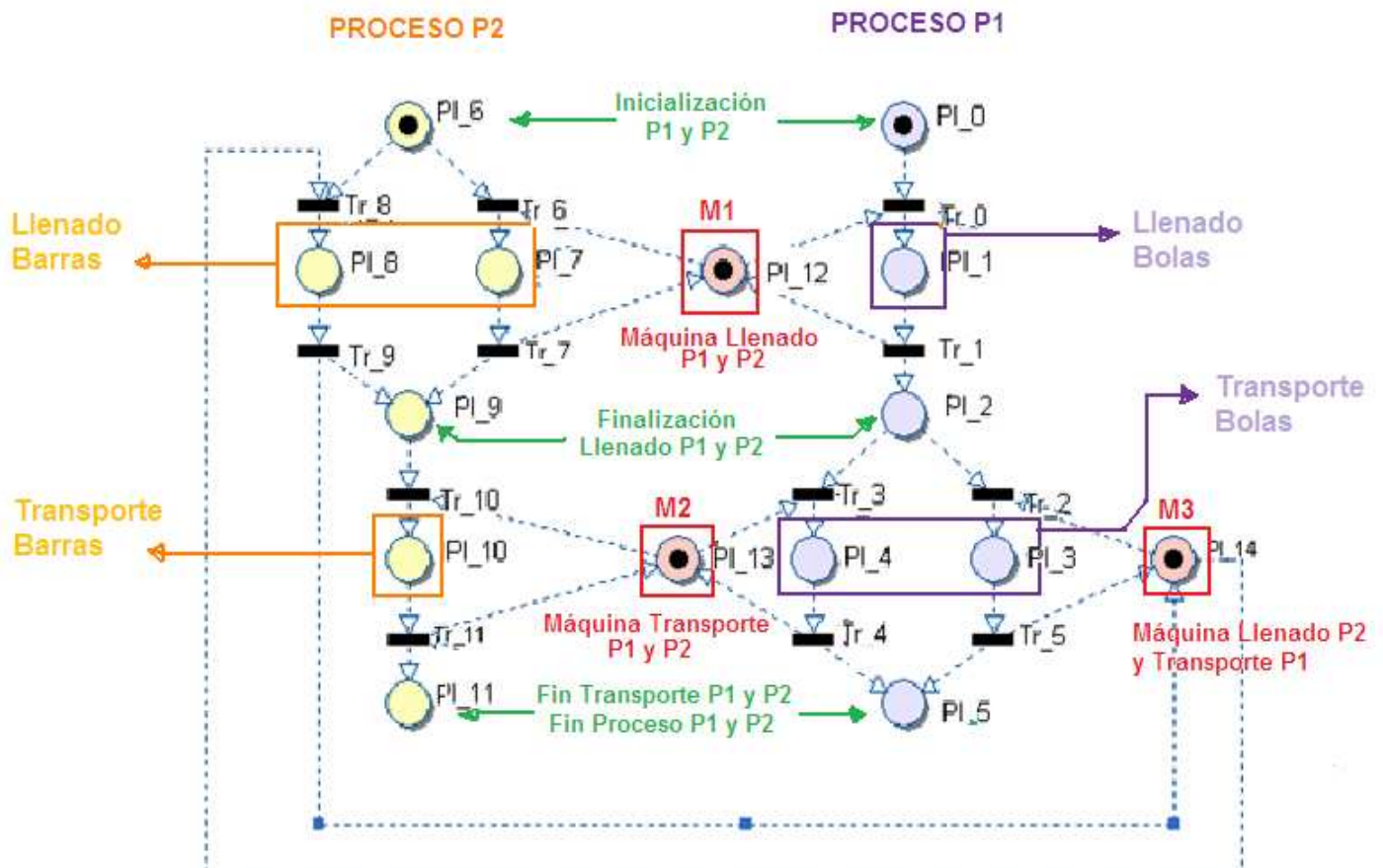
1. Llenado para P1 y P2
2. Transporte P1 y P2
3. Llenado P2 y Transporte P1

Fig. 6.1. Sistema de producción de chocolates para el modelo hipotético

El sistema de producción de chocolates cuenta con tres máquinas o recursos compartidos: la máquina uno (1) realiza operaciones de llenado para los dos procesos, la máquina dos (2) efectúa la etapa de transporte para ambos productos y la máquina tres (3) realiza llenado para el proceso P2 y transporte para el proceso P1.

6.1.1 RdP para el Sistema Modelado

Para modelar el sistema propuesto se plantea una RdP que consta de dos Secuencias de Actividades (SA) o procesos, P1 y P2. Se recuerda al lector que en P2, producción de barras, existen dos opciones para realizar la operación de llenado (M1 y M3); de la



misma manera sucede en P2, producción de bolas, debido existen dos alternativas para la operación de transporte (M2 y M3). Teniendo en cuenta la descripción del sistema y teniendo en cuenta sus restricciones de funcionamiento, se diseñó RdP apreciada en la Fig 6.1, la cual modela el sistema de producción de barras y bolas de chocolates planteado anteriormente.

Fig. 6.2. RdP para el Sistema de Producción de Chocolates de barra y de bola

En donde el proceso P1 se encuentra representado por los estados en color violeta (PI_0, PI_1, PI_2, PI_3, PI_4 y PI_5), el proceso P2 por los estados en color amarillo (PI_6, PI_7, PI_8, PI_9, PI_10 y PI_11) y las máquinas por los estados en color rojo (PI_12, PI_13, PI_14).

Para esta red, los lugares de inicialización de proceso para P1 y P2 son PI_0 y PI_6 respectivamente, así mismo, los lugares de finalización para dichos procesos son PI_5 y PI_11 respectivamente. Los lugares PI_2 y PI_9 indican finalización de operación. Los lugares de operación están representados por PI_1, PI_3 y PI_4 para el primer proceso y PI_7, PI_8 y PI_10 para el segundo proceso.

Existen cuatro posibles secuencias o caminos que la red puede tomar en el momento de evolucionar, tal como se muestra a continuación:

- Secuencia 1 (SD1) → **P1**: PI_1 - PI_3 y **P2**: PI_7 - PI_10
- Secuencia 2 (SD2) → **P1**: PI_1 - PI_4 y **P2**: PI_7 - PI_10
- Secuencia 3 (SD3) → **P1**: PI_1 - PI_3 y **P2**: PI_8, PI_10
- Secuencia 4 (SD4) → **P1**: PI_1 - PI_4 y **P2**: PI_8, PI_10

El objetivo será entonces, encontrar la secuencia óptima para el sistema modelado, que dependerá de los tiempos de operación de cada proceso y de la estructura propia de la red implementada.

6.1.2 Caracterización del Sistema

Para la caracterización del sistema, se hace necesario, en primera instancia, establecer y denominar mediante una tabla el significado físico de cada uno de los estados compuestos por la red, en segunda instancia, identificar las secuencias que puedan ser

útiles para el modelamiento del sistema y en tercer lugar, desarrollar las ecuaciones dinámicas de acuerdo a las características de la RdP modelada.

En las tablas 6.1.1 y 6.1.2 de se distribuye y asigna los lugares que conforman la red implementada (Fig. 6.1).

Lugar	PROCESO 1 (P1): Producción Barras	Lugar	PROCESO 2 (P2): Producción Bolas
PI_0	Inicialización P1	PI_6	Inicialización P2
PI_1	O1 – P1: Llenado de chocolate en molde (barra) por Máquina 1	PI_7	O1 – P2: Llenado de chocolate en molde (bola) por Máquina 1
PI_2	Operación 1 de P1 concluida	PI_8	O1 – P2: Llenado de chocolate en molde (bola) por Máquina 3
PI_3	O2 – P1: Transporte de barra de chocolate por Máquina 3	PI_9	Operación 1 de P2 concluida
PI_4	O2 – P1: Transporte de barra de chocolate por Máquina 2	PI_10	O2 – P2: Transporte de bola de chocolate por Máquina 3
PI_5	Operación 2 de P1 concluida. Finalización de P1	PI_11	Operación 2 de P2 concluida. Finalización P2

Tabla 6.1.1. Asignación de lugares para la RdP de la Fig. 6.1

Lugar	MAQUINAS
PI_12	Máquina 1
PI_13	Máquina 2
PI_14	Máquina 3

Tabla 6.1.2. Lugares de recurso para la RdP de la Fig. 6.1

Los tiempos asociados a las operaciones (τ) para cada uno de los procesos se presenta en la tabla 6.2, en la cual se aprecia la distribución de las máquinas con el respectivo tiempo de utilización.

	Proceso 1	τ	Lugar	Proceso 2	τ	Lugar
Operación 1	M1	4	PI_1	M1	3	PI_7
				M3	2	PI_8

Operación 2	M2	3	PI_4	M1	4	PI_10
	M3	2	PI_3			

Tabla 6.2. Relación de tiempos (τ)

Se ha mencionado en la sección 6.1.1 las secuencias del sistema modelado de producción de chocolates. A continuación se analizará físicamente cada una de las secuencias seleccionadas, con la finalidad de tener aclarar el funcionamiento de las mismas.

En la secuencia SD1, tanto el llenado de las barras (P2) como el de las bolas (P1) de chocolate se lleva a cabo mediante la máquina M1, es importante recalcar que dichas operaciones comparten un recurso (M1), motivo por el cual no podrán ejecutarse de forma simultánea. La operación de transporte para las barras, se efectúa con la máquina M2, mientras que para los chocolates mediante la máquina M3.

Para SD2, la operación de llenado para las barras y las bolas se realiza con la máquina M1 y la operación de transporte se lleva a cabo con la máquina M2. Analizando la estructura de la red y el comportamiento de la misma, se puede llegar a la concluir que la secuencia SD2 es la menos apropiada para el objetivo de optimización de la producción del sistema, debido a que tanto para P1 como P2 los recursos son compartidos para todas las operaciones efectuadas, inutilizando los recursos de máquina de M3.

La secuencia SD3 permite el llenado de las barras con la máquina M3 y el de bolas con la máquina M1, con lo cual optimizamos la operación de llenado de los productos. Sin embargo, para la etapa de transporte, tanto P1 como P2 comparten el recurso M2.

En SD4, la operación de llenado para las barras de chocolate se efectúa mediante la máquina M3, mientras que el llenado para las bolas de chocolate se hace mediante la máquina M1. Para la operación de transporte de las barras de chocolate se utiliza la máquina M2, mientras que para las bolas de chocolate se realiza mediante la máquina M3.

De acuerdo al comportamiento y evolución de la RdP modelada mediante las secuencias estudiadas, es posible afirmar que SD4 resulta ser la secuencia óptima del sistema, debido a que disminuye el tiempo de producción de chocolates, aprovechando al máximo los recursos de funcionamiento de cada una de las máquinas.

Una vez estructurada la parametrización del sistema y estudiadas las posibles secuencias de funcionamiento, se pasará a la etapa de modelado matemático, de la RdP planteada, según las ecuaciones de estado para tiempo de proceso remanente y

para tiempo de proceso acumulado, las cuales fueron enunciadas en las expresiones 3.24 y 3.30 respectivamente.

A continuación se evaluará la evolución de la ecuación 3.24 utilizada para el cálculo de tiempos remanentes para el sistema modelado con la RdP de la Fig. 6.1. En primer lugar se expondrán los parámetros de la ecuación para posteriormente pasar a su evaluación y seguimiento.

$$X_r(k+1) = \begin{bmatrix} M(k+1) \\ M_r(k+1) \end{bmatrix} = \begin{bmatrix} I_n & O_n \\ -\delta(k) \cdot P & I_n \end{bmatrix} \cdot \begin{bmatrix} M(k) \\ M_r(k) \end{bmatrix} + \begin{bmatrix} A \\ T \cdot A^+ \end{bmatrix} \cdot u(k) \quad (6.1)$$

Matriz de lugares de operación:

$$P = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(6.2)

Lugares $\rightarrow n = 15$

Transiciones $\rightarrow m = 12$

I_n = Matriz identidad de [15x15]

O_n = Matriz de ceros de [15x15]

Matriz de tiempos de operación:

$$A = \left[\begin{array}{cccccc|cccccc} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ \hline -1 & 1 & 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{array} \right]$$

Proceso

Proceso

Recursos Compartidos

(6.5)

El árbol de cobertura del sistema se presenta en la gráfica 6.3, se obtuvo mediante el toolbox de Redes de Petri de Matlab.

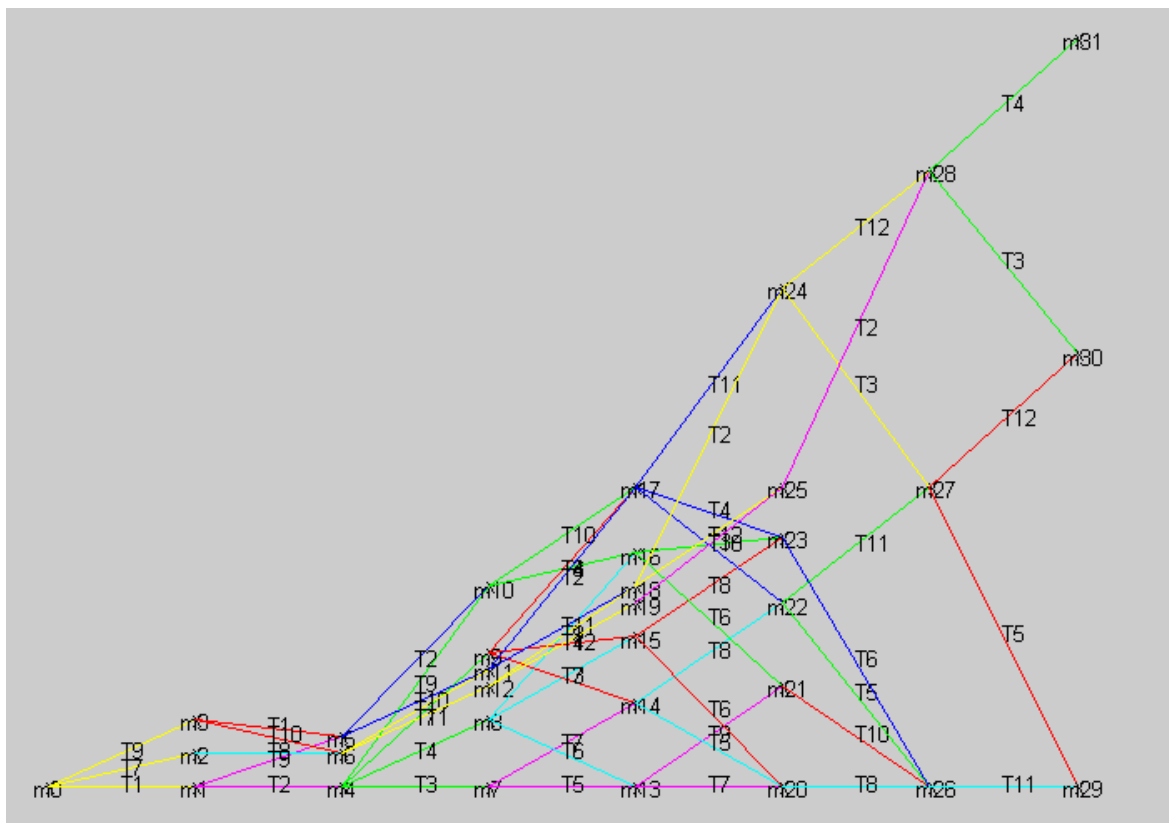


Fig. 6.3. Árbol de cobertura del sistema

El vector de marcado inicial para el sistema modelado se presenta en la expresión 6.6

$$M(k) = [1,0,0,0,0,0,1,0,0,0,0,0,1,1,1]^T \quad (6.6)$$

Para el análisis de las ecuaciones dinámicas de tiempo remanente y tiempo acumulado se tendrá en cuenta el vector de tiempo remanente inicial (6.7), el vector de tiempo acumulado inicial (6.8) y los vectores de control para las secuencias dadas (6.9),(6.10),(6.11),(6.12). Es importante que el lector tenga en cuenta que las ecuaciones de tiempo relacionadas en este documento representan los casos ideales dentro de los cuales no existen tiempos de espera entre procesos.

$$M_r(k) = [6,6,2,2,0,0,6,0,6,4,4,0,0,0,0]^T \quad (6.7)$$

$$M_T(k) = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]^T \quad (6.8)$$

Los vectores de control o secuencias de disparo se presentan a continuación.

$$u_1(k) = [1,1,1,0,1,0,1,1,0,0,1,1]^T \quad (6.9)$$

$$u_2(k) = [1,1,0,1,0,1,1,1,0,0,1,1]^T \quad (6.10)$$

$$u_3(k) = [1,1,1,0,1,0,0,0,1,1,1,1]^T \quad (6.11)$$

$$u_4(k) = [1,1,0,1,0,0,0,0,1,1,1,1]^T \quad (6.12)$$

Tomando el vector de control dado para la secuencia cuatro $u_4(k)$ y los parámetros hallados anteriormente, se obtuvo como resultado para la ecuación de tiempo remanente, la expresión denotada en (6.13)

$$X_r(k+1) = \begin{bmatrix} M(k+1) \\ M_r(k+1) \end{bmatrix} = \begin{bmatrix} [0,0,0,0,0,1,0,0,0,0,0,1,1,1,1]^T \\ [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]^T \end{bmatrix} \quad (6.13)$$

Como se puede observar, el tiempo remanente final para cualquier proceso dará siempre cero, razón por la cual implementamos la ecuación de tiempo acumulado que se presenta en (6.14), que presentará la información del tiempo total de procesos del sistema.

$$X_T(k+1) = \begin{bmatrix} M(k+1) \\ M_T(k+1) \end{bmatrix} = \begin{bmatrix} M(k) \\ M_T(k) \end{bmatrix} + \begin{bmatrix} A \cdot u(k) \\ T \cdot M(k) \end{bmatrix} \quad (6.14)$$

Tomando el vector de control $u_4(k)$ y los demás parámetros hallados a lo largo de la sección, se obtuvo el resultado enunciado en la expresión 6.15

$$X_T(k+1) = \begin{bmatrix} M(k+1) \\ M_T(k+1) \end{bmatrix} = \begin{bmatrix} [0,0,0,0,0,1,0,0,0,0,0,1,1,1]^T \\ [0,4,0,3,0,0,0,0,2,0,4,0,0,0]^T \end{bmatrix} \quad (6.15)$$

6.2 Implementación del Sistema

El sistema hipotético se llevó a simulación física mediante los elementos que se encuentran en el Laboratorio de Automatización de la Universidad Militar Nueva Granada. Se efectuó un montaje electro-hidroneumático en el que se simula la RdP de la figura 6.2.

La parte neumática representa los procesos o secuencias de actividades del sistema global, mientras que el montaje hidráulico representa las máquinas o recursos compartidos del sistema.

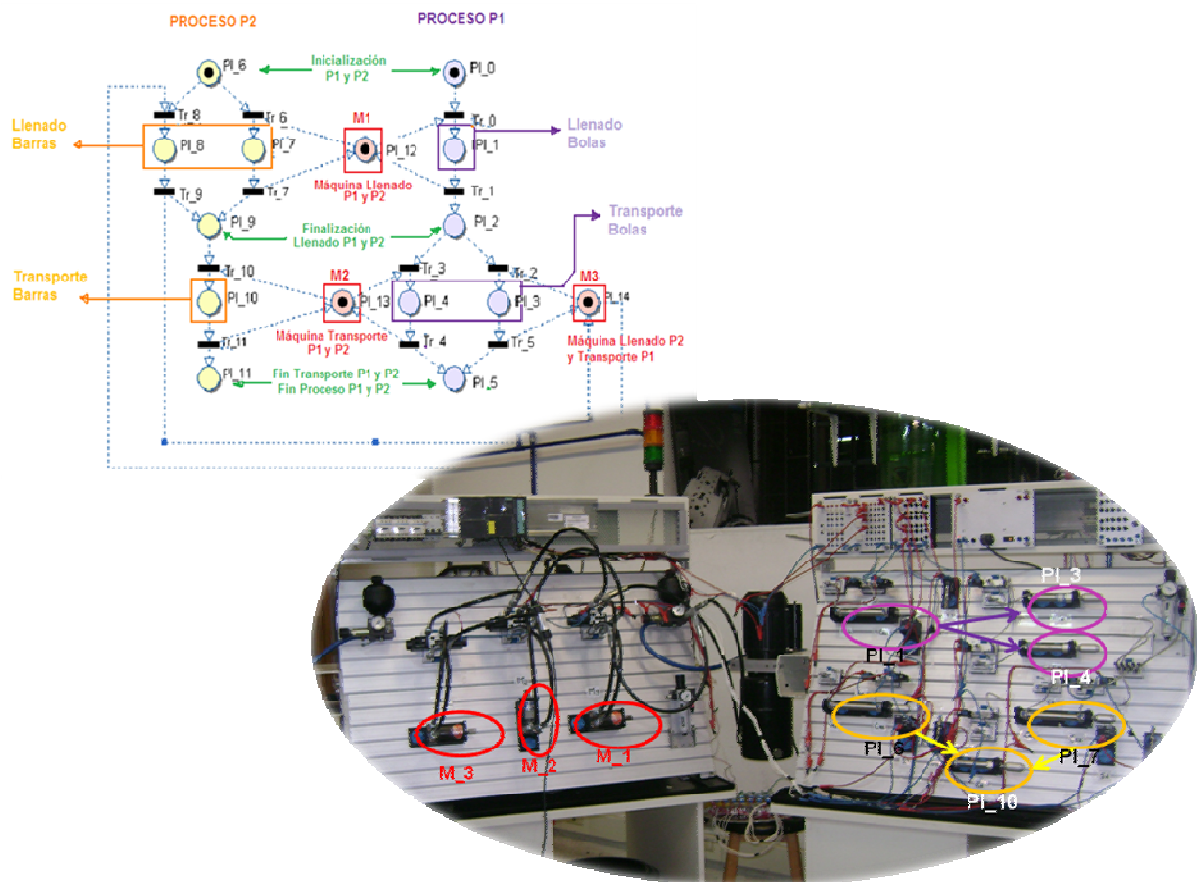


Fig. 6.3. Equivalencia entre la RdP diseñada y la implementación efectuada

6.2.1 Simulación Electro - Neumática

Para simular el sistema hipotético, se realizó una simulación que inicialmente solo incluía la parte electro-neumática, mediante el software simulación FluidSim. Se partió de este diseño a fin de evaluar el comportamiento físico de la implementación de los procesos que conforman el sistema de producción de chocolates anteriormente planteado.

Con el fin de comprender con mayor facilidad la simulación realizada, se presenta en la Fig. 6.4 un esquema que se asemeja al modelo de la RdP planteada en la Fig. 6.2. La simulación completa puede apreciarse en el Anexo 2.

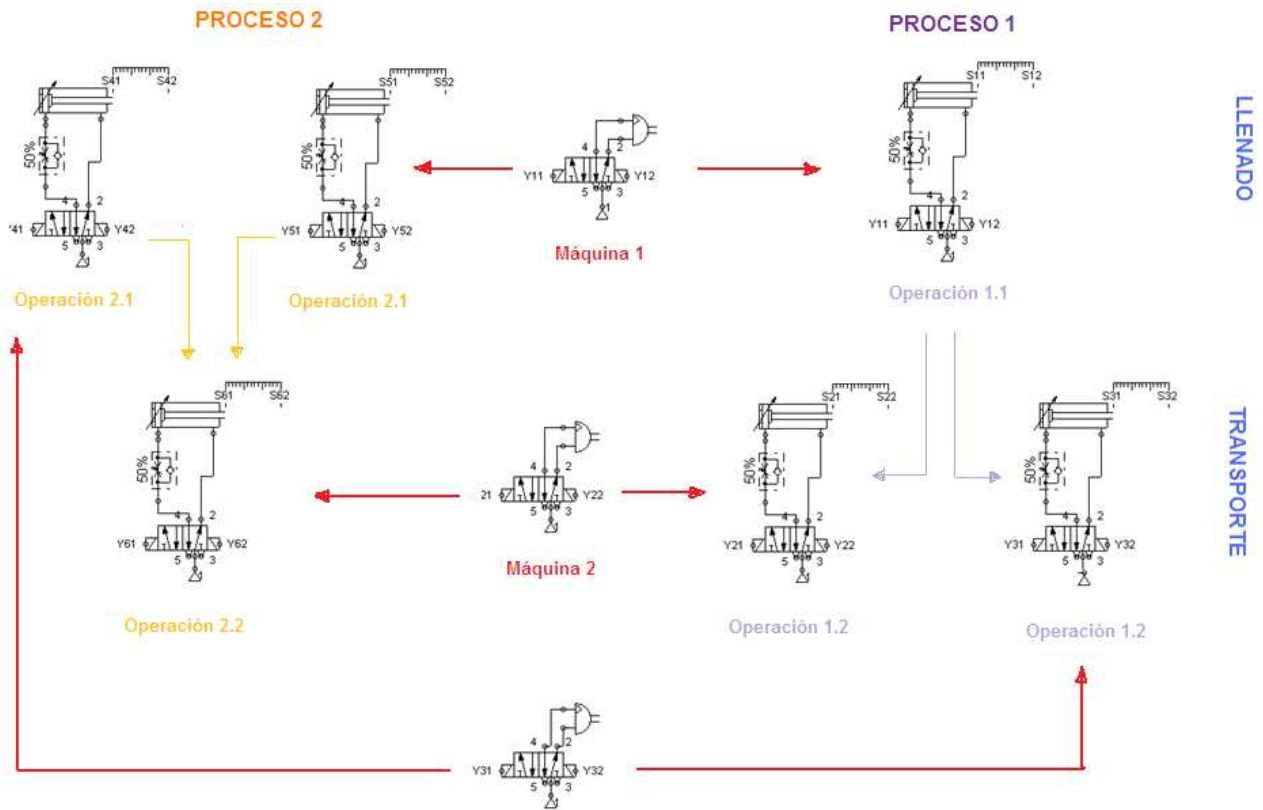


Fig. 6.4. Esquema de funcionamiento de la simulación electro-neumática

En la simulación se utilizaron 6 cilindros de doble efecto y tres motores, cada uno de estos elementos está controlado por una válvula 5/2, adicionalmente, cada secuencia se acciona manualmente mediante pulsadores.

6.2.2 Montaje Físico Electro - Hidroneumático

Para la implementación física, se tuvo en cuenta la simulación que se muestra en la Fig. 6.4, la diferencia radica en que los motores utilizados pasaron a ser hidráulicos, con el fin de tener un sistema híbrido que junta la neumática con la hidráulica. El esquema que vincula el modelo teórico de la RdP con la implementación física se muestra en la Fig. 6.3.

Se establecieron algunas convenciones útiles en el proceso de clasificación y presentación de los elementos involucrados en la implementación del modelo hipotético que se muestran en la tabla 6.3.

CONVENCIÓN	DESCRIPCIÓN
N	Neumático
HB	Hidráulico Bi-estable
HM	Hidráulico Monoestable
E	Eléctrico
CSE	Cilindro Simple Efecto
CDE	Cilindro Doble Efecto
M	Motor
SO	Sensor Óptico
SI	Sensor Inductivo
SM	Sensor Mecánico
P	Pulsador

Tabla 6.3. Convenciones utilizadas en la caracterización del modelo

Para el montaje neumático se utilizaron 3 cilindros de doble efecto para las operaciones de llenado y 3 cilindros de simple efecto para las operaciones de transporte. Se utilizaron 2 sensores ópticos, 1 inductivos y 3 finales de carrera para las señales de los cilindros de doble efecto.

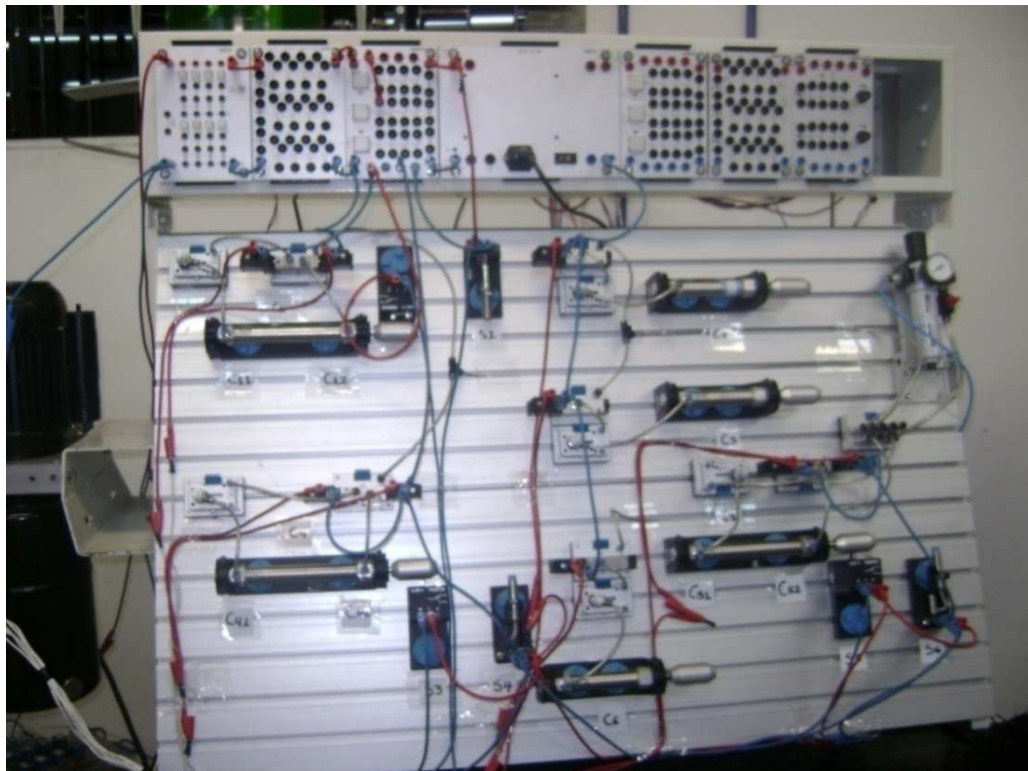


Fig. 6.5. Montaje neumático del sistema implementado

El montaje hidráulico está compuesto por tres motores y tres válvulas, dos de ellas bi-estables de tres posiciones y una monoestable de dos posiciones. En el momento en que se accione un motor significará que una máquina está siendo utilizada dentro del sistema modelado.



Fig. 6.6. Montaje hidráulico del sistema implementado

La RdP que se muestra a continuación presenta la nomenclatura que fue utilizada para el montaje electro hidroneumático.

Con el fin de simular el funcionamiento de las máquinas, además de los motores, se programaron luces industriales que permiten visualizar mediante 3 diferentes colores el funcionamiento de las mismas.

Las señales de entrada y de salida del sistema híbrido, fueron llevadas al PLC y desde allí se implantaron los GRAFCET de control para obtener un funcionamiento automático y uno manual (paso a paso). La tabla 6.4 presenta la relación de entradas del sistema, mientras que la tabla 6.5 representa la relación de los elementos de salida.

Variables Entrada (eventos)	Descripción	Tipo Elemento
S1	Sensor_1	E - SM
S2	Sensor_2	E - SO
S3	Sensor_3	E - SM
S4	Sensor_4	E - SO
S5	Sensor_5	E - SM
S6	Sensor_6	E - SI
START	Pulsador de Inicio	E - P
P1	Selector Secuencia Proc_1	E - P
P2	Selector Secuencia Proc_2	E - P

Tabla 6.4. Variables de entrada para el sistema electro-hidroneumático

Variables Salida (estados)	Descripción Montaje	Descripción Modelo	Tipo Elemento
C11	Cilindro_1 Sale	Inicio llenado P1 con M1	N - CDE
C12	Cilindro_1 Entra	Fin llenado P1 con M1	
C2	Cilindro_1 Sale	Transporte P1 con M3	N - CSE
C3	Cilindro_1 Sale	Transporte P1 con M2	N - CSE
C41	Cilindro_1 Sale	Inicio llenado P2 con M1	N - CDE
C42	Cilindro_1 Entra	Fin llenado P2 con M1	
C51	Cilindro_1 Sale	Inicio llenado P2 con M3	N - CDE
C52	Cilindro_1 Entra	Fin llenado P2 con M3	
C6	Cilindro_1 Sale	Transporte P2 con M2	N - CSE
M1	Motor_1	M1 operando	HM
M2	Motor_2	M2 operando	HB
M3	Motor_3	M3 operando	HB

Tabla 6.5. Variables de salida para el sistema electro-hidroneumático

Los tiempos de operación del sistema implementado se muestran en la Fig. 6.7. Debe tenerse en cuenta que en la implementación física se programaron los tiempos en segundos, sin embargo, el análisis teórico se encuentra en Unidades de Tiempo (UT), esto con el fin de facilitar la apreciación del sistema.

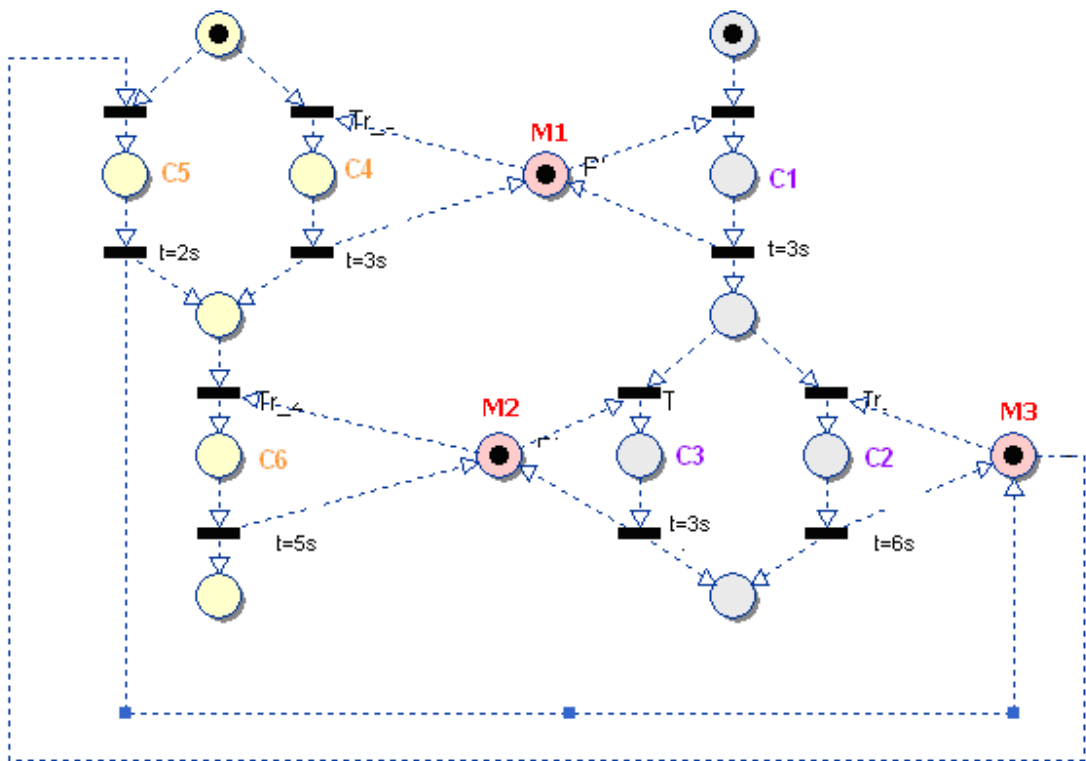


Fig. 6.7. RdP con los tiempos de operación del sistema implementado

En las figuras 6.8, 6.9, 6.10 y 6.11 se plasman las cuatro secuencias de funcionamiento seleccionadas analizando que secuencia resulta ser la óptima.

Secuencia 1 (SD1) → P1: P1 - P1₃ y P2: P1 - P1₇ - P1₁₀

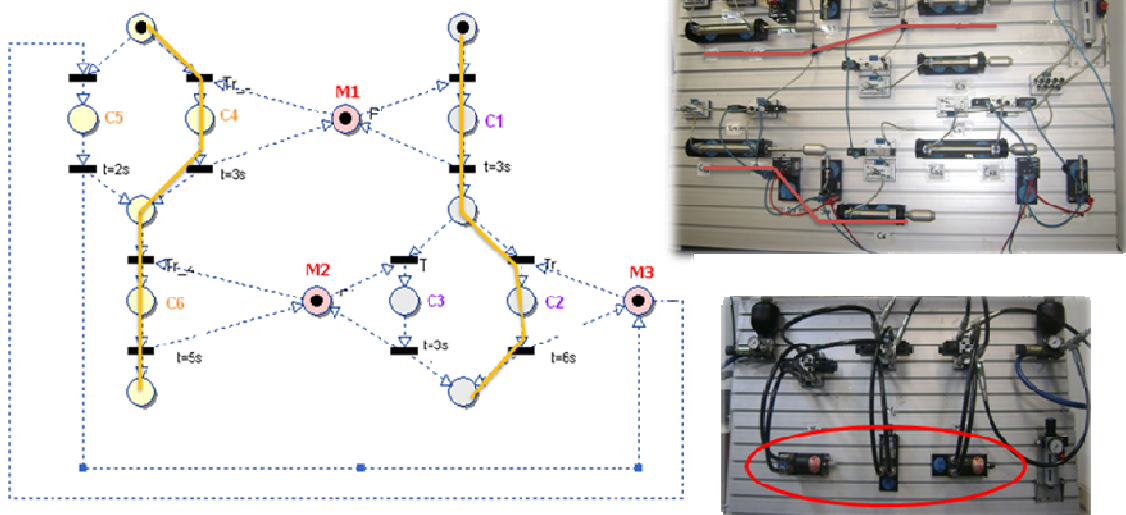


Fig. 6.8. Secuencia No.1 de Funcionamiento

Secuencia 2 (SD2) → P1: PI_1 - PI_4 y P2: PI_7 - PI_10

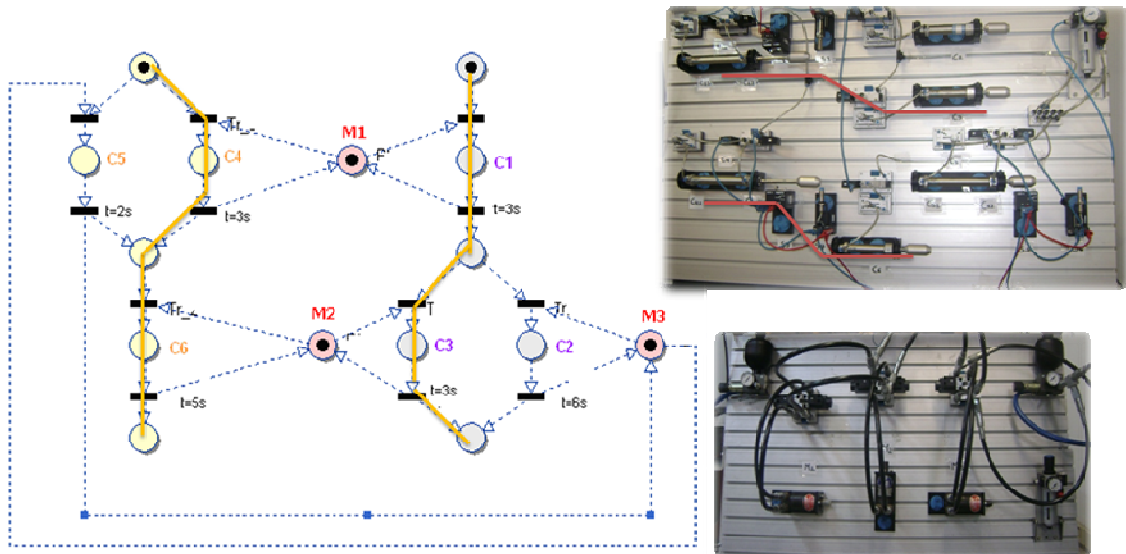


Fig. 6.9. Secuencia No.2 de Funcionamiento

Secuencia 3 (SD3) → P1: PI_1 - PI_4 y P2: PI_8, PI_10

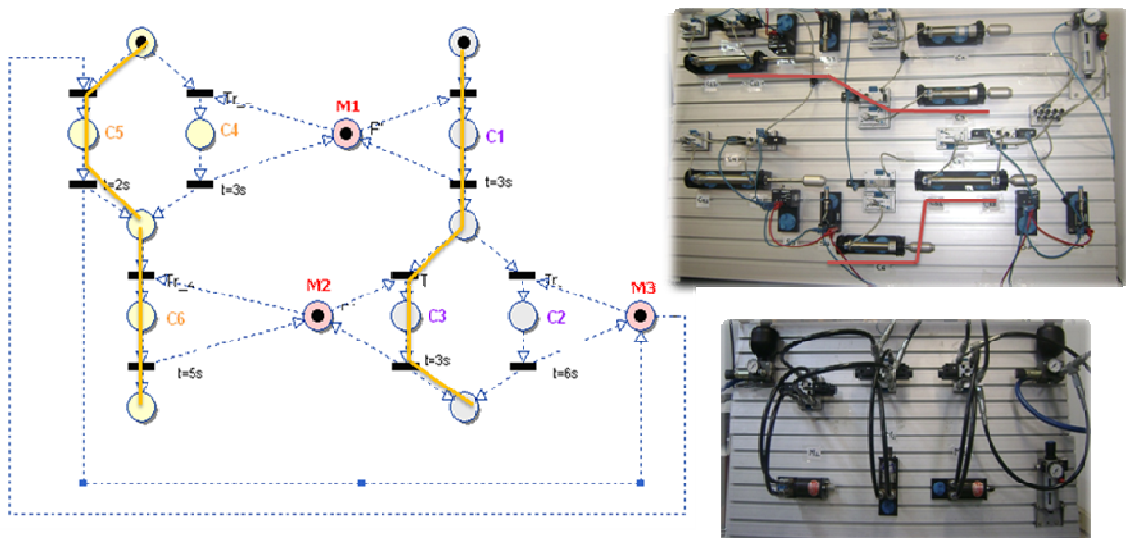


Fig. 6.10. Secuencia No.3 de Funcionamiento

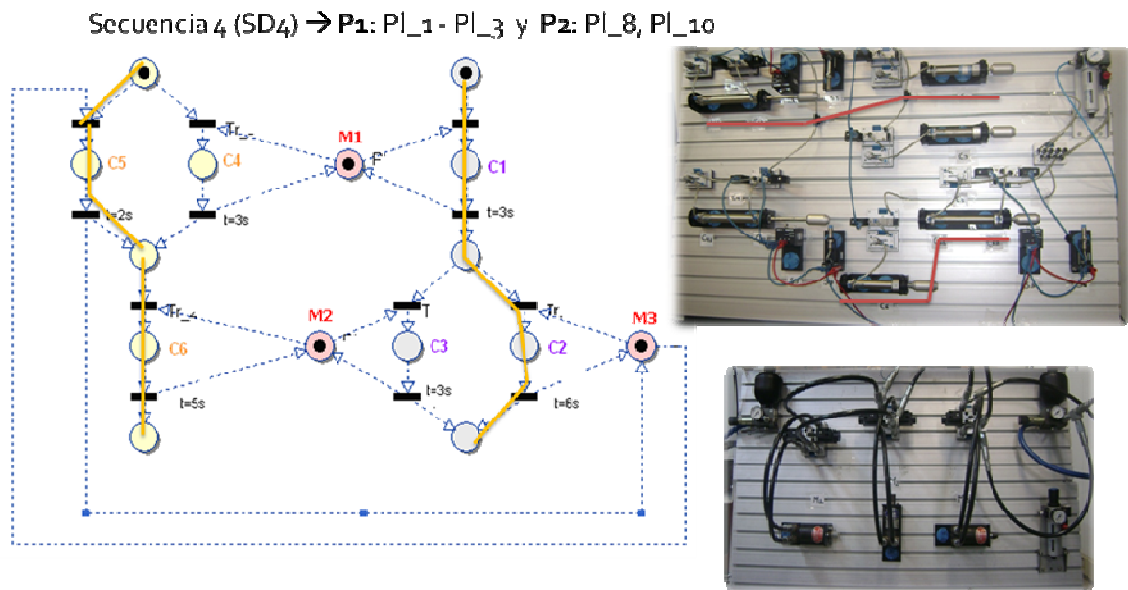


Fig. 6.11. Secuencia No.4 de funcionamiento – secuencia óptima

Comparando los resultados esperados que fueron comentados en la sección 6.1.1, se pudo verificar que la secuencia No. 4.

6.2.3 Control mediante GRAFCET – PLC SIEMENS

El sistema implementado se controla en su totalidad mediante el PLC Siemens S7-300. Se realizó toda la programación en GRAFCET aplicando las teorías aprendidas en el capítulo 4.

Los GRAFCET de control que se programaron en el PLC se muestran en el Anexo 3, esto con el fin de no extender demasiado el contenido del capítulo, sin embargo, en la sección 6.2.3, se hará una esquema explicativo de la implementación efectuada.

Existen dos programas de control del sistema uno para funcionamiento paso a paso y otro para funcionamiento automático. Sin embargo, para cualquiera de estos, se diseñaron GRAFCET independientes para los dos procesos implementados y para las máquinas simuladas.

EPILOGO

A. Conclusiones Generales

El trabajo de grado presentando contribuye a la formación de la comunidad neogranadina en las temáticas relacionadas con Sistemas a Eventos Discretos (SED), específicamente a la aplicación práctica de teorías de Redes de Petri (RdP), GRAFCET (G) y Optimización. La teoría estudiada puede ser utilizada e implementada en cursos de automatización o control para los estudiantes del Programa de Mecatrónica de la Universidad Militar Nueva Granada. Este documento exhibe una primera aproximación a la implementación y desarrollo de las teorías de Sistemas a Eventos Discretos.

Importancia en la industria

Se propuso un modelo matemático con el cual es posible evaluar dinámicamente el tiempo total acumulado de procesos para un sistema flexible, considerando las características de modelado de sistemas mediante RdP. Dicho modelo evalúa los tiempos de proceso por secuencia individual.

Se implementó un proceso utilizando los recursos de la UMNG, en el cual se pudo demostrar, mediante tiempos la secuencia óptima del sistema que se había modelado según la ecuación desarrollada. Esto fue posible debido a que el sistema implementado y modelado era considerablemente pequeño comparado con un sistema real de manufactura flexible.

B. Trabajos Futuros

Se hace alusión a que este trabajo es una primera aproximación a las teorías de RdP, Grafcet y Optimización, por lo que se plantea para trabajos posteriores la profundización de dichas teorías.

El modelo matemático planteado permite calcular el tiempo total dinámico de proceso para un sistema según una secuencia dada, sin embargo hace falta desarrollar ya sea mediante técnicas de optimización o sea mediante técnicas heurísticas, un algoritmo que permita encontrar la secuencia óptima dentro de un proceso industrial sin tener en cuenta el número de estados u operaciones que este pueda llegar a tener.

ANEXO 1: Software de Simulación

A1.1 QUEST y IGRIP:

Tanto Quest como Igrip pertenecen a la casa DENEb (The Digital Manufacturing Company) y están especialmente diseñados para la simulación de procesos industriales.

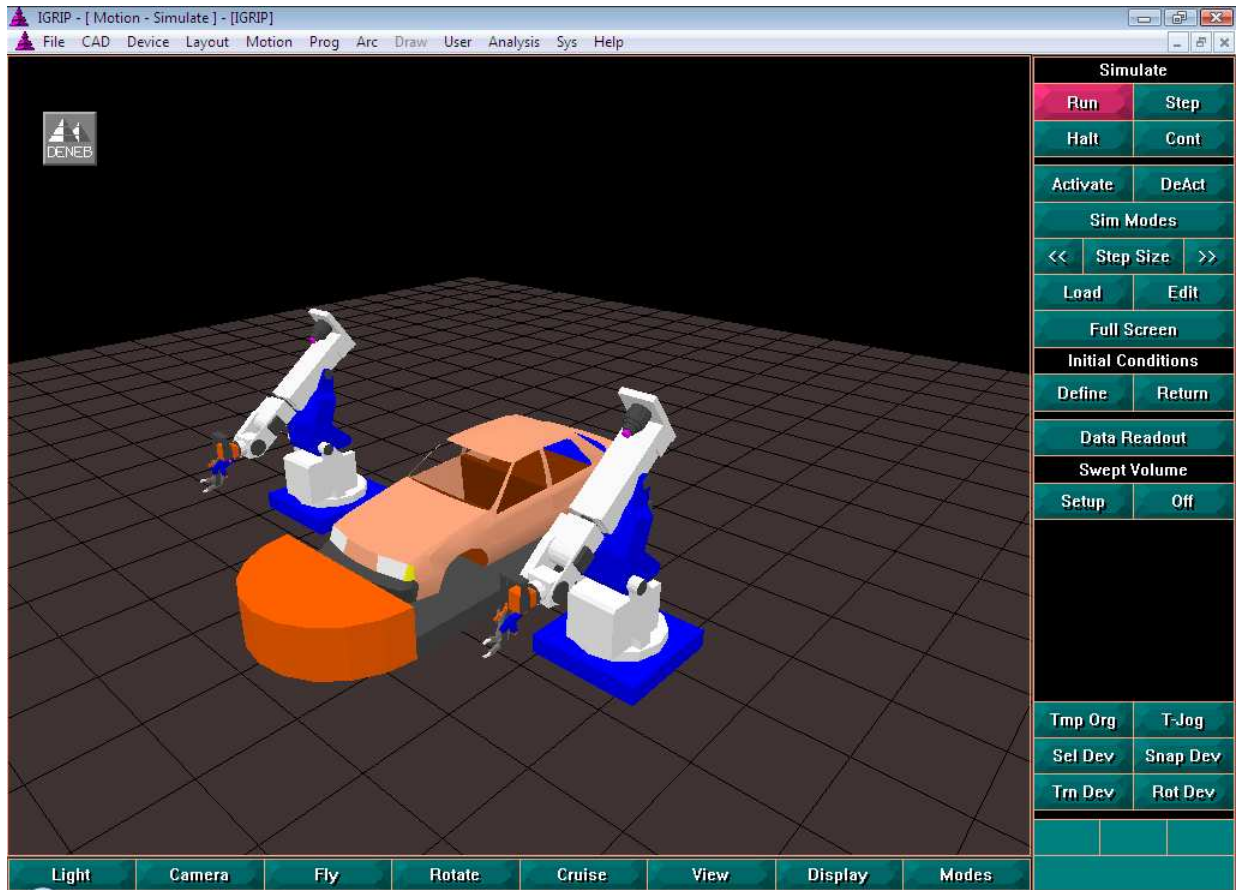


Fig. A1.1. Visualización del Software IGRIP

Estas herramientas de modelado 3D, son ideales para la simulación de eventos. En la figura A1.1 se puede visualizar la ventana de operación del software Igrip, en la cual se esta simulando un proceso de manufactura de automóviles, mientras que en la figura A1.2 se puede observar la simulación de un sistema mediante el software de Quest.

Es importante resaltar que con Igrip se puede visualizar con mayor facilidad los las operaciones de los procesos paso a paso, con Quest se evalúan los procesos a nivel macro, sin entrar al detalle de cada operación realizada dentro del sistema global.

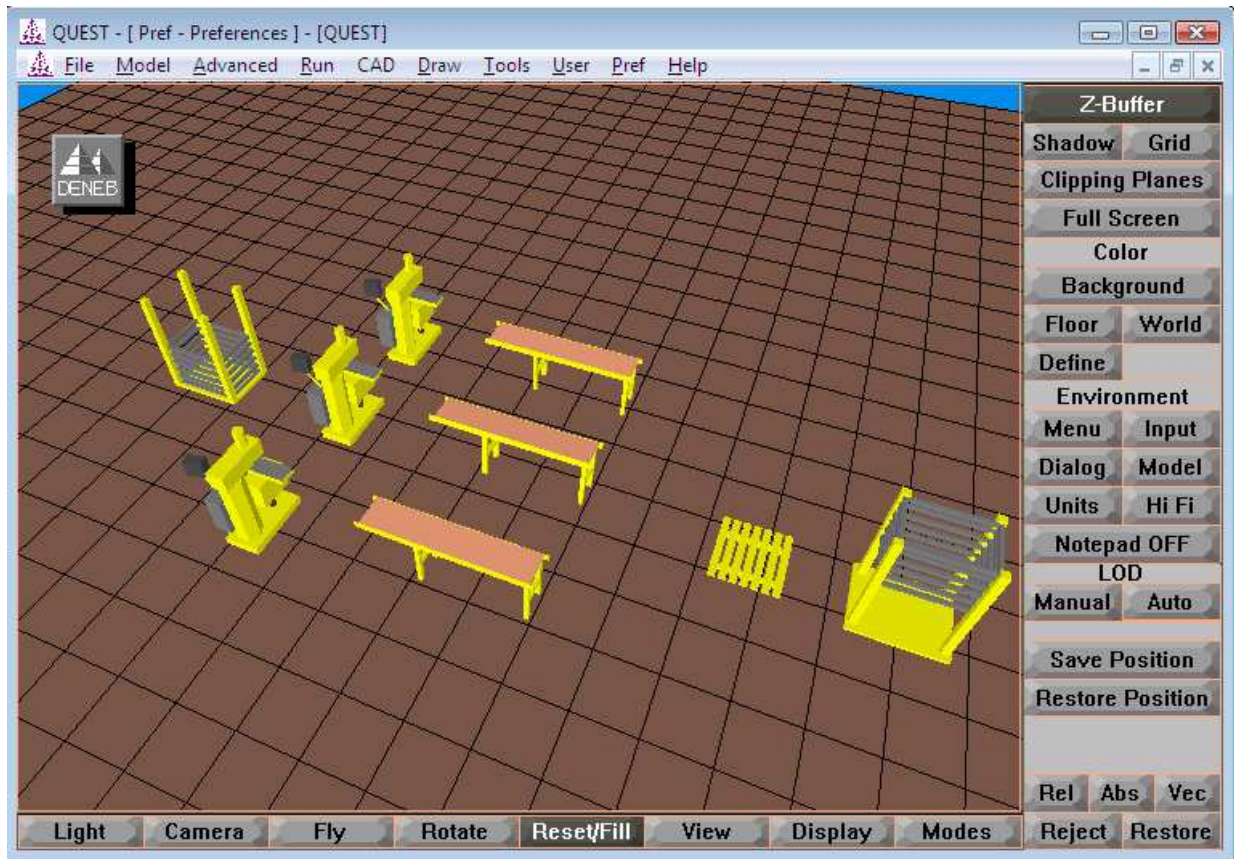


Fig. A1.2. Sistema modelado mediante QUEST

Para el desarrollo de los ejemplos trabajados a lo largo del documento, se trabajó con las licencias del Laboratorio de Realidad Virtual de la Universidad Militar Nueva Granda.

Si el lector está interesado en aprender a manejar este software de simulación 3D deberá adquirir su respectiva licencia, en la cual viene toda la documentación de ayuda que permite el aprendizaje de manejo del software de Igrip y de Quest. En este documento incluimos algunas referencias [21] [22] [23], que pueden complementar el estudio de estas herramientas de simulación.

A1.2 HILES DESIGNER, PM EDITEUR - PETRI NET - TOOLBOX MATLAB

Durante el capítulo 3, en el que se estudió la teoría relacionada con RdP, fueron utilizadas algunas herramientas de software para el modelo y análisis de las RdP implementadas.

Se trabajó HilesDesigner, un editor de RdP que conjuga el modelado de sistemas físicos dentro de su estructura interna, a fin de analizar el comportamiento intrínseco de los

mismos e identificando las características de funcionamiento de una determinada red modelada.

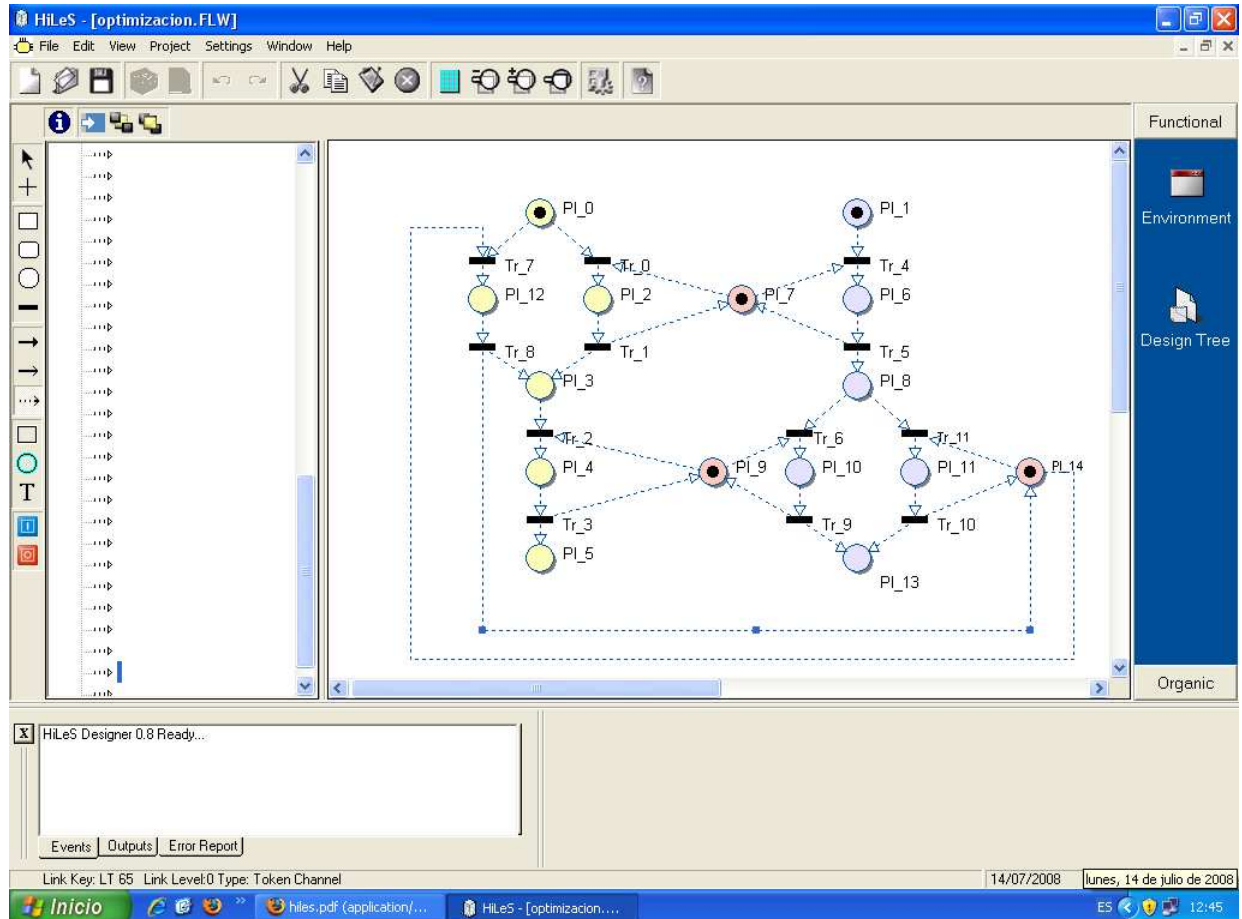


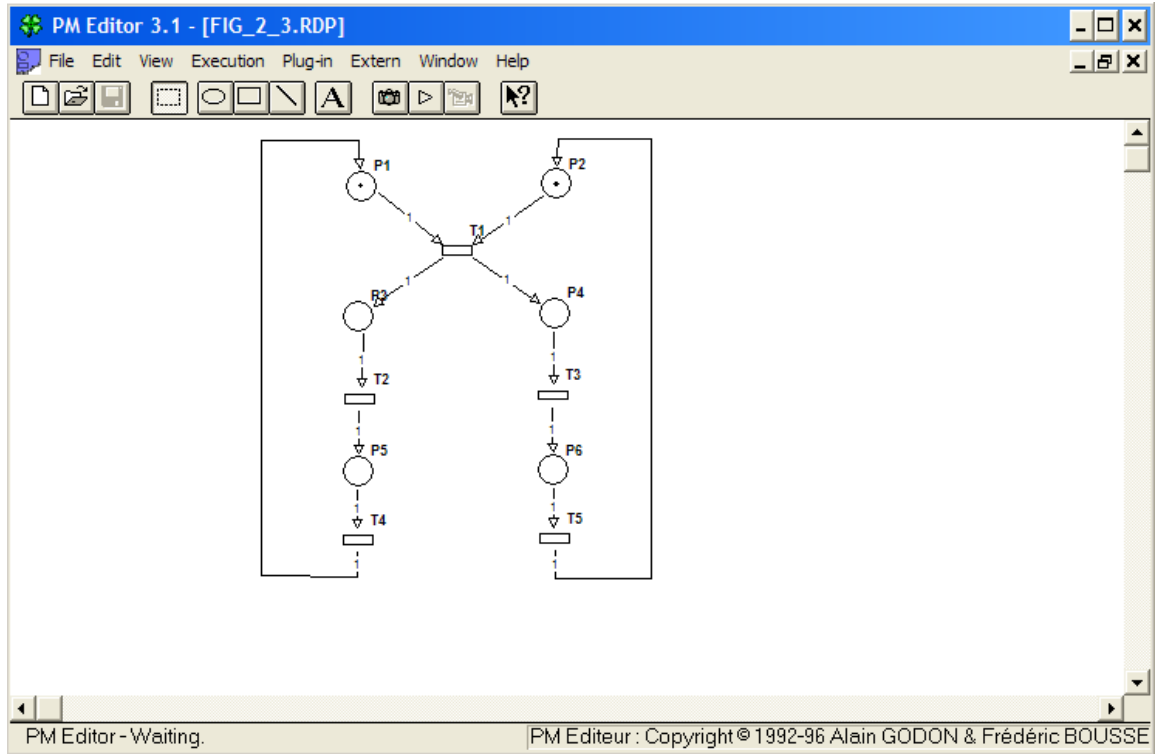
Fig. A1.3. Interfaz gráfica de HilesDesigner

HiLeS DESIGNER trabaja en conjunto con la herramienta TINA (Time petri Nets Analyzer), ambos desarrollados por el LAAS en Toulouse Francia [10] [11]. La figura A1.3 muestra el editor del Hiles Designer.

Otro editor de RdP es PM Editor 3.1 (Petri Maker Package) [12], que fue desarrollado por el departamento de informática de la Universidad de Hamburgo – Alemania. Este editor permite exportar las redes modeladas a Matlab mediante archivos '.rdp'. La figura A1.4 muestra la interfaz gráfica de este editor. Esta herramienta contiene internamente tutoriales que permiten identificar el funcionamiento básico el software.

También se trabajo con el toolbox de RdP para Matlab (Petri Net Toolbox) [13], desarrollado por el Departamento de Control Automático e Informática Industrial de la Universidad Técnica Gheorghe Asachi – Rumania. Mediante este Toolbox, Matlab

recibe los archivos .rdp y analiza las características de la RdP modelada en el PM Editor.



A1.4 Interfaz gráfica de PM Editeur

Este toolbox permite hallar la matriz de incidencia de entrada, la matriz de incidencia de salida, la matriz de incidencia total de la red, adicionalmente viene varias opciones de análisis para RdP, una vez instalado el toolbox, se puede acceder a la ayuda del mismo y consultar las diferentes opciones de análisis. En la web se encuentra información sobre el manejo del mismo, algunas aplicaciones implementadas y donde descargarlo, sin embargo, dentro de la bibliografía propuesta se han incluido algunos links para que el autor profundice en el manejo de esta poderosa herramienta de análisis de RdP. [27] [28] [29].

Existe también para Matlab una librería ejecutable para Simulink, que permite editar directamente la RdP desde Matlab, sin embargo en el desarrollo de este trabajo no se implemento dicha herramienta [29].

Cabe recalcar que el software implementado para el análisis de RdP que se utilizó en este proyecto es totalmente libre, todos pueden descargarse de la web. Claro está que para poder hacer uso del toolbox de redes de petri, será necesario tener acceso a la licencia de Matlab.

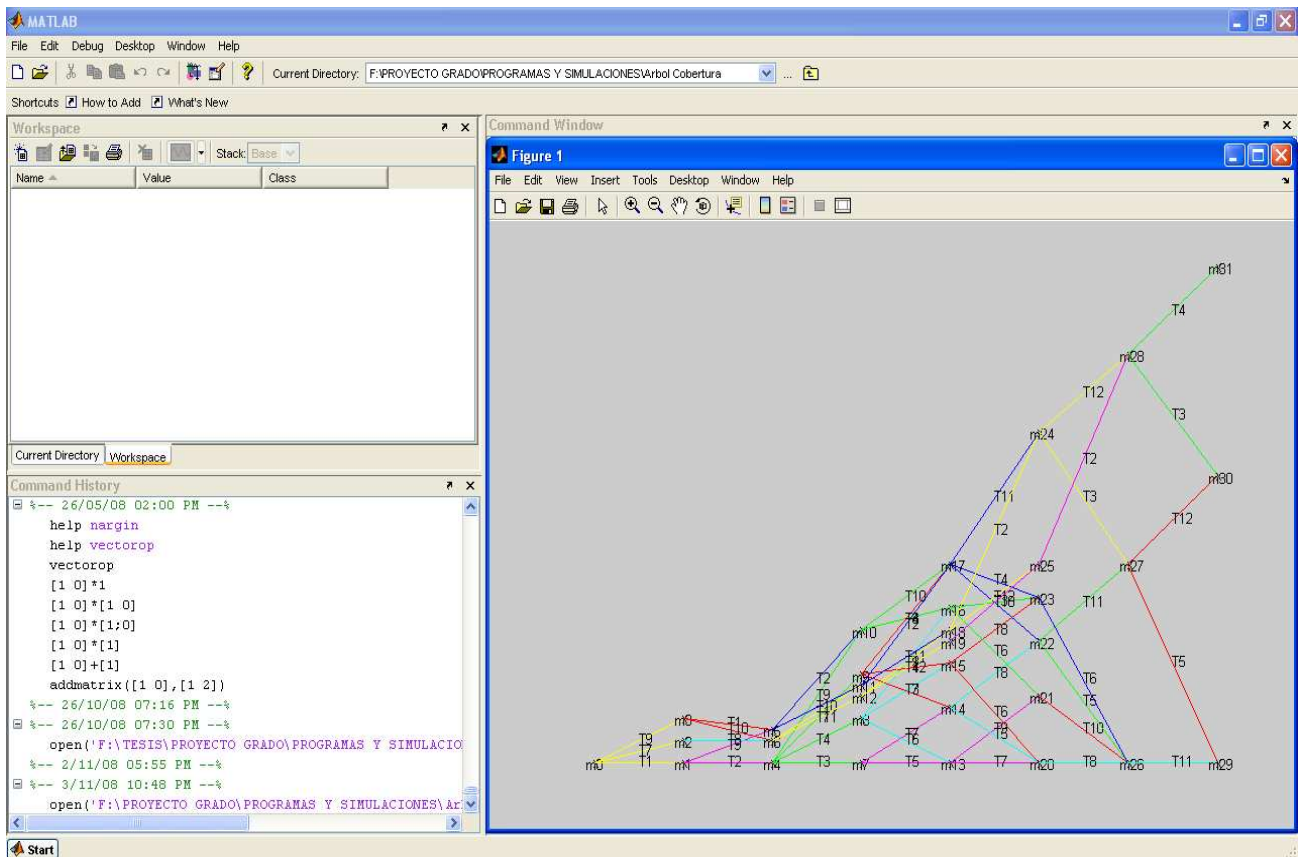


Fig. A1.5 Toolbox de Redes de Petri - Matlab

A1.3 SIMATICS – S7 SIEMENS - S7-GRAPH

Para la programación de los PLC de Siemens S7 – 300, se utilizó el software Simatic de Siemens. En el manejo de este software se debe contar con la licencia que viene en el momento de comprar el PLC S7 – 300. En este software se programan en diferentes lenguajes tales como ladder, listado de instrucciones, variables lógicas y GRAFCET, adicionalmente viene con un simulador interno que permite al usuario realizar las pruebas que sean necesarias antes de efectuar la implementación física.

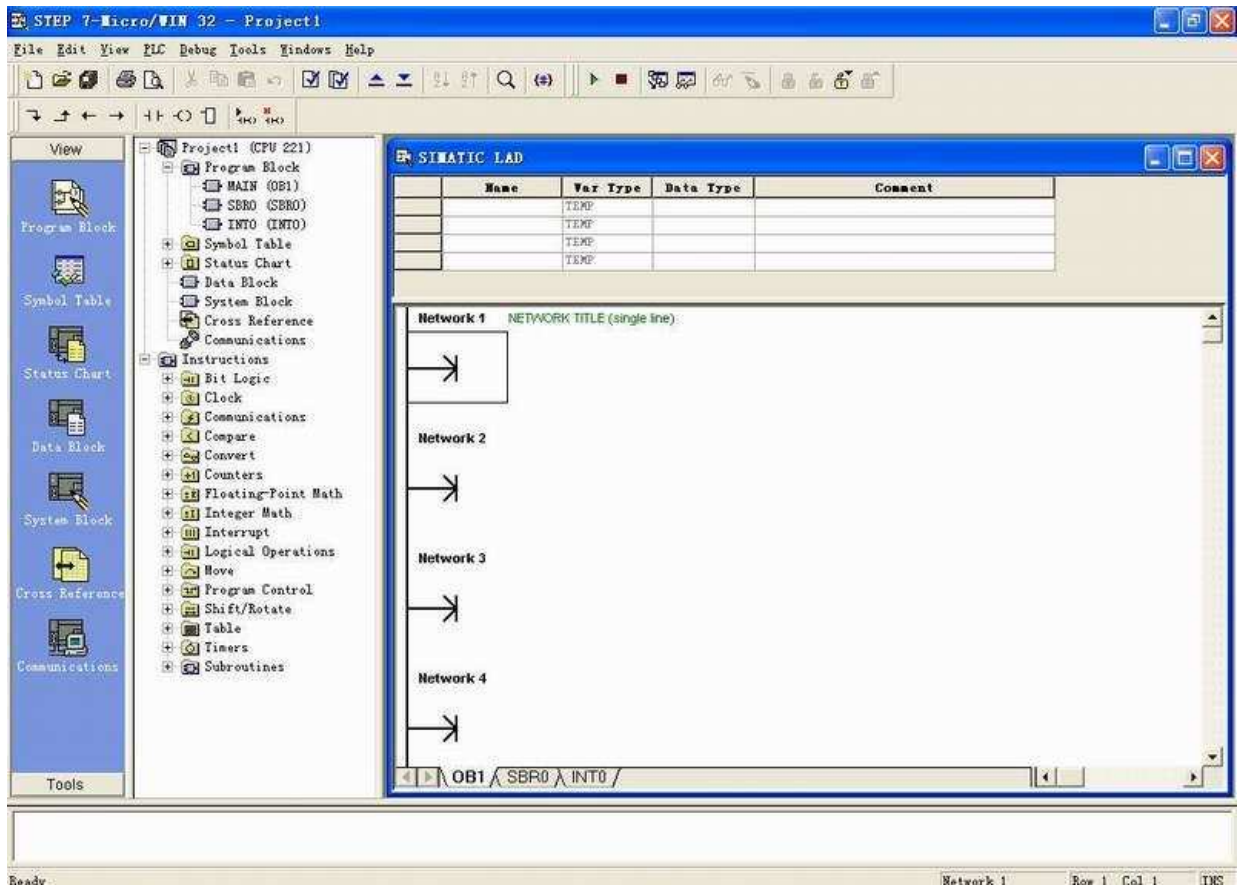


Fig. A1.6 Interfaz de Simatic de Siemens

Para la programación en GRAFCET debe utilizarse un módulo específico del Simatic llamado Step 7, para esto deberá crearse un módulo independiente dentro del proyecto, al cual se le especificará el lenguaje de programación, que para este caso particular será GRAFCET y posteriormente será llamado por el bloque de función principal.

Las figuras A1.6 muestra la interfaz del Simatic para la programación en lader, mientras que la figura A1.6 muestran la interfaz de programación en GRAFCET utilizando el módulo de STEP 7.

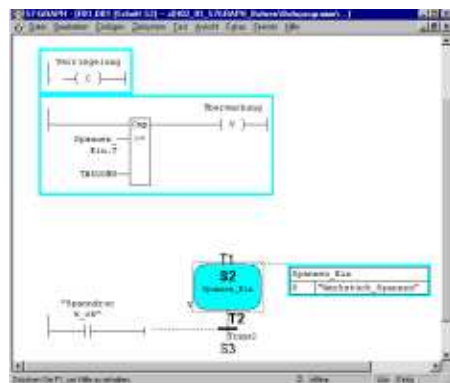
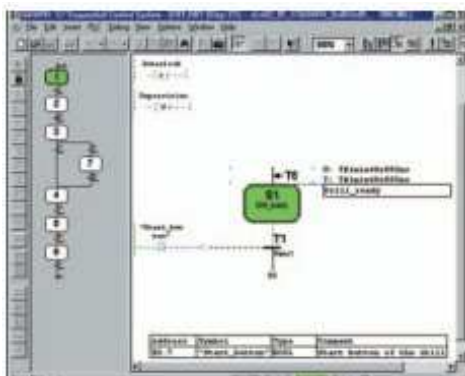


Fig. A1.7. Interfaz del S7 Graph de Siemens

En el anexo 3 se muestra y especifica los GRACET diseñados para el control del modelo hipotético planteado en el capítulo 6. Si el lector está interesado en aprender a manejar esta herramienta, deberá dirigirse a la ayuda que viene implícita en el software. En la web también se encuentran algunos tutoriales sobre el manejo del mismo [24].

A1.4 FLUIDSIM

FluidSim es un software especializado en la simulación de sistemas neumáticos, tal como se aprecia en la Fig. A1.7. Para conocer y aprender su funcionamiento se pueden consultar los tutoriales que vienen internamente en el programa y complementar con la información que se encuentra en la web [25] [26].

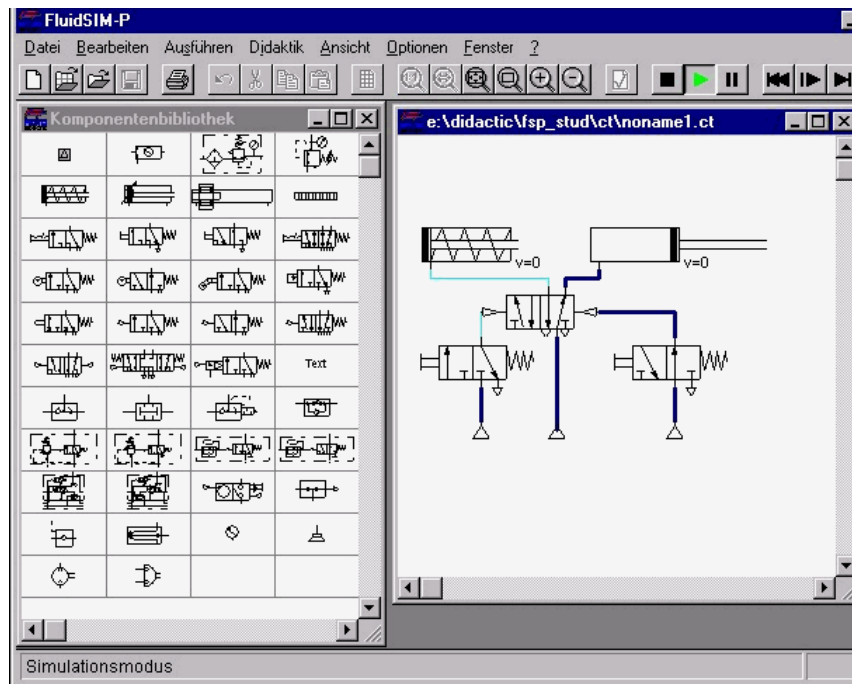
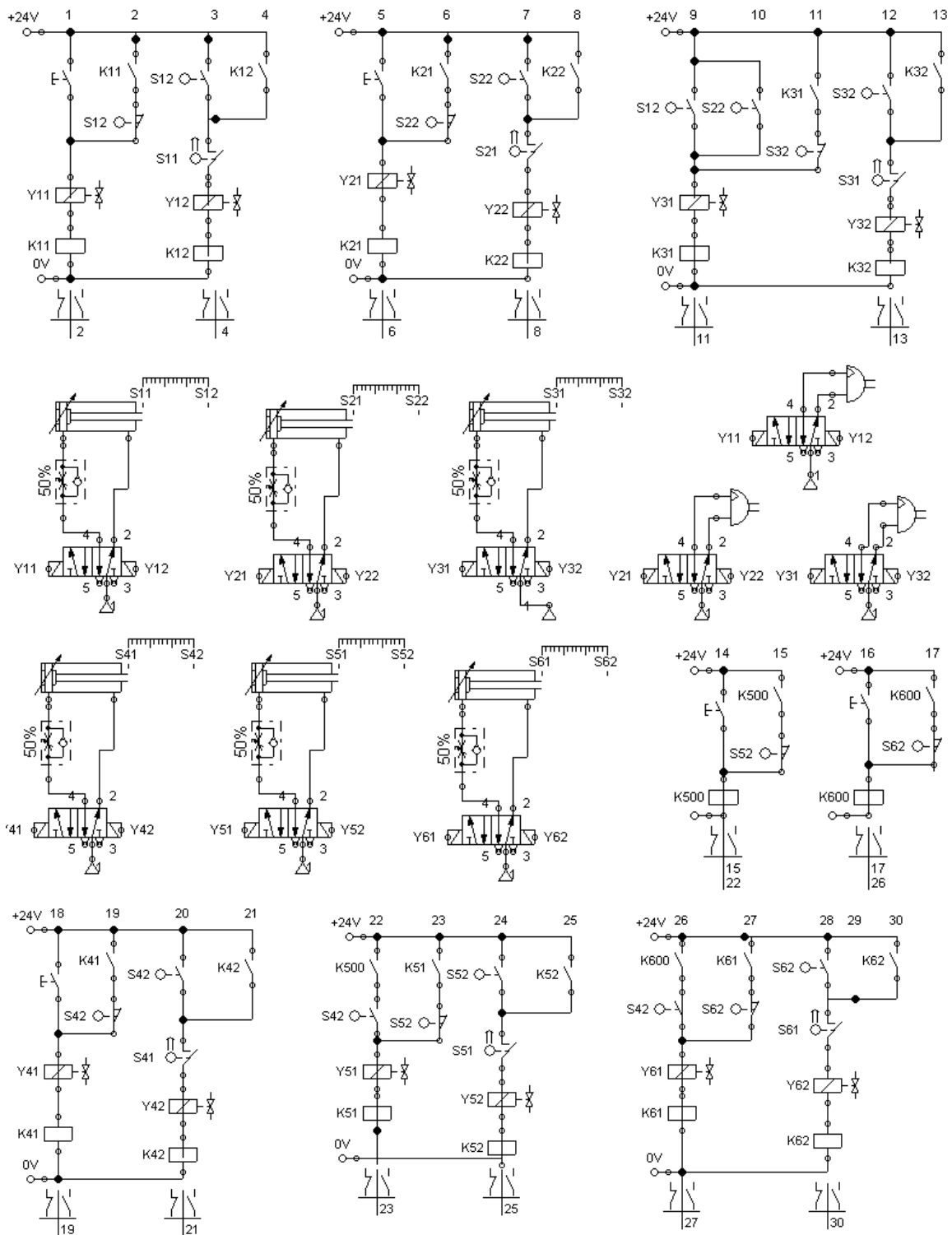


Fig. A1.8 Interfaz gráfica de FluidSim

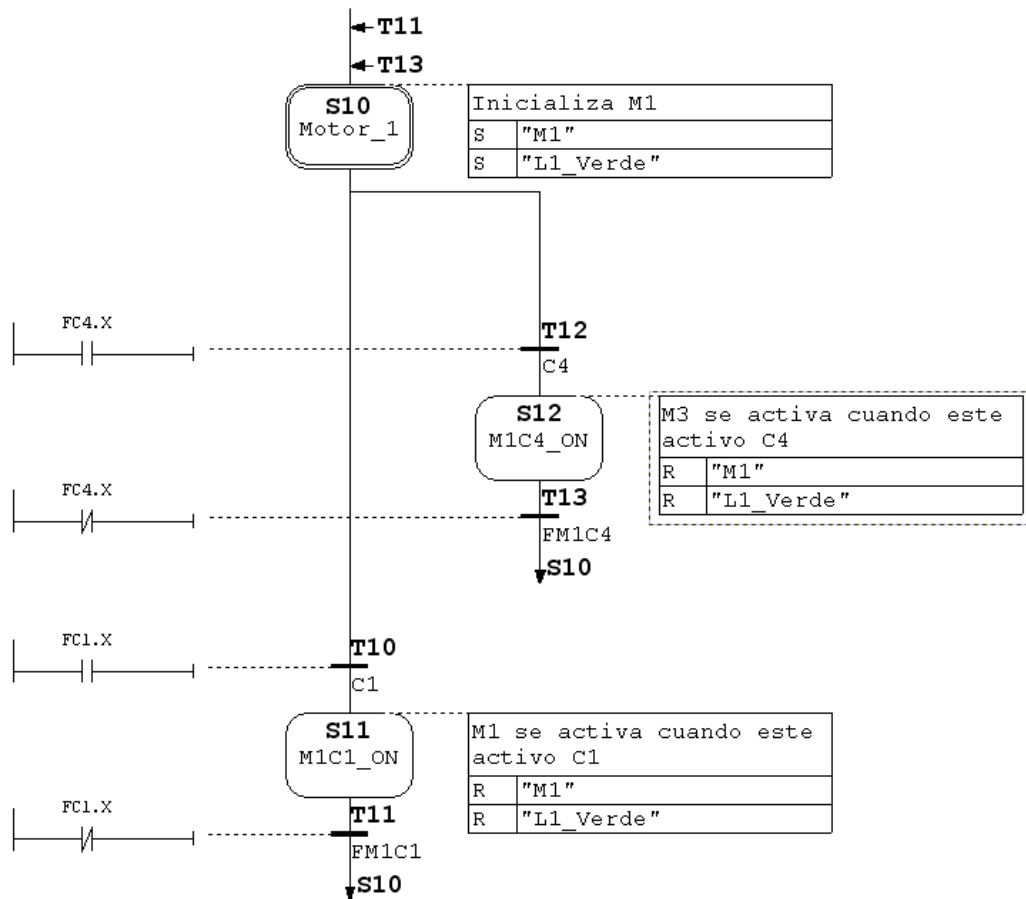
ANEXO 2: Simulación neumática con FluidSim



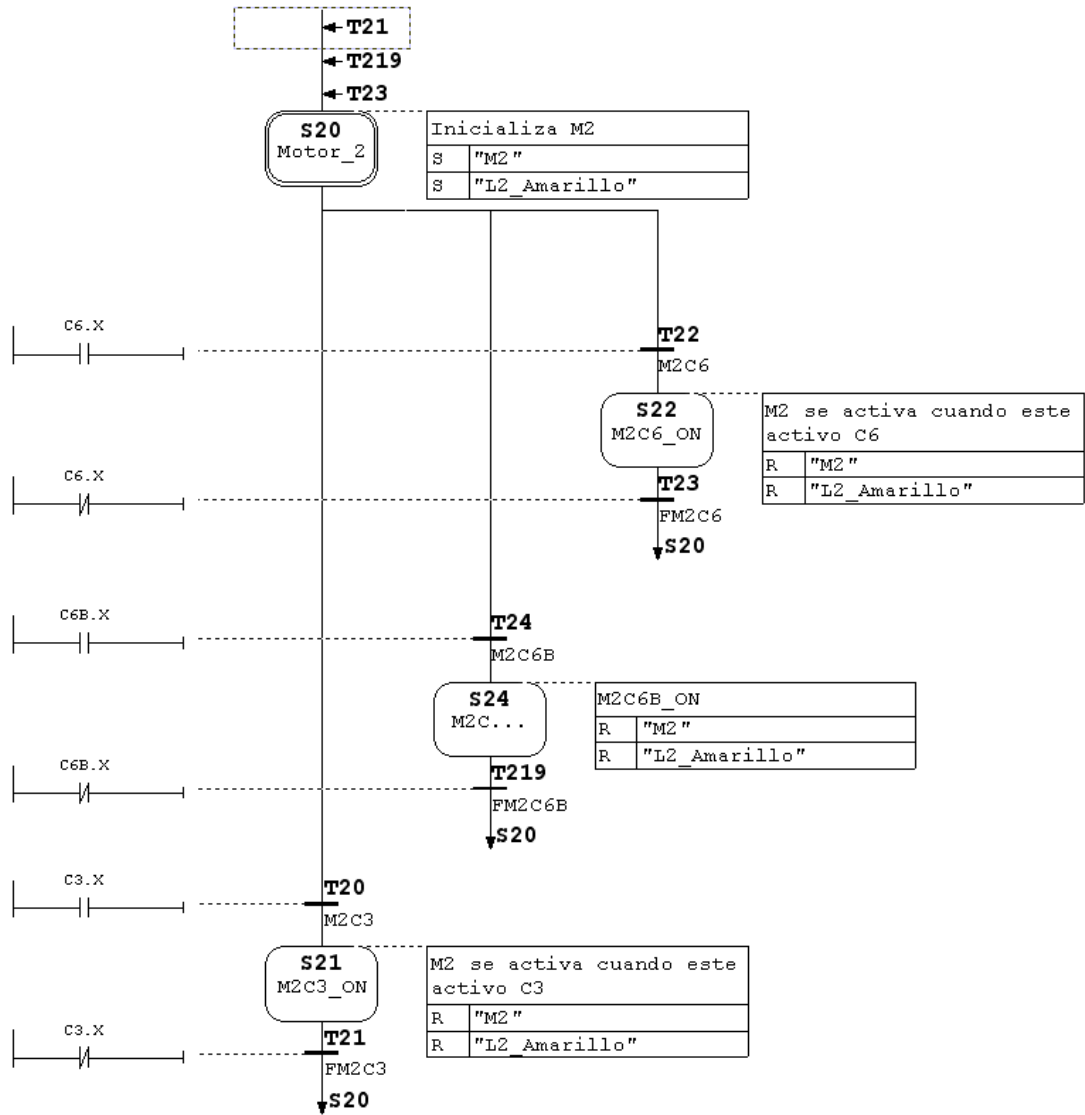
ANEXO 3: Graficet de control para el sistema físico implementado

MODO AUTOMÁTICO

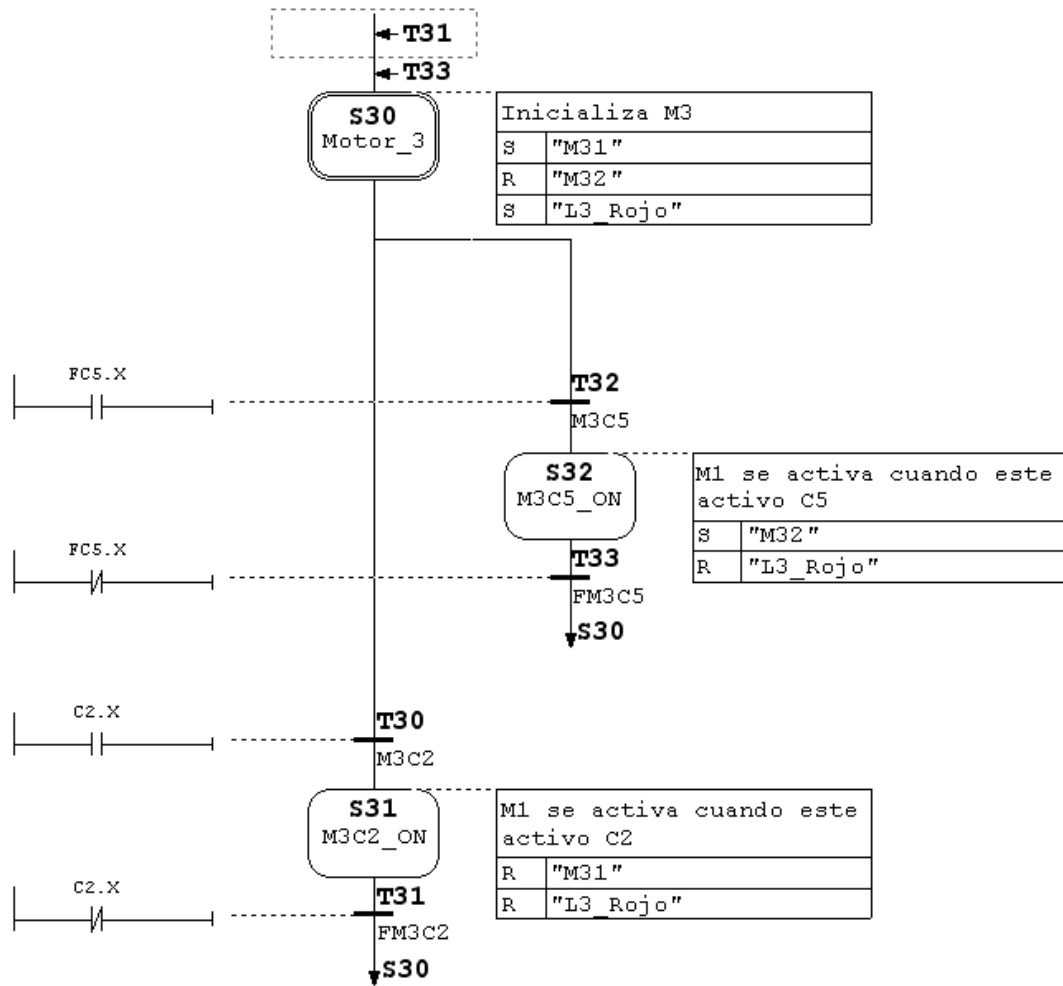
GRAFICET Máquina 1:



GRAFSET Máquina 2:

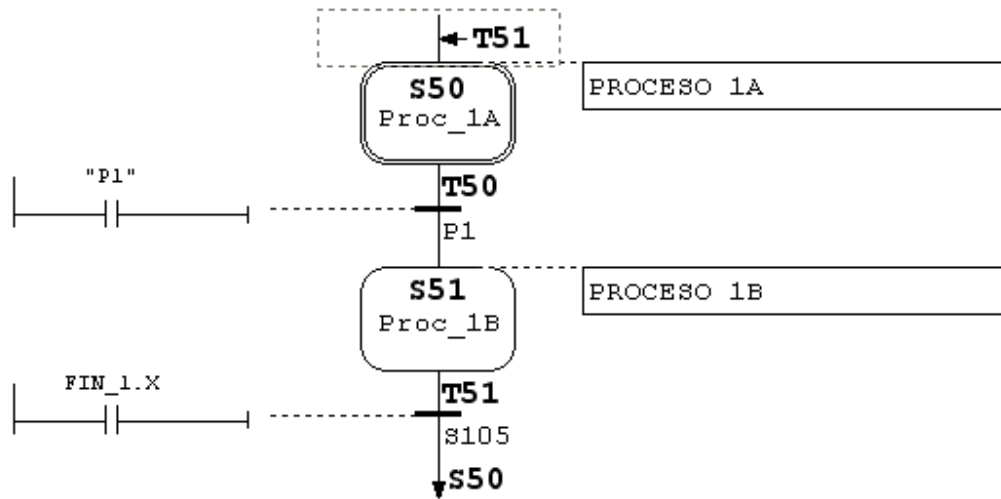


GRAFSET Máquina 3:

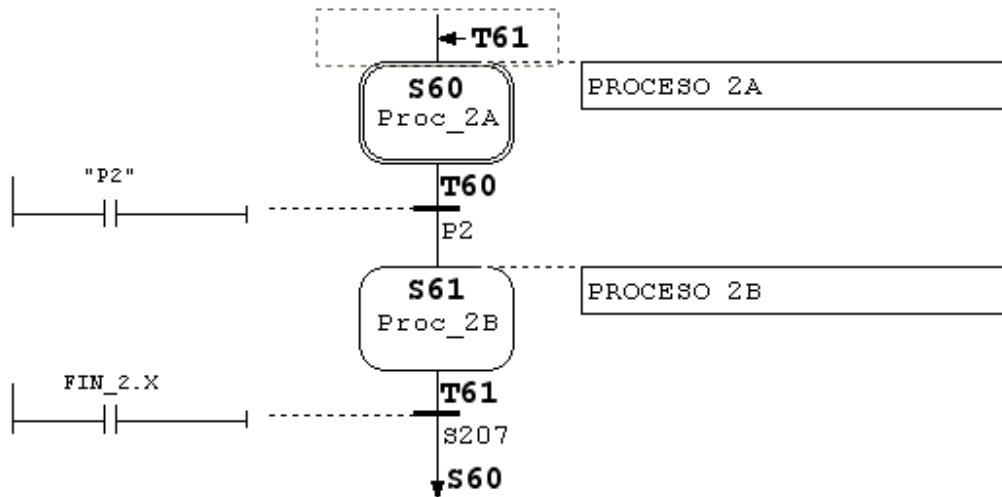


GRAFICET de Selección de Secuencia:

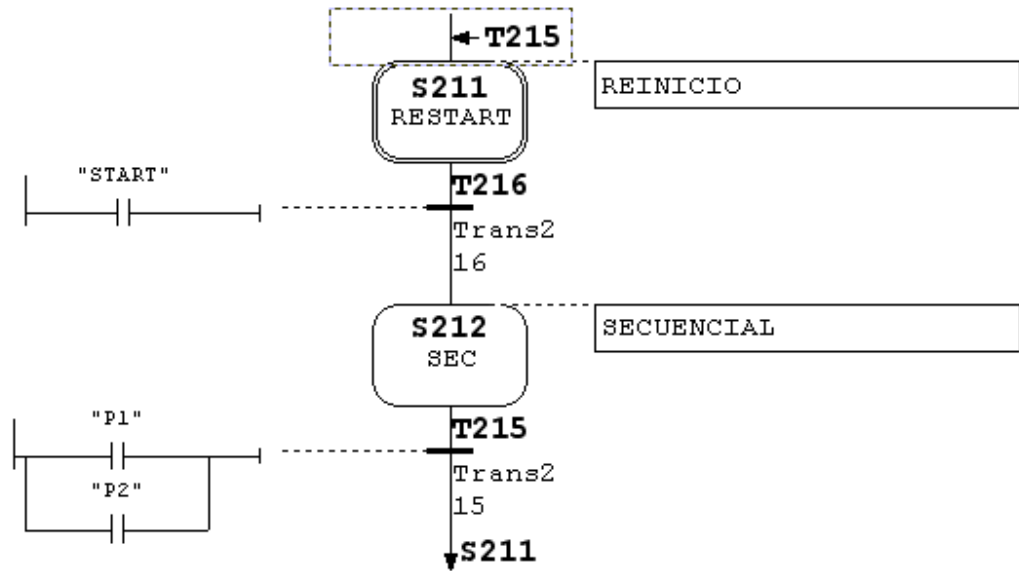
Proceso 1:



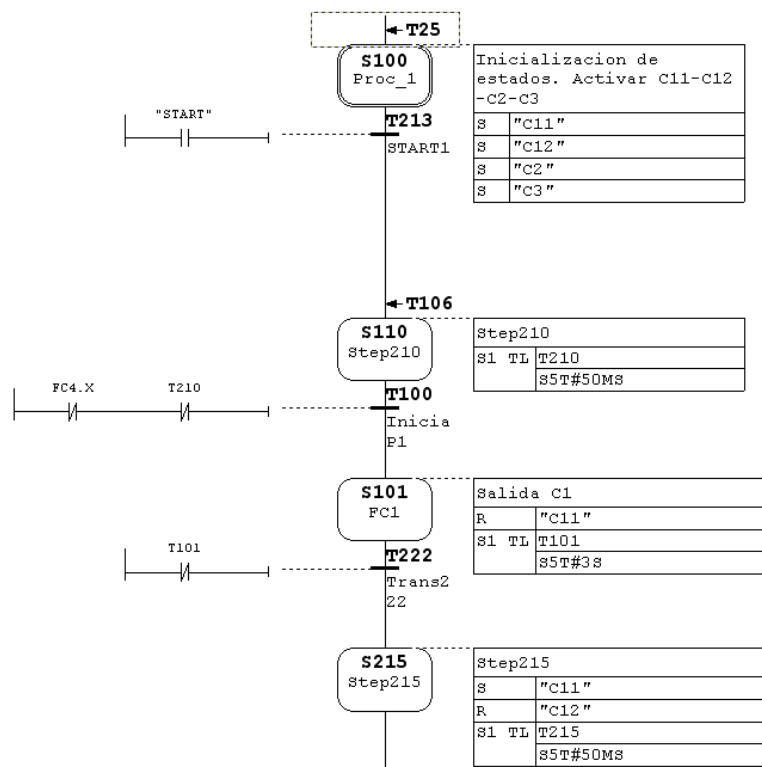
Proceso 2:

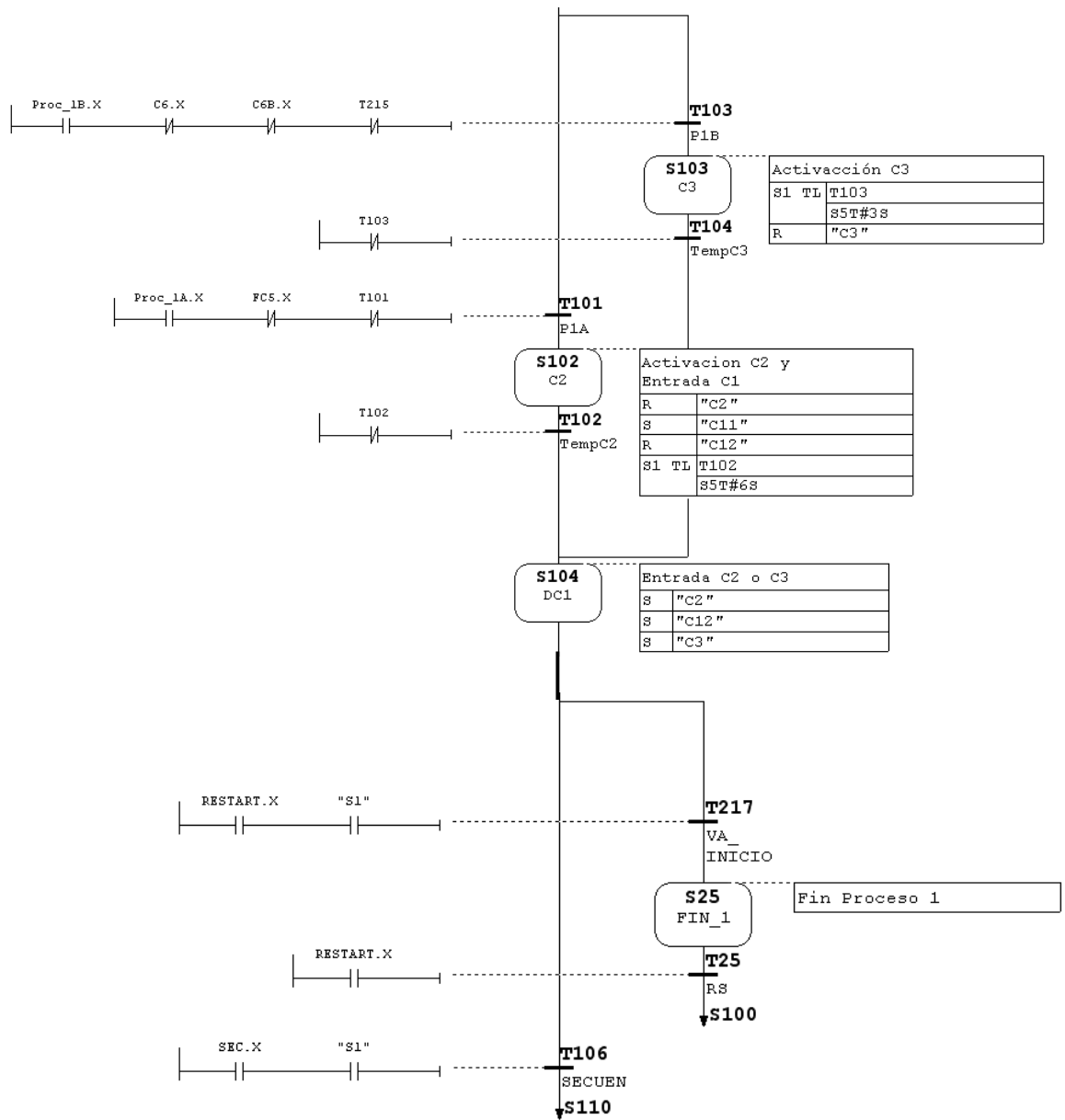


GRAFSET Parada Secuencia Automática:

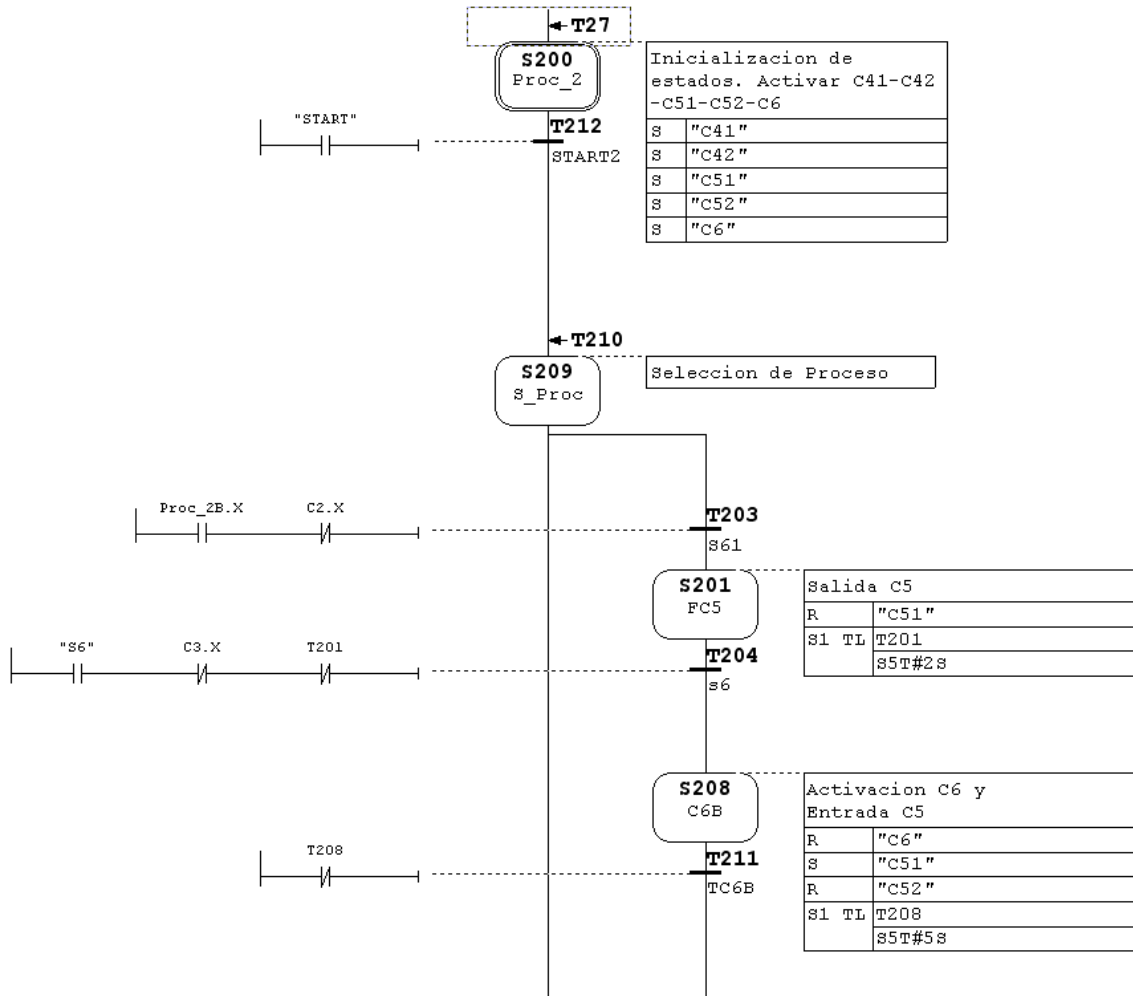


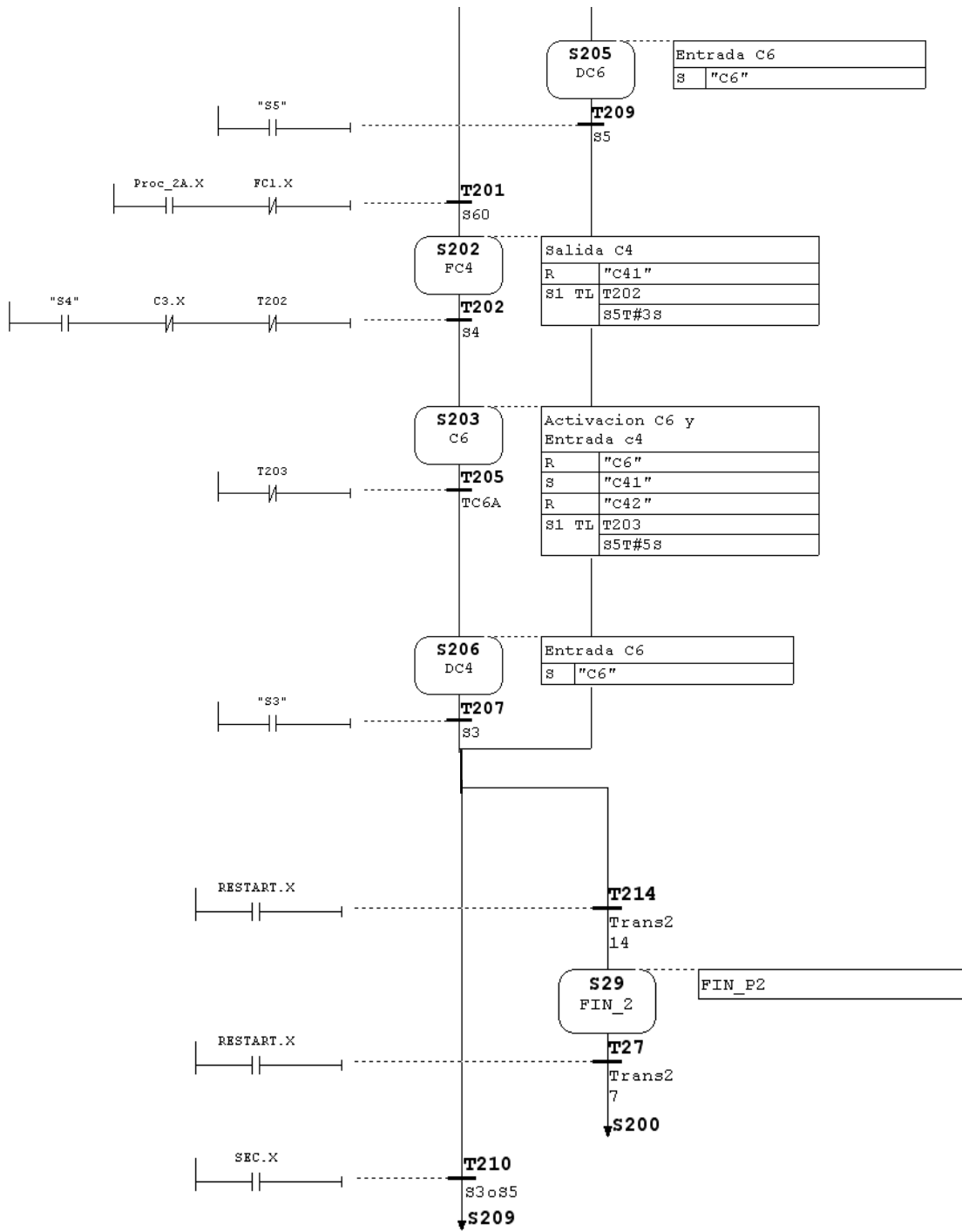
GRAFSET Proceso 1:



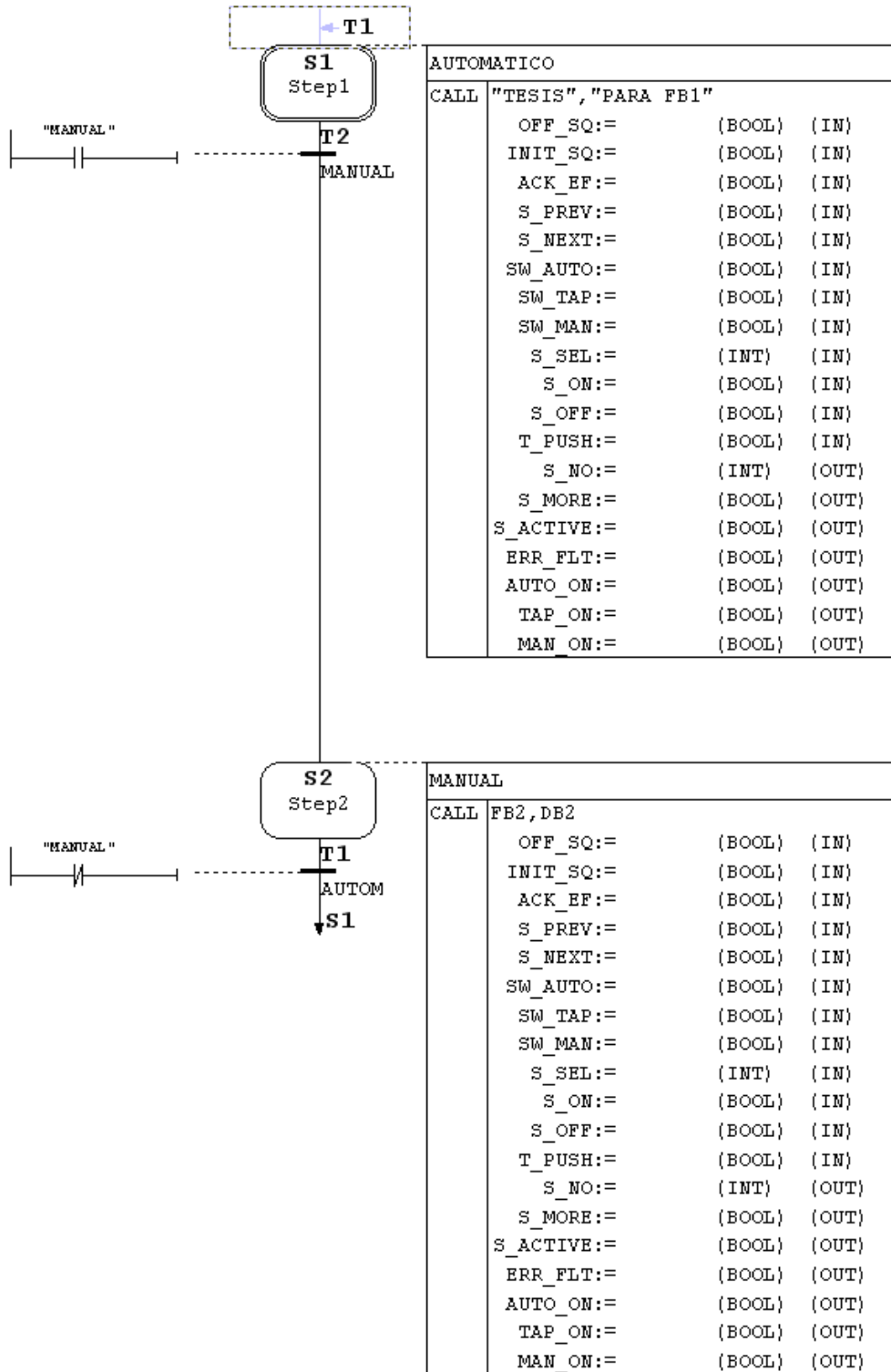


GRAFSET Proceso 2:



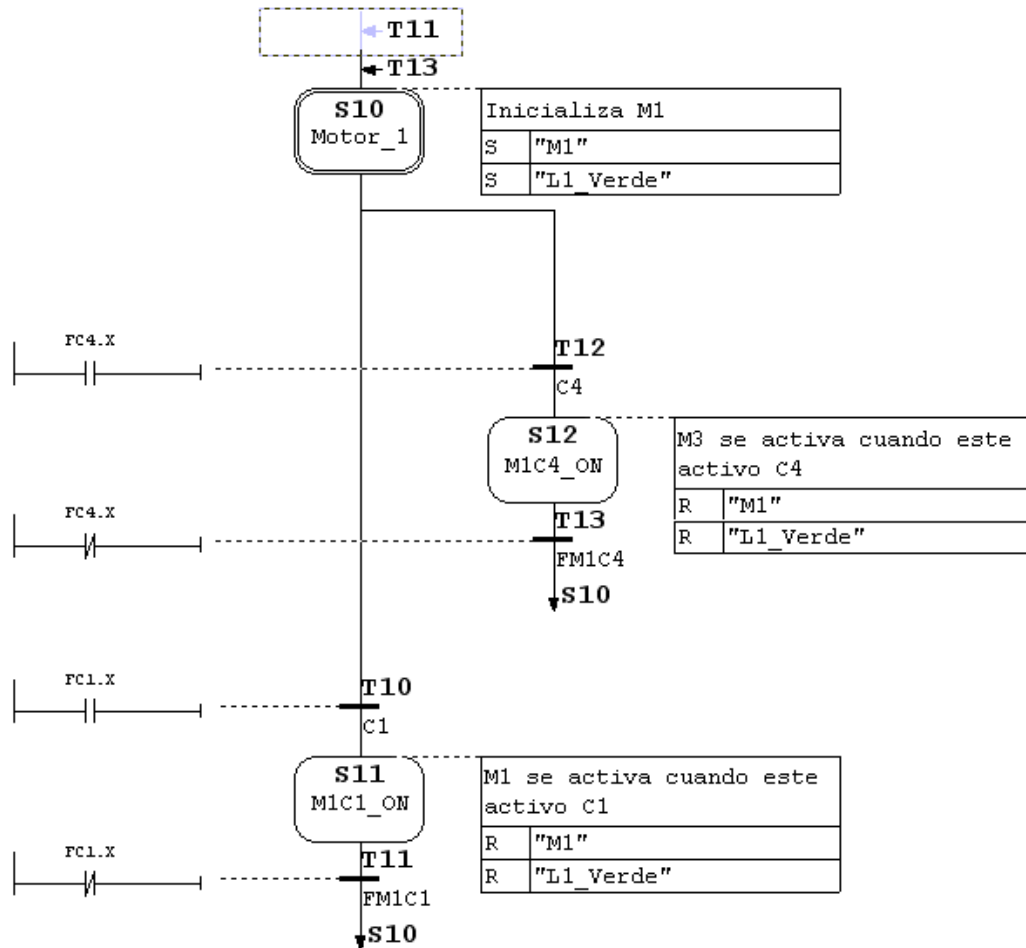


GRAFSET Selección de Modo (Automático - Paso a Paso)

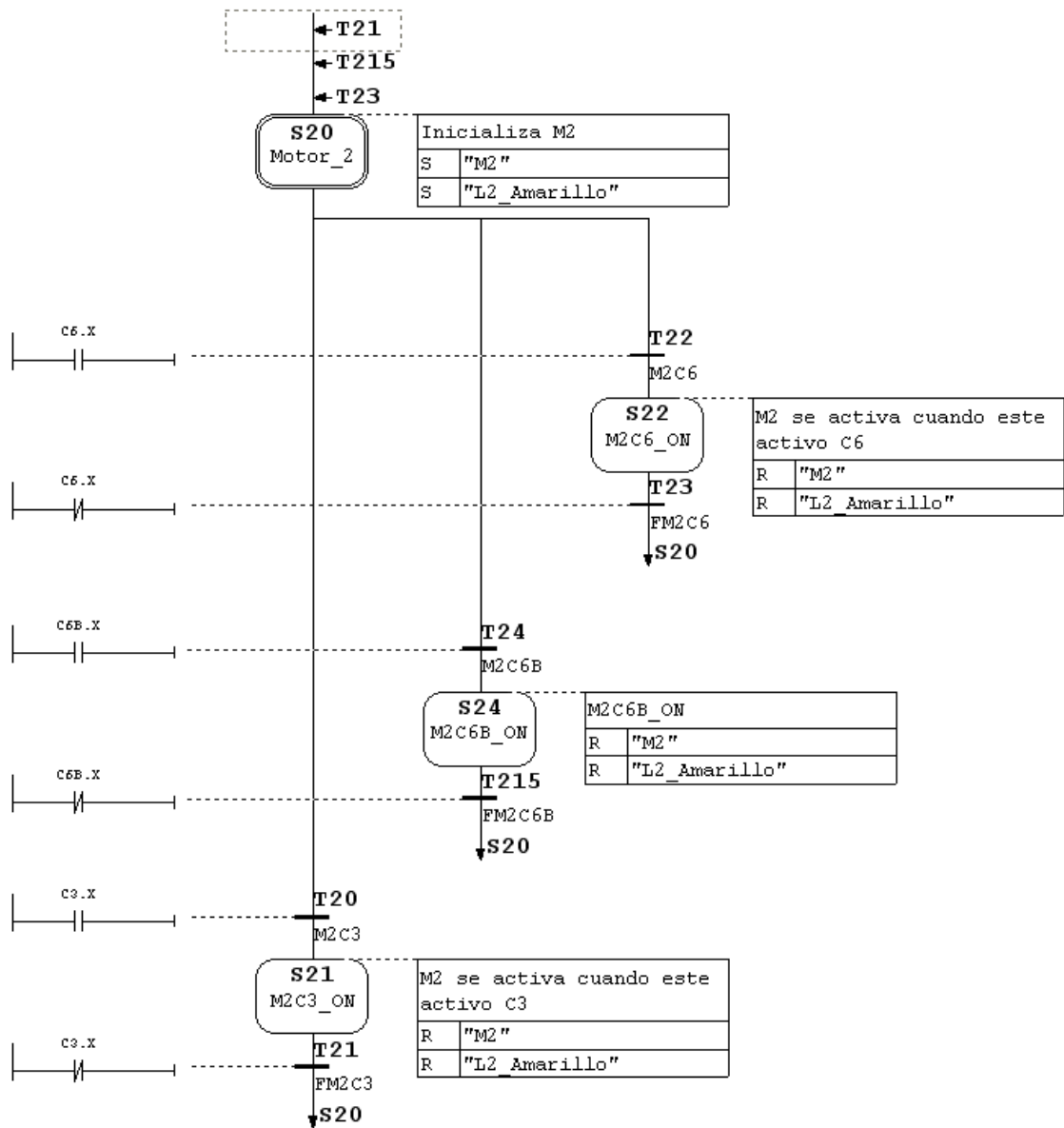


MODO MANUAL (Paso a Paso)

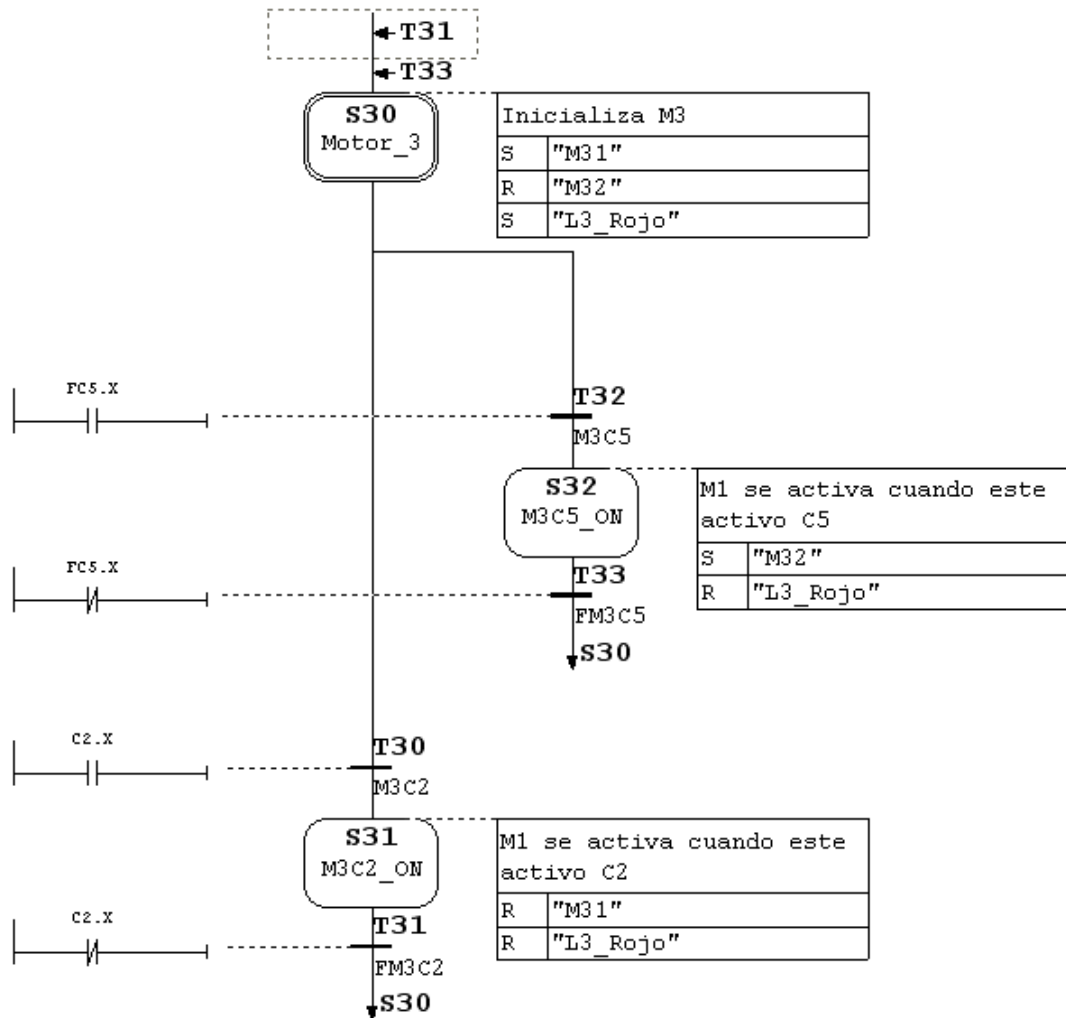
GRAFSET Máquina 1:



GRAFSET Máquina 2:

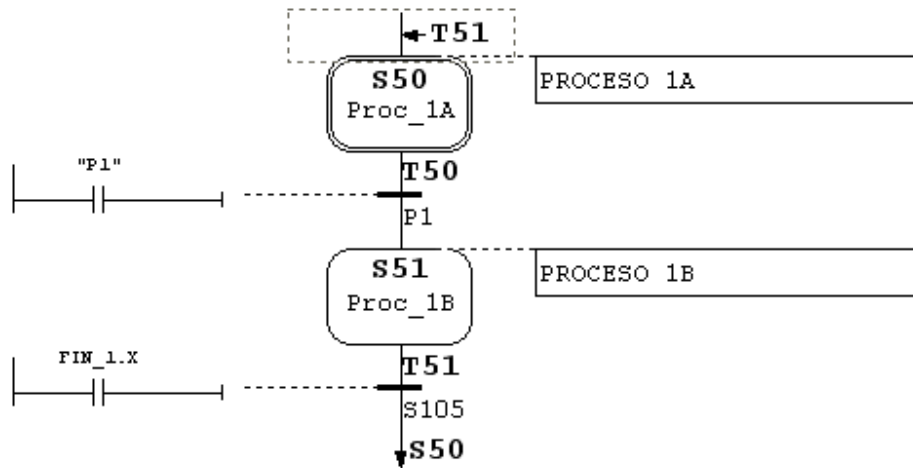


GRAFSET Máquina 3:

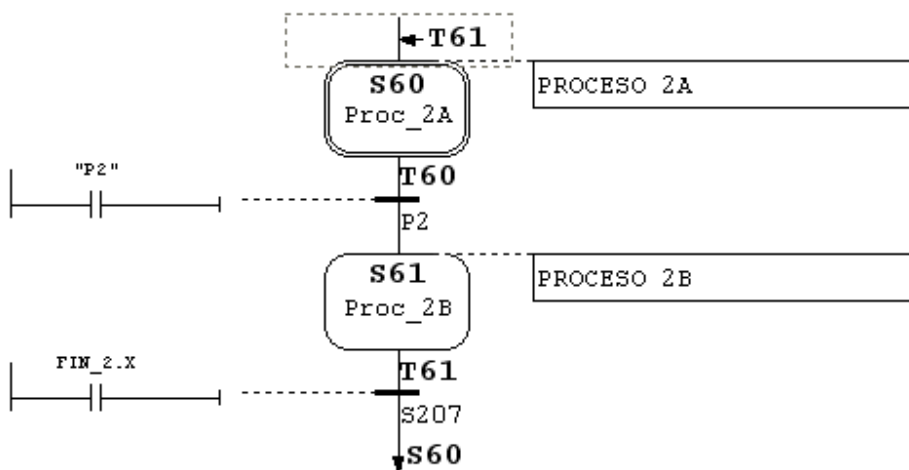


GRAFSET de Selección de Secuencia:

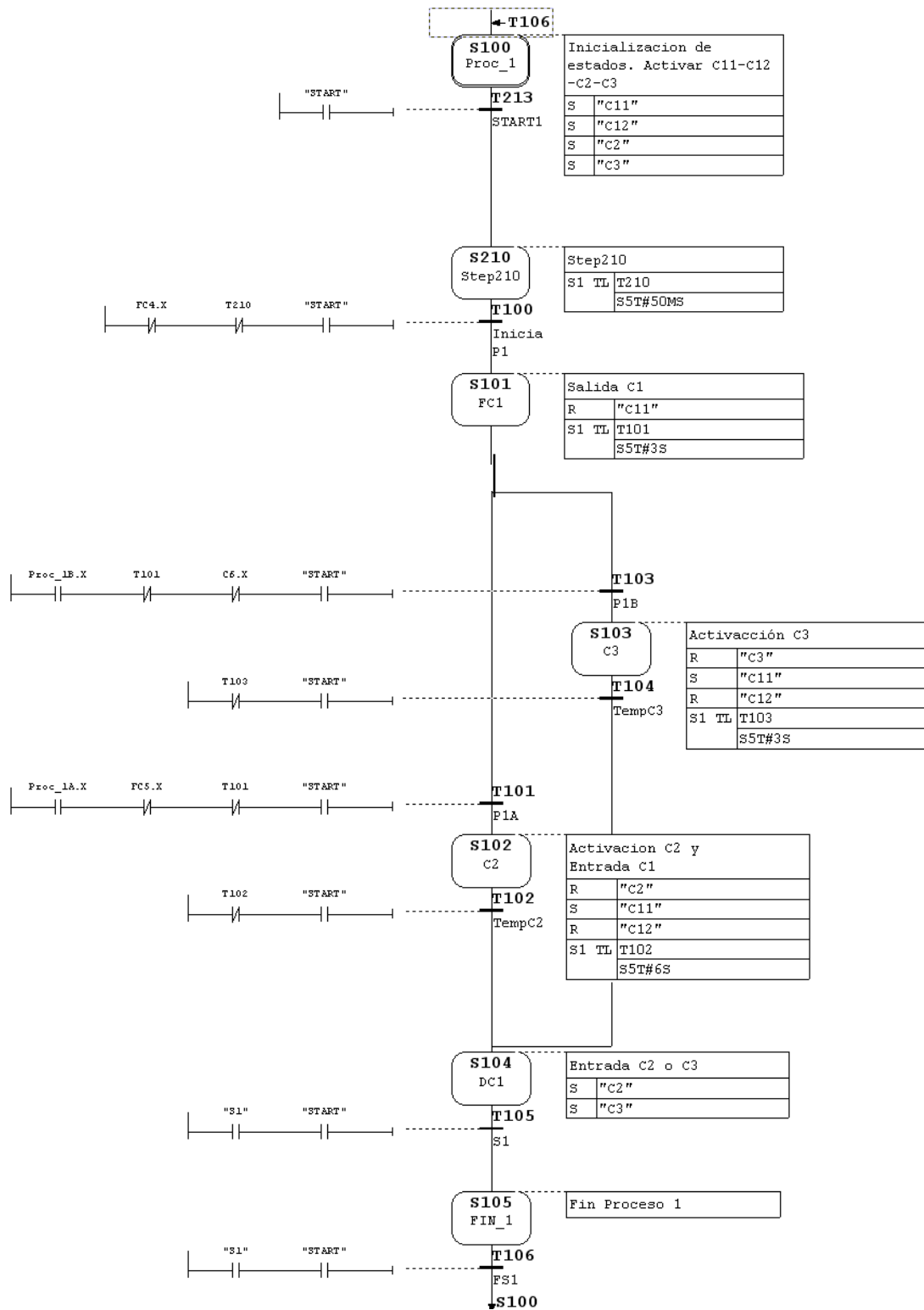
Proceso 1:



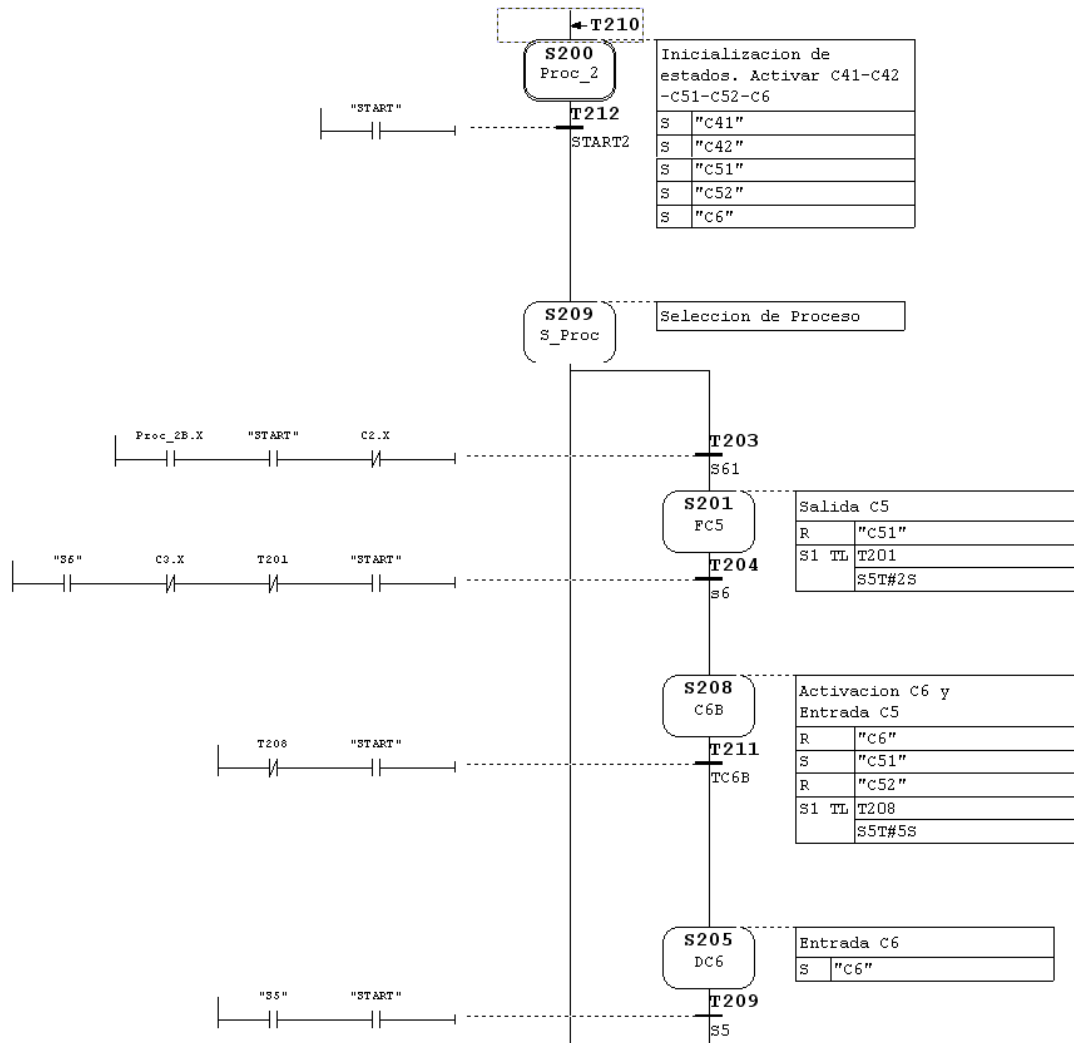
Proceso 2:

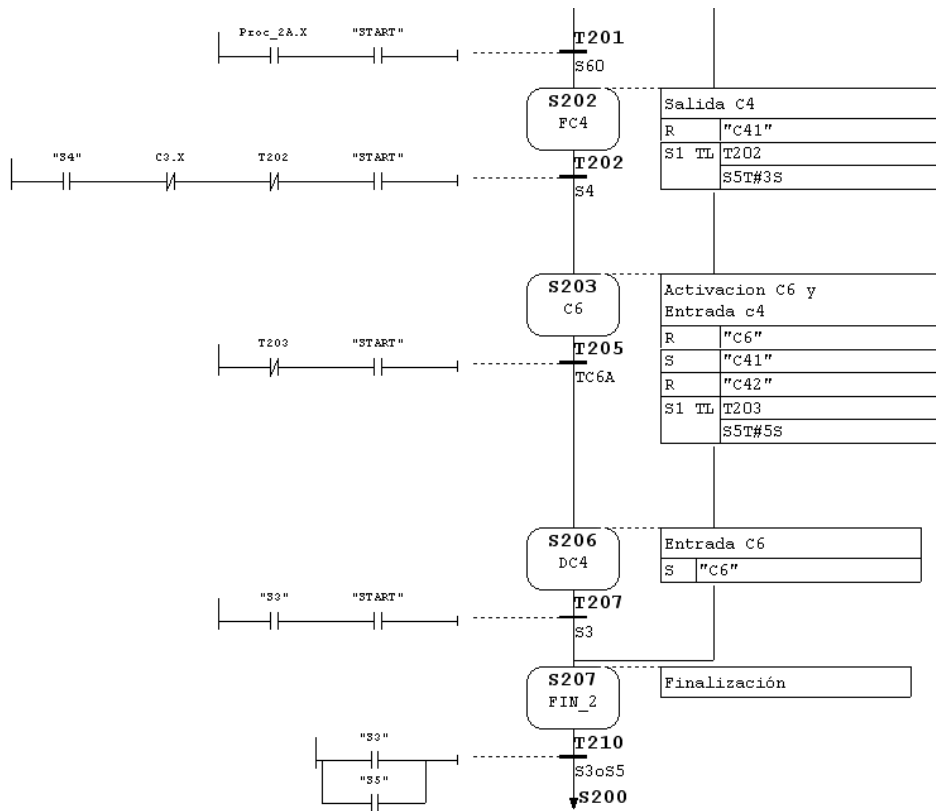


GRAFSET Proceso 1:



GRAFSET Proceso 2:





BIBLIOGRAFIA

- [1] D. E. Kirt. *Optimal Control Theory. An Introduction*. Ed. Prentice-Hall, INC. 1970. CCCN 74-161636
- [2] D. G. Luenberger. *Programación Lineal y no Lineal*. Standford University, California. Versión en español de Manuel López Mateos. Universidad Nacional Autónoma de México. Ed. Addison Wesley Longman. ISBN 968-444-362-5
- [3] Tadao Murata. *Petri Nets: Properties, Analysis and Applications*. Proceedings of the IEEE, Vol. 77, No. 4, April 1989.
- [4] J. Balcells. J.L. Romeral. *Autómatas Programables*. Ed. Alfaomega marcombo 1998. ISBN 970-15-0247-7
- [5] R. Simarro. *Apuntes de clase Automatización Industrial*. Universidad Politécnica de Valencia. personales.upv.es/rausifer/
- [6] PETRI NETS 2000. 21st International Conference on Applications and Theory of Petri Nets. Aarhus, Denmark, June 26-30, 2000.
- [7] Liu, C.S., Ma, Y.H., y Odrey, N.G. *Hierarchical Petri Net Modeling for System Dynamics and Control of Manufacturing Systems*. International Conference on Flexible Automation Intelligent Manufacturing (FAIM.97), Sponsored by University of Teesside, Middlesbrough, UK, June 25-27, 1997.
- [8] Mu Der Jeng, Reyu Wen Jaw, Pen Li Hung. *Scheduling FMS with Due Dates Based on Petri State Equations*. Proceedings of the 1997IEEE. p 2724, 5p
- [9] H. Yu, A. Reyes, S. Cang and S. Lloyd. *Combined Petri net modeling and AI based heuristic hybrid search for flexible manufacturing systems*. Part 1 Petri net modeling and heuristic search. Computers & Industrial Engineering, 44(2003) 527-543
- [10] Toolsys Hiles Designer. Laboratorio LAAS – Toulouse Francia. <http://www.laas.fr/toolsys/hiles.htm>
- [11] TINA. Laboratorio LAAS – Toulouse Francia. <http://www.laas.fr/tina/>
- [12] Petri Maker Editor. Departamento de Informática – Universidad de Hamburgo Alemania. <http://www.informatik.uni-hamburg.de/TGI/PetriNets/>
- [13] Petri Net Toolbox for Matlab. Departamento de Control Automático e Informática Industrial. Universidad Técnica Gheorghe Asachi – Rumania. <http://www.ac.tuiasi.ro/pntool/>

- [14] A. L. Castro Lopez y G. Mejía. *Combinación de Métodos Heurísticos con Redes de Petri para la Programación de Sistemas de Manufactura Flexible*. Departamento de Ingeniería Industrial. Universidad de los Andes. Bogotá – Colombia.
- [15] E. Vallejo. *Sistemas Flexibles de Manufactura*. Ingeniería & Desarrollo. Universidad del Norte. 3-4: 43–49, 1998. Barranquilla – Colombia.
- [16] G. Mejía. and N.G. Odrey. *Petri Net Models and Heuristic Search for scheduling of Manufacturing Systems: A comparative Study*. Departamento de Ingeniería Industrial. Universidad de los Andes. Bogotá – Colombia. Departamento de Ingeniería Industrial y de Sistemas. Universidad de Lehigh. Bethlehem, PA, USA.
- [17] Varios autores. GEMMA (Guide d'Étude des Modes de Marches et d'Arrêts). ADEPA (Agence pour le Développement de la Productique Appliquée), 13/17 Rue P{erier, BP 5492123, Montrouge, Cedex.
- [18] Curso web sobre GRAFCET y GEMMA. Informática Industrial II - Universidad de Oviedo. Oviedo – España. <http://www.isa.uniovi.es/~vsuarez/ii2/index.htm>.
- [19] G. Mejía. *Timed Petri net modeling and optimization with heuristic search for flexible manufacturing workstations*. Emerging Technologies and Factory Automation 2003. IEEE Conference Proceedings. ETFA apos;03. Volume 1, Issue , 16-19 Sept. 2003 Page(s): 211 - 217 vol.1 DOI 10.1109/ETFA.2003.1247708.
- [20] Plantas Químicas de España. Proceso de pasteurización de la leche. España. <http://plantasquimicas.iespana.es/Agroindustria/dpa19.htm>
- [21] M. Laguna. *Metaheuristic Opimization with Evolver, Genocop and OptQuest*. Graduate School of Business Administration. Colorado University, Boulder, CO 80309-0419. 1997.
- [22] L. I. Goldman, H. Emmett. *Crystal Ball for Six Sigma Tutorial*. Proceedings of the 2003 Winter Simulation Conference.
- [23] M. R. Barnes. *An Introduction to Quest*. Proceedings of the 1997 Winter Simulation Conference. Deneb Robotics, Inc. U.S.A
- [24] http://www.automation.siemens.com/simatic/industriesoftware/html_76/products/simatic-s7-graph-maintenance.htm. *S7 Graph Siemens*.
- [25] *FluidSim*. <http://www.artelab.uni-bremen.de/Projects/eugabe/KonzeptReality.html>.
- [26] *FluidSim*. http://www.fluidsim.de/fluidsim/index4_e.htm.

- [27] M. H. Matcovschi, C. Mahuela, O. Pastravanu. *Petri Net Toolbox for MATLAB*. <http://med.ee.nd.edu/MED11/pdf/papers/t4-006.pdf>
- [28] M. Svádová, Z. Hanzálek. *Matlab Toolbox for Petri Nets*. Center for Applied Cybernetics , DCE FEE. Czech Technical University in Prague. <http://dce.felk.cvut.cz/hanzalek/publications/Hanzalek01a.pdf>
- [29] Libreria de Simulink para modelado de RdP en Matlab. http://www.irt.rwth-aachen.de/typo3/fileadmin/IRT/Download/NetLab/HilfeHTML/netlab_toolbox.html