

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

“Modelado de Fallos en un Quadcopter utilizando Redes de Petri”

Autor: Julio Cesar Bravo Cortés

Tutores: Ing. Ailet Abreu López

Dr.C Eduardo Izaguirre Castellanos

Dr.C José Rafael Abreu García

Santa Clara

2017

"Año 59 de la Revolución"

Universidad Central “Marta Abreu” de Las Villas

Facultad de Ingeniería Eléctrica

Departamento de Automática y Sistemas Computacionales



TRABAJO DE DIPLOMA

“Modelado de Fallos en un Quadcopter utilizando Redes de Petri”

Autor: Julio Cesar Bravo Cortés

jbravo@uclv.cu

Tutores: Ing. Ailet Abreu López

aileta@uclv.cu

Dr.C Eduardo Izaguirre Castellanos

izaguirre@uclv.edu.cu

Dr.C José Rafael Abreu García

abreu@uclv.edu.cu

Santa Clara

2017

"Año 59 de la Revolución"

PENSAMIENTO

No vayas donde guía el camino, ve donde no hay camino y deja un rastro.

Ralph Waldo Emerson

DEDICATORIA

A la memoria de mis abuelos Blas Lorenzo Bravo Sánchez y Víctor Cortés Pérez,

por el ejemplo que me dejaron.

A Noel David,

por su candidez, siempre lo recordamos.

A mis padres,

por el apoyo incondicional y por la guía.

A mi novia,

por compartir conmigo cada momento en estos años.

A mi hermano y mi familia,

por estar ahí siempre que los he necesitado.

AGRADECIMIENTOS

A mi mamá porque sabe mejor que nadie todo lo recorrido hasta llegar aquí.

A mi papá por un día explicarme el significado de la palabra Ingeniero.

A mi hermano por compartir mis preocupaciones y soñar junto conmigo este día.

A mi novia Yadira y a mi prima Ileen por estar pendientes de cada nota, por las alegrías.

A mis hermanos del cuarto: Rolo, Luisi, Addiel gracias por estar todos estos años
juntos como una familia.

A mi familia de Vueltas, mi segunda casa, tía Aida, abuela Argelia, tío Noel, Luis E, Maite,
Amandita, Mavi, Nimia y al Vili.

A la familia de Manicaragua: tío Rubén, Teresa que ha sido como una madrina todos estos
años, a mi abuela Ramona y al Maco que es también como mi abuelo; a Vitico, Lily y
Emilito.

A *Chuncha* que es una fortuna inmensa tenerla y que vea como todos sus nietos y bisnietos
la quieren tanto.

A Omar, Olga y Liset que tanto nos apoyan y se preocupan.

A los compañeros de grupo en estos cinco años de carrera: Soca, Claudia, Vicente, Rubén.

A los compañeros de grupo que vienen conmigo desde del IPVCE y la gente de Camagüey
que tanto hemos compartido.

A José Omar Padrón por sus consejos que nunca olvidamos.

A todos los profesores del Departamento de Automática que tanto nos han enseñado,
siempre desde la humildad, a mis tutores: Ailet, Izaguirre y Abreu; y también a los
maestros que me han educado desde la enseñanza primaria.

RESUMEN

El empleo de las Redes de Petri en el modelado de Sistemas de Eventos Discretos se ha hecho extensivo desde el área de las Ciencias Computacionales hasta campos como la Ingeniería Automática. El Diagnóstico de Fallos en sistemas automatizados es una necesidad debido a la creciente complejidad que presentan los mismos. En esta investigación se propone un modelo de fallos basado en Redes de Petri para un Quadcopter utilizando las ventajas que presentan las mismas para este tipo de aplicación. Para la construcción del modelo se tienen en cuenta los principales fallos registrados en este tipo de aeronave en la literatura especializada; además se tiene presente en el modelo que los fallos se pueden presentar tanto antes del despegue (durante el chequeo de funcionamiento de los componentes) como en pleno vuelo. El modelo con Redes de Petri permitirá diseñar un Diagnosticador que posibilita detectar los fallos del Quadcopter a partir de las mediciones de los sensores a bordo.

TABLA DE CONTENIDOS

| | |
|--|-----|
| PENSAMIENTO | i |
| DEDICATORIA | ii |
| AGRADECIMIENTOS | iii |
| RESUMEN | iv |
| INTRODUCCIÓN | 1 |
| Organización del informe | 5 |
| CAPÍTULO 1. Fundamentos teórico-conceptuales de las Redes de Petri | 6 |
| 1.1 Conceptualización de las Redes de Petri y su aplicación en el diagnóstico de fallos | 6 |
| 1.1.1 Propiedades representativas de las Redes de Petri..... | 8 |
| 1.1.2 Álgebra lineal en Redes de Petri..... | 9 |
| 1.1.3 Redes de Petri No-Autónomas | 11 |
| 1.1.4 Redes de Petri Autónomas | 14 |
| 1.2 Metodología para la Construcción del Modelo de Fallos en Sistemas Híbridos ... | 17 |
| 1.2.1 Proceso de Construcción del Modelo | 18 |
| 1.3 Conclusiones parciales del capítulo | 23 |
| CAPÍTULO 2. Fallos en un Quadcopter | 24 |
| 2.1 Descripción funcional del Quadcopter GARP X4..... | 24 |
| 2.1.1 Elementos del Quadcopter GARP X4 | 25 |
| 2.2 Conceptos y definiciones generales asociados a fallos | 29 |
| 2.2.1 Análisis de los principales fallos en el Quadcopter | 30 |
| 2.3 PNTTool como herramienta de simulación de RdP | 33 |

| | |
|--|-----------|
| 2.3.1 Interfaz Gráfica de Usuario (GUI) del PNTTool | 34 |
| 2.4 Conclusiones parciales del capítulo | 36 |
| CAPÍTULO 3. Modelo de Fallos para el Quadcopter GARP X4 utilizando Redes de Petri | |
| 37 | |
| 3.1 Aplicación de la Metodología para el Modelado de Fallos..... | 37 |
| 3.2 Verificación del Modelo | 51 |
| 3.2.1 Árbol de Alcanzabilidad | 52 |
| 3.2.2 Ecuación de Estados | 54 |
| 3.2.3 Simulaciones con el PNTTool de MatLab | 54 |
| 3.3 Análisis económico y medioambiental | 56 |
| 3.4 Conclusiones parciales del capítulo | 57 |
| CONCLUSIONES Y RECOMENDACIONES | 58 |
| Conclusiones | 58 |
| Recomendaciones | 59 |
| REFERENCIAS BIBLIOGRÁFICAS | 60 |
| ANEXOS | 63 |
| Anexo I Redes de Petri Refinadas para cada subsistema..... | 63 |
| Anexo II Modelo General Refinado Redes de Petri..... | 65 |
| Anexo III Capturas de pantalla de las simulaciones | 66 |
| Anexo IV Estados y Transiciones del Modelo en RdP..... | 67 |

INTRODUCCIÓN

El desarrollo de sistemas automatizados cada vez más complejos ha traído consigo la necesidad de diseñar sistemas más fiables en los que la detección de fallos, hecho muy común dada la complejidad de estos, constituye una tarea vital en aras de evitar roturas y paradas innecesarias de la planta. El diagnóstico preciso ayuda a tomar decisiones correctas en acciones de emergencia y reparación.

En el Diagnóstico de Fallos de Sistemas de Eventos Discretos (DES's, por sus siglas en inglés) existe una evolución implementada en lenguajes normados: Gráficos de Estado, Máquinas de Estado Finito (FSM's, por sus siglas en inglés) (Sampath et al., 1996) y el más empleado, Redes de Petri (RdP) (Ramirez et al., 2007). Además, se encuentran otras investigaciones donde las ventajas de FSM's y RdP se conjugan (Giua and Seatzu, 2005) (Cheng and Jeng, 2003) (Ushio et al., 1998). Los fundamentos de los trabajos mencionados están hechos de FSM's (Sampath et al., 1996). Este modelo tiene una gran limitante que es el número de estados de la composición del modelo (que está dado por la multiplicación de los eventos de los componentes del sistema), si los componentes del sistema se incrementan la construcción del modelo es irrealizable. En general esta metodología presenta varios inconvenientes: es rígida (los fallos tienen que ocurrir de una forma determinada), solo permite el diagnóstico de un fallo, para múltiples fallos, simultáneos y dependientes no puede ser aplicado; y por último, la mayor desventaja son las *explosiones combinatoriales*, esto significa que solo puede ser aplicado a procesos pequeños, cuando la complejidad de los procesos se incrementa, es imposible aplicar esta metodología.

Las Redes de Petri (RdP) constituyen una herramienta de modelado gráfico y matemático aplicada a muchos sistemas. Tiene una gran proyección en el terreno de la Automática, en la

cual se puede estudiar y describir información de sistemas de procesamiento que son caracterizados por ser concurrentes, paralelos, asíncronos, distribuidos, no determinísticos y/o estocásticos. Como herramienta gráfica las RdP pueden ser empleadas como una ayuda de comunicación visual, similar a las cartas de flujo, diagramas de bloques y redes. En adición las marcas son utilizadas en estas redes para simular la dinámica y actividades de concurrencia de sistemas. Como herramienta matemática es posible conformar ecuaciones de estado, ecuaciones algebraicas y otros modelos que representan el comportamiento de sistemas.

En las últimas dos décadas se ha incrementado el número de investigaciones y proyectos relacionados con los Vehículos Aéreos No Tripulados (UAVs, por sus siglas en inglés). Un UAV es un vehículo sin piloto a bordo equipado con armamento o equipamiento de vigilancia (en el caso de aplicaciones militares), o cámaras para grabación de videos, cámaras multiespectrales etc. (en el caso de aplicaciones civiles como lo son la exploración de zonas intrincadas y riego inteligente de cultivos). Pueden ser controlados por un operador desde una estación de control en tierra o pueden operar de manera autónoma mediante un controlador a bordo (Johry, 2016).

Numerosas son las investigaciones existentes en diversas universidades y centros de investigación en el mundo sobre el Diagnóstico de Fallos en los Vehículos Aéreos No Tripulados: Universidad Politécnica de Madrid (Martinez, 2016); Universidad de Minnesota (Hu and Seiler, 2015); Universidad de Concordia, Montreal (Chamseddine et al., 2013); Instituto para el Control y Ciencias Computacionales de la Academia de Ciencias de Hungría (Venkataraman et al., 2013); Universidad del Sur de Georgia (Bupe, 2015); Centro de Investigación en Automática de Nancy (CRAN, por sus siglas en francés) (Witczak et al., 2016); Universidad Tecnológica de Dinamarca (Hansen and Blanke, 2012); Université Libanaise (Saied et al., 2015).

En Cuba, el estudio de fallos en los UAVs es un tema en el cual todavía no existen investigaciones sistemáticas, no obstante, el Grupo de Automatización, Robótica y Percepción (GARP) de la Universidad Central “Marta Abreu” de Las Villas ha venido trabajando en los últimos años en el diseño de estrategias de control de vuelo en Vehículos Aéreos No Tripulados (Avila, 2008) (Martinez, 2009) (Perez, 2014).

La empresa GEOCUBA ha empleado UAVs en los levantamientos aerofotográficos mediante técnicas de Control Fotográfico lo que agiliza en gran medida el proceso de levantamiento topográfico (García et al., 2016) (Ricaño, 2016).

A pesar de la actualidad en estas investigaciones se observa que no se ha indagado sobre el empleo de herramientas para modelado de fallos en este tipo de vehículos; por tanto, el **objeto de estudio** se centra en el empleo de las Redes de Petri como herramientas para el modelado de fallos.

Mientras que el **campo de aplicación** engloba el empleo de las Redes de Petri como herramienta para el modelado de fallos en los sistemas de actuadores de vehículos UAV.

Por consiguiente, se puede identificar el siguiente **problema científico**: No existe un modelo de fallos para el Quadcopter GARP X4 que permita representar los fallos más comunes en el vehículo.

En la literatura consultada se reportan numerosas investigaciones que demuestran la eficacia de las Redes de Petri como herramienta de utilidad para el modelado de fallos en sistemas (Trigos et al., 2015), por lo que se puede plantear la siguiente hipótesis.

Hipótesis:

Resulta posible efectuar el procedimiento de modelado de fallos en vehículos aéreos autónomos mediante el empleo de las Redes de Petri como herramienta de diseño, con vistas a un mejor control de este tipo de sistemas.

En este contexto, la presente investigación está dirigida a enfocar la contribución de los resultados a los vehículos aéreos que se tienen disponibles en el grupo de investigaciones GARP. En este caso, se puede mencionar como una de las recientes adquisiciones al vehículo Quadcopter GARP X4.

El GARP X4 no presenta un Sistema de Detección de Fallos a bordo que sea capaz de detectar, aislar y diagnosticar averías en los motores, en el sistema de comunicación con la estación de tierra y en los sensores. Por tal motivo se necesita contar con un modelo para la detección de fallos en el GARP X4 que permita en trabajos posteriores elaborar un Diagnosticador de Fallos que contribuya a detectar, aislar o diagnosticar a tiempo los mismos en el UAV, dándole sentido práctico a la presente investigación.

Bajo estas consideraciones, se pueden plantear los objetivos que se indican a continuación.

Objetivo General:

Modelar los fallos principales del Quadcopter GARP X4 empleando Redes de Petri.

Objetivos específicos:

1. Revisar la literatura especializada dentro del objeto de investigación.
2. Estudiar los fundamentos teóricos conceptuales relacionados con el Modelado de Fallos de UAV basados en Redes de Petri (RdP).
3. Evaluar las herramientas para el modelado de fallos basados en RdP en el contexto de aplicación relacionado con los vehículos autónomos aéreos.
4. Caracterizar la operación del UAV Quadcopter GARP X4 haciendo hincapié en la identificación de los parámetros de funcionamiento y principales fallos, sobre todo los relacionados con los actuadores.
5. Modelar los principales fallos en el UAV Quadcopter GARP X4 utilizando Redes de Petri.
6. Simular y verificar el modelo de fallos del UAV Quadcopter GARP X4 empleando la herramienta *PNTool* del MatLab.

Tareas de investigación:

1. Consulta de la literatura especializada dentro el objeto de estudio.
2. Análisis de los fundamentos teóricos conceptuales relacionados con el Modelado, de Fallos de Sistemas basados en RdP.
3. Estudio de las herramientas para el modelado de fallos basados en RdP. Clasificación, características, ventajas y desventajas.
4. Estudio de la operación del UAV Quadcopter GARP X4. Identificación de los parámetros de funcionamiento. Fallos en los actuadores y sensores.
5. Modelar los principales fallos en el UAV Quadcopter GARP X4 utilizando Redes de Petri.

6. Simulación y verificación del modelo de fallos en el UAV Quadcopter GARP X4 empleando herramienta PNTTool del Matlab.
7. Redacción del Informe de Tesis.

El principal aporte de esta tesis es la propuesta de un modelo de fallos para el futuro diagnóstico de fallos en el UAV Quadcopter GARP X4, el cual permitirá diseñar en trabajos posteriores un diagnosticador para poder detectar los fallos presentes en los actuadores y sensores del vehículo.

Organización del informe

En la Introducción queda definida la importancia, actualidad y necesidad del tema que se aborda y se dejan explícitos los elementos teóricos empleados en el modelado de fallos con Redes de Petri.

Los temas que se abordan en cada capítulo se encuentran estructurados de la forma siguiente:

Capítulo 1: Se presentan los fundamentos teóricos de las Redes de Petri y su empleo en el modelado de fallos. A partir de estos se presenta la metodología a emplear en el modelado de fallos con Redes de Petri para sistemas híbridos.

Capítulo 2: Se comienza describiendo de forma breve los elementos que conforman el vehículo y sus funciones dentro del conjunto. A partir de las utilidades del Mission Planner se mencionan los fallos que se diagnostican con este software y con ayuda de la literatura especializada se definen los fallos a modelar con Redes de Petri. Se termina el capítulo describiendo las potencialidades de la herramienta PNTTool para la simulación de Redes de Petri.

Capítulo 3: Se modelan los fallos descritos en el capítulo anterior siguiendo la metodología descrita en el Capítulo 1, con Redes de Petri empleando el Toolbox de MatLab PNTTool. Se analizan los resultados finales del modelado de fallos en el Quadcopter después de la obtención del Modelo General Refinado de Redes de Petri, y se verifican las principales características de la RdP para comprobar la funcionalidad del modelo.

CAPÍTULO 1. Fundamentos teórico-conceptuales de las Redes de Petri

En este Capítulo 1 se expondrán los conceptos, principios y características de la utilización de las Redes de Petri, así como los fundamentos matemáticos en los que se sustenta esta herramienta. Se presenta además la metodología a emplear para el modelado de fallos con Redes de Petri en Sistemas Híbridos.

1.1 Conceptualización de las Redes de Petri y su aplicación en el diagnóstico de fallos

Las Redes de Petri (RdP) son una herramienta de modelado gráfico y matemático aplicada a muchos sistemas. Poseen una alta proyección en el campo de la automática, con las cuales se pueden estudiar y describir información sobre sistemas de procesamiento que son caracterizados por ser concurrentes, paralelos, asíncronos, distribuidos, no determinísticos y/o estocásticos. Como herramienta gráfica las RdP pueden ser usadas como una ayuda de comunicación visual similar a las cartas de flujo, diagramas de bloques y redes. Las marcas son usadas en estas redes para simular la dinámica y actividades de concurrencia de sistemas. Como herramienta matemática es posible obtener ecuaciones de estado, ecuaciones algebraicas y otros modelos que gobiernan el comportamiento de sistemas (Murata, 1989).

Una Red de Petri (RdP) tiene dos tipos de nodos, llamados lugares y transiciones. Un lugar (P) es representado por un círculo y una transición (T) por una barra, ver la figura 1.1. Los lugares y transiciones son conectados por arcos. El número de lugares y transiciones son finitos y diferentes de cero. Un arco es directamente conectado de un lugar a una transición o de una transición a un lugar. En otras palabras, una RdP es un grafo bipartito, es decir, que los lugares y transiciones alternan en un camino hecho de arcos consecutivos, como se aprecia en la figura 1.1, por lo cual, es obligatorio que cada arco tenga un nodo en cada una de sus terminaciones, haciendo referencia al hecho de identificar transiciones de entrada y

transiciones de salida, asociadas a un lugar de entrada o salida en particular de la red (Murata, 1989).

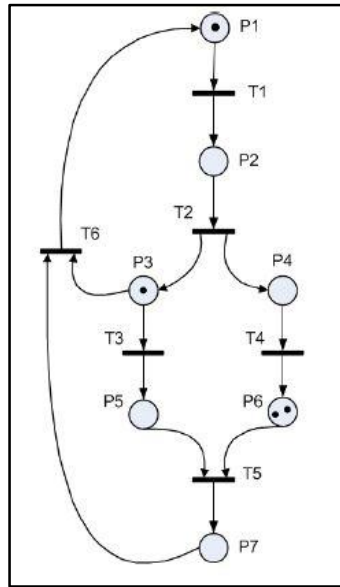


Figura 1.1: Ejemplo de una Red de Petri.

Marcado. Cada lugar contiene un número entero (positivo o cero) de marcas. El número de marcas contenidas en un lugar P_i es llamado $m(P_i)$ o m_i . La red marcada, m , es definida por el vector de ese marcado, es decir, $m = (m_1, m_2, \dots, m_n)$. El marcado en un cierto momento define el estado de la RdP, o para ser más precisos el estado del sistema descrito por la RdP. La evolución del estado por lo tanto corresponde a la evolución del marcado, causada por el disparo de transiciones.

Redes de Petri Autónomas y no-autónomas

Cuando una RdP describe el funcionamiento de un sistema evolucionando de una manera autónoma, es decir, cuyos instantes de disparo están entre no conocidos y no indicados, entonces se puede decir que es una RdP autónoma. Una RdP No-Autónoma describe el funcionamiento de un sistema cuya evolución es condicionada por eventos externos y/o tiempo. Una RdP No-Autónoma es sincronizada y/o temporizada (Martínez, 2016).

1.1.1 Propiedades representativas de las Redes de Petri

Son aquellas características que definen la correcta utilización de las RdP cuando se modela cualquier sistema. Las propiedades más importantes se encuentran vinculadas a los conceptos de limitación, redes seguras, vivacidad de la red, estados sumideros y conflictos.

Un lugar P_i se dice que es limitado por un marcado inicial m_0 , si hay un entero natural k , tal que para cada marcado alcanzable desde m_0 , el número de marcas en P_i no es mayor que k . Una RdP es limitada por un marcado inicial m_0 , si todos los lugares que la conforman, son limitados por m_0 .

Una RdP se dice que es segura para un marcado inicial m_0 , si para todos los marcados alcanzables, cada lugar contiene cero o una marca. Una transición T_j se considera viva por un marcado inicial m_0 , si para cada marcado alcanzable $m_i \in \mu(m_0)$, existe una secuencia de disparos s desde m_i que contiene a T_j . Una RdP es viva para el marcado inicial m_0 , si todas sus transiciones son vivas para m_0 .

Un bloqueo o estado sumidero de la red, es un marcado tal que ninguna transición está habilitada. Una RdP se dice que no contiene estado sumidero o libre de bloqueos para un marcado inicial m_0 , si ninguno de los marcados alcanzables $m_i \in \mu(m_0)$, es estado sumidero (Murata, 1989).

Un conflicto efectivo $K^E = \langle P_i, \{T_1, T_2, \dots\}, m \rangle$, es la existencia de un conflicto en la estructura de la red $K = \langle P_i, \{T_1, T_2, \dots\} \rangle$ y del marcado m , tal que las transiciones en el conjunto $\{T_1, T_2, \dots\}$ son habilitadas por m y el número de marcas en P_i es menor que la suma del peso de los arcos $P_i \rightarrow T_1, P_i \rightarrow T_2, \dots$ véase figura 1.2

Un conflicto general $K^G = \langle P_i, \{T_1, T_2, \dots\}, m \rangle$, es la existencia de un conflicto estructural $K = \langle P_i, \{T_1, T_2, \dots\} \rangle$ y del marcado m , tal que el número de marcas en P_i no es suficiente para habilitar todas las transiciones de salida de P_i de acuerdo a su grado de activación (Trigos et al., 2015).

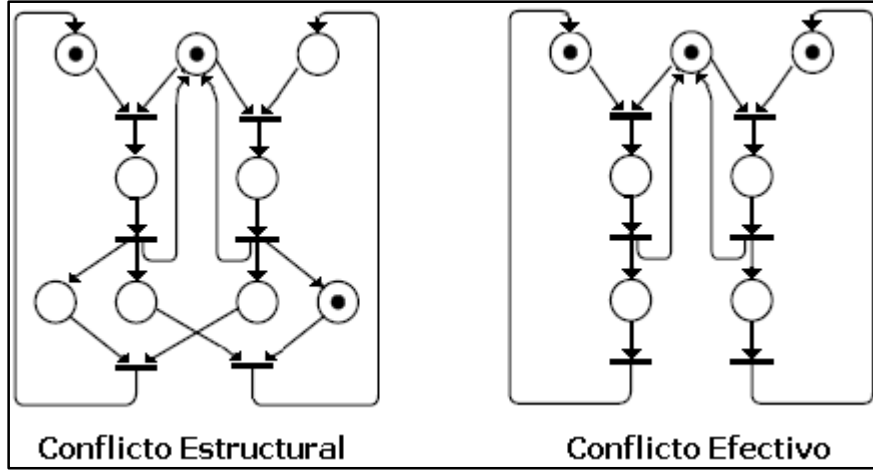


Figura 1.2: Conflicto estructural y conflicto efectivo.

1.1.2 Álgebra lineal en Redes de Petri

Una RdP Ordinaria no marcada está representada por la cuádrupla $Q = \langle P, T, P_{re}, P_{ost} \rangle$ tal que:

$P = \{P_1, P_2, \dots, P_n\}$ es el conjunto de lugares finito y no vacío;

$T = \{T_1, T_2, \dots, T_m\}$ es el conjunto de transiciones finito y no vacío;

$P \cap T = \emptyset$, es decir que los conjuntos P y T son disjuntos;

$P_{re}: P \times T \rightarrow \{0, 1\}$ es la aplicación de la incidencia de entrada;

$P_{ost}: T \times P \rightarrow \{0, 1\}$ es la aplicación de la incidencia de salida.

$P_{re}(P_i, T_j)$ es el peso del arco $P_i \rightarrow T_j$. El peso es 1 si el arco existe y 0 si no.

$P_{ost}(P_i, T_j)$ es el peso del arco $T_j \rightarrow P_i$. P_{re} y P_{ost} así relaciona a la transición T_j del par (P_i, T_j) .

Una RdP no marcada generalizada es definida como una RdP ordinaria no marcada, excepto

$$P_{re}: P \times T \rightarrow \mathbb{N}$$

$$P_{ost}: P \times T \rightarrow \mathbb{N} \quad (1.1)$$

Las siguientes notaciones serán usadas:

${}^{\circ}T_j = \{P_i \in P \mid P_{re}(P_i, T_j) > 0\}$ = el conjunto de lugares de entrada de T_j ;

$T = \{P_i \in P \mid P_{ost}(P_i, T_j) > 0\}$ = el conjunto de lugares de salida de T_j ;

${}^aP_i = \{T_j \in T \mid P_{re}(P_i, T_j) > 0\}$ = el conjunto de transiciones de entrada de P_i ;

$P_0 = \{T_j \in T \mid P_{ost}(P_i, T_j) > 0\}$ = el conjunto de transiciones de salida de P_i .

Una RdP marcada es un par $R = \langle Q, m_0 \rangle$ en el cual Q es una RdP no marcada y m_0 es el marcado inicial (Murata, 1989). Las condiciones de validación pueden ser expresadas como sigue: una transición T_j es habilitada para un marcado m_k si $m_k(P_i) \geq P_{re}(P_i, T_j)$ para cada $P_i \in {}^aT_j$.

La matriz de incidencia de entrada es $W^- = [w_{ij}^-]$ donde $w_{ij}^- = P_{re}(P_i, T_j)$; la matriz de incidencia de salida es $W^+ = [w_{ij}^+]$ donde $w_{ij}^+ = P_{ost}(P_i, T_j)$; entonces la matriz de incidencia es:

$$W = W^+ - W^- = [w_{ij}] \quad (1.2)$$

Ecuación fundamental. Sea S una secuencia de transiciones, la cual puede ser realizada desde el marcado m_i , la cual puede ser escrita como $m_i \xrightarrow{S}$. El vector característico de secuencia S , escrito como s , es el vector de m -componentes cuyo componente número j corresponde al número de disparos de la transición T_j en la secuencia S . Si el disparo de la secuencia S es tal que $m_i \xrightarrow{S} m_k$, entonces la ecuación fundamental es obtenida (Ruiz-Beltran et al., 2005):

$$m_k = m_i + W \cdot s \quad (1.3)$$

El conjunto de alcanzabilidad de una RdP es el conjunto de todos los marcados alcanzables desde m_0 disparando solo transiciones habilitadas; este conjunto es denotado por $R \langle Q, m_0 \rangle$. Una secuencia de transiciones de disparo de una RdP $\langle Q, m_0 \rangle$, es una secuencia de transición $S = T_i, T_j, \dots, T_k, \dots$ tal que $m_0 \xrightarrow{T_i} m_1 \xrightarrow{T_j} m_x \xrightarrow{T_k} \dots$

El conjunto de todas las secuencias de disparo es llamado el lenguaje $L(Q, m_0)$:

$$L(Q, m_0) = \{S = T_i, T_j, \dots, T_k, \dots \wedge m_0 \xrightarrow{T_i} m_1 \xrightarrow{T_j} m_x \xrightarrow{T_k} \dots\} \quad (1.4)$$

1.1.3 Redes de Petri No-Autónomas

Las RdP autónomas permiten un análisis cualitativo, a continuación se presenta una extensión de las RdP con la que es posible describir no solo “qué pasa” sino también “cuándo pasa” un hecho. Las RdP No-autónomas habilitan el sistema a ser modelado cuyos disparos están sincronizados con eventos externos y/o cuyas evoluciones son dependientes del tiempo.

Redes de Petri Sincronizadas

En una RdP autónoma, se sabe que una transición puede ser disparada si esta es habilitada, pero no se conoce cuando esta podría ser disparada. En una RdP sincronizada, un evento es asociado con cada transición, y el disparo de esta transición ocurrirá si la transición es habilitada, cuando los eventos asociados ocurren. Los eventos externos corresponden a un cambio en el estado de la palabra externa (incluyendo el tiempo); por el contrario, un cambio en el estado interno, un cambio en el marcado, podría ser llamado un evento interno. La ocurrencia de un evento no tiene duración (Trigos et al., 2015).

Una RdP Sincronizada es un Triple $\langle R, E, Sync \rangle$ tal que:

- R es una RdP marcada.
- E es un conjunto de eventos externos.
- $Sync$ es una función del conjunto T de las transiciones de R de $E \cup \{e\}$, en el cual e es siempre el evento que ocurre o vacío.

$E = \{E_1, E_2, \dots\}$ es un conjunto de eventos externos. La notación E_i corresponde al nombre de un número de evento externo. La notación E_j corresponde al evento asociado a la transición E_j .

Redes de Petri Interpretadas

Una Red de Petri Interpretada exhibe características tales como sincronización, lugares temporizados (P-timed) y tiene una parte para procesamiento de datos. Las entradas están asociadas con las transiciones y las salidas están asociadas con los lugares. Como se puede ver en la figura 1.3, el evento E_j y la condición C_j son asociados con la transición T_j . La condición C_j es una función booleana dependiendo de la parte de procesamiento de datos y el ambiente (Huayna et al., 2009). El evento E_j está entre un evento externo derivado del

ambiente y un evento e siempre que ocurra la transición T_j que será disparada si: la transición T_j está habilitada y si la condición C_j es verdadera, cuando el evento E_j ocurra.

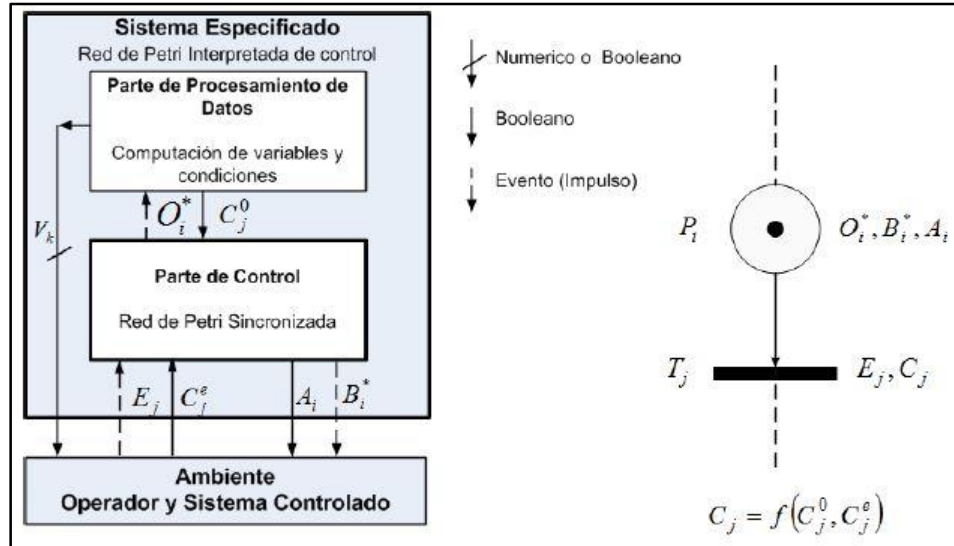


Figura 1.3: Red de Petri Interpretada.

El producto $R_j = E_j \cdot C_j$ es llamado receptividad de la transición T_j . La acción denotada en la figura O_i^* , B_i^* , y A_i está asociada al lugar P_i . Cuando una marca es depositada en el lugar P_i , en el instante t , la operación O_i^* es realizada y la acción impulso B_i^* es enviada al ambiente. La salida Booleana A_i tiene el valor booleano 1 tanto tiempo como hay una marca en el lugar P_i .

Una RdP Interpretada de control exhibe las siguientes cinco características (1 a 3 necesariamente y 4 y 5 posiblemente).

1. Es sincronizada a eventos externos y estable.
2. Es segura.
3. Es determinística.
4. Tiene una parte de procesamiento de datos cuyo estado es definido por un conjunto de variables $V = \{V_1, V_2, \dots\}$. Este estado es modificado por operaciones O_i^* las cuales son asociadas a los lugares. Este determina el valor de los predicados C_j^0 .
5. Recibe información booleana C_e del ambiente. Envía acciones de nivel A_j y acciones impulso B_j , asociadas con los lugares, al ambiente.

Redes de Petri Temporalizadas

Las RdP temporalizadas son útiles para evaluar comportamiento y habilitar un sistema a ser descrito cuando su funcionamiento es dependiente del tiempo. Por ejemplo, un evento puede durar cierto tiempo entre el inicio y el final de una operación. Si un marcado en un lugar indica que esta operación está en progreso, una RdP temporalizada permite al tiempo ser tenido en cuenta. Hay dos métodos principales para modelar el tiempo: el tiempo asociado con los lugares (lugares temporalizados) y el tiempo asociado con las transiciones (transiciones temporalizadas) (Genc and Lafortune, 2007).

Redes de Petri P-timed. Un tiempo d_i de valor cero si no es especificado, es asociado con cada lugar P_i . Se debe considerar el caso donde d_i es un valor constante, pero en general puede ser variable. Una RdP P-timed es el par $\langle R, T_{empo} \rangle$ tal que:

- R es una RdP marcada.
- T_{empo} es una función del conjunto de lugares P del conjunto de números racional cero o positivo. $T_{empo}(P_i) = d_i =$ tiempo asociado con el lugar P_i .
- Cuando una marca es depositada en un lugar P_i , esta marca debe mantenerse en este lugar al menos por un tiempo d_i . Esta marca se dice que es no disponible para este tiempo. Cuando el tiempo d_i ha transcurrido, entonces la marca se convierte en disponible.

Redes de Petri T-timed. Un tiempo d_j , de valor cero si no es especificado, es asociado con la transición T_j . Una RdP T-timed es el par $\langle R, T_{empo} \rangle$ tal que:

- R es una RdP marcada.
- T_{empo} es una función del conjunto de transiciones T del conjunto de números racional cero o positivo. $T_{empo}(T_j) = d_j =$ tiempo asociado con la transición T_j .
- Cuando una transición T_j se convierte en habilitada, esta no es disparada de forma inmediata. Esta será disparada cuando el tiempo d_j haya transcurrido después de la habilitación, excepto en caso de un conflicto. Una marca puede tener dos estados: puede ser reservada para el disparo de la transición T_j o puede no ser reservada.

Redes de Petri Estocásticas. Una RdP estocástica puede ser considerada como una RdP Temporizada en el cual los tiempos tienen valores estocásticos (Cheng and Jeng, 2003).

Una RdP Estocástica es el par $\langle R, Rate \rangle$ tal que:

- R es una RdP marcada;
- $Rate$ es una función desde el conjunto de transiciones T al conjunto de números reales positivos finitos. $Rate(T_j) = \mu_j = \text{rating de disparo asociado con } T_j$. μ que representa la función de probabilidad de disparo.

1.1.4 Redes de Petri Autónomas

La Red de Petri continua es un modelo en el cual el número de marcas en el lugar es representado por números reales en vez de enteros (Murata, 1989).

Redes de Petri Continuas

Se considera un RdP R , (autónoma, discreta, ordinaria o generalizada) definida por su grafo Q (lugares, transiciones, arcos) y su marcado m . Se considera aplicar una transformación, la cual consiste en dividir cada marca, en k partes iguales (sin cualquier otra modificación de la RdP) y en donde se consigue el disparo de una transición T_j , removiendo una de estas k partes (denominados también tokens) de un lugar de entrada P_i , para adherirlo, después del disparo, a un lugar de salida (Murata, 1989).

Una RdP continua autónoma es una quintupla $R = \langle P, T, P_{re}, P_{ost}, m_0 \rangle$ tal que:

- $P = \{P_1, P_2, \dots, P_n\}$ es el conjunto de lugares finito y no vacío;
- $T = \{T_1, T_2, \dots, T_m\}$ es el conjunto de transiciones finito y no vacío;
- $P \cap T = \emptyset$, es decir que los conjuntos P y T son disjuntos;
- $P_{re}: P \times T \rightarrow Q_+$ es la aplicación de la incidencia de entrada;
- $P_{ost}: T \times P \rightarrow Q_+$ es la aplicación de la incidencia de salida;
- $m_0: P \rightarrow R_+$ es el marcado inicial.

El peso de los arcos puede ser definido como números reales (Manyari-Rivera et al., 2007). El marcado de un lugar debe ser un número real. Los lugares y transiciones son representados por una doble línea. En una RdP continua, el grado de habilitación de una transición T_j para el marcado m , denotado por q o $q(T_j, m)$ es un número real q tal que:

$$q = \min_{i: P_i \in {}^oT_j} \left(\frac{m(P_i)}{P_{re}(P_i, T_j)} \right) \quad (1.5)$$

Si $q > 0$, transición T_j es habilitada; esto para decir q -habilitada (Martinez, 2016).

Redes de Petri Híbridas

Una RdP híbrida está compuesta por lugares y transiciones continuas (C) y lugares y transiciones discretas (D). Como se observa en la figura 1.4, la representación de los lugares y transiciones de la parte discreta y continua es diferente, además, el marcado de un lugar continuo es representado por un número real a diferencia de un lugar discreto que mantiene los puntos negros (Trigos et al., 2015).

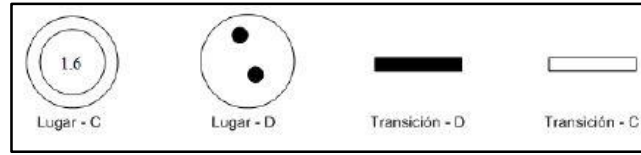


Figura 1.4: Elementos de una RdP Híbrida.

Una RdP Híbrida autónoma marcada es una séxtupla $R = \langle P, T, P_{re}, P_{ost}, m_0, h \rangle$ cumpliendo las siguientes condiciones:

- $P = \{P_1, P_2, \dots, P_n\}$ es un conjunto de lugares finito y no vacío;
- $T = \{T_1, T_2, \dots, T_m\}$ es un conjunto de transiciones finito y no vacío;
- $P \cap T = \emptyset$, es decir que los conjuntos P y T son disjuntos;
- $h: P \cup T \rightarrow \{D, C\}$, llamada función Híbrida, indica para cada nodo si este es un nodo discreto (conjunto P^D y T^D) o continuo (conjunto P^C y T^C).
- $P_{re}: P \times T \rightarrow \mathbb{Q}_+ \text{ o } \mathbb{N}$ es la aplicación de la incidencia de entrada;
- $P_{ost}: T \times P \rightarrow \mathbb{Q}_+ \text{ o } \mathbb{N}$ es la aplicación de la incidencia de salida;
- $m_0: P \rightarrow \mathbb{R}_+ \text{ o } \mathbb{N}$ es el marcado inicial.

En las definiciones de P_{re} , P_{ost} y m_0 , \mathbb{N} corresponde al caso donde $P_i \in P^D$, y \mathbb{Q}_+ o \mathbb{R}_+ corresponde al caso donde $P_i \in P^C$. Una transición discreta en una RdP híbrida es habilitada si cada lugar P_i en oT_j encuentra la condición $m(P_i) \geq P_{re}(P_i, T_j)$. Una transición continua en una RdP híbrida es habilitada si cada lugar P_i en oT_j encuentra la condición $m(P_i) \geq$

$P_{re}(P_i, T_j)$, si P_i es un lugar discreto (D), o $m(P_i) > 0$, si P_i es un lugar continuo (C) (Martinez, 2016).

La Matriz de Incidencia de una RdP Híbrida puede ser escrita como:

$$W = \begin{bmatrix} W^D & 0 \\ W^{CD} & W^C \end{bmatrix} \quad (1.6)$$

donde W^D corresponde a los arcos entre nodos discretos, W^C a los arcos entre nodos continuos, y W^{CD} a los arcos entre los lugares C y transiciones D . Los arcos entre los lugares D y transiciones C corresponden a la submatriz 0.

Las funciones P_{re} y P_{ost} deben mantener los siguientes criterios: si P_i y T_j son un lugar y una transición tal que $h(P_i) = D$ y $h(T_j) = C$, entonces $P_{re}(P_i, T_j) = P_{ost}(P_i, T_j)$ debe ser verificada.

Esta última condición establece que un arco debe unir una transición C a un lugar D tan pronto como un arco recíproco exista. Esto asegura que el marcado de un lugar D , es un entero cualquiera que sea la evolución que ocurra (Martinez, 2016).

Una Transición C es habilitada si las dos condiciones siguientes son encontradas:

1. Para cada lugar D P_i en ${}^oT_j, M(P_i) \geq Pre(P_i, T_j)$.
2. Para cada lugar C P_i en ${}^oT_j, M(P_i) > 0$.

Para una transición C , la clase de lugar precediendo la transición, debe ser especificado porque las condiciones que habilitan son diferentes si esta es un lugar C o un lugar D .

Sea S una secuencia de disparo y s sea el vector característico de S . La dimensión del vector s es igual al número m de transiciones. El j -ésimo componente de m representa el número de disparos de transiciones T_j y será denotado por n_j . Si T_j es una transición D , entonces n_j es un entero y si T_j es una transición C , entonces n_j es un número real (Murata, 1989).

Un marcado m puede ser deducido de un marcado m_0 debido a una secuencia S , usando la Ecuación de Estados, ver ecuación 1.3.

La relación fundamental de una RdP Híbrida es igual a la relación fundamental de una RdP Discreta. Entonces, se puede deducir que cada propiedad de RdP Discretas resultante de esta relación puede ser transpuesta a RdP Híbridas (Martinez, 2016).

Esta parte de procesamiento de datos es definido por un conjunto de variables. El estado es modificado por operaciones $O = O_1, O_2, \dots$, asociadas a los lugares. Las mismas determinan el valor de las condiciones $C = C_1, C_2, \dots$ que son asociadas a las transiciones (Murata, 1989).

1.2 Metodología para la Construcción del Modelo de Fallos en Sistemas Híbridos

Un sistema híbrido es un sistema en el cual el comportamiento de interés es determinado por la interacción de procesos de diferentes características, en particular, interactuando de forma dinámica y discreta. Los sistemas híbridos generan señales mezcladas que consisten en combinaciones de señales de valor discreto y continuo. Además, estas señales de valor discreto o continuo dependen de variables independientes tales como el tiempo, el cual también puede ser de valor discreto o continuo. Otra distinción que puede ser hecha, es que algunas de las señales pueden ser manejadas por el tiempo, mientras que otras pueden ser manejadas por eventos, en una manera asincrónica. El comportamiento dinámico de tales sistemas híbridos es capturado en modelos híbridos (Martinez, 2016).

La complejidad de un sistema híbrido es la razón fundamental para que se tome la decisión de migrar de una herramienta de modelado de fallos diseñada inicialmente solo para Sistemas de Eventos Discretos (SED's) a sistemas híbridos (Martinez, 2016).

La teoría fundamental de las RdP se basa en la identificación individual de los componentes del sistema (SED's) y la relación entre ellos, en el modelo se debe incluir el comportamiento normal del proceso, junto con el comportamiento de fallo, similar al presentado por (Sampath, 1995) (Ramirez et al., 2007).

Sea $R = (P, T, P_{re}, P_{ost}, m_0)$, la RdP descrita en el epígrafe 1.1 que representa el modelo de eventos discretos del sistema a modelar. Las transiciones que componen el sistema son clasificadas en el conjunto de transiciones observables T_o y el conjunto de transiciones no observables T_{ou} , $T = T_o \cup T_{ou}$, cuando se hace referencia a observable, se plantea que estas transiciones pueden ser observadas por cualquier agente del sistema, las cuales son dadas por

los eventos de control T_c (comandos del supervisor) y por la instrumentación implementada en el proceso, tales como las transiciones dadas por los sensores T_s , $T_o = T_c \cup T_s$ (Cabasino et al., 2010).

No observable se refiere a transiciones que suceden desde cualquier estado o lugar del sistema P_i y no está contemplado en el modelo que describe el funcionamiento del mismo. En las transiciones no observables es donde están incluidas las transiciones de fallo T_f , $T_f \subseteq T_{uo}$. El objetivo que se plantea cualquier sistema de diagnóstico de fallos es el de identificar las T_{uo} , porque las T_o puede ser identificadas fácilmente (E.Garcia et al., 2005); las T_f se clasifican en conjuntos disjuntos correspondientes a los diferentes tipos de fallo que se pueden presentar en el sistema, siendo importante poder distribuir los fallos en grupos que permitan facilitar su identificación, por lo tanto, el conjunto de las transiciones de fallo T_f está compuesto de los diferentes subconjuntos de fallo dados en el proceso, $T_f = T_{f1} \cup \dots \cup T_{fN}$. N es el número de componentes que integran el sistema y sea Π_f la distribución de fallos, dicha partición viene justificada por:

- No es necesario identificar la transición de fallo de forma particular, es preferible de un conjunto de fallos, identificar alguno que conduzca a diagnosticar el fallo, suceda el que suceda, esto le permite al sistema ser más práctico.
- La instrumentación inadecuada del proceso hace imposible de forma simple diagnosticar cada posible fallo.

Por consiguiente, cuando un fallo de tipo F_i ha ocurrido, significa que alguna transición del conjunto T_{fi} ha ocurrido (Sampath, 1995).

1.2.1 Proceso de Construcción del Modelo

Paso 1. Dividir el Sistema en Subsistemas

El sistema H puede ser dividido en M subsistemas con cercana relación entre ellos (Martinez, 2016), dependiendo de su comportamiento. Esto permite reducir la complejidad de los sistemas híbridos. Así:

$$H_j = H_1 \cup H_2 \cup \dots \cup H_M \quad (1.7)$$

$j = 1, \dots, M$, M es el número de subsistemas que componen el sistema.

Paso 2. Construir el Modelo de RdP de los componentes de cada subsistema

En cada uno de los subsistemas identificados $H_j, j = 1, \dots, M$ dependiendo si es un subsistema discreto o continuo, la forma de representar el modelo varía. En los modelos se debe tener en cuenta el comportamiento normal y de fallo. Al mismo tiempo se tienen que definir los fallos de los componentes en cada subsistema.

Sea $R_i^j = (P_i, T_i, P_{re}, P_{ost}, m_0)$ la RdP marcada del i -componente correspondiente al subsistema j , $i = 1, \dots, N$, N es el número de componentes en el subsistema j . P_i es el conjunto de lugares, T_i es el conjunto de transiciones, P_{re}/P_{ost} son las aplicaciones de la incidencia de entrada/salida y m_0 es el marcado inicial. Es importante notar que se debe mantener la sincronía de funcionamiento del proceso completo. Para diferenciar los lugares y las transiciones de fallo, estas se deben representar con círculos y barras sombreados respectivamente (Martinez, 2016).

Paso 3. Operación de integración

La siguiente expresión se utiliza para denotar la operación de integración para el subsistema $\widetilde{Q}^j = (\widetilde{P}^j, \widetilde{T}^j, \widetilde{P}_{re}^j, \widetilde{P}_{ost}^j, \widetilde{m}_0^j)$. Por lo tanto \widetilde{Q}^j representa el comportamiento del subsistema H_j a través de un único modelo de RdP, el cual incluye N_j diferentes modelos de RdP correspondientes a sus componentes. El mismo integra el comportamiento normal y de fallo del sistema, las transiciones (observable T_o y no observable T_{ou}) pueden ocurrir en cualquier lugar. Sea \widetilde{P}^j y \widetilde{T}^j los conjuntos de lugares y transiciones del componente respectivo, como sigue:

$$\widetilde{P}^j = \bigcup_{i=1}^N P_i \quad y \quad \widetilde{T}^j = \bigcup_i T_i \quad (1.8)$$

El siguiente proceso tiene que ser complementado para construir el modelo de integración (Martinez, 2016). Así, para cada uno de los subsistemas M es requerido:

Paso 3.1: Definir el conjunto de fallos F , basado en los fallos definidos para cada componente del sistema, $F = \{F_1, F_2, \dots, F_N\}$.

Paso 3.2: Definir el lugar inicial p_0^j . Este brinda los lugares de cada componente del subsistema $P_0^j = \{p_0^1, p_0^2, \dots, p_j^N\}$.

Paso 3.3: Asignar la marca al lugar de inicio p_0^j , este define el punto de inicio para la evolución del subsistema representado.

Paso 3.4: Construir la rama correspondiente al comportamiento normal R_N^j . Esta designa la interacción y evolución de los componentes. Las transiciones son solamente compuestas por comandos del controlador T_c . El conjunto de lugares del subsistema está compuesto de los lugares de cada componente.

Paso 3.5: Adicionar los lugares de fallo P_f^j . Un lugar por cada fallo desde el conjunto F tiene que ser generado.

Paso 3.6: Adicionar las transiciones de fallo T_f^j . En cada lugar de la RdP general del subsistema j se puede presentar la evolución de los fallos, es decir, desde cada lugar p_i de la RdP general, una transición de fallo t_f^j es por lo tanto adicionada: $\forall p_i^j: p_i \in P^j \rightarrow \exists (p_i, t_j) \in F$, donde F es el conjunto de fallos del sistema.

Paso 4. Refinar el Modelo General

Es necesario considerar solo la parte observable del modelo general \widetilde{Q}^j . Por lo tanto, $\widetilde{Q}^j = (\widetilde{P}^j, \widetilde{T}^j, \widetilde{P}_{re}^j, \widetilde{P}_{ost}^j, \widetilde{m}_0^j)$ debe ser transformado al modelo general refinado $Q^j = (P^j, T^j, P_{re}^j, P_{ost}^j, m_0^j)$. Este modelo está solo compuesto por lugares y transiciones observables. Acorde a esto, las transiciones de fallo tienen que ser reemplazadas por lecturas de los sensores T_s . El siguiente proceso debe ser seguido para refinar el modelo:

Paso 4.1: Identificar los sensores del sistema. Cada subsistema tiene N_{sj} sensores. La idea fundamental de la herramienta es aprovechar la información disponible del subsistema leída por los sensores, con esta información se identifica la aparición de posibles condiciones anormales del sistema.

Paso 4.2: Construir el conjunto discreto de salidas del sensor. Se debe suponer que el sistema de sensores debe entregar solo variables discretas. Dado el conjunto N_s sensores del subsistema de interés, identificar sus salidas y con ellas construir la combinación posible de lecturas sensoriales. Y representa el número de combinaciones de las lecturas de sensores. Desde ahí, Y son las entradas para la tabla de integración de sensores, $|Y| = 2^{N_s}$. Para una mejor ilustración de la construcción de las posibles salidas de sensores, se supone que el

sistema posee dos sensores, el estado discreto de cada sensor es L o \bar{L} (1 o 0) según corresponda. Para el ejemplo tendremos $2^2 = 4$ combinaciones de salidas de sensores, representados en la tabla 1.1. La combinación Y dada anteriormente se convierte en las entradas de la tabla de integración de sensores.

Tabla 1.1: Tabla de Salida de Sensores.

| | |
|-------------|-------------|
| S_1 | S_2 |
| \bar{L}_1 | \bar{L}_2 |
| \bar{L}_1 | L_2 |
| L_1 | \bar{L}_2 |
| L_1 | L_2 |

Paso 4.3: Definir las salidas de la tabla de integración. Una salida en la tabla de integración de sensores es adicionada para cada lugar de la rama de comportamiento normal en la RdP de cada subsistema. Cada lugar representa el estado actual de las lecturas de los sensores, por ejemplo, el lugar P_3^j representa las lecturas de los componentes X, \bar{Y} , entonces, $P_3^j = (X, \bar{Y})$, muestra que en el lugar normal P_3 , X está en este lugar en un estado positivo y Y mantiene un estado negativo.

Tabla 1.2: Salidas de la Tabla de Integración de Sensores de un Subsistema.

| | | | | | |
|-------------|-------------|---------------------|---------------|---------------------|---------------------|
| S_1 | S_2 | $P_1^j(\bar{X}, Y)$ | $P_2^j(X, Y)$ | $P_3^j(X, \bar{Y})$ | $\dots P_i^j(X, Y)$ |
| \bar{L}_1 | \bar{L}_2 | | | | |
| \bar{L}_1 | L_2 | | | | |
| L_1 | \bar{L}_2 | | | | |
| L_1 | L_2 | | | | |

Paso 4.4: Construir la tabla de integración de sensores $h_j = \tilde{P} \rightarrow \tilde{Y}_j, j = 1, \dots, M$, donde \tilde{Y}_j denota el conjunto discreto de posibles salidas del j -ésimo sensor, esto define:

$$Y = \prod_{j=1}^N Y_j \quad (1.9)$$

h_j es definida por: $h(P) = (h_1(P), h_2(P), \dots, h_N(P))$. Y corresponde a la tabla de integración de sensores. Las entradas de la tabla de integración Y y las lecturas esperadas de los sensores de cada lugar de la rama normal deben ser comparadas, es decir, si las lecturas del sensor y las lecturas esperadas son la misma, entonces, $h_j(P_j)$ es clasificada como normal N . Por otra parte, si las lecturas son diferentes, estas pueden indicar un fallo F o un valor indeterminado X , debido a que ellas no tienen información útil para utilizar, $Y_j = Y_{jN} \cup Y_{jF} \cup Y_{jX}$. Por lo tanto, en cada lugar del modelo se pueden definir las salidas de sensores como N (normal), F (fallo) y X (no determinado).

Tabla 1.3: Tabla de Integración de Sensores de un Subsistema.

| S_1 | S_2 | $P_1^j(\bar{X}, Y)$ | $P_2^j(X, Y)$ | $P_3^j(X, \bar{Y})$ | $\dots P_i^j(X, Y)$ |
|-------------|-------------|---------------------|---------------|---------------------|---------------------|
| \bar{L}_1 | \bar{L}_2 | F | F | X | F |
| \bar{L}_1 | L_2 | N | X | F | X |
| L_1 | \bar{L}_2 | X | X | N | N |
| L_1 | L_2 | F | N | F | N |

Paso 4.5: Reemplazar las transiciones de fallos y remover los lugares no alcanzables. Las transiciones de fallo del modelo de RdP general \tilde{Q}^j tienen que ser reemplazadas por las lecturas de los sensores. Los lugares de fallo que no pueden ser identificados porque la información disponible no permite habilitar el marcado del lugar, deben ser removidos de \tilde{Q}^j .

$$\forall p_i: p_i \in P \rightarrow M(p_i) \notin R(Q, M_0) \quad (1.10)$$

Finalmente, el modelo de integración refinado de cada subsistema está compuesto de lugares normales $P_i^j = p_j^1 \cup p_j^2 \cup \dots \cup p_j^n$ y lugares de fallo P_f dados en Π_f , $P = P_N \cup P_F$. Las transiciones T^j incluyen eventos de control o supervisor T_C y las lecturas de sensores consideradas en la tabla de integración T_S , $T = T_C \cup T_S$. En conclusión, el modelo refinado de cada subsistema estará compuesto solo por lugares observables (Martinez, 2016).

1.3 Conclusiones parciales del capítulo

Las Redes de Petri constituyen una herramienta eficaz para la representación de sistemas de eventos discretos, continuos e híbridos, gracias a su potencialidad en el modelado y el poderoso basamento matemático que poseen.

La metodología para el modelado de fallos en sistemas híbridos es factible para aplicar en el presente caso dado que es una extensión de la empleada en sistemas discretos, y existen evidencias de resultados concretos en la práctica avalada en varios trabajos científicos.

La construcción del modelo de fallos para el Quadcopter con RdP se debe efectuar a partir del conocimiento profundo del funcionamiento del sistema a modelar, sobre todo en lo relacionado a los parámetros de funcionamiento y sus especificaciones funcionales asociadas.

CAPÍTULO 2. Fallos en un Quadcopter

En este capítulo se describen las partes del Quadcopter GARP X4. También con la ayuda del Mission Planner y de la literatura especializada se logra caracterizar el comportamiento en fallo de los diferentes subsistemas del Quadcopter.

2.1 Descripción funcional del Quadcopter GARP X4

El Quadcopter es un helicóptero multirotor que es propulsado por cuatro rotores. A diferencia de la mayoría de los helicópteros, los Quadcopter utilizan dos conjuntos de hélices fijas idénticas, dos que giran en sentido de las agujas del reloj y otras dos en sentido contrario, estas hélices emplean una variación de las RPM para controlar la elevación y la dirección del desplazamiento. El control de movimiento del vehículo se consigue mediante la modificación de la velocidad de rotación de uno o más discos de rotor, cambiando así su carga de par y las características de elevación y empuje (Saied et al., 2015).

El funcionamiento de estos dispositivos para vuelo con control remoto está representado por el diagrama de la figura 2.2. A través de un sistema de comunicación inalámbrico por radio frecuencia se comunica al controlador la nueva dirección a tomar lo cual, junto con las medidas obtenidas de los elementos de medición incluidos en la Unidad Inercial de Medidas (IMU) del controlador de vuelo (giróscopos, acelerómetros, magnetómetros) así como elementos de posicionamiento externos como el Módulo del Sistema de Posicionamiento Global (GPS), permiten actuar sobre las velocidades de los motores variando la tensión aplicada a los mismos (Carrillo et al., 2013).



Figura 2.1: Modelo de Quadcopter.

El giro de las palas de los rotores del Quadcopter generan un flujo de aire descendente que proporciona una fuerza de empuje sobre el cuerpo que sustenta el vehículo durante el vuelo. Variando las fuerzas de empuje individuales de cada rotor, a través de la tensión aplicada a los motores, es posible modificar los ángulos de alabeo (roll) y cabeceo (pitch), así como, variando el par de los motores, el ángulo de guiñada (yaw) (Haddadi et al., 2015).



Figura 2.2: Esquema de Control de Vuelo para un Quadcopter.

2.1.1 Elementos del Quadcopter GARP X4

En esta sección se presentan y describen los distintos componentes que constituyen el Quadcopter GARP X4. Elemento a elemento, se analizará su funcionamiento, así como su función dentro del sistema completo de forma que logre un mejor entendimiento del conjunto.

Estructura o marco

El marco es la estructura a la que se van a acoplar los demás elementos. El marco debe ser rígido de forma que se minimicen las vibraciones generadas en los rotores. El material del GARP X4 es de fibra de carbono.



Figura 2.3: Marco de fibra de carbono.

Motores

Los motores DC sin escobillas (brushless) son motores síncronos que son alimentados por una fuente eléctrica DC (Batería LIPo). A diferencia de los motores DC con escobillas, los motores brushless requieren de un conmutador electrónico para generar una señal de control que energice cada una de las bobinas en forma secuencial para dar paso a la rotación del motor. Estos motores presentan grandes ventajas respecto a los motores con escobillas: la obvia ausencia de escobillas que representan una gran desventaja debido a su desgaste y mantenimiento; los brushless son más livianos comparados con los que poseen escobillas de su misma potencia; y también presentan eficiencias y pares superiores por cada Watt consumido (Gómez, 2016).

El motor se alimenta con una señal de Modulación por Ancho de Pulso (PWM) que regula el voltaje promedio de entrada cambiando la velocidad de giro, de tal manera que cuando se requiera la máxima velocidad, el ciclo útil del PWM será del 100%, generando una entrada de voltaje promedio igual al bus de alimentación del sistema, mientras que cuando se requiera una velocidad de aproximadamente la mitad de la máxima alcanzada por el motor, el ciclo útil del PWM será de 50%, generando un voltaje promedio igual a la mitad del bus de alimentación del sistema (Delgado, 2013).



Figura 2.4: Motor brushless 2212 960Kv.

Controlador electrónico de velocidad

Los Controladores Electrónicos de Velocidad (Electronic Speed Controller, ESC) son los dispositivos encargados de coordinar y dar paso a la rotación y control de los motores brushless. Estas unidades se encargan de generar la secuencia de pulsos trifásicos que son requeridos para provocar el movimiento de los motores. La velocidad de giro se varía mediante una señal PWM. Los controladores típicos poseen tres salidas bidireccionales las cuales consisten en un arreglo de transistores de efecto de campo (FET's) controlados por un circuito lógico. En la mayoría de los casos este circuito es un microcontrolador que se encarga de administrar la aceleración, controlar la velocidad de rotación y aumentar la eficiencia adaptándose al sistema en el cual se encuentra trabajando el controlador (Gómez, 2016).



Figura 2.5: Controlador Electrónico de Velocidad Simonk.

Batería

Uno de los principales problemas de los vehículos aéreos multirrotores lo constituye el sistema de alimentación de energía. Dada la construcción de los motores que se usan en este tipo de dispositivos y la presencia de bobinas en los mismos (las bobinas se oponen a cambios en la corriente que circula por ellas), la fuente de alimentación debe suministrar unas intensidades de corrientes relativamente grandes para lo que suelen dar las baterías estándar. Además, conviene que el peso añadido por las baterías sea el menor posible, porque los motores tendrán que vencer una fuerza de peso mayor. En consecuencia con lo anterior, la batería empleada en el GARP X4 es 3DR 4S 6000 mAh 14.8 V 35C (Mahony et al., 2012).



Figura 2.6: Batería LiPo.

Hardware para el Control de Vuelo

El módulo de hardware integrado 3DR PX4 Pixhawk constituye el Controlador de Vuelo sobre el cual se montan los demás elementos actuadores y sensores: motores brushless, los controladores electrónicos de velocidad (ESC), módulo de antena GPS + brújula digital, sistema FPV y GoPro de cámaras, el receptor para control por radio frecuencias.



Figura 2.7: Controlador de Vuelo.

Unidad Inercial de Medidas (IMU)

El controlador de vuelo 3DR PX4 Pixhawk posee una Unidad Inercial de Medidas que no es más que un conjunto de sensores capaces de medir la velocidad, orientación y aceleraciones del vehículo. Estas mediciones son procesadas por el controlador de vuelo y son empleadas para calcular los cambios en las velocidades de los motores. La IMU dispone de una combinación de un Acelerómetro de 3 ejes, Giróscopo de 3 ejes, un Magnetómetro y un Barómetro.

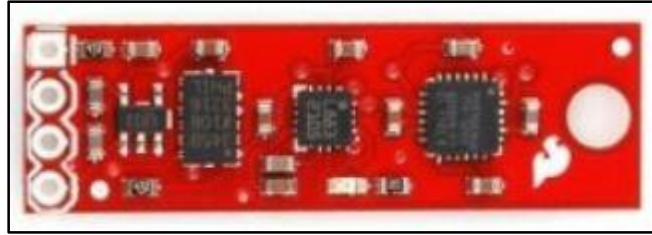


Figura 2.8: Unidad Inercial de Medidas (IMU).

2.2 Conceptos y definiciones generales asociados a fallos

Los Vehículos Aéreos No Tripulados son empleados desde su creación en misiones donde es necesario garantizar la seguridad de vuelo, es decir, el sistema debe trabajar correctamente a pesar de la presencia de fallos y averías durante su operación. Por tanto, se hace necesario definir los conceptos de fallos y averías (Sadeghzadeh and Zhang, 2011).

Fallo: es una desviación de al menos una característica o propiedad (comportamiento) del sistema de una condición estándar o de un comportamiento deseado (Johry, 2016).

Por ejemplo, la temperatura de un motor. Si la temperatura del motor sobrepasa un valor umbral máximo, existe un fallo en el sistema. Puede que no afecte el funcionamiento total del UAV durante un tiempo, pero puede llevar a un daño de ciertos componentes del motor y hacer que el sistema colapse. Para los sensores, una desviación de los datos adquiridos sobre un valor aceptable, se considera un fallo. El ruido excesivo, o valores falsos son catalogados como fallos en los sensores (Correcher et al., 2012).

Avería: es una interrupción permanente que impide al sistema mantener la capacidad de comportarse bajo ciertas especificaciones de operación (Johry, 2016).

Provenientes de uno o varios fallos, las averías se presentan de tal forma que degradan el funcionamiento del sistema tal y como se había comportado. El Control de Velocidad por Voltaje es empleado en el Quadcopter para accionar los motores brushless, en caso que mediante el ESC no se puedan accionar los motores, se puede decir que están averiados y no pueden usarse más.

Por último, pueden existir daños estructurales que pueden afectar de forma irreversible la estructura del UAV tales como la pérdida de un motor del Quadcopter u otra pieza del vehículo (Schafroth et al., 2010).

2.2.1 Análisis de los principales fallos en el Quadcopter

Los fallos a modelar en la aeronave se han seleccionado a partir de las características de los componentes del sistema y según la bibliografía consultada, también se ha tenido en cuenta el criterio de expertos del GARP con la ayuda de datos de los vuelos realizados con el Quadcopter mediante la monitorización de las diferentes variables con el Mission Planner.



Figura 2.7: Captura de pantalla principal Mission Planner.

Mission Planner es una aplicación de control y monitorización de vuelo para la estación de control en tierra compatible con el sistema ArduPilot (software del Quadcopter) que funciona sobre el sistema operativo Windows (Team, 2016). Puede ser empleado como una herramienta útil en la configuración de vuelo o como un suplemento de control dinámico para vehículos autónomos. Una de las características más importantes del software es la capacidad de controlar la aeronave sin la ayuda del mando a distancia y sin necesidad de preparar de antemano la ruta de vuelo (Modo Guiado).

Además, Mission Planner permite ejecutar los datos de vuelo para analizar los registros de las variables asociadas a los cinco fallos más comunes que afectan al Quadcopter durante una misión de vuelo; esto es posible de dos maneras: cargando el archivo Dataflash logs en la memoria *MicroSD* abordo del Pixhawk, de la cual se pueden descargar los datos de vuelo después de culminada la misión; y la otra, mediante el Telemetry logs (“tlogs”) con la cual los

datos de vuelo son grabados por Mission Planner cuando se conecta el ArduPilot a la computadora en la estación de tierra mediante el enlace con el Radio Telemetry (Team, 2016).

Los cinco fallos más comunes que pueden ser monitorizados con la ayuda del Mission Planner son:

1. Fallos Mecánicos: incluye fallos en los motores y en los controladores electrónicos de velocidad (ESC), entre ellos, deslizamientos de las propelas o desajustes. Su diagnóstico puede hacerse a partir de la observación de la diferencia entre las mediciones reales de los ángulos Roll y Pitch y los valores deseados de estos ángulos. Estas discrepancias pueden verse mejor en los Dataflash logs.
2. Fallos en los sensores por alto nivel de vibraciones mecánicas: los altos niveles de vibraciones mecánicas en la aeronave causan fallos en la medición del acelerómetro y la altura alcanzada. La mejor manera de visualizar el efecto de las vibraciones indeseadas es mediante el Dataflash de la IMU.
3. Fallos en las mediciones de la Brújula digital por interferencias magnéticas de los actuadores: están dadas por los efectos del sistema de distribución de potencia a bordo (Batería y conectores), motores, controladores de velocidad electrónicos y otros dispositivos electrónicos que pueden causar interferencias en las mediciones de la brújula digital (Digital Compass).
4. Fallo en la medición de la posición (GPS glitches): cuando el Quadcopter está volando en modos autónomos (Loiter, RTL, AUTO) los errores de posición dados por fallos en el GPS pueden provocar que la aeronave siga una ruta equivocada debido a los problemas en el cálculo de la posición y altura que son dados por el GPS; causando que el Quadcopter trate de corregir la ruta de vuelo programada de manera errónea.
5. Fallo en el suministro de energía: Las variaciones del voltaje suministrado al Pixhawk por encima de 0,15V son un signo de problemas con la energía. La introducción del módulo de potencia 3DR fue concebido con el objetivo de mejorar el suministro de energía en este tipo de aeronaves. Con esto se ha logrado reducir los apagones de energía (brownouts).

Si está habilitada la comunicación Radio Telemetry vía inalámbrica y está conectado al ArduPilot mediante el Mission Planner, el archivo de log de datos de vuelo se grabará automáticamente como un tlog en la carpeta Mission Planner's logs. Mientras se reproduce el archivo log se puede analizar el comportamiento de las variables que caracterizan cada uno de los componentes del Quadcopter. El comportamiento de las variables puede ser chequeado durante el tiempo que transcurrió el vuelo (Team, 2016), como se muestra en la figura 2.8.



Figura 2.8: Análisis de los datos de vuelo.

Después de analizar datos de diferentes misiones de vuelo real del Quadcopter y de revisar la bibliografía especializada (Martinez, 2016) (Johry, 2016) (Chamseddine et al., 2013) (Bupe, 2015) (Haddadi et al., 2015) (Hystad, 2015), los fallos a modelar son:

- Fallo en el Suministro de Energía (FSE): Ocurre cuando el voltaje suministrado por la batería LIPo del sistema es menor de 12V, si el voltaje de la batería cae por debajo de este umbral, existe una condición de fallo.
- Fallo Nivel de Voltaje del ESC (FNV): Se presenta cuando el voltaje de salida de los controladores de velocidad electrónicos de los motores es inferior a 7V debido a que este es voltaje mínimo de trabajo de los motores brushless 2212 960Kv.

- Fallo por Calentamiento del Motor (FCM): El umbral de temperatura de los motores del Quadcopter es 40 °C, por tanto, cuando sobrepase este valor aparecerá una condición de fallo.
- Fallo por Atascamiento del Motor (FAM): El umbral mínimo de RPM para que el motor funcione adecuadamente es de 6720 RPM que se corresponde con el voltaje mínimo del motor funcionando adecuadamente. Si la velocidad cae por debajo de este límite se presenta el fallo.
- Fallo en los Sensores de la IMU (FS): Está dado por un error en el funcionamiento de cualquiera de los tres sensores (Acelerómetro, Giróscopo, Magnetómetro) de la IMU. Mediante una señal de malfuncionamiento de uno de estos tres sensores se sabrá que existe un fallo en la IMU.
- Fallo en la Medición de Altura (FMA): Como el encargado de medir la altura del Quadcopter es el módulo GPS+Compass, un error en chequeo de la señal de recepción de los satélites (como mínimo debe estar sintonizado con cuatro satélites) encargados de ubicar la aeronave, conduce a un Fallo en la Medición de Altura.

2.3 PNTTool como herramienta de simulación de RdP

El Toolbox Petri Net (PNTTool) de MatLab es una herramienta de software para la simulación, análisis y diseño de Sistemas de Eventos Discretos, Continuos e Híbridos basados en Redes de Petri; creada por profesores del Departamento de Automática e Informática Aplicada de la Universidad “Gheorghe Asachi” de Iasi Rumanía. Mediante esta herramienta se pueden representar cinco tipos de Redes de Petri (sin temporizar, transiciones temporizadas, lugares temporizados, estocásticas y estocásticas generalizadas) con capacidad finita o infinita. El toolbox está equipado con una interfaz de usuario amigable y posee tres modos de simulación acompañados de animación gráfica. Sus funciones cubren los métodos claves de análisis de Redes de Petri tales como Árboles de Cobertura, Propiedades estructurales y dinámicas (incluyendo las invariantes P y T), los índices de comportamiento dependientes del tiempo, representaciones del espacio-estado con la función Max-plus (Matcovschi, 2003). La integración con la filosofía de MatLab amplía el uso de este software en el entorno de los Sistemas de Eventos Discretos. La aplicabilidad de este toolbox en varios dominios de la ingeniería demuestra su vigencia.

2.3.1 Interfaz Gráfica de Usuario (GUI) del PNTool

Existen dos modos en los cuales el PNTool puede ser empleado, el primero Modo Diseño (Draw Mode), que permite al usuario construir un nuevo Modelo de RdP o incluso modificar las propiedades de una red existente; y el otro, Modo de Exploración (Explore Mode) que posibilita al usuario acceder a las herramientas de simulación, análisis y diseño. La interfaz exhibe ocho paneles de control (ver figura 2.9): Barra de Menú (1), Barra de Herramientas de Rápido Acceso (2), Área de Diseño (3), Panel de Diseño (4), Switch Diseño/Exploración (5), Panel de Simulación (6), Panel de Estado (7) y Caja de Mensajes (8). A continuación, cada uno de los paneles y menús son descritos de manera breve.

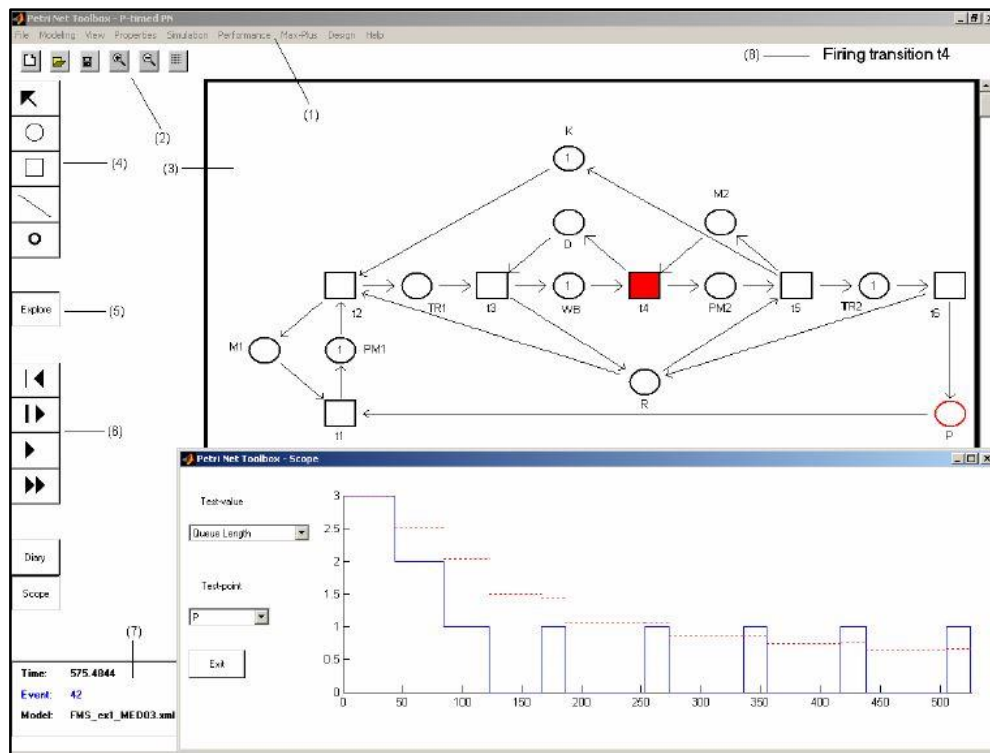


Figura 2.9: Captura de Pantalla del GUI Petri Net Toolbox.

La Barra de Menú (1) muestra un conjunto de seis submenús, desde los cuales se puede acceder a todas las facilidades disponibles en el toolbox. Estos submenús están habilitados en concordancia con el modo de explotación del PNTool.

El Menú de Archivos ofrece facilidades para las operaciones de manejo de archivos. Este es el único menú habilitado cuando el GUI del PNTool se inicia. El Menú de Modelado provee herramientas para la edición gráfica (nodos, arcos, tokens, labels) de modelos en el Área de

Diseño. El Menú de Vista permite elegir las condiciones específicas para la visualización del modelo actual. El Menú de Propiedades provee herramientas de cómputo para el análisis del comportamiento y propiedades estructurales del modelo de RdP actual. A través del Menú de Simulación el usuario puede controlar el progreso de la simulación y grabar los resultados. Al final del experimento de la simulación, el Menú de Desempeño permite la visualización del comportamiento global de los índices que son almacenados en un archivo de formato HTML. Estos índices son grabados separadamente para las transiciones y los lugares de la red. El Menú Max-plus permite representar la simulación y el análisis de un evento graficado (grafico marcado) basado en su modelo espacio-estado Max-plus. Si una nueva RdP en MatLab es abierta, todas las facilidades están disponibles para el análisis con la función Max-plus y la simulación. El Menú de Diseño es utilizado para la síntesis de modelos temporizados de RdP; esto permite simulaciones para diferentes tipos de parametrizaciones consideradas en la arquitectura de la RdP. El Menú de Ayuda proporciona información para la utilización del PNTTool y permite la visualización de cuatro Flash demos inicializándose el usuario en el empleo del PNTTool (Matcovschi, 2003).

El Área de Diseño (3) esta provista con una cuadrícula (grid), donde los nodos de la RdP serán ubicados, y con dos scrollbars (a la derecha y debajo) para moverse a través de toda la red y visualizarla. El Área de Diseño es un objeto de MatLab con ejes y está organizado como una matriz de celdas con 50 filas y 50 columnas. En una celda el usuario puede dibujar un nodo (lugar o transición).

El Panel de Diseño (4) presenta cinco botones con imágenes que le facilita al usuario acceder a los comandos: Edición de Objetos, Agregar un Lugar, Agregar una Transición, Agregar un Arco y Agregar Tokens. De forma similar, el Panel de Simulación (6) presenta botones de comando como Reinicializar (Reset), Paso (Step), Correr Lento (Run Slow) y Correr Rápido (Run Fast). Además, provee dos instrumentos para la visualización del progreso de la simulación: Diary y Scope (ver figura 2.9).

El toolbox cuenta con cuatro ejemplos tutoriales básicos para la mejor comprensión de la herramienta. Estos cuatro ejemplos cubren un área extensa de tópicos relacionados con automatismos e incorporan una animación siendo beneficioso especialmente para iniciados en el uso de estas herramientas. El Demo 1 se refiere a un Sistema Computacional con dos

procesadores compartiendo dos discos duros (en paralelo) el cuales una versión del conocido problema de “Los Dos Filósofos” (Dijkstra, 1968). El Demo 2 trata sobre un Sistema de Manufactura con un robot secuencial compartido (Al-Jaar, 1993) (DiCesare, 1993). El Demo 3 se refiere a un sistema de flujo de compras en una tienda con tres máquinas. Mientras el Demo 4 trata sobre una red abierta markoviana de colas (Cassandras, 1993).

Varios son los trabajos con RdP en los que se han empleado esta herramienta de simulación entre ellos destacan (Martinez, 2016) (Ramirez et al., 2007) (Giua and Seatzu, 2005).

2.4 Conclusiones parciales del capítulo

La caracterización de los fallos en este tipo de UAVs es importante como premisa para la construcción del modelo de fallos con Redes de Petri.

El Mission Planner posibilita monitorizar cinco fallos genéricos en el Quadcopter haciendo uso de los datalogs para el post procesamiento de vuelos, el cual permite caracterizar los parámetros de funcionamiento de cada subsistema.

Las ventajas que posee el PNTTool de MatLab como herramienta de simulación para las Redes de Petri están dadas por el amplio espectro de funciones que posee para el análisis de las principales propiedades de la RdP.

CAPÍTULO 3. Modelo de Fallos para el Quadcopter GARP X4 utilizando Redes de Petri

En este capítulo se procede a modelar los fallos mencionados en el Capítulo 2 utilizando las Redes de Petri (RdP). Siguiendo la misma metodología expuesta en el epígrafe 1.2 se describe el proceso de modelado de fallos con RdP, empleando el Toolbox de MatLab PNTTool para la simulación de la Red de Petri Integrada del Sistema, y a partir de esta, se verifican las principales propiedades de la RdP para corroborar el adecuado funcionamiento del modelo según las especificaciones del comportamiento del Quadcopter.

Además, se hace un análisis económico mencionando la ventaja que representa el poseer un Modelo de fallos para el Quadcopter, así como las pérdidas que se pueden evitar en daños materiales a la aeronave.

3.1 Aplicación de la Metodología para el Modelado de Fallos

Como se describió en el Capítulo 2 el UAV elegido para el modelo de fallos es un Quadcopter. Esta aeronave ha sido empleada en el Grupo de Automática, Robótica y Percepción (GARP) de la Universidad Central “Marta Abreu” de Las Villas en diversas misiones de vuelo para el mapeo de cultivos y reconocimiento de zonas geográficas. Un fallo en el mismo, como ha sucedido en varias ocasiones, puede ocasionar que fracasen las misiones que se planifiquen.

A continuación se procede a aplicar la metodología para el Modelado de Fallos. Previo a la aplicación del algoritmo, se deben tener en cuenta los siguientes supuestos:

- El Quadcopter debe ser encendido en modo manual.

- Se asume que ningún fallo ocurre en el Controlador de Vuelo, es decir no ocurren errores por concepto de programación.
- No existen fallos en el Sistema de Comunicaciones.

Para el modelado de fallos en el Quadcopter se ha centrado la atención en cinco subsistemas fundamentales dada la importancia que tienen en el funcionamiento de la aeronave. Estos son: Subsistema Batería, Subsistema Controlador Electrónico de Velocidad (ESC), Subsistema Motor, Subsistema Unidad Inercial de Medidas (IMU) y Subsistema Módulo GPS. Basado en la importancia de los cinco subsistemas fundamentales que conforman el Quadcopter, es en estos donde se enfoca la atención, seguido se realiza el modelo de RdP de los componentes que conforman a cada uno de los subsistemas, en cada dispositivo se tiene en cuenta su estado normal y de fallo. Entonces se construye el modelo general de RdP de cada subsistema, que se convertirá en un modelo refinado libre de transiciones no observables T_{uo} .

Paso 1. Clasificación en Subsistemas

El Quadcopter es dividido en cinco subsistemas fundamentales, el Subsistema Batería (H_1), el Subsistema Controlador Electrónico de Velocidad (H_2), el Subsistema Motor (H_3), el Subsistema IMU (H_4) y el Subsistema GPS (H_5).

- Subsistema Batería: es el encargado de suministrar la potencia a los motores y a los sensores y controlador de vuelo del Quadcopter. Está compuesto por una batería LIPo de 14.8V.
- Subsistema Controlador Electrónico de Velocidad: son cuatro, uno por cada motor del Quadcopter, son los encargados de convertir la corriente DC suministrada por la batería en una corriente trifásica que mueve los motores brushless.
- Subsistema Motor: son cuatro motores brushless que son accionados por sus respectivos ESC. Los motores son los encargados de darle suspensión y movimiento en cualquier dirección según la variación de voltaje, a la aeronave.
- Subsistema Unidad Inercial de Medidas: está integrada en el controlador de vuelo, pero por su función se considera un subsistema aparte. Los tres sensores que están integrados en su circuito (giroscopo 3-ejes, acelerómetro 3-ejes y un magnetómetro) son los encargados de medir posición, velocidad y aceleración del vehículo.

- **Subsistema Módulo GPS:** está compuesto por el GPS y una brújula digital integrada. Este subsistema es el encargado de posicionar la aeronave y de calcular la altura a la que se encuentra, a partir de la señal recibida de distintos satélites.

Para el funcionamiento de la electrónica en el Quadcopter se requiere una fuente de alimentación constante y con un valor de voltaje mínimo.

Paso 2. Construir el modelo de RdP de los componentes de cada subsistema

A continuación, se procede a la elaboración de los modelos de RdP de cada uno de los dispositivos que componen los subsistemas. Se seguirán las pautas definidas en el epígrafe 1.2. El sistema general cuenta con cinco subsistemas, $M = 5$.

Subsistema Batería H_1 : el subsistema tiene un solo componente a modelar: la batería LIPo H_1^1 , con un máximo de voltaje de 14.8V, el voltaje no puede caer por debajo de un umbral mínimo de 12V bajo ningún concepto, porque puede quemarse y quedaría inutilizable. Basado en la literatura especializada la única variable a evaluar será:

Voltaje de la Batería: estando cargada con 12.3V es suficiente para que el Quadcopter pueda retornar al punto de lanzamiento, según las prácticas de vuelo realizadas y los criterios de especialistas, por lo que se establece como condición de fallo un umbral de 12V según las experiencias de vuelos con el Quadcopter. Por tanto, el fallo a modelar es:

Fallo en el Suministro de Energía (FSE): el nivel máximo de voltaje de la batería LIPo es de 14.8V, para garantizar la misión sin daños, se debe mantener como nivel mínimo 12V porque en caso de caer por debajo de este umbral todos los sistemas del vehículo quedarían sin energía y la aeronave se caería al instante sin posibilidades de aterrizar sin daños.

En la figura 3.1 se observa el Modelo de RdP de la Batería, en él se manifiesta la integración del funcionamiento normal y del fallo nombrado. Se puede apreciar que las transiciones de fallo son no-observables y junto con los lugares de fallo son representados por barras y círculos en rojo.

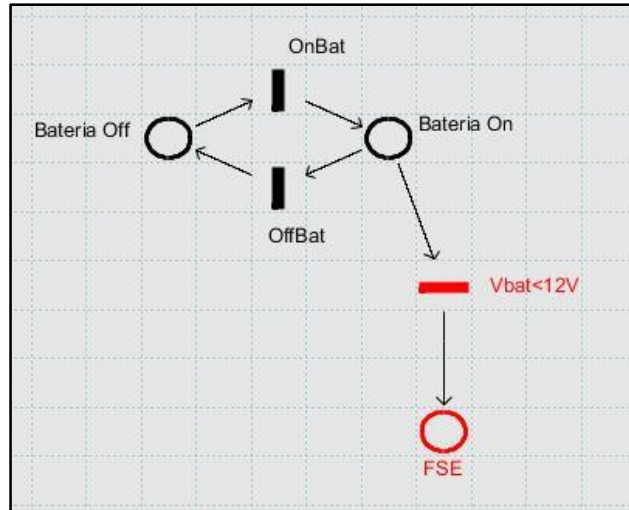


Figura 3.1: Modelo RdP Batería.

En el modelo de RdP la batería puede estar (comportamiento normal) en dos estados: encendida (Batería On) o apagada (Batería Off), solo se cambia de un lugar a otro mediante los mandos recibidos por el controlador de vuelo para el encendido (OnBat) y apagado de la batería (OffBat). Por otro lado se observa cómo se llega al lugar de fallo (FSE) a través de la transición $V_{bat} < 12V$.

Subsistema Controladores de Velocidad Electrónicos H_2 : el subsistema tiene un solo componente a modelar, el controlador electrónico de velocidad H_2^1 (son cuatro, uno por cada motor, pero se modelará uno porque los demás son iguales). El ESC tiene que suministrar una corriente DC para alimentar las tres fases del motor brushless, en un rango de 7-12V. La variable a evaluar será:

Voltaje de Salida del ESC: Como el rango de trabajo del motor es 7-12V entonces el ESC correspondiente tiene que alimentarlo con una corriente DC que no se salga de ese rango. Caso de salirse de esos límites el fallo a modelar es:

Fallo Bajo Nivel de Voltaje (FNV): este fallo aparecerá a consecuencia de un funcionamiento incorrecto (rotura de algún componente electrónico) del ESC que traiga como consecuencia un bajo nivel de voltaje de entrega al motor ($V_{out} < 7V$).

En la figura 3.2 se observa el modelo de RdP del Controlador Electrónico de Velocidad (ESC), presentándose al igual que el modelo de RdP de la batería, el comportamiento normal y de fallo de este subsistema.

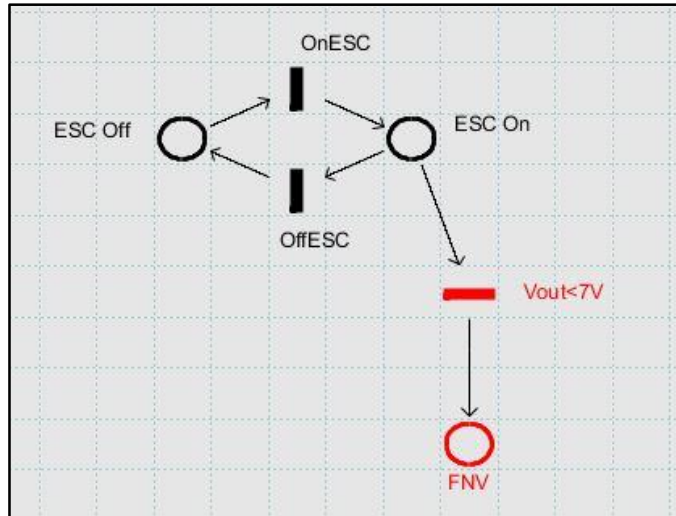


Figura 3.2: Modelo RdP ESC.

En el modelo de RdP del ESC este dispositivo puede estar en dos estados (normales): encendido (ESC On) o apagado (ESC Off), solo se cambia de un lugar a otro mediante los mandos de encendido (OnESC) y apagado (OffESC). Mientras que se llega al lugar de fallo (FNV) mediante la transición ($V_{out} < 7V$).

Subsistema Motores H_3 : está compuesto por un solo elemento a modelar: el motor brushless (son cuatro pero como son idénticos se modelará uno como muestra). Trabajan en un rango de temperatura de -5°C a 40°C y con velocidades características entre 11510rpm(12V) a ~6720rpm(7V). Por lo que las variables a seguir serán:

Temperatura de trabajo del Motor: como el rango de temperatura permisible en el motor es -5°C a 40°C , entonces un valor por encima de los 40°C sería una condición de fallo.

Velocidad del Motor (RPM): el motor necesita unas RPM mínimas (6720) que garanticen el giro de las palas y unas RPM máximas (11520) que no sobrecargue la aeronave, se hace necesario medirlas y mantener un nivel mínimo en vuelo normal. Las variaciones de las RPM evidencian posibles problemas en el nivel de voltaje que se le suministra al motor.

En consecuencia, los fallos a modelar en los motores del Quadcopter son:

Fallo por Calentamiento del Motor (FCM): el umbral de temperatura normal del motor es de 40°C si el motor pasa por encima de este umbral existe una condición de fallo.

Fallo por Atascamiento del Motor (FAM): el número de RPM mínimas que garantizan un adecuado funcionamiento del motor, y por consiguiente del vehículo, es de 6720 rpm velocidad que se corresponde con el valor de voltaje mínimo con que trabaja el motor del Quadcopter. Por tanto, en caso de disminuir la velocidad de los motores por debajo de este valor umbral aparece un fallo.

En la figura 3.3 se muestra el modelo de RdP del Subsistema Motor diferenciándose ambos comportamientos de manera clara: normal y de fallo; tal y como se representó en los casos anteriores.

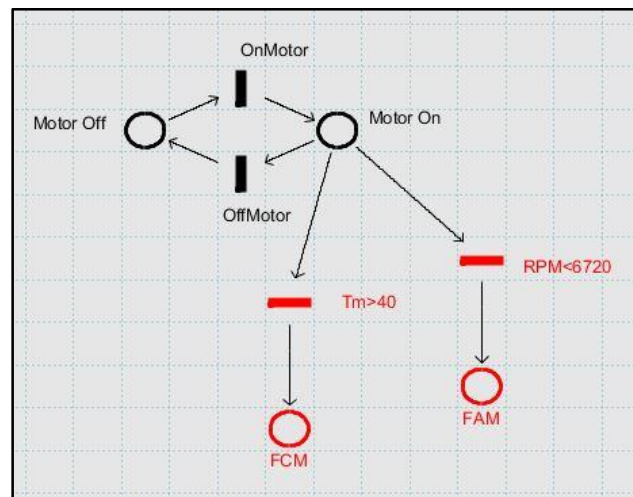


Figura 3.3: Modelo RdP Motor.

En el modelo de RdP del motor se observan dos estados en el comportamiento normal: Motor encendido (Motor On) y Motor apagado (Motor Off), se cambia de un estado a otro mediante los mandos recibidos del controlador de vuelo: encendido (OnM) y apagado (OffM). Los lugares de fallo están representados por FCM (Fallo Calentamiento Motor) y FAM (Fallo por Atascamiento del Motor) con sus respectivas transiciones.

Subsistema Unidad Inercial de Medidas H_4 : está compuesta por un solo elemento a modelar: la IMU. Esta tiene integrado en un mismo circuito los tres sensores básicos de una IMU: giróscopo, acelerómetro y magnetómetro. Esta unidad entrega al Pixhawk las lecturas de los sensores. El elemento a validar será el funcionamiento correcto de cada uno de los sensores que la conforman ($A * G * M$).

Fallo en los Sensores de la IMU: elementos como el magnetómetro son muy sensibles a golpes que pueden ocasionar una rotura del mismo, siendo necesario chequear los datos de las lecturas de cada elemento, para conocer si existen fallos en los sensores de la IMU que son los encargados de medir posición, velocidad y aceleración del vehículo. De no validarse las lecturas en cada sensor se presenta una condición de fallo. Como se supuso al inicio del Capítulo 3 que no existen errores en la programación del controlador de vuelo y los dispositivos conectados a él (entre los que se encuentra la IMU), se descarta un error en la comunicación que provoque una invalidación de la trama de datos con las lecturas de los sensores que entrega la IMU al controlador de vuelo.

En la figura 3.4 se representa el modelo de RdP del Subsistema IMU. Destacan dos estados en comportamiento normal: encendida (IMU On) y apagada (IMU Off), se cambia de un estado a otro mediante los mandos OnIMU y OffIMU. El lugar de fallo es (FS) Fallo en los sensores de la IMU.

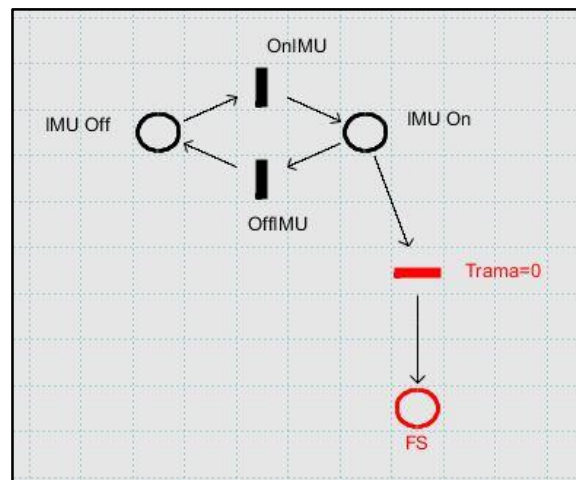


Figura 3.4: Modelo RdP IMU.

Subsistema Módulo GPS H_4 : está compuesto por un solo elemento a modelar el GPS a pesar de poseer una brújula digital integrada, porque para nuestro propósito solo es necesario el GPS; este es el encargado de ubicar la aeronave en el espacio pero también es quien mide la altura del Quadcopter.

Como el GPS funciona según la conexión que este posea con varios satélites, el elemento a validar es si existe o no conexión del módulo con los cuatro satélites (conexión/ no conexión).

Fallo en la Medición de Altura (FMA): la conexión del módulo GPS con al menos cuatro satélites es necesario para que este pueda ubicar el vehículo y medir la altura, si se verifica esta conexión y se detecta que no existe, aparece una condición de fallo y no se logra medir correctamente la altura a la que vuela el Quadcopter.

En la figura 3.5 se representa el modelo de RdP del Subsistema Módulo GPS. Destacan dos estados en comportamiento normal: encendida (GPS On) y apagada (GPS Off), se cambia de un estado a otro mediante los mandos OnGPS y OffGPS. El lugar de fallo es (FMA) Fallo en la Medición de Altura.

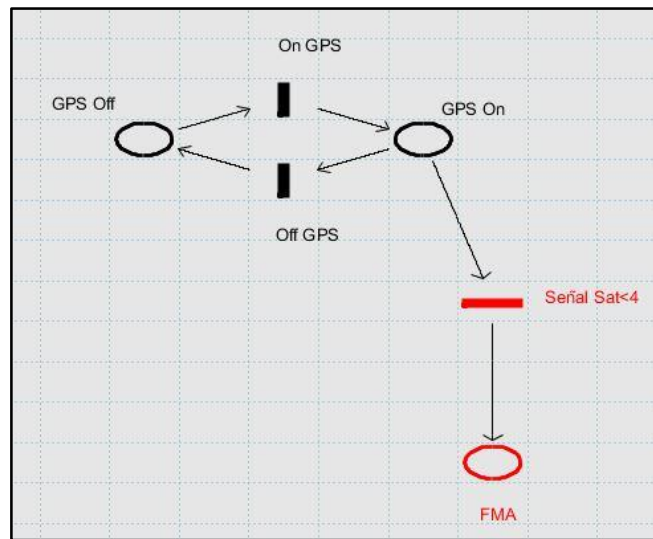


Figura 3.5: Modelo RdP Módulo GPS.

Paso 3. Operación de Integración

No es necesario integrar el modelo de cada subsistema debido a que los lugares normales están unidos y los de fallo bien identificados, como se explica en el epígrafe 1.2.

Paso 4. Operación de Refinamiento

Posterior a la representación del modelo general de cada subsistema se hace necesario refinar el modelo de RdP General, porque se deben sustituir las transiciones de fallo (no observables T_{uo}) por transiciones observables T_o ; este proceso se realiza basado en las variables medidas T_s (sensores) con que cuenta el sistema. Para ello se construye la tabla de integración de sensores, que es la productoria de las salidas de las lecturas sensoriales, comparadas con las lecturas esperadas del lugar. Cuando el subsistema está en cualquier lugar de funcionamiento

normal, los sensores pueden entregar medidas diferentes a las esperadas, indicando la presencia de fallo. La lectura sensorial reemplaza a la T_{uo} de fallo y el modelo de RdP es así como queda compuesto solo de T_o y por lo tanto, queda refinado. La tabla de integración de sensores se construye individual para cada subsistema.

Paso 3.1 Identificar los Sensores N_s para cada subsistema

Esto se hace acorde a los sensores requeridos para la monitorización de las variables deseadas. El conjunto de sensores del Motor está compuesto por el sensor de temperatura (T) y el sensor de velocidad (RPM); por tanto el conjunto N_s para el subsistema motor está definido como sigue: $N_{s1} = T, RPM$. Siguiendo el mismo procedimiento el conjunto de sensores del ESC está compuesto solo por el sensor de voltaje (V_{out}), siendo $N_{s2} = V_{out}$. Mientras el conjunto de sensores de la IMU está dado por: Giróscopo (G), Acelerómetro (A) y Magnetómetro (M); por tanto $N_{s3} = G, A, M$. El conjunto de sensores del subsistema Batería está compuesto solo por sensor de voltaje de la batería (V_{bat}), siendo $N_{s3} = V_{bat}$. Por último el subsistema Módulo GPS está compuesto, para el caso particular de esta investigación, por el GPS; $N_{s3} = GPS$.

Paso 3.2 Construir el Conjunto Discreto de Salidas de los Sensores

Cada sensor del sistema entregará una medida discreta que define su estado; con estas medidas se construye el conjunto Y de salidas de los sensores y permite establecer con cuales lecturas se cuenta en la evaluación de cada subsistema.

- Subsistema Motor: compuesto por el sensor de velocidad (RPM) y el sensor de temperatura (T), $|Y| = 2^{N_s} = 2^2 = 4$ combinaciones.
- Subsistema ESC: compuesto por el sensor de voltaje salida del ESC (V_{out}), siendo $|Y| = 2^1 = 2$.
- Subsistema IMU: compuesto por acelerómetro, giróscopo y magnetómetro, teniendo $|Y| = 2^3 = 8$.
- Subsistema Batería: compuesto por el sensor de voltaje batería (V_{bat}), siendo $|Y| = 2^1 = 2$.
- Subsistema Módulo GPS: compuesto por el GPS, teniendo $|Y| = 2^1 = 2$.

Tabla 3.1: Conjunto Discreto de Salidas de Sensores del Motor.

| Combinaciones | Lecturas Motor | |
|---------------|----------------|------------------|
| | T | RPM |
| 1 | \bar{T} | \overline{RPM} |
| 2 | \bar{T} | RPM |
| 3 | T | \overline{RPM} |
| 4 | T | RPM |

Tabla 3.2: Conjunto Discreto de Salidas de Sensores del ESC.

| Combinaciones | Lectura ESC |
|---------------|----------------------|
| | V_{out} |
| 1 | V_{out} |
| 2 | $\overline{V_{out}}$ |

Tabla 3.3: Conjunto Discreto de Salidas de Sensores de la Batería.

| Combinaciones | Lecturas Batería |
|---------------|----------------------|
| | V_{bat} |
| 1 | $\overline{V_{bat}}$ |
| 2 | V_{bat} |

Tabla 3.4: Conjunto Discreto de Salidas de Sensores de la IMU.

| Combinaciones | Lecturas IMU |
|---------------|---------------------------------------|
| | $A \quad G \quad M$ |
| 1 | $\bar{A} \quad \bar{G} \quad \bar{M}$ |
| 2 | $\bar{A} \quad \bar{G} \quad M$ |
| 3 | $\bar{A} \quad G \quad \bar{M}$ |
| 4 | $A \quad \bar{G} \quad \bar{M}$ |
| 5 | $A \quad G \quad \bar{M}$ |
| 6 | $A \quad \bar{G} \quad M$ |
| 7 | $\bar{A} \quad G \quad M$ |
| 8 | $A \quad G \quad M$ |

Tabla 3.5: Conjunto Discreto de Salidas de Sensores del GPS.

| Combinaciones | Lecturas GPS |
|---------------|------------------|
| | GPS |
| 1 | \overline{GPS} |
| 2 | GPS |

Paso 3.3 Definir las Salidas de la Tabla de Integración de Sensores

Para cada lugar del Modelo General RdP de cada subsistema se agrega una salida en la tabla de integración de sensores, esta salida contiene la representación de las lecturas del subsistema en un estado de funcionamiento normal.

- Subsistema Motor: dos salidas han sido adicionadas para el subsistema (P_3 y P_4). Bajo condiciones de funcionamiento normal el estado de la lectura de temperatura \bar{T} (temperatura del motor está por debajo de los 40°C) y el estado de velocidad del motor debe ser RPM (las revoluciones por minuto están por encima del umbral mínimo 6720 RPM).

Tabla 3.6: Tabla de Integración de Sensores Motor.

| Lecturas de Sensores | $P_3 (\bar{T}, RPM)$ | $P_4(\bar{T}, RPM)$ |
|--------------------------------|----------------------|---------------------|
| $\bar{T} \quad \overline{RPM}$ | $N \quad FAM$ | $N \quad FAM$ |
| $\bar{T} \quad RPM$ | $N \quad N$ | $N \quad N$ |
| $T \quad \overline{RPM}$ | $FCM \quad FAM$ | $FCM \quad FAM$ |
| $T \quad RPM$ | $FCM \quad N$ | $FCM \quad N$ |

- Subsistema IMU: dos salidas han sido definidas para este subsistema (P_9 y P_{10}). Bajo condiciones de operación normal la lectura del acelerómetro será A (correcta lectura aceleración en 3 ejes), la lectura del giróscopo G (correcta lectura de la velocidad angular en 3 ejes) y la lectura del Magnetómetro M (correcta lectura de orientación magnética).

Tabla 3.7: Tabla de Integración de Sensores IMU.

| Lecturas Sensores | $P_9 (A, G, M)$ | $P_{10}(A, G, M)$ |
|---------------------------------------|------------------------|------------------------|
| $\bar{A} \quad \bar{G} \quad \bar{M}$ | $FS \quad FS \quad FS$ | $FS \quad FS \quad FS$ |
| $\bar{A} \quad \bar{G} \quad M$ | $FS \quad FS \quad N$ | $FS \quad FS \quad N$ |
| $\bar{A} \quad G \quad \bar{M}$ | $FS \quad N \quad FS$ | $FS \quad N \quad FS$ |

| | | |
|---------------------------------|-----------------------|-----------------------|
| $A \quad \bar{G} \quad \bar{M}$ | $N \quad FS \quad FS$ | $N \quad FS \quad FS$ |
| $A \quad G \quad \bar{M}$ | $N \quad N \quad FS$ | $N \quad N \quad FS$ |
| $A \quad \bar{G} \quad M$ | $N \quad FS \quad N$ | $N \quad FS \quad N$ |
| $\bar{A} \quad G \quad M$ | $FS \quad N \quad N$ | $FS \quad N \quad N$ |
| $A \quad G \quad M$ | $N \quad N \quad N$ | $N \quad N \quad N$ |

- Subsistema ESC: dos salidas han sido definidas para este subsistema (P_5 y P_6). Bajo condiciones normales de funcionamiento la lectura de voltaje de salida del ESC V_{out} (voltaje de salida del ESC por encima de los 7V).

Tabla 3.8: Tabla de Integración de sensores ESC.

| Lecturas de Sensores | $P_5 (V_{out})$ | $P_6 (V_{out})$ |
|----------------------|-----------------|-----------------|
| $\overline{V_{out}}$ | FNV | FNV |
| V_{out} | N | N |

- Subsistema Batería: dos salidas han sido definidas para este subsistema (P_7 y P_8). Bajo condiciones normales de funcionamiento la lectura de voltaje de batería V_{bat} (voltaje de la batería por encima de los 12V).

Tabla 3.9: Tabla de Integración de Sensores Batería

| Lecturas de Sensores | $P_7 (V_{bat})$ | $P_8 (V_{bat})$ |
|----------------------|-----------------|-----------------|
| $\overline{V_{bat}}$ | FSE | FSE |
| V_{bat} | N | N |

- Subsistema GPS: dos salidas han sido incorporadas a la tabla (P_{11} y P_{12}). La lectura del GPS deber estar en GPS (señal de 4 satélites adquirida por el módulo GPS) para que esté en funcionamiento correcto.

Tabla 3.10: Tabla de Integración de Sensores GPS.

| Lecturas de los sensores | $P_{11}(GPS)$ | $P_{12}(GPS)$ |
|--------------------------|---------------|---------------|
| \overline{GPS} | FMA | FMA |
| GPS | N | N |

Paso 3.4 Construcción de las tablas de integración de sensores

Las lecturas de los sensores (Y) tienen que verificarse de manera cruzada con la correspondiente lectura de funcionamiento normal de cada subsistema. Las lecturas de los sensores permiten identificar qué lectura de los sensores es Normal (N), de Fallo (F) o es una lectura que no brinda información útil alguna (X). Este paso requiere un profundo conocimiento del sistema.

En el subsistema Motor se utiliza la información de cada sensor, solo o en unión con los otros sensores para determinar los posibles fallos, por ello, estando el subsistema Motor en el estado P_4 sus lecturas de comportamiento normal son (\bar{T}, RPM) . Si en algún momento estas lecturas cambian, mostrando que exista o no exista calentamiento del motor T , por ejemplo; esto significa que ha ocurrido un Fallo por Calentamiento del Motor, por el contrario, cuando las lecturas de los sensores coinciden con las esperadas, se registra en la tabla como un evento normal. Del mismo modo se realizan todas las combinaciones y lecturas de los demás sensores.

Paso 3.5 Reemplazar las Transiciones de Fallo y Eliminar los Lugares de Fallo No Alcanzables

Este paso consiste en reemplazar las transiciones no observables (transiciones de fallo) por lecturas sensoriales identificadas en la tabla de integración de sensores.

- Subsistema Motor: en el modelo de RdP se procede a sustituir la transición de FCM por la lectura de temperatura (T) y la transición de FAM por la lectura de velocidad (RPM).
- Subsistema IMU: el subsistema IMU depende del funcionamiento de los tres sensores integrados: Acelerómetro, Giróscopo y Magnetómetro, por ello en el modelo de RdP

se sustituyen la transición FS por el producto de las lecturas de los sensores $A * G * M$.

- Subsistema ESC: como el funcionamiento del ESC depende de la lectura de voltaje a la salida del mismo, se procede a sustituir la transición FNV por la lectura del sensor V_{out} .
- Subsistema Batería: la lectura de voltaje de la batería es la que identifica el funcionamiento del subsistema, entonces se sustituye la transición FSE por la lectura del sensor de voltaje de la batería V_{bat} .
- Subsistema Módulo GPS: en el modelo de RdP se ha sustituido el FMA por la lectura del GPS .

La nueva RdP refinada de estos subsistemas se puede apreciar en el Anexo I, mientras que la tabla donde se definen los estados y las transiciones aparece en el Anexo IV.

3.2 Verificación del Modelo

Uno de los principales objetivos del modelado de sistemas concurrentes y paralelos empleando Redes de Petri es su análisis automático. En general, verificar la corrección de un sistema modelado consiste en determinar que ciertas propiedades se satisfacen. A continuación se mostrará el significado de algunas de ellas:

- Una Red de Petri está *acotada* cuando cada uno de sus lugares está acotado, es decir, cuando el número de tokens que contiene es menor que cierta cota para cualquier marcado alcanzable (cuando dicha cota es 1 se dice que la red es segura).
- Una Red de Petri se dice que está *viva* cuando cada una de sus transiciones está viva, es decir, cuando desde cualquier marcado alcanzable puede dispararse cualquier transición.
- Una Red de Petri está *libre de interbloqueos* cuando en cualquier marcado alcanzable hay alguna transición habilitada.
- Un marcado es un *estado home* cuando es alcanzable desde cualquier estado alcanzable y una red es *reversible* cuando el marcado inicial es un *estado home*.
- Dos lugares están en *exclusión mutua* cuando nunca están marcados simultáneamente.

La *acotabilidad* excluye desbordamientos, la *vivacidad* asegura que ninguna acción en el sistema pueda resultar inalcanzable, la existencia de *estados home* informa de la posibilidad de volver a ciertos estados y la *exclusión mutua* se requiere entre lugares que representan el uso de un recurso común o la presencia en una sección crítica.

Los principales métodos de análisis para la verificación son:

1. Análisis por reducción o transformación.
2. Árbol de alcanzabilidad.
3. Método de Eliminación de Gauss-Jordan.
4. Ecuación de Estado.
5. Método del Teorema del rango para redes de libre elección.

Todas estas técnicas no se entienden mutuamente exclusivas, sino que deben combinarse para un análisis práctico. A continuación, se emplearán las técnicas de Árbol de alcanzabilidad y Ecuación de Estados para verificar el cumplimiento de algunas propiedades de la red.

3.2.1 Árbol de Alcanzabilidad

En el árbol de alcanzabilidad de una red deben representarse los cambios de estados debido a los disparos de cada una de las transiciones. Como se ha visto en el Capítulo 1, si se dispara una transición, se ve afectado el marcado de la RdP. La enumeración correcta de todos los marcados alcanzables de una red permite lograr una adecuada descripción del sistema modelado. El árbol de alcanzabilidad es una herramienta de análisis de sistemas que toma como raíz el marcado inicial de la RdP y, a partir de ahí, mediante la ejecución de las transiciones, se llega a enumerar todos los marcados posibles de la red.

Se comprenderá que, en dependencia del número de marcados que presente la red, será el tamaño del árbol de alcanzabilidad y que este podría crecer indefinidamente para redes complejas o ilimitadas y hacer imposible la construcción del mismo. Debido a la complejidad de la Red de Petri General Refinada (ver Anexo II) a la que se arriba como modelo de fallos para el Quadcopter, solo se presentará un árbol de alcanzabilidad parcial que representa los marcados principales que describen el funcionamiento del Quadcopter en cada una de sus etapas de vuelo.

En este caso se describen los marcados posibles entre el marcado inicial (M_i) que presenta los tokens (marcas) de la siguiente manera: P_4 (3), P_6 (2), P_8 (2), P_9 (2), P_{11} (2). La dinámica representada en el árbol de alcanzabilidad muestra la evolución del sistema pasando por los estados P_{13} (Listo para volar), P_{14} (Vuelo sin fallos), P_{15} (Vuelo con fallos) hasta llegar al estado P_{30} (Retorno al punto de lanzamiento).

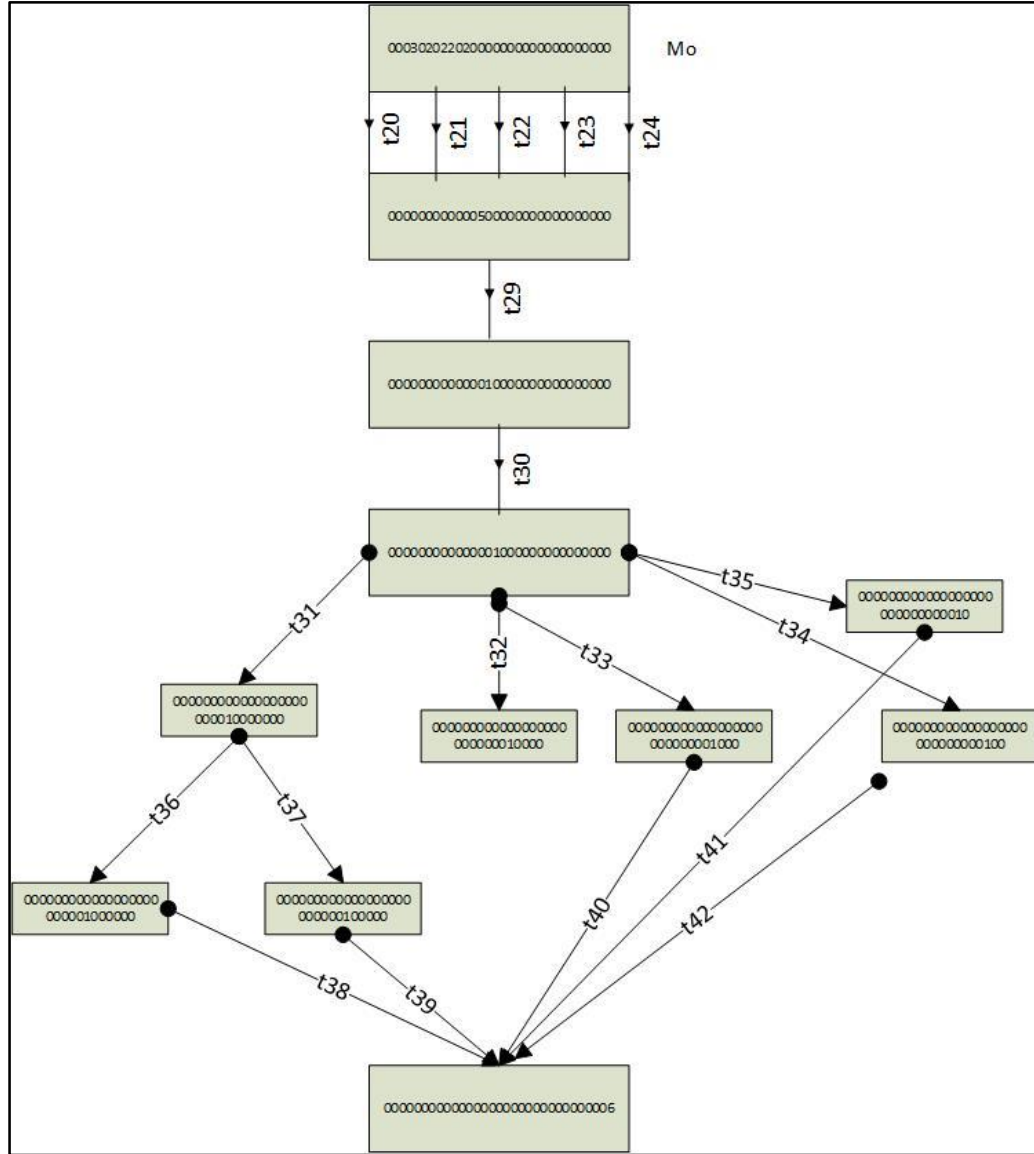


Figura 3.6: Árbol de alcanzabilidad parcial de la Red de Petri.

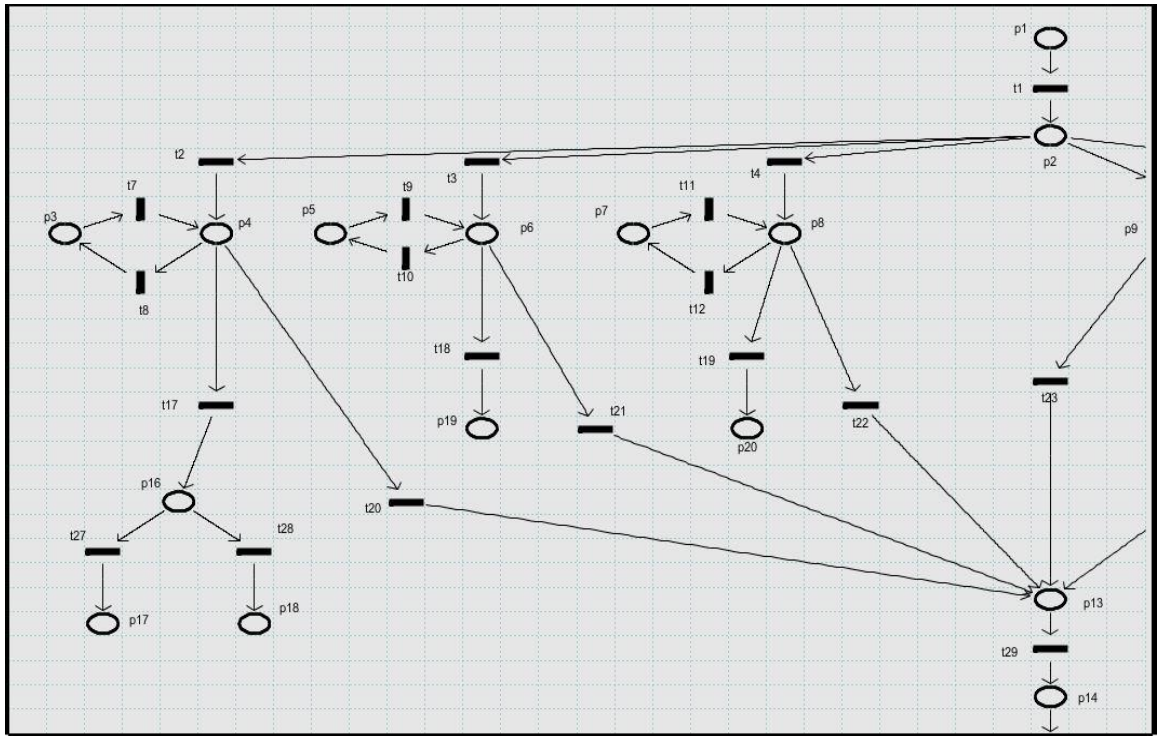


Figura 3.7: Captura de pantalla del espacio de trabajo del PNTTool.

El procedimiento para modelar en el Petri Net Toolbox de MatLab comienza por crear un nuevo modelo en el Menú de Archivos que es el único habilitado, luego aparece una ventana donde se muestran las opciones a escoger del tipo de RdP a diseñar (Estocástica, Estocástica Generalizada, Lugares Temporizados, Transiciones Temporizadas, Sin temporizar) se marca una de ellas y aparecen habilitadas las opciones del Panel de Diseño.

Después de terminar el diseño de la Red de Petri se pueden modificar las probabilidades de la ocurrencia de conflictos estructurales en la red, mediante el Menú de Modelado se accede a la opción de Resolución de Conflictos en Transiciones; por defecto el toolbox le da una probabilidad igual de ocurrencia a cada una de las transiciones en conflicto estructural, por ejemplo, en la figura 3.8 se aprecia que t8, t17 están en conflicto estructural y funcional, dándole el software una probabilidad de ocurrencia a cada transición de 0.5.

Como muestra de las simulaciones se han presentado dos marcados que describen el funcionamiento del Quadcopter en las diferentes etapas de vuelo (ver Anexo III). Las capturas de pantallas muestran dos situaciones: la primera Fallos en los subsistemas Motor, Batería y GPS, en estado previo al vuelo (este marcado se corresponde con el ejemplo de la

ecuación de estado); mientras el segundo responde a Fallos en el Motor, en el Controlador Electrónico de Velocidad y en el GPS, en estado de vuelo.

3.3 Análisis económico y medioambiental

El modelo de fallos basado en Redes de Petri propuesto para el Quadcopter GARP X4 constituye la base para el diseño del diagnosticador, en su conjunto (Modelo y Diagnosticador) trabajan en la monitorización y diagnóstico de fallos del Quadcopter y prevén la aparición de estos, evitando pérdidas económicas sustanciales por concepto de compras de piezas de repuestos para el UAV, ver tabla 3.11. En cuanto al tema medioambiental, el hecho de poder realizar misiones exitosas previendo los fallos a partir de un Modelo y Diagnosticador, posibilitará efectuar misiones de reconocimiento en zonas intrincadas donde se preserve la fauna, la prevención de incendios forestales, el mapeo de cultivos para el riego inteligente.

Tabla 3.11: Precios de los componentes del Quadcopter GARP X4.

| Elementos | Precios |
|---|------------|
| Frame DIY de 4 ejes | 17.70 USD |
| Módulo de 4 Motores Brushless 2212 920Kv + 4 ESC Simonk | 41.32 USD |
| Controlador de Vuelo Pixhawk | 100 USD |
| Módulo GPS+Digital Compass (Ublox) | 20 USD |
| Módulo Radio Telemetría | 30 USD |
| Batería LIPo 4S 6000 mAh 35C | 98.98 USD |
| Total | 308.09 USD |

3.4 Conclusiones parciales del capítulo

El modelado de fallos con Redes de Petri es posible a partir de un estudio profundo y la caracterización de los fallos más comunes que se presentan en la aeronave.

Los fallos fueron modelados con Redes de Petri partiendo de un correcto estudio de los parámetros que describen el funcionamiento de cada uno de los subsistemas del Quadcopter, siendo posible esto con la ayuda del Mission Planner.

El uso de la herramienta de simulación PNTTool de MatLab posibilitó verificar las principales características de las RdP propuesta, posibilitando demostrar la funcionabilidad de la misma y de la capacidad de describir el funcionamiento del sistema (Quadcopter) en estado normal y de fallo.

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

Como resultado final de esta investigación se propone un modelo de fallos basado en Redes de Petri capaz de representar el comportamiento normal y en fallo del Quadcopter GARP X4, lo cual queda demostrado mediante la simulación con el software Petri Net Toolbox de MatLab. A partir de estos resultados se plantean las conclusiones generales siguientes:

- 1 Las Redes de Petri, como herramienta de modelado de Sistemas Híbridos, posibilitan el modelado de fallos en el Quadcopter GARP X4 siendo posible representar las condiciones de fallos en cada subsistema a partir de un estudio de los parámetros asociados a cada tipo de fallo.
- 2 Con el empleo de la herramienta de software Petri Net Toolbox de MatLab se verifica el funcionamiento adecuado de la Red de Petri haciéndose un análisis de las principales características fundamentales de las RdP tales como la vivacidad, alcanzabilidad, reversibilidad y acotamiento.
- 3 El modelo propuesto cumple las condiciones planteadas para la representación del comportamiento normal y en fallo del Quadcopter GARP X4 quedando demostrado que la implementación con Redes de Petri satisface los requisitos de comportamiento del sistema.
- 4 La Simulación y Verificación del modelo de fallos del UAV Quadcopter GARP X4 empleando la herramienta PNTTool del MatLab demostró la efectividad del mismo como aplicación para la descripción del funcionamiento del sistema.

Recomendaciones

Con el fin de lograr mejoras en el modelo de fallos propuesto en el presente trabajo se hacen las siguientes recomendaciones:

- 1 Incorporar los resultados del post procesamiento de cada una de las variables que describen el funcionamiento en pleno vuelo del Quadcopter, con la ayuda del Mission Planner; y a partir de un análisis profundo, identificar aquellas variables que pueden aportar información a la hora de definir aquellas transiciones del modelo que no han sido debidamente caracterizadas.
- 2 Diseñar el Diagnosticador a partir del modelo de fallos propuesto en esta investigación teniendo en cuenta consideraciones dinámicas del sistema en cuestión.
- 3 Integrar los conocimientos de tesis relacionadas con el diagnóstico y sistematización de fallos en el Quadcopter para enriquecer esta investigación y mejorar el modelo de fallos propuesto.

REFERENCIAS BIBLIOGRÁFICAS

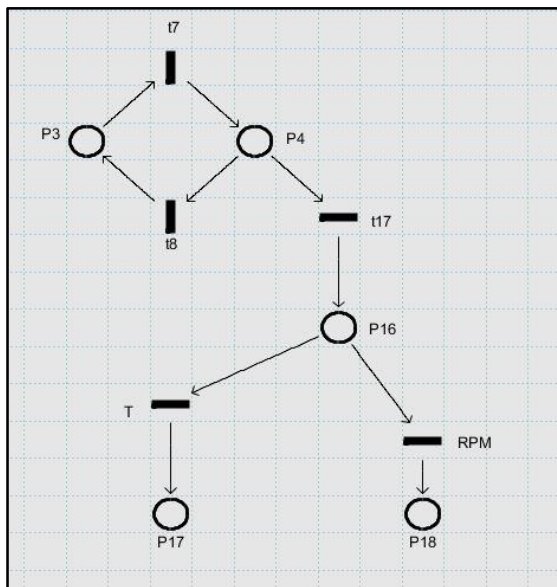
- AL-JAAR 1993. Modeling and Control of Automated Manufacturing Systems. *IEEE Computer Society Press*.
- AVILA, D. D. 2008. Estrategias de Control basado en modelo para minihelicópteros no tripulados.
- BUPE, P. 2015. Ultra-Fast, Autonomous, Reconfigurable Communication System.
- CABASINO, M. P., GIUA, A. & SEATZU, C. 2010. Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*, 46.
- CARRILLO, L. R. G., LÓPEZ, E. D., LOZANO, R. & PÉGARD, C. 2013. Quad Rotorcraft Control. *Springer-Verlag*.
- CASSANDRAS, C. G. 1993. Discrete Event Systems: Modeling and Performance Analysis.
- CHAMSEDDINE, A., THEILLIOL, D., ZHANG, Y. M., CJOIN & RABBATH, C. A. 2013. Active fault-tolerant control system design with trajectory re-planning against actuator faults and saturation: Application to a quadrotor unmanned aerial vehicle. *International Journal of Adaptive Control and Signal Processing*.
- CHENG, S. & JENG, M. 2003. Failure Diagnosis: A case study on Modelling and Analysis by Petri Nets. *IEEE*.
- CORRECHER, A., GARCIA, E., MORANT, F., QUILES, E. & RODRIGUEZ, L. 2012. Intermittent failure dynamics characterization. *IEEE Transactions on Reliability*.
- DELGADO, J. D. 2013. Control de Velocidad para Motor Brushless DC sin sensores.
- DICESARE, F. 1993. Petri Net Synthesis for Discrete Event Control of Manufacturing Systems.
- DIJKSTRA, E. W. 1968. Co-operating sequential processes. *Programming Languages, New York Academic*, 43-112.
- E.GARCIA, A.CORRECHER, F.MORANT, E.QUILES & R.BLASCO 2005. Modular Fault Diagnosis Based on Discrete Event Systems. *Discrete Event Dynamic Systems*.
- GARCÍA, E. P., MARTINEZ, P., MARTINEZ, D. O. & LORENZO, F. B. 2016. Vehículos Aéreos No Tripulados en Cuba aplicados a la Geomática. Perspectivas y Desarrollo.
- GENC, S. & LAFORTUNE, S. 2007. Distributed Diagnosis of Place-Bordered Petri Nets. *IEEE Transactions on Automation Science and Engineering*, 4.

- GIUA, A. & SEATZU, C. 2005. Fault detection for discrete event systems using Petri Nets with unobservable transitions. *44th IEEE Conference on Decision and Control, Seville*.
- GÓMEZ, R. C. 2016. Construcción de un cuadricóptero de bajo coste basado en herramientas de software libre.
- HADDADI, S. J., ZARAFSHAN, P. & NIROUMAND, F. J. 2015. Dynamics Modelling and Implementation of an Attitude Control on an Octorotor. *Proc. of IEEE 28th Canadian Conference on Electrical and Computer Engineering*.
- HANSEN, S. & BLANKE, M. 2012. Diagnosis of Airspeed Measurement Faults for Unmanned Aerial Vehicles. *SM IEEE*.
- HU, B. & SEILER, P. 2015. Pivotal Descomposition for Analysis of Fault Tolerant Control Systems on Unmanned Aerial Vehicles. *Reliability Engineering and System Safety*.
- HUAYNA, A. M., CORTEZ, A. & VEGA, H. 2009. Aplicación de las Redes de Petri a la Simulación Discreta de sistemas. *Revista de Ingeniería de Sistemas e Informática*, 6.
- HYSTAD, A. V. 2015. Model, Desing and Control of a Quadcopter. *Norwegian University of Science and Technology*.
- JOHRY, A. 2016. Unmanned Aerial Vehicle (UAV): Fault Tolerant Design. *International Journal of Engineering Technology Science and Research*, 3.
- MAHONY, R., KUMAR, V. & CORKE, P. 2012. Multirotor Aerial Vehicles: Modeling, Estimation and Control of a Quadrotor. *IEEE Robotics and Automation Magazine* 19, 20-32.
- MANYARI-RIVERA, M., BASILIO, J. C. & BHAYA, A. 2007. Integrated Fault Diagnosis Based on Petri Net Models. *16th IEEE International Conference on Control Applications*.
- MARTINEZ, M. A. T. 2016. Las Redes de Petri Interpretadas en el Diagnóstico de Fallos de Sistemas Híbridos.aplicación a un Helicóptero no Tripulado., 182.
- MARTINEZ, M. E. 2009. Desarrollo de Sistemas de Control para autopiloto de avión de pequeño porte.
- MATCOVSCHI, M. 2003. Petri Net Toolbox for MatLab. *The 11th Mediterranean Conference on Control and Automation*.
- MURATA, T. 1989. Petri Nets: Propierties, Analysis and Applications. *Proceedings of the IEEE* 77.
- PEREZ, H. D. 2014. Arquitectura de Hardware y Software basada en Smartphone para el control de UAV.
- RAMIREZ, A., RUIZ, E., RIVERA, I. & LÓPEZ, E. 2007. Online Fault Diagnosis of Discrete Event Systems. A Petri Net Based Approach. *IEEE Trans. in Autom. Science and Engineering*, 4.
- RICAÑO, R. R. 2016. El Control Fotográfico orientado a la creación de Mapas con Vehículos Aéreos No Tripulados.

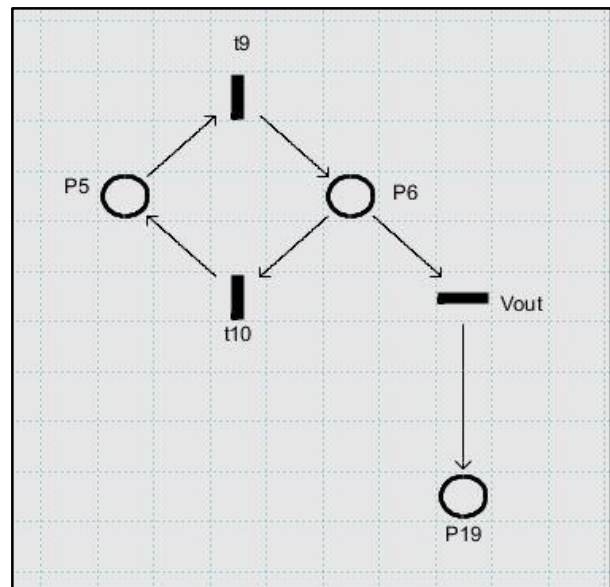
- RUIZ-BELTRAN, E., E.LOPEZ-MELLADO & M.MEDA-CAMPAÑA 2005. Fault detection and location in DES using Petri nets. *2005 IEEE International Conference In Systems, Man and Cybernetics*, 2.
- SADEGHZADEH, I. & ZHANG, Y. 2011. A Review on Fault-Tolerant Control for Unmanned Aerial Vehicles. *Concordia University, Montreal*.
- SAIED, M., SHRAIM, H., FRANCIS, C., FANTONI, I. & LUSSIER, B. 2015. Controllability Analysis and Motors Failures Symmetry in a Coaxial Octorotor. *Proc. of 2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering*.
- SAMPATH, M. 1995. A discrete event systems approach to failure diagnosis. *PhD thesis, Wayne State University*.
- SAMPATH, M., SENGUPTA, R., LAFORTUNE, S., SINNAMOHIDEEN, K. & TENEKETZIS, D. 1996. Failure diagnosis using discrete event models. *IEEE Trans On Contr. Systems*, 4, 19.
- SCHAFROTH, D., BERMES, C., BOUABDALLAH, S. & SIEGWART, R. 2010. Modeling, System Identification and Robust Control of a Coaxial Micro Helicopter. *Control Engineering Practice*.
- TEAM, A. D. 2016. *Copter Home* [Online]. [Accessed April].
- TRIGOS, M. A., BARRIENTOS, A. & CERRO, J. D. 2015. Systematic Process for Building a Fault Diagnoser Based on Petri Nets Applied to a Helicopter. *Hidawi Publishing Corporation*, 2015, 13.
- USHIO, T., ONISHI, I. & OKUDA, K. 1998. Fault Detection Based on Petri Net models with Faulty Behaviors. *IEEE*.
- VENKATARAMAN, R., LUKÁTSI, M., VANEK, B. & SEILER, P. 2013. Reliability Assessment of Actuator Architectures for Unmanned Aircraft
- WITCZAK, P., PAZERA, M., WITCZAK, M., KORBICZ, J. & THEILLIOL, D. 2016. Thrust balance estimation of an unmanned aerial vehicle: application to fault detection. *3rd Conference on Control and Fault-Tolerant Systems, SysTol2016*.

ANEXOS

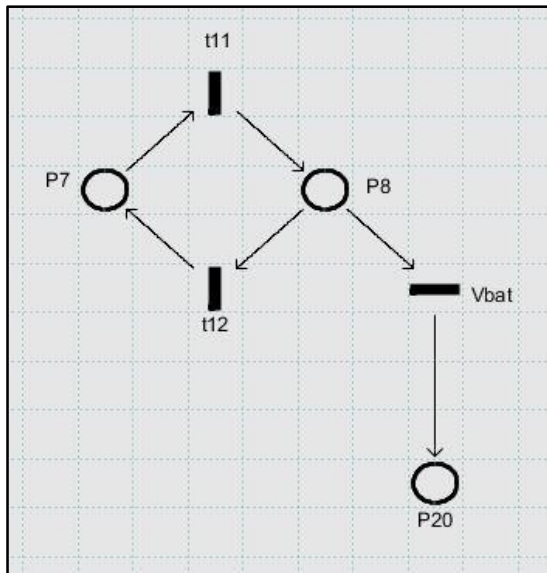
Anexo I Redes de Petri Refinadas para cada subsistema



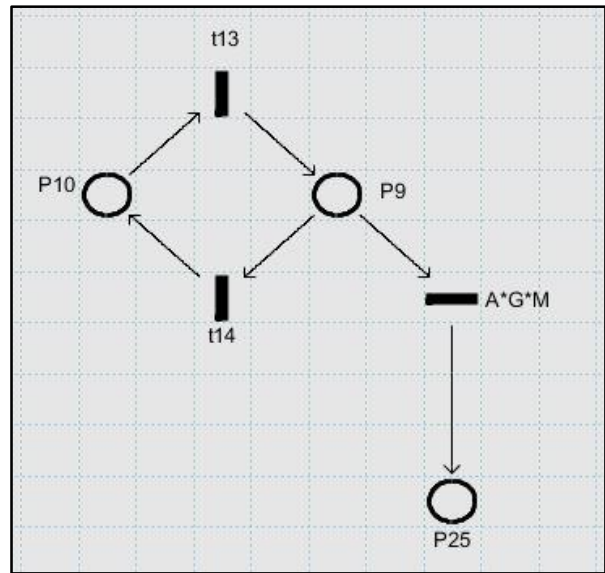
Modelo Refinado Subsistema Motor



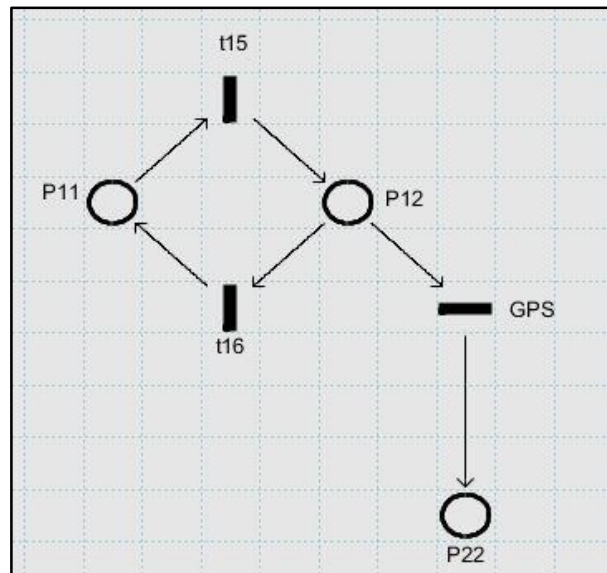
Modelo Refinado Subsistema ESC



Modelo Refinado Subsistema Batería

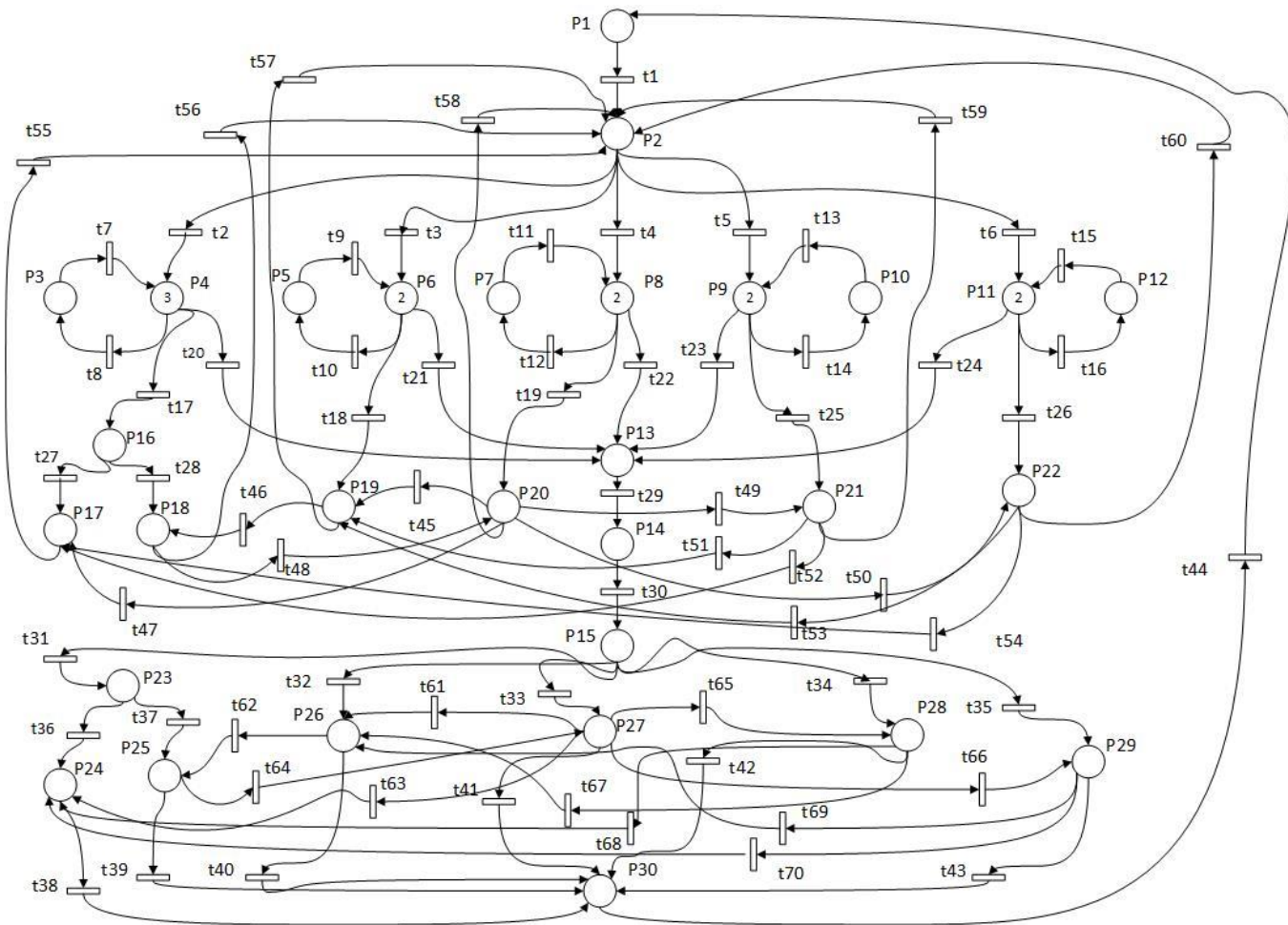


Modelo Refinado Subsistema IMU

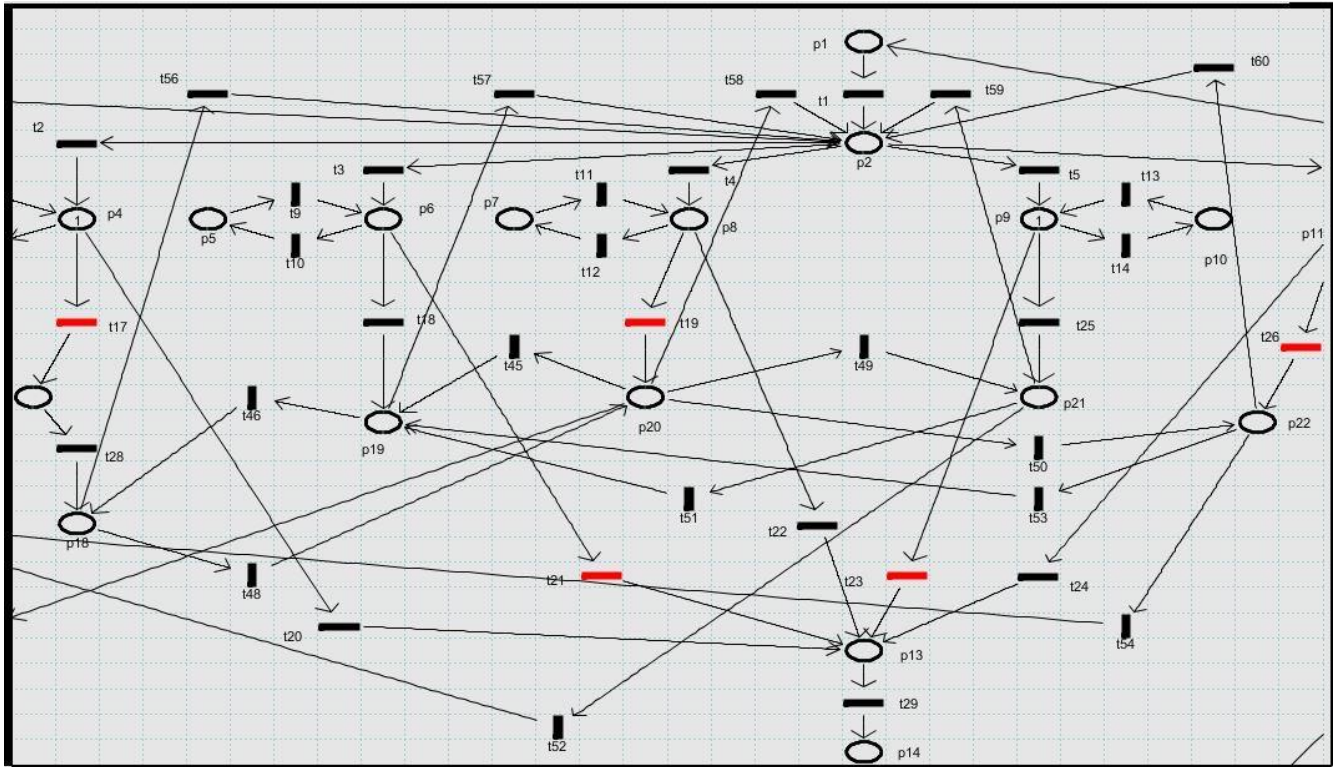


Modelo Refinado Subsistema GPS

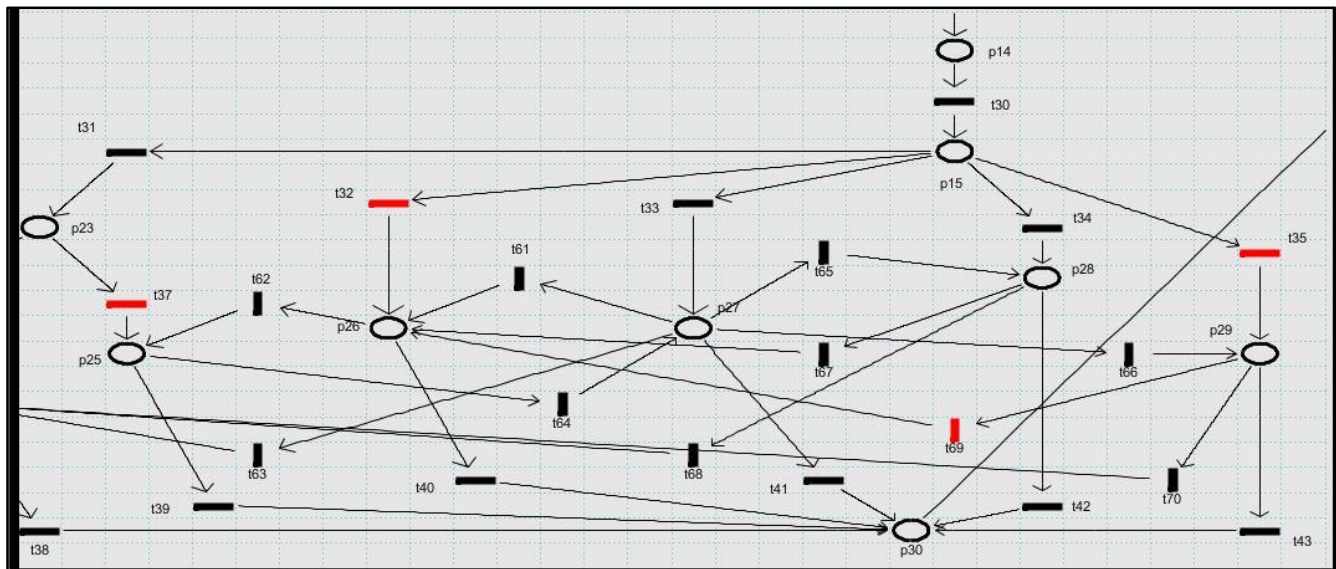
Anexo II Modelo General Refinado Redes de Petri



Anexo III Capturas de pantalla de las simulaciones



Fallos en los subsistemas Motor, Batería y GPS en estado previo al vuelo.



Fallos en los subsistemas Motor, ESC y GPS en estado de vuelo.

Anexo IV Estados y Transiciones del Modelo en RdP

| ESTADOS | TRANSICIONES |
|---|---------------------------------------|
| P1 Quadcopter apagado | T1 Mando encender Quadcopter (manual) |
| P2 Quadcopter encendido | T2 Chequeo Motor correct. armado |
| P3 Motor apagado | T3 Chequeo ESC correct. conectado |
| P4 Motor encendido (correct. armado) | T4 Chequeo Batería correct. conectada |
| P5 ESC apagado | T5 Chequeo IMU correct. funcionando |
| P6 ESC encendido (correct. conectado) | T6 Chequeo GPS correct. conectado |
| P7 Batería apagada | T7 Mando encender motor (OnMotor) |
| P8 Batería encendida (correct. conectada) | T8 Mando apagar motor (OffMotor) |
| P9 IMU encendida (correct. funcionando) | T9 Mando encender ESC (OnESC) |
| P10 IMU apagada | T10 Mando apagar ESC (OffESC) |
| P11 GPS encendido (correct. funcionado) | T11 Mando encender batería (OnBat) |
| P12 GPS apagado | T12 Mando apagar batería (OffBat) |
| P13 Quadcopter listo para volar (RTF) | T13 Mando encender IMU (OnIMU) |
| P14 Quadcopter en vuelo normal | T14 Mando apagar IMU (OffIMU) |
| P15 Quadcopter en vuelo con fallos | T15 Mando encender GPS |
| P16 Fallo Motor (Alarma) | T16 Mando apagar GPS |
| P17 Fallo Calentamiento Motor (sin volar) | T17 Condición de fallo motor (alarma) |

| | |
|--|--|
| P18 Fallo Atascamiento del Motor (sin volar) | T18 Condición de Fallo ESC ($V_{out} < 7V$) |
| P19 Fallo Nivel de Voltaje ESC (si volar) | T19 Condición de fallo batería ($V_{bat} < 12V$) |
| P20 Fallo Suministro de Energía (sin volar) | T20 Motor listo para vuelo |
| P21 Fallo Sensores IMU (sin volar) | T21 ESC listo para vuelo |
| P22 Fallo Medición Altura (sin volar) | T22 Batería lista para vuelo |
| P23 Fallo Motor (en vuelo) | T23 IMU lista para vuelo |
| P24 Fallo Calentamiento Motor (vuelo) | T24 GPS listo para vuelo |
| P25 Fallo Atasco Motor (vuelo) | T25 Condición fallo IMU ($A * G * M$) |
| P26 Fallo Nivel de Voltaje ESC (vuelo) | T26 Condición de fallo GPS (GPS) |
| P27 Fallo Suministro de Energía (vuelo) | T27 Condición de fallo calent. Motor (\bar{T}) |
| P28 Fallo Sensores IMU (vuelo) | T28 Condición de atascamiento del motor (RPM) |
| P29 Fallo Medición de Altura (vuelo) | T29 Mando de vuelo del Quadcopter |
| P30 Retorno al Punto de Lanzamiento (RTL) | T30 Condición Vuelo con Fallos ($T30=T21*T22*T23*T24$) |
| | T31 Condición de Fallo en el Motor (vuelo) |
| | T32 Condición de Fallo Nivel de Voltaje ESC (vuelo) |
| | T33 Condición de Fallo Batería (vuelo) |

| | |
|--|--|
| | T34 Condición de Fallo IMU (vuelo) |
| | T35 Condición de Fallo GPS (vuelo) |
| | T36 Condición Fallo Calent. Motor (vuelo) |
| | T37 Condición de Fallo Atascamiento Motor (vuelo) |
| | T38 Condición RTL desde el fallo calent. Motor (persistencia del fallo durante 30 seg) |
| | T39 Condición RTL desde el fallo atascamiento motor (persistencia del fallo durante 30 seg) |
| | T40 Condición RTL desde el fallo nivel voltaje ESC (persistencia del fallo durante 30 seg) |
| | T41 Condición RTL desde el fallo suministro de energía (persistencia del fallo durante 30 seg) |
| | T42 Condición RTL desde el fallo en los sensores de la IMU (persistencia del fallo durante 30 seg) |
| | T43 Condición RTL desde el fallo en la medición de altura (persistencia del fallo durante 30 seg) |
| | T44 Condición de apagado del Quadcopter |

| | |
|--|--|
| | T45 Relación entre fallo suministro de energía y fallo nivel de voltaje ESC (sin volar) |
| | T46 Relación entre fallo nivel de voltaje ESC y fallo por atascamiento del motor (sin volar) |
| | T47 Relación entre fallo suministro de energía y fallo calent. Motor (sin volar) |
| | T48 Relación entre el fallo por atascamiento del motor y fallo por suministro de energía (sin volar) |
| | T49 Relación entre el fallo por suministro de energía (sin volar) |
| | T50 Relación entre el fallo por suministro de energía y fallo medición de altura (sin volar) |
| | T51 Relación entre el fallo de los sensores de la IMU y fallo en el nivel de voltaje ESC (sin volar) |
| | T52 Relación entre fallo de los sensores de la IMU y fallo por calentamiento del motor (sin volar) |
| | T53 Relación ente el fallo en la medición de altura y fallo nivel de voltaje ESC (sin volar) |

| | |
|--|--|
| | T54 Relación entre el fallo en la medición de altura y fallo por calentamiento del motor (sin volar) |
| | T55 Condición de persistencia del fallo por calentamiento del motor (sin volar) durante 30 seg |
| | T56 Persistencia del fallo por atasco del motor (sin volar) durante 30 seg |
| | T57 Persistencia del fallo de nivel de voltaje del ESC (sin volar) durante 30 seg |
| | T58 Persistencia del fallo suministro de energía (sin volar) durante 30 seg |
| | T59 Persistencia del fallo e los sensores de la IMU (sin volar) durante 30 seg |
| | T60 Persistencia del fallo en la medición de altura (sin volar) durante 30 seg |
| | T61 Relación entre el fallo de los sensores de la IMU y el fallo nivel de voltaje ESC (en vuelo) |
| | T62 Relación entre el fallo de nivel de voltaje ESC y el fallo por atascamiento del motor (en vuelo) |
| | T63 Relación entre el fallo suministro de energía y fallo por calentamiento del motor (en vuelo) |

| | |
|--|---|
| | T64 Relación entre el fallo por atascamiento del motor y fallo en el suministro de energía (en vuelo) |
| | T65 Relación entre el fallo en el suministro de energía y el fallo en los sensores de la IMU (en vuelo) |
| | T66 Relación entre el fallo en el suministro de energía y el fallo en la medición de altura (en vuelo) |
| | T67 Relación entre el fallo en los sensores de la IMU y el fallo en el nivel de voltaje ESC (en vuelo) |
| | T68 Relación entre el fallo en los sensores de la IMU y el fallo por calentamiento del motor (en vuelo) |
| | T69 Relación entre el fallo en la medición de altura y el fallo en el nivel de voltaje ESC (en vuelo) |
| | T70 Relación entre el fallo en la medición de altura y el fallo por calentamiento del motor (en vuelo) |