

Scheduling en Celdas Autónomas de Ensamblaje Basado en un Algoritmo de Tiempo Mínimo en RdP

A. Ramírez-Treviño, J. L. Villarroel, M. Silva y P.R. Muro-Medrano

Dpto. de Ingeniería Eléctrica e Informática
Centro Politécnico Superior, Universidad de Zaragoza
María de Luna 3, E-50015 ZARAGOZA, SPAIN

Abstract: A minimum time algorithm for

The algorithm appears specially effective for schedule generation in assembly cells with shared resources. It has a polynomial response time in the data input and can be used in real time systems. The improvement of this algorithm, compared with other dispatch rules, is due to the use of a heuristic function that takes into account the structural and temporal features of the autonomous system as well as a limited search in the PN reachability tree.

Resumen: Este artículo presenta un algoritmo de tiempo mínimo para Redes de Petri (RdP). El algoritmo resulta especialmente efectivo para la creación de schedules en celdas de ensamblaje con recursos compartidos. El algoritmo tiene un tiempo de respuesta polinomial en los datos de entrada y puede ser usado en tiempo real. La mejora que se tiene frente a otras reglas de despacho se debe a que trabaja con una función heurística que contempla tanto características estructurales y temporales del sistema autónomo así como una búsqueda limitada en el árbol de alcanzabilidad de una RdP.

Palabras clave: Scheduling, Ensamblaje, Kanban, Redes de Petri, Tiempo de ciclo.

Agradecimientos: Este trabajo ha sido parcialmente financiado por los proyectos IT 10/91 del CONAI de la Diputación General de Aragón y ROB91-0949 de la Comisión Interministerial de Ciencia y Tecnología del Estado Español. El primer autor ha sido apoyado por CONACYT-México.

1 Introducción

El problema de scheduling [1, 5, 13] puede enunciarse como la asignación de recursos (máquinas, herramientas, robots, etc.) a operaciones a lo largo del tiempo. Las operaciones deben ser planificadas de forma que satisfagan las restricciones tecnológicas y de precedencia del proceso [11], optimizando al mismo tiempo los objetivos de producción (mínimo coste, mínimo tiempo, justo-a-tiempo, optimizar inventarios, etc.) [14] [9]. El problema de cálculo de un plan óptimo resulta ser, excepto para casos elementales, NP-completo, incluso bajo hipótesis académicas que contemplen fuertes simplificaciones en las restricciones tecnológicas y en los objetivos a optimizar.

Las aproximaciones a su solución siguen, en general, dos pasos, en el primero se aborda el modelado de sistema, cuya complejidad dependerá de las informaciones que se deseen contemplar en el modelo (conurrencia, exclusión, sincronizaciones, relaciones causales, objetivos de producción, etc.), el segundo tiene como objetivo la tarea de scheduling propiamente dicha utilizando el modelo anterior.

Todo programa de control que deba gestionar un sistema de fabricación autónomo, no sólo debe asegurarse de generar acciones que mantengan al sistema en un funcionamiento seguro, sino que además debe tratar de optimizar los objetivos de producción basando sus acciones en las secuencias y tiempos generados por un scheduler.

El trabajo que aquí se expone forma parte de un proyecto de mayor envergadura sobre generación de programas de control para sistemas flexibles de fabricación. En este proyecto se ha partido de un contexto de diseño de orientado a objeto, donde se han integrado herramientas formales de la familia de redes de Petri (RdP) con el objetivo de especificar las restricciones de comportamiento del sistema. En este trabajo se exploran posibilidades de solución al problema de scheduling a partir de modelos de sistema basados en redes de Petri. Centrados en un problema concreto, consistente en encontrar un plan que minimice

propone un algoritmo que permite encontrar una solución "suficientemente buena" (en los casos analizados esta solución corresponde con la solución óptima) en un tiempo "razonable". Se propone también la utilización de varias heurísticas que permite obtener el schedule óptimo o, en casos muy complejos, un schedule muy bueno en poco tiempo (se dice que la relación *calidad/tiempo* es muy buena [8]).

La exposición está organizada de la siguiente forma. En la sección 2 se introducen los conceptos básicos sobre redes de Petri ordinarias y temporizadas así como tiempos de ciclo y cotas. Esto sirve de base a la sección 4 para detallar el algoritmo de obtención de un schedule de mínimo tiempo. En la sección 5 se presenta un ejemplo ilustrativo de funcionamiento del algoritmo. Y, finalmente, en la sección 6 se enumeran algunas conclusiones.

2 Conceptos y terminología básica sobre Redes de Petri

Una red de Petri (RdP) [7, 10] es un grafo bipartito con un estado inicial llamado marcaje inicial. Está caracterizada por la quintupla $N = \{P, T, F, W, M_0\}$, donde $P \cup T$ es el conjunto de vértices (P es el conjunto de lugares y T es el conjunto de transiciones); $F \in \{(P \times T) \cup (T \times P)\}$, $W : F \rightarrow \{0, 1\}$ es una función que asocia a cada arco un 0 si el arco no existe y un 1 si dicho arco existe, y $M_0 :$

$P \rightarrow N$ es el marcaje inicial. Denotamos por $\bullet x$ (respec. x^\bullet) el conjunto de los predecesores (respec. sucesores) de $x \in P \cup T$. El estado de la RdP está representado por su marcaje M , que viene dado por $M : P \rightarrow N$.

La dinámica de la RdP está determinada por el disparo de las $t_i \in T$. El disparo de t_i se puede efectuar ssi $M(p) \geq 1 \forall p \in \bullet t_i$. El disparo de t_i implica la eliminación de una marca de cada $\bullet t_i$ y la adición de una marca a cada t_i^\bullet .

Una red de Petri temporizada, *RdPT*, es una RdP a la cual se le ha asociado un tiempo a cada transición, esto es, existe $\tau : T \rightarrow \mathbb{R}^+$, $\tau(t_i) = \tau_i \in \mathbb{R}^+$. Este tiempo representa cuanto tarda la transición en dispararse. La matriz de pre-incidencia (post-incidencia) $C^- = [C_{ij}^-]$ ($C^+ = [C_{ij}^+]$) es una matriz $n \times m$ con $C_{ij}^- = W(p_i, t_j)$ ($C_{ij}^+ = W(t_j, p_i)$). La matriz de incidencia es $C = C^+ - C^-$. Su ecuación de estado viene dada por: $M_{k+1} = M_0 + C \cdot v_k$. Una solución entera de $C \cdot X = 0$, $X \geq 0$ ($Y^T \cdot C = 0, Y \geq 0$) es llamado un T-semiflujo (P-Semiflujo).

3 Modelado de celdas de ensamblaje

Con objeto de poder comprobar la aplicabilidad del algoritmo, es necesario primero mostrar el método utilizado para la realización de los modelos de celdas de ensamblaje mediante RdP. Para ello, se muestra a continuación un ejemplo ilustrativo de modelado. La figura 1 muestra una celda de ensamblaje de dos piezas, ambas piezas son pre-procesadas antes del ensamblaje llevado a cabo por la máquina 6. La máquina 4 es un recurso compartido por ambas piezas. Todas las operaciones son modeladas como en la figura 2 a), y en el caso de operaciones con máquinas compartidas se modelan como en la parte b) de la misma figura.

Cuando se tienen secuencias de operaciones basta con fundir el lugar de salida de la transición con el lugar de entrada a la transición de la siguiente operación. El modelo total de este ejemplo se observa en la red original de la figura 3. En caso de estar la celda

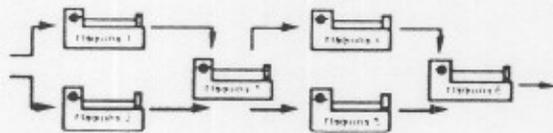


Figura 1: Celda de ensamblaje.

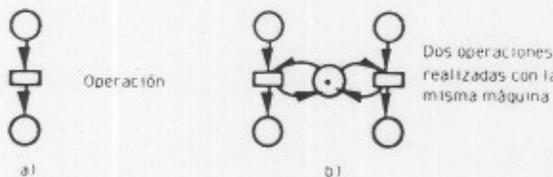


Figura 2: Modelado de operaciones.

integrada dentro de un sistema Kanban, el lugar p_1 de dicha figura nos sirve para mandar la etiqueta o "kanban" a la etapa anterior cuando el proceso ha terminado.

4 Algoritmo de tiempo mínimo

El objetivo del algoritmo presentado en esta sección es el de obtener un schedule de tiempo mínimo para celdas de ensamblaje que han sido modeladas como se describe en la sección 3. Como el modelo queda expresado como una RdP, el algoritmo está basado en resultados y heurísticas obtenidas de dicho modelo. A continuación se introducen algunos resultados sobre tiempos de ciclo y cotas que son útiles en el algoritmo.

4.1 Bases previas

En redes cíclicas es posible calcular el tiempo en que se vuelve a tener el marcaje inicial después de haber disparado una secuencia de transiciones fija. Si la RdPT es un grafo marcado fuertemente conexo el problema está totalmente resuelto, pero si existen varias secuencias (i.e. existe flexibilidad) el problema es muy difícil de resolver (NP-Completo) [6]. El siguiente problema de programación lineal proporciona el tiempo de ciclo exacto para el caso de grafos marcados [12, 2] (para redes más

generales este problema proporciona una cota inferior del tiempo de ciclo).

$$\pi = \text{maximizar } Y^T \cdot C^{-1} \cdot D$$

$$\text{sujeto a } Y^T \cdot C = 0$$

$$Y^T \cdot M_0 = 1 \text{ Normalización. } Y \geq 0$$

4.2 El algoritmo

Inicialización

$$M_i = M_0; \pi = \phi; \text{schedule} = \phi; \text{tiempo} = 0$$

Realizar

M_s es el conj. de sucesores de M_i en el ESC.

tomar un elemento $m_{si} \in M_s$

Agregar a la RP t_{m+1} que alcance m_{si} de M_0 en tiempo 0.

Obtener el tiempo de ciclo π_i para esta red

$$\Pi = \Pi \cup \pi_i$$

$$M_s = M_s - m_{si}$$

Fin-Repetir

$$\pi_j = \min_{x_i \in \Pi} \{x_i + \text{tiempo} + \text{peso del arco } [M_i, m_{si}]\}$$

$$\text{schedule} = \text{schedule} + (t_j, \text{tiempo}), M_i[t_j > m_{sj}]$$

$$\text{tiempo} = \text{tiempo} + \text{peso del arco } [M_i, m_{sj}]$$

$$M_i = m_{si}$$

Fin-Realizar (hasta $M_i = M_0$)

4.2.1 Explicación intuitiva

Primero el algoritmo se inicializa, las condiciones iniciales son el marcaje inicial y todos los demás elementos a cero.

Dentro de un ciclo general se calcula el "earliest state graph" [3] para el siguiente nivel solamente. A continuación se analiza cada uno de los nodos hijos, el análisis se lleva a cabo solamente a nivel de tiempos de ciclo.

Es muy importante notar que la cota de Sifakis da una aproximación al tiempo de ciclo, pero nosotros desearíamos obtener una estimación de lo que le falta a la red para completar su ciclo. Por este motivo se introduce una transición bypass $m+1$ a la red, de manera que nos lleve de M_0 al marcaje deseado en un solo paso y con tiempo cero. Esto se ilustra en las figuras 3 y 4. Claramente el cálculo del tiempo de ciclo es una estimación del tiempo que hace falta por terminar, porque la transición bypass tiene un tiempo asociado igual a cero.

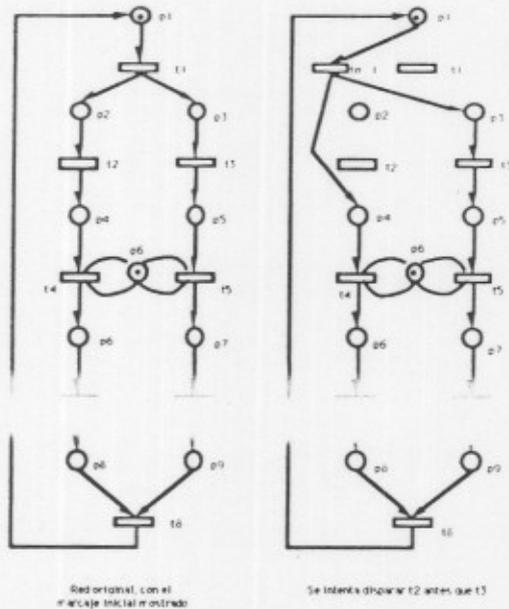


Figura 3: La red original con su transformación.

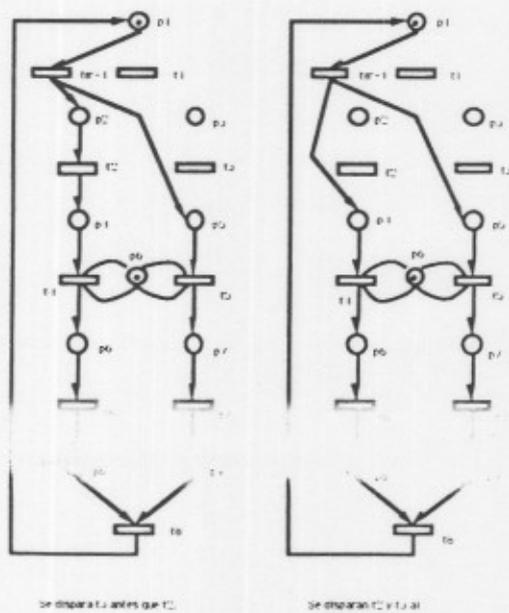


Figura 4: Dos transformaciones más de la red.

Con esta estimación del tiempo que le falta por terminar y el tiempo transcurrido hasta el momento es posible determinar cual es mejor camino a seguir. El algoritmo continúa la búsqueda hasta que se llega al marcaje inicial nuevamente.

Debido a la expansión parcial del grafo de alcanzabilidad es posible obtener un schedule, el cual resulta en óptimo si se han probado todos los hijos generados, pero en algunos casos el número de hijos es muy grande y es imposible probarlos todos, por lo cual existe una interrupción que dado un cierto tiempo detiene el mecanismo de estimación

con la información de que se disponga en ese momento.

Implementación: El algoritmo fue implementado en C, a la heurística prevista se le añadió la posibilidad de que en caso de empate seleccionara el estado que realiza más tareas. Si al evaluar está segunda heurística todavía se tiene empate, entonces se toma cualquier nodo como el siguiente.

5 Ejemplo de aplicación a celdas de ensamblaje

El ejemplo que tomamos es el de una celda de ensamble que pertenece a un sistema Kanban. La estrategia de un sistema kanban se basa en que una vez que cada celda está libre solicita a la etapa anterior una pieza para continuar el procesamiento. Como ha sido probado, un sistema kanban puede ser modelado por una RdP [4] con un lugar que limita la capacidad de cada etapa o celda, y que le indica a la etapa anterior su estado por medio de una etiqueta o kanban.

La celda a analizar consta de un ensamblado de dos tipos diferentes de piezas, las cuales se fabrican alternativamente y tienen una sección de procesamiento común. La celda tiene 6 máquinas. Antes del ensamble, en la M_6 , cada pieza debe ser preprocesada. Existiendo tres secuencias de preprocesados. Ambos ensamblados tienen en común la secuencia M_3 , M_4 y M_6 , pero el primer tipo de ensamblado

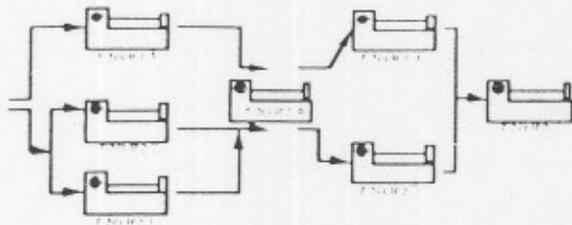


Figura 5: Celda de ensamblado.

(llamado S2) tiene adicionalmente la secuencia M_1 , M_4 y M_5 , mientras que el segundo tipo de ensamblado (llamado S1) tiene la secuencia

observarse la M_4 es un recurso compartido por las dos secuencias.

El sistema ha sido probado por simulación. A los resultados se añaden los resultados de una simulación utilizando una política del tipo "lo antes posible" para contrastar resultados y resaltar el interés por el uso del método propuesto.

La primera se realizó con los siguientes tiempos de trabajo: $M_1 = 1$, $M_2 = 3$, $M_3 = 2$, $M_4 = 2$ (S2 y S3), $M_4 = 1$ (S1), $M_5 = 1$, $M_6 = 3$.

La siguiente tabla muestra el tiempo en que son visitadas las máquinas.

Máquina	tiempo
1 y 3	0
4 (S1)	2
4 (S2) y 6	3
5 (6 en proceso)	5
2 y 3	6
4 (S1) (3 en proceso)	8
5 (S2) y 6	9
5 (6 en proceso)	11
Finalización	12

En el caso de la regla de despacho se seleccionó el peor schedule posible, con la temporización seleccionada se tuvo un tiempo de finalización de 13 unidades.

6 Conclusiones

En este artículo se presentó un algoritmo que permite obtener un schedule de tiempo mínimo para una RdPT. Este algoritmo puede ser usado por un sistema autónomo como guía para la toma de decisiones de scheduling. Las decisiones proporcionadas por el algoritmo son especialmente eficientes en sistemas de fabricación gestionados por políticas Kanban, que dispongan de celdas de ensamblaje con secuencias de producción con recursos compartidos.

Los caminos que va a seguir cada pieza, evidenciando los posibles cuellos de botella y saturación de almacenes. La utilización de este algoritmo resulta una mejora con respecto a las reglas de despacho tradicionales por realizar una búsqueda más exhaustiva. Sin embargo el problema de la complejidad computacional todavía está presente, por lo que habrá algunos casos en los que el algoritmo de una respuesta muy buena, pero no la óptima. Creemos que se podrán obtener mejores aproximaciones a partir de una caracterización la estructura de la RdP subyacente en el modelo.

Bibliografía

- [1] R. Bellman, A. Esogbue, and I. Nabeshima. *Mathematical Aspects of Scheduling & Applications*, volume 4 of *Modern Applied Mathematics and Computer Science*. Pergamon Press, Great Britain, 1982.
- [2] J. Campos. *Performance Bounds for Synchronized Queuing Networks*. PhD thesis, Universidad de Zaragoza., María de Luna 3 E-50015 Zaragoza, España., 1990.
- [3] J. Carlier and Ph Chretienne. Timed petri net schedules. In G. Rozenberg, editor, *Advances in Petri Nets 1988*, volume 340 of *Lecture Notes in Computer Sciences*, pages 62-84. Springer-Verlag, Berlin, Germany, 1988.

- [4] M. Di Mascolo. *Modélisation et Évaluation de Performances de Systèmes de Production Gérés en Kanban*. PhD thesis, Institut National Polytechnique Grenoble., 46 avenue F. Viallet - 38031 Grenoble, France., 1990.
- [5] S. French. *Sequencing and Scheduling, An Introduction to the Mathematics of the Job-Shop*. John Wiley & Sons, 605 Third Avenue, New York, NY 10010 USA, 1982.
- [6] J. Magott. New np-complete problems in performance evaluation of concurrent systems using petri nets. *IEEE Transactions* 581, May 1987.
- [7] T. Murata. Petri nets: properties, analysis, and applications. In *Proceedings of the IEEE*, pages 77(4):541-580, April 1989.
- [8] S. J. Noronha and V. V. S. Sarma. Knowledge-based approaches for scheduling problems: A survey. *IEEE Transactions on Knowledge and data Engineering*, 3(2):160-171, June 1991.
- [9] S. Panwalkar and W. Iskander. A survey of scheduling rules. *Operations Research*, 25(1), January-February 1977.
- [10] J.L. Peterson. *Petri Net Theory and the Modeling of Systems*. Prentice Hall, NJ USA, 1981.
- [11] J. Proth and H. Hillion. *Mathematical Tools in Production Management*. Plenum Press New York, 233 Spring Street, New York, NY 10013 USA, 1990.
- [12] C. Ramamoorthy and G. Ho. Performance evaluation of asynchronous concurrent systems using petri nets. *IEEE Transactions on Software Engineering*, 6(5):440-449, September 1980.
- [13] M. Silva and Valette R. Petri nets and flexible manufacturing. In Rozenberg, editor, *Advances in Petri Nets 1989*, volume 266 of *Lecture Notes in Computer Sciences*, pages 375-417. Springer-Verlag, Berlin, Germany, 1989.
- [14] H. van Dyke. Characterizing the manufacturing scheduling problem. *Journal of Manufacturing Systems*, 10(3):241-259, May 1991.



E AUTOMATIZAÇÃO

REVISTA DE INFORMAÇÃO TÉCNICA E CIENTÍFICA

PUBLICAÇÃO TRIMESTRAL

N. 10 • Novembro 1992

ALEXANDRE TATO
Director

TENREIRO MACHADO
Director Científico

LUIS SÁ
Director Adjunto

ANTÓNIO MALHEIRO
Editor

PAULO ANDREZ

PAULO ALVES
Design

ALBANO PEREIRA
Marketing
Tel. (02) 510 05 18

SÉRGIO CASTRO
Assinaturas
Tel. (02) 57 77 28

CONSELHO EDITORIAL:
CAMARINHA MATOS
Robótica e CIM
EUGÉNIO OLIVEIRA
Inteligência Artificial
PINA CABRAL
Automação
PEREIRA DIAS
Electrónica Industrial
AMORIM MARQUES
Informática

ORGANIZAÇÃO E EDIÇÃO DESTE NÚMERO:
CAMARINHA MATOS

COLABORAÇÃO ESPECIAL:
STEIGER GARCÃO

PROPRIEDADE E ADMINISTRAÇÃO:
PUBLINDÚSTRIA
Produção de Comunicação, Lda.
Rua Ferreira dos Santos, 124
4300 PORTO
Tel. (02) 56 54 82 - Fax (02) 510 05 17

BRITISH REPRESENTATION
ROBERT G. HORSFIELD
EDWARD J. KANIA
International Publishers Representatives
Daisy Bank - Chinley
Via Stockport SK126DA - ENGLAND
Tel. (0663) 75 02 42 - Fax (0663) 75 09 73

COMPOSIÇÃO FOTOLITO E MONTAGEM:
PUBLINDÚSTRIA

IMPRESSÃO E ACABAMENTO:
PUBLICOR
Serzedo - V. N. GAIA
Tel. (02) 762 64 85

TIRAGEM:
3.000 EXEMPLARES

DEPÓSITO LEGAL:
113164



Os trabalhos assinados
são da exclusiva responsabilidade
dos seus autores

SUMÁRIO

3 EDITORIAL

5 APRESENTAÇÃO

6 CURRÍCULAVITAE

ANÁLISE DE SISTEMAS
DE MANUFATURA COM BASE

11 INTEGRACIÓN DE REDES DE PETRI
EN EL DISEÑO ORIENTADO
A OBJETO DE SISTEMAS
DE FABRICACIÓN

19 ANÁLISE DE SISTEMAS INTEGRADOS
DE MANUFATURA BASEADA
NA METODOLOGIA MFG/PFS

25 ANÁLISE SEMI-AUTOMÁTICA
DE MARK FLOW GRAPHS

31 ESPECIFICAÇÃO E REALIZAÇÃO
DE CONTROLADORES UTILIZANDO
REDES DE PETRI COLORIDAS
E SINCRONIZADAS INTEGRANDO
LÓGICA IMPRECISA

ESCALONAMENTO E CONTROLE

39 UMA METODOLOGIA
PARA IMPLEMENTAÇÃO
DE SISTEMAS DE CONTROLE
E PROJEÇÃO TEMPORAL
DE ATIVIDADES

45 UMA APROXIMAÇÃO
COM BASE EM REGULAÇÃO

55 SCHEDULING EN CELDAS
AUTÓNOMAS DE ENSAMBLAJE
BASADO EN UN ALGORITMO
DE TIEMPO MÍNIMO EN R&P

PLANEAMENTO E MONITORAÇÃO
EM ROBOTICA

61 PROGRAMACION IMPLÍCITA
DE ROBOTS
UNA APROXIMACION FORMAL

69 PLANEAMENTO EFICIENTE

75 EXECUÇÃO, MONITORAÇÃO
SENSORIAL E REACÇÃO
A SITUAÇÕES DE EXCEÇÃO
EM ROBOTICA DE MONTAGEM

CONCEPÇÃO DE SISTEMAS
DE MANUFATURA

81 ENTORNO DE DESARROLLO
DEL CONTROL DE SISTEMAS
DE MANUFATURA

87 DISEÑO DE CELULAS FLEXIBLES
DE MANUFATURA

91 SADS INTELIGENTES
PARA PLANEJAMENTO
EM MANUFATURA
UMA ABORDAGEM VIA MODELAGEM
VISUAL INTERATIVA

PROGNÓSTICO, VEÍCULOS
AUTÓNOMOS E SEQUENCIAMENTO
DE TRÁFEGO

95 CARACTERIZAÇÃO AUTOMÁTICA
DE PADRÕES EM SENSORES
DE MÁQUINAS CNC

107 ANÁLISE DE SISTEMAS
DE CONTROLO DE MÁQUINAS
CONTROLADOS
POR COMPORTAMENTOS

113 UMA APROXIMAÇÃO "MUNDOS
POSSÍVEIS" AOS PROBLEMAS
DE ESCALONAMENTO DE VOOS

Agradecemos à UNISYS a excelente colaboração
que prestou na informatização dos nossos serviços