



HAL
open science

Une approche formelle pour l'optimisation de systèmes à événements discrets

Juan Cardillo Albarran

► **To cite this version:**

Juan Cardillo Albarran. Une approche formelle pour l'optimisation de systèmes à événements discrets. Automatic. Université Paul Sabatier - Toulouse III, 2004. Español. NNT : . tel-00010477

HAL Id: tel-00010477

<https://theses.hal.science/tel-00010477>

Submitted on 7 Oct 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Préparée au

Laboratoire d'Analyse et d'Architecture de Systèmes (LAAS) du CNRS,
Toulouse – France et à la
Facultad de Ingeniería, Departamento de Sistemas de Control de la ULA,
Mérida - Venezuela

En vue de l'obtention du

Doctorat de l'Université Paul Sabatier de Toulouse et de l'Université de
Los Andes (Convention de cotutelle 2002)

Spécialité

Automatique

Par

Juan CARDILLO-ALBARRÁN

Un Approche Formelle pour l'Optimisation de Systèmes a Événements Discrets

Soutenue le 5 février 2004

Président

Z. VARGA

J-L CALVET

F SZIGETI

J-C HENNET

Rapporteurs

L. SOLE

J. ZUBIRI

Une approche formelle pour l'optimisation de Systèmes à événements discrets

Juan Cardillo-Albarran

RÉSUMÉ

Cette thèse est centrée sur l'optimisation de systèmes dynamiques à événements discrets sur un domaine fini.

Deux approches fondamentales de l'optimisation sont considérées : le principe du minimum et la programmation dynamique. Dans chaque cas une méthode de calcul formel est développée pour l'optimisation d'une forme polynomiale sur un domaine de définition fini.

Le résultat de l'approche basée sur le principe du minimum est établi sous la forme d'un théorème dans lequel la condition (nécessaire) d'optimalité est définie en terme d'une inégalité variationnelle sous une forme séparable par rapport aux étapes (instants).

Celle-ci est explicitement fonction de la variable de commande à l'instant courant, (commande courante), de la forme paramétrée de celle-ci et de l'état du système auquel peuvent s'ajouter des paramètres exogènes impliqués dans le modèle d'évolution.

La résolution de cette inégalité variationnelle paramétrique est développée selon une procédure de type Min-Max avec une minimisation par rapport aux commandes courantes et une maximisation par rapport aux autres commandes.

L'algorithme correspondant est désigné par le symbole SCDO (Symbolic Computation for Discrete Optimisation).

L'approche basée sur la programmation dynamique exploite le caractère paramétrique de cette méthode de décomposition. L'intégration de l'algorithme SCDO dans les deux phases de ce processus d'optimisation permet ici encore d'exprimer la séquence de commandes optimales sous une forme explicite de l'état du système et des autres paramètres exogènes.

Dans ce mémoire, nous considérons également le principe de relaxation pour transformer le problème discret en un problème continu de résolution classique. Ainsi, pour une famille particulière de processus discrets linéaires par rapport à l'état et de fonction de coût concave, nous obtenons une condition de type principe du minimum, équivalente pour l'optimum relaxé et l'optimum de problème original. Dans le cas de fonctions de coût linéaires, la condition obtenue est celle du principe du minimum classique.

Un enfoque formal para la optimización de sistemas a eventos discretos

Juan Cardillo-Albarran

RESUMEN

El desarrollo de esta tesis se centra en la optimización de Sistemas Dinámicos a Eventos Discretos sobre dominios finitos.

Para tal fin, se abordaron los dos principios básicos de la optimización: “El Principio de Mínimo” y “la Programación Dinámica”. Sobre cada uno de ellos, una extensión basado en el cálculo formal se realizó considerando dominios de definición finitos y parámetros exógenos de evolución.

El resultado del enfoque basado sobre el principio está dado en la forma de un teorema, en donde la condición (necesaria) de optimalidad es definida en términos de una desigualdad variacional sobre una forma separable en etapas (instantes).

Esta es una función explícita de la variable de control en el instante actual (control actual) en forma paramétrica de esta y el estado del sistema el cual puede contener parámetros exógenos en el modelo de evolución

La resolución de esta desigualdad variacional paramétrica es desarrollada según un procedimiento del tipo Min-Max con una minimización en función del control de la etapa actual y una maximización en función del control óptimo de esa etapa.

El algoritmo correspondiente para realizar la minimización es designado por el símbolo SCDO (Symbolic Computation for Discrete Optimization), y obtiene mínimos (parametrizados) de una función definida sobre un hiper-cubo con vértices en $\{-1,1\}$.

El enfoque basado sobre la Programación Dinámica explota el carácter paramétrico de este método de descomposición. La integración del algoritmo SCDO en las dos fases de este proceso de optimización permite expresar la secuencia de controles óptimos bajo una forma explícita del estado del sistema y los otros parámetros exógenos.

En esta memoria nosotros también consideramos el principio de relajación para transformar el problema discreto en un problema continuo de resolución clásica. Así para una familia particular de procesos discretos lineales en el estado y una función de costo cóncava, obtenemos una condición del tipo principio de mínimo para la condición de equivalencia entre el óptimo relajado y el óptimo del problema original. Para el caso donde la función de costo es lineal, la condición de equivalencia obtenida es una condición del tipo principio de mínimo clásico.

A Formal Approach to Discrete Event Systems Optimization

Juan Cardillo-Albarran

SUMMARY

This thesis is centered on optimization of discrete event dynamical systems over finite domains. Two basic optimization methods have been considered: the Minimum Principle and Dynamic Programming.

The result of the approach based on the Minimum Principle has been established in the form of a Theorem in which the necessary optimality condition is defined in terms of a variational inequality separable with respect to stages. This inequality is an explicit function of control variables at the current time, of state variables and possibly exogenous parameters involved in the evolution model. A resolution technique for such a parametric variational inequality has been developed under a « min-max » type process, respectively for the current control and the other controls.

A method of symbolic computation is proposed in the manuscript to optimize a polynomial form defined over a finite discrete set. The corresponding algorithm is denoted by the acronym SCDO (Symbolic Computation for Discrete Optimization). The outcome of the application of this algorithm to solving the above mentioned « Min-Max » problem is the expression of the symbolic control sequence in a symbolic form explicit in the system state (and the exogenous parameters).

The approach based on Dynamic Programming directly exploits the parametric character and the decomposition framework of this method. Integration of the SCDO algorithm in both stages of the process (optimization and trajectory reconstruction) here again generates an expression of the optimal control sequence explicit in the state.

TABLA DE CONTENIDO

Introducción

INTRODUCCIÓN	I-1
I.1. SISTEMAS A EVENTOS DISCRETOS	I-1
I.2 BREVE DESCRIPCIÓN DEL DESARROLLO DE ESTA TESIS	I-2

Capítulo 1

MODELOS FORMALES DE SDED	1-1
1.1. MODELOS FORMALES EN LEGUAJES CERRADOS: CASOS GENERALES DE SISTEMAS LINGÜÍSTICOS	1-1
1.1.1. Lenguajes Cerrados y Realización Mínima.....	1-2
1.1.2. Ejemplo: Simplificación Matemática por cálculo formal.....	1-8
1.1.3. Ejemplo: Supervisor para la primera etapa blanqueo de sebo	1-12
1.2. CONCLUSIONES	1-23

Capítulo 2

COMPUTACIÓN SIMBÓLICA EN OPTIMIZACIÓN DISCRETA	2-1
2.1. INTRODUCCIÓN A LA OPTIMIZACIÓN DE FUNCIONES SOBRE DOMINIOS DISCRETOS	2-1
2.2. OPTIMIZACIÓN SIMBÓLICA DE FUNCIONES BOOLEANAS	2-1
2.2.1. Caso monovariable.....	2-2
2.2.2. Caso multivariable	2-2
2.3. OPTIMIZACIÓN SOBRE DOMINIO DISCRETO FINITO	2-4
2.3.1. Representación polinómica canónica.....	2-5
2.3.2. Optimización simbólica.....	2-8
2.3.3. Mejoras en el algoritmo simbólico.....	2-9
2.4. EJEMPLOS ILUSTRATIVOS	2-10
2.4.1. Ejemplo 1.....	2-10
2.4.2. Ejemplo 2.....	2-13
2.5. CONCLUSIONES	2-15

Capítulo 3

ENFOQUE SIMBÓLICO DEL PRINCIPIO DE MÍNIMO	3-1
3.1. ECUACIONES CANÓNICAS DE MOVIMIENTO EN EL FORMALISMO HAMILTONIANO.....	3-2
3.2 EL PRINCIPIO DE MÍNIMO PARA PROCESOS CONTINUOS	3-2
3.2.1. Preliminares	3-2
3.2.2. El Principio de Máximo: El problema de control con tiempo final fijo.	3-4
3.3. EL PRINCIPIO DE MÍNIMO PARA PROCESOS A EVENTOS DISCRETOS SOBRE DOMINIOS FINITOS	3-6
3.3.1 Preliminares	3-6
3.3.2. Procesos discretos enteros finitos	3-8
3.3.3. Principio de Mínimo de Pontryaguin sobre dominios finitos (PMPDF).....	3-9
3.3.4. Algoritmo simbólico para aplicar PMPDF.....	3-11
3.3.4.1. Algoritmo simbólico	3-11
3.3.4.2. Ejemplo Ilustrativo.....	3-12
3.3.4.3. Algoritmo Simbólico detallado para PMPDF	3-14
3.4. EJEMPLO:	3-15

3.5. RESOLUCIÓN DE DESIGUALDADES VARIACIONALES	3-21
3.5.1. Solución de las desigualdades variacionales vía solución Min-Max.	3-21
3.5.2. Ejemplo: Optimización paramétrica usando PMPDF	3-22
3.6. EL MÉTODO DE RELAJACIÓN APLICADO A SISTEMAS DISCRETOS.....	3-28
3.6.1. El problema de optimización de procesos a eventos discretos.....	3-29
3.6.2 Resultados	3-30
3.6.3. Ejemplos.....	3-32
3.6.3.1. Ejemplo 1	3-32
3.6.3.2. Ejemplo 2.....	3-33
3.7. CONCLUSIONES	3-34

Capítulo 4

PROGRAMACION DINAMICA PARAMETRICA	4-1
4.1. LA ECUACIÓN DE BELLMAN Y LA FUNCIÓN VALUADA	4-1
4.2 PROGRAMACIÓN DINÁMICA DISCRETA	4-4
4.3. UN ENFOQUE SIMBÓLICO.....	4-4
4.3.1. Modelo de optimización dinámica.....	4-5
4.3.2. Optimización simbólica en el paso k	4-7
4.4. PROGRAMACIÓN DINÁMICA PARAMETRICA	4-11
4.4.1. Algoritmo programación dinámica paramétrica (PDPA).....	4-11
4.5. EJEMPLO	4-12
4.6. CONCLUSIONES	4-14

Conclusiones Generales

CONCLUSIONES GENERALES	C-1
-------------------------------------	------------

Anexo A

FUNDAMENTOS SOBRE LEGUAJES FORMALES Y REALIZACIÓN MÍNIMA	A-1
DEFINICIONES	A-1
OPERACIONES:	A-1
PROPIEDADES DEL ALFABETO.....	A-1
CLASE DE EQUIVALENCIA	A-2
EL SISTEMA DINÁMICO INDUCIDO POR LA CONCATENACIÓN:	A-2
Propiedades de la dinámica	A-3
Transformación de estado inducida	A-3
Función de salida	A-3
REALIZACIÓN MÍNIMA	A-4
Procedimiento 1	A-4
Procedimiento 2	A-6

Referencias

REFERENCIAS.....	R-1
-------------------------	------------

LISTA DE FIGURAS

Capítulo 1

FIGURA 1.1: CLASES DE EQUIVALENCIA	1-3
FIGURA 1.2: RELACIÓN ENTRE LA CAJA NEGRA Y EL SISTEMA LINGÜÍSTICO	1-4
FIGURA 1.3: SCADA PRIMERA ETAPA BLANQUEO DE SEBO.....	1-13
FIGURA 1.4: PANEL DE CONTROL.....	1-13
FIGURA 1.5: AUTÓMATA DE LA BOMBA, DE LA VÁLVULA Y DEL TANQUE PULMÓN.....	1-14
FIGURA 1.6: AUTÓMATA DEL LLENADO DEL TANQUE PULMÓN.....	1-14
FIGURA 1.7: AUTOMATAS DE LAS BOMBAS 2, 3, 4, DE LAS VÁLVULAS 2, 3, 4, LOS SENSORES 1, 2, Y.....	1-16
FIGURA 1.8: AUTOMATA DEL PORCESO DE SECADO	1-16

Capítulo 2

FIGURE 2.1: UN AUTÓMATA CON 4 ESCOGENCIAS.....	2-13
--	------

Capítulo 3

FIGURA 3.1: GRAFO CON $P=5$	3-28
FIGURA 3.2: GRAFO CON $P=0$	3-28
FIGURA 3.3: SIGNIFICADO GEOMÉTRICO DE LA RELAJACIÓN.....	3-33

Capítulo 4

FIGURA 4.1: TRAYECTORIAS DESCRITAS POR UNA SECUENCIA DE ESTADOS DISCRETOS EN ETAPAS	4-5
FIGURA 4.2: GRAFO PARA EL CASO DE UNA SIMPLE DECISIÓN BINARIA	4-7
FIGURA 4.3: GRAFO DEL CASO GENERAL	4-8
FIGURA 4.4: DESCOMPOSICIÓN EN ESCALERA.....	4-8
FIGURA 4.5: EJEMPLO ILUSTRATIVO Y FIGURA 4.6: GRAFO EN ESCALERA	4-9
FIGURA 4.7: GRAFO DE TRANSICIONES MODIFICADO	4-11
FIGURA 4.8: GRAFO CON $P=0$ Y FIGURA 4.9: GRAFO CON $P=5$	4-14

LISTA DE TABLAS

Capítulo 1

TABLA 1.1: MARCACIÓN VS. OPERACIÓN	1-9
--	-----

Capítulo 2

TABLA 2.1: TABLA PARA EL AUTÓMATA DE 4 ESCOGENCIAS	2-13
--	------

Capítulo 3

TABLA 4.1: VALORES DE LOS POLINOMIOS	3-16
--	------

Capítulo 4

TABLA 4.1: TABLA DE TRANSICIÓN PARA EL ESTADO x_{ki}	4-7
TABLA 4.2: TABLA DE TRANSICIÓN DESDE EL ESTADO x_{k1}	4-8
TABLA 4.3: TABLA DE TRANSICIÓN DESDE EL ESTADO \tilde{x}_{k1}	4-9
TABLA 4.4: TABLA DE TRANSICIÓN DEL ESTADO x_k , DEL GRAFO DE LA FIGURA 4.3.....	4-10
TABLA 4.5: DECISIONES BINARIAS EXTENDIDAS.....	4-11

INTRODUCCIÓN

El principal interés de esta tesis está focalizado en la optimización de Sistemas Dinámicos a Eventos Discretos (SDED). Para ello, el desarrollo se centra en extensiones tanto en el principio de mínimo como en la programación dinámica, las cuales hemos denominado: Principio de Mínimo de Pontryaguin sobre Dominios Finitos (PMPDF) y Programación Dinámica Paramétrica (PDPA) respectivamente. Así, una breve descripción de lo SDED se dará, luego presentamos un resumen de los desarrollos realizados en la tesis y finalizaremos con una descripción de cada capítulo.

I.1. Sistemas a Eventos Discretos

En los años recientes tanto en la industria como en el trabajo académico se ha tenido más y más interés en técnicas para modelar, analizar, y controlar sistemas complejos tales como sistemas de manufactura flexible, redes de comunicaciones, sistemas de procesamiento paralelo, sistemas de control de tráfico, entre otros. Esta clase de sistemas son típicamente ejemplos de sistemas a eventos discretos, SED, este tema es una disciplina que emerge en teoría de sistemas de control. Los SED son una clase de modelos que contienen sistemas hechos por el hombre, estos consisten en un número finito de recursos (esto es: máquinas, canales de comunicaciones, procesadores, etc.) que son compartidos por varios usuarios (esto es: tipos de productos, paquetes de información, trabajos) los cuales contribuyen al logro de una meta (esto es: ensamblaje de productos, transmisión punto-a-punto de un conjunto de paquetes de información, computación paralela, entre otros).

Una de las características de los SED es que sus dinámicas son manejadas por eventos en oposición a los manejados por tiempo: *“El comportamiento de los SED es gobernado por eventos en vez del tic del reloj”*. Un evento corresponde al comienzo o finalización de una actividad. Si nosotros consideramos sistemas de producción, entonces, posibles eventos son: La culminación de una pieza en la máquina, la parada de una máquina, el vaciado de un buffer, etc. Los eventos ocurren en instantes de tiempo discretos. Los intervalos entre eventos no son necesariamente idénticos (SDED asíncronos), además ellos pueden ser determinísticos o estocásticos.

En la mayoría de los casos, los sistemas dinámicos a eventos discretos, SDED son caracterizados por la sincronización y la concurrencia. La sincronización requiere de la disponibilidad de varios recursos al mismo tiempo (esto es: antes de que podamos ensamblar un producto en una máquina, la máquina tiene que estar disponible al igual que las piezas; antes de especificar el trabajo que se debe ejecutar en un sistema de procesamiento en paralelo, el procesador y todos los datos de entrada deben estar disponibles, etc.). La concurrencia aparece cuando en un cierto tiempo, un usuario tiene que elegir entre varios recursos (esto es: en un sistema de producción un trabajo puede ser ejecutado en una de las máquinas que están disponibles en ese tiempo; en un sistema de procesamiento en paralelo de manejo de datos es un trabajo que puede ser ejecutado en uno de los procesadores que estén disponibles en el tiempo o el que más pronto llegará a estarlo). Los SED típicamente exhiben comportamientos asíncronos con mucho paralelismo e interacciones con su ambiente. Ellos usualmente tienen una estructura compleja y jerárquica.

Cuando nosotros queremos diseñar o rediseñar un control para asegurar que un sistema reúna ciertos requerimientos o cuando queremos verificar u optimizar el comportamiento de un sistema, nosotros podríamos representar nuestro conocimiento sobre las propiedades y el comportamiento del sistema en un modelo que nos permita estudiar y predecir el desempeño del sistema.

Hay muchas técnicas de modelado y análisis de SED, tales como teoría de colas, máquinas de estado, álgebra Max-Plus, lenguajes formales, autómatas, lógica temporal, procesos semi Markovianos generalizados, redes de Petri, análisis de perturbaciones, simulación en computadora, y así

sucesivamente. Para más información y para artículos tutoriales sobre SED y las técnicas mencionadas anteriormente el lector interesado es referido a [FB92], [LB95], [CC95], [GC94], [GC85], [GC94], [YH92], [PK93], [MB93], [TM89], [JP77], [JP81], [PI89], [MZ91] y las referencias citadas en estos. Todas estas técnicas de modelado y análisis tienen particulares ventajas y desventajas y esto realmente depende del sistema que deseamos modelar y de las metas que queramos alcanzar, para luego escoger la metodología que mejor satisfaga nuestras necesidades. Cuando tenemos que elegir una metodología apropiada, quizás los aspectos primordiales a considerar son el poder del modelado y el poder de decisión: el modelo más exacto, es lo menos que nosotros podemos decir analíticamente acerca de sus propiedades del sistema. Por ejemplo, si consideramos las Redes de Petri, una de las técnicas más poderosas para el modelado matemático de SED, se tiene que una solución analítica completa usualmente no es disponible. Además algunos de los problemas encontrados en análisis de redes de Petri son indecibles [TM89], [JP81], esto quiere decir que para estos problemas no puede existir un algoritmo general aplicable que resuelve el problema. El enfoque max-álgebra permite determinar y analizar muchas propiedades de los sistemas, pero este enfoque solo puede ser aplicado a la subclase de SED que pueden ser descritos por modelos invariantes en el tiempo max-lineales, podríamos decir que esta subclase corresponde a la subclase de SED invariantes en el tiempo determinista en el cual la sincronización y la no concurrencia ocurren.

La técnica mas ampliamente usada para estudiar SED ciertamente es la simulación por computadora. Una de las mayores desventajas de la simulación por computadora es que computacionalmente requiere bastantes recursos, ya que hay un alto grado de detalle en el modelo. Sin embargo, esto también lleva a un alto grado de correspondencia entre el modelo y el sistema real. Otra desventaja de la simulación por computadora es que no se tiene un real entendimiento del efecto de los cambios de los parámetros sobre las propiedades del sistema tales como robustez, estabilidad, optimalidad del funcionamiento del sistema, y así sucesivamente. Por consiguiente frecuentemente preferimos usar modelos matemáticos que están más preparados para el análisis matemático. La principal razón de una descripción matemática de un sistema dado, queda en la existencia de algoritmos eficientes para evaluar el desempeño del sistema. Esta es una significativa ventaja sobre el tiempo consumido y los costos de simulaciones, lo cual es usualmente requerido para obtener la misma información. Además, las técnicas analíticas también proveen un buen punto de vista en el efecto de cambios de parámetros sobre las propiedades del sistema. Sin embargo, como ya hemos mencionado, tenemos que tomar en cuenta que hay un intercambio entre la precisión de nuestro modelo por un lado y la técnica disponible para el análisis por otro lado. Por consiguiente, normalmente se usa una mezcla de métodos analíticos aproximaciones, heurística, y simulaciones cuando queremos estudiar las propiedades y el comportamiento de SED ó cuando queremos diseñar un controlador para este.

Hay tres niveles principales para modelado de SED, el nivel lógico, el nivel temporal, y el nivel estocástico. Los modelos lógicos (tales como (no medidos en tiempo): redes de Petri, máquinas de estado finito) son usados para estudiar las propiedades que conciernen solamente al ordenamiento de eventos. Si estamos interesados en el instante de tiempo en el cual el evento ocurre entonces tenemos que usar modelos temporales (tales como redes de Petri temporizadas y modelos max-algebraicos). Los modelos estocásticos (tales como procesos de semi-Markov generalizados) si nosotros queremos determinar el comportamiento esperado sobre alguna condición estadística dada. Los modelos presentados en esta tesis, por ser modelos formales podrían estar en cualquiera de los niveles anteriores, pero solo el nivel lógico y el temporal serán considerados.

I.2 Breve Descripción del Desarrollo de esta Tesis

Para la extensión realizada al Principio de Mínimo, PMPDF, no es considerado ningún criterio de convexidad, preservando la optimización multiobjetivo (con algún criterio de ordenamiento o prioridad) y el formalismo Hamiltoniano. El resultado de la condición de optimalidad, sobre un espacio de estados y control finito, se traduce en un conjunto de desigualdades variacionales en términos del control óptimo $u^*(\cdot)$, el control variado v y el estado como parámetro. Así, en principio,

las funciones de síntesis en la optimización deberían ser dadas por la resolución, hacia atrás en el tiempo, de la desigualdad variacional, generando una función paramétrica en el estado como control óptimo en cada etapa. El proceso continua hasta que la condición inicial es substituida y a tiempo directo se obtiene los valores de control óptimo. Es de resaltar que no existe un único método para dar solución al conjunto de desigualdades variacionales. Por lo que el mayor esfuerzo en este trabajo se centra en dar solución a este conjunto de desigualdades variacionales. Por ello un procedimiento denominado Min-Max para resolver las desigualdades variacionales es propuesto y desarrollado. El procedimiento Min-Max obtiene primero la expresión del control variado v que minimiza la desigualdad y luego la expresión del control óptimo $u^*(.)$ que maximiza la desigualdad. Así, una función es obtenida en cada etapa. La obtención del mínimo y el máximo en el procedimiento Min-Max, son dados por un procedimiento desarrollado el cual denominamos algoritmo SCDO. El algoritmo SCDO, permite obtener los mínimos de una función cuyo dominio de definición es el conjunto producto $\{-1,1\}^{k+1}$, por lo que el uso del algoritmo SCDO pasa por utilizar la representación diádica sobre el dominio de definición de v y $u^*(.)$. Si bien es cierto que en primera instancia el principio de mínimo lleva las condiciones de óptimo en lazo abierto, en nuestro caso la solución de óptimo son obtenida a lazo cerrado, pero en forma simbólica. Debido a la naturaleza simbólica del algoritmo SCDO en el procedimiento Min-Max, es posible incorporar parámetros exógenos de evolución tanto en el sistema como en la función de costo, permitiendo obtener un control óptimo en función del estado y parámetros exógenos de evolución haciendo atractivo este desarrollo respecto a los métodos clásicos de optimización conocidos.

En el caso de la programación dinámica, la extensión es dada por el uso del algoritmo SCDO en cada una de las etapas en el proceso de optimización discreta definida sobre espacio de estado y control finitos. Así, de la misma manera como en el PMPDF, SCDO permite obtener la forma explícita de la control óptimo, inclusive cuando parámetros exógenos de evolución, que afectan al sistema o a la función de costo, son requeridos.

Los alcances de lo desarrollado en esta tesis se presentan en cuatro capítulos y un anexo. Así, en el **Capítulo 1: Modelos Formales de SED**, la principal meta es la obtención de modelos formales a partir de las descripciones clásicas de SDED. Así sobre dos familiar particulares de SED, *lenguajes formales* y *autómatas de estado finito*, se muestran procedimientos para obtener estos modelos formales. La intención de este capítulo al obtener modelos formales es que las extensiones de los métodos clásicos de optimización, Principio de Mínimo y Programación Dinámica, necesitan estas descripciones. En el **Capítulo 2: Cálculo Simbólico en Optimización Discreta**, se muestra el desarrollo progresivo de un método para calcular mínimos explícitos paramétricos de una función con dominio discreto finito. Este método desemboca en un algoritmo simbólico el cual denominamos SCDO. SCDO permite obtener los mínimos de una función paramétrica en donde las variables están definidas sobre el hiper-cubo $\{-1,1\}^k$ y los parámetros están definidos en un dominio dado. Así el dominio discreto finito de la función a minimizar es representado en los vértices de un hyper-cubo $\{-1,1\}^k$, y el algoritmo SCDO puede ser aplicado. Varias propiedades interesantes de funciones Booleanas son descubiertas aquí y son relacionadas a una particular representación, sobre los vértices de un hiper-cubo $\{-1,1\}^k$. El algoritmo simbólico propuesto tiene esencialmente la misma complejidad computacional que los métodos de búsqueda clásica, con la ventaja de su posible aplicación a la optimización paramétrica. En el **Capítulo 3: Enfoque Simbólico del Principio de Mínimo**, esta dividido en tres partes, fundamentalmente. En la primera parte, la extensión del Principio de mínimo para sistemas dinámicos a eventos discretos, Teorema PMPDF, se presenta. Así una breve descripción del Principio de Mínimo en continuo es dada y luego nuestra extensión es desarrollada con detalle. El resultado de aplicar PMPDF es una desigualdad variacional, del control variado v , el control óptimo $u^*(.)$, parametrizada en cada etapa por el estado, como condición necesaria para el óptimo. Un algoritmo simbólico para la implantación de la técnica es desarrollado y ejemplos son mostrados. La segunda parte se centra en la resolución de las desigualdades variacionales. Así, el algoritmo Min-Max, basado en el algoritmo

SCDO, es propuesto y desarrollado como método para obtener la solución (simbólica) de la desigualdad variacional. Ejemplos de ello son mostrados. Como un tópico adicional, es el uso del método de relajación, en la optimización de problemas discretos ya que este tiene una relación estrecha con el principio de mínimo. Dos casos considerados y resueltos; con funciones de costo convexo y cóncavo sobre el estado final muestran esta relación. En el **Capítulo 4: Programación Dinámica Paramétrica**, se da una extensión del método de programación dinámica cuando el uso del algoritmo SCDO es usado para resolver en cada etapa la ecuación discreta de Bellman. El resultado es una forma explícita (simbólica) del control óptimo en función del estado. Así, el método de programación dinámica en continuo y su equivalente en discreto se presentan. Luego dos enfoques basadas en la optimización simbólica en un paso se desarrollan y por último el algoritmo para aplicar la programación dinámica paramétrica es desarrollado y aplicado a unos ejemplos. En el **Anexo A: Fundamentos sobre Lenguajes Formales y Realización Mínima**, aquí presentamos algunos fundamentos matemáticos requeridos para obtener los modelos formales del Capítulo 1. Un conjunto de definiciones, y teoremas para lenguajes formales son mostrados, así como procedimientos para obtener una realización mínima de modelos formales. Y por último en **Conclusiones y Problemas Abiertos**, una descripción de los alcances obtenidos en el desarrollo de esta tesis, así como algunas perspectivas de trabajos futuros y problemas abiertos son expuestos.

Introducción

INTRODUCCIÓN	1
I.1. SISTEMAS A EVENTOS DISCRETOS	1
I.2 BREVE DESCRIPCIÓN DEL DESARROLLO DE ESTA TESIS	2

MODELOS FORMALES DE SDED

Como se dijo en la introducción, el principal interés de esta tesis está focalizado en la optimización de SDED. Para ello, el desarrollo se centra en extensiones tanto en el principio de mínimo como en la programación dinámica, las cuales hemos denominado: Principio de Mínimo para procesos a Eventos Discretos sobre dominios finitos y Programación Dinámica Paramétrica respectivamente.

En ambas extensiones, las descripciones formales (entrada-estado-salida) de SDED son requeridas. Es por ello, que este capítulo lo dedicamos al desarrollo de modelos formales y consideremos dos familias particulares de SDED. Estas familias corresponden a aquellas descritas por

- Lenguajes formales,
- Autómatas de estado finito.

En el caso de lenguajes formales, el modelo formal es obtenido a través de sistemas dinámicos inducidos por concatenación. Así una realización de estos sistemas es mostrada y una realización mínima del modelo se establece, basada en dos construcciones que se desarrollan.

En el caso de modelos descritos por autómatas de estado finito, el modelo formal se obtiene cuando la dinámica del autómata (conexión de nodos por arcos) es atrapada en una descripción vía interpolación, particularmente interpolación de Lagrange. Así un sistema dinámico con lado derecho polinómico es obtenido.

En cada uno de los casos anteriores, un modelo formal (entrada-estado-salida) es obtenido.

A continuación damos al lector una idea de como representar a esta clase de sistemas a eventos discretos por modelos formales. Ejemplos ilustrativos son también mostrados. Así, el modelo de un sistema dinámico a eventos discretos, en nuestro caso, está definido por la quintupla

$$G(X, \Omega, Y), S(\Omega, Y, X, \varphi(\cdot, \cdot), \eta(\cdot, \cdot)). \quad (1.1)$$

donde:

- Ω es el conjunto no vacío de eventos, esto es el conjunto de símbolos tales como alfabeto, números, etc.
- Y es el conjunto no vacío de valores de salida.
- X es el conjunto no vacío de estados o valores del estado.
- $f(\cdot, \cdot): X \times U \rightarrow X$, la función del próximo estado.
- $\eta(\cdot, \cdot): X \times U \rightarrow Y$, es la función de la próxima salida.

1.1. Modelos Formales en Lenguajes Cerrados: Casos Generales de Sistemas Lingüísticos

Para la obtención de modelos formales en lenguajes cerrados daremos algunos fundamentos matemáticos que nos permiten mostrar las propiedades de los lenguajes y como éstas pueden ser usadas para la obtención de modelos formales. Así, la inducción de un sistema dinámico vía concatenación y la realización mínima de modelos serán tratadas. Un ejemplo sobre la simplificación de una expresión matemática, muestra como una descripción lingüística puede conformar un modelo formal de tal manera de proponer una estrategia óptima para realizar dicha simplificación.

Los fundamentos matemáticos aquí presentados son complementados en el anexo A

1.1.1. Lenguajes Cerrados y Realización Mínima

Clase de Lenguajes

Sea $L \subset \Omega^*$ un lenguaje. Supongamos que para una palabra $u \in L$, $u = u_0 u_1 \dots u_i$, para todos los $i=0,1,\dots,k-1$, entonces podemos generar palabras que se encuentren dentro del lenguaje por concatenación de la siguiente manera: Si $u = u_0 u_1 \dots u_{k-1}$ y $x_0 = \theta$, entonces

$$\begin{aligned} x_1 &= x_0 u_0 = u_0 \in L \\ x_2 &= x_1 u_1 = u_0 u_1 \in L \\ &\vdots \\ x_{i+1} &= x_i u_i = u_0 u_1 \dots u_i \in L \\ &\vdots \\ x_k &= x_{k-1} u_{k-1} = u_0 u_1 \dots u_{k-1} \in L \end{aligned}$$

Prefijos

$u \in \Omega^*$, es prefijo de una palabra $v \in \Omega^*$, si para algún $w \in \Omega^*$, $v=uw$.

Así, $x_k = u_0 u_1 \dots u_{k-1}$ y los $x_i = u_0 u_1 \dots u_{i-1}$ son los prefijos de la palabra u .

Lenguaje Cerrado

Se dice que un lenguaje es cerrado $\Leftrightarrow \forall u \in L$, los prefijos de u también pertenecen al lenguaje. Así úsi definimos la clausura de L como

$$\bar{L} = \{v : uv \in L, \text{ para algún } v \in \Omega^*\}, L \subset \Omega^*$$

Para este tipo de lenguaje se definirán conceptos similares a los descritos en el anexo A. Consideremos de nuevo una función P como sigue:

$$\begin{aligned} p : L &\rightarrow Y, L \subset \Omega^* \\ u \in L, L(u) &= \{v : uv \in L\} \subset \Omega^* \\ \Rightarrow u\theta &= u \in L \Rightarrow \theta \in L \end{aligned}$$

Otro caso particular de interés es $L(\theta) = L$. En efecto, $\theta u \in L \Leftrightarrow u \in L$.

Propiedades del Lenguaje Cerrados

Distinguibilidad: Si se tienen dos palabras $u, v \in L$, son distinguibles, si :

1. $L(u) \neq L(v)$ y
2. $w \in L(u) = L(v) \Rightarrow p(uw) \neq p(vw)$.

Indistinguibilidad: Si se tienen dos palabras $u, v \in L$, son indistinguibles, si :

1. $L(u) = L(v)$ ó
2. $\exists w \in L(u) = L(v) \Rightarrow p(uw) = p(vw)$.

Esto se denota uNv . En los lenguajes cerrados, la indistinguibilidad es una relación de equivalencia, para esto es necesario chequear que N es simétrica y transitiva.

Transitiva: Si se tienen dos palabras $u, v \in L$, tenemos que probar que si uNz y $zNv \Rightarrow uNv$.

Demostración

$$L(u) = L(z) \text{ y } L(z) = L(v) \Rightarrow L(u) = L(v) \quad \forall w \in L(u) = L(v)$$

$$p(uw)=p(zw) \text{ y } p(vw)=p(zw) \Rightarrow p(uw)=p(vw) \Rightarrow uNv.$$

Simetría: La prueba es inmediata de la definición de indistinguibilidad (ver anexo A).

Clases de equivalencia: Consideramos el conjunto de las palabras de L , que son equivalentes (indistinguibles) y lo denotamos como $[u]$, gráficamente se muestra en la Figura 1.1.

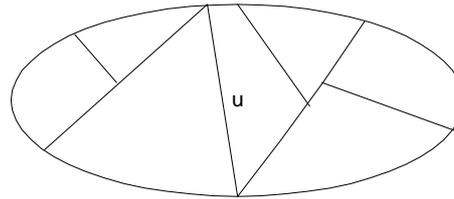


Figura 1.1: Clases de equivalencia

Como se puede observar la relación de equivalencia hace una clasificación de los elementos. Entonces $[u] = \{v : uNv\} \subset L$ y por lo tanto debe cumplirse los siguientes casos:

1. $[u]=[v]$
2. $[u] \cap [v]=?$

Conjunto X de todas las Clases de Equivalencia

Se define al conjunto X de todas las clases de equivalencia como:

$$X = \{[u] : u \in L\}.$$

Sobre X podemos establecer algunas definiciones como para cada $x \in X$:

$$L(x)=L([u])=L(u),$$

quedando bien definida ya que $L(x)$ no depende del representante u de la clase $x=[u]$. Se puede considerar también uNu^* , esto implica que $L(u)=L(u^*)=L(x), \forall x \in X \subset \Omega^*$.

Aplicación Parcial (Dinámica)

Sea $\theta \in L \Rightarrow x_0 = [\theta], \phi : X \times \Omega^* \rightarrow X$. Donde el dominio de ϕ esta dado por:

$$\text{Dom}\phi = \cup_{x \in X} \{x\} \times L(x),$$

y $\{x\} \times L(x)$ contiene pares del tipo (x,v) tal que $v \in L(x)$.

El dominio de ϕ no es completo lo que indica que es una aplicación parcialmente definida.

Función de Transición

La función de transición ϕ_u está parcialmente definida como , $u \in \Omega^*, \phi_u : X \rightarrow X$,

$$\phi_u(x) = [uv] \text{ donde } x=[v].$$

Función de salida

La función de salida esta definida como sigue, $\sigma : X \rightarrow Y, \sigma(x) = \sigma([u]) = p(u)$.

De la misma forma como se definió la relación entre la caja negra y el conjunto de ecuaciones para el alfabeto y el sistema dinámico inducido por la concatenación en el anexo A, se hará para el lenguaje cerrado y la dinámica parcialmente definida.

Si la igualdad $\sigma(\phi_u(x_0)) = p(u)$ se cumple se habrá obtenido la relación entre ellas. Reescribiendo tenemos $L(x_0) = L([\theta]) = \{u : \theta, u \in L\} = L$. Así gráficamente la relación entre la caja negra y el sistema lingüístico se muestra en la Figura 1.2.

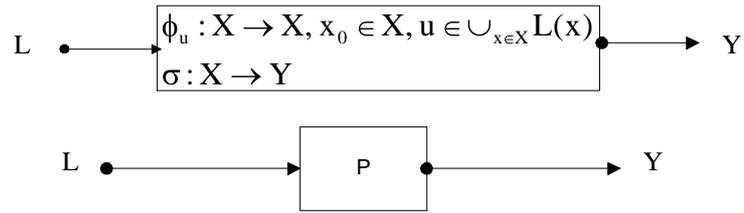


Figura 1.2: Relación entre la caja negra y el sistema lingüístico

Lo que significa que $\sigma(\phi_u(x_0)) = \sigma(\phi(x_0, u)) = \sigma((\phi[\theta], u)) = \sigma([\theta u]) = \sigma([u]) = p(u)$, la ecuación anterior representa la función de entrada salida por medio de una dinámica con entrada y una función de salida, lo que implica que ella es una realización. Así el sistema lingüístico (dinámico) y la caja negra tienen la misma notación, respectivamente:

$$(X, x_0, \phi_u, \sigma),$$

donde :

- X: Espacio de estado.
- x_0 : Condición inicial.
- ϕ_u : Dinámica.
- σ : Salida

$$(\Omega, Y, P),$$

donde:

- Ω : Alfabeto.
- Y: Salida.
- P: Proceso.

Accesibilidad del cero o del estado inicial

Un sistema dinámico (X, x_0, ϕ_u, σ) es x_0 -accesible si para cada $x \in X$ existe $u \in L$ con $x = \phi_u(x_0)$. Por lo tanto el ser accesible considera que si $x \in X$, entonces $x=[u] \Rightarrow x = \phi_u(x_0)$, probando la igualdad tenemos:

$$\phi_u(x_0) = \phi_u([\theta], u) = [\theta u] = [u] = x.$$

En general podemos decir que un sistema dinámico $(\tilde{X}, \tilde{x}_0, \tilde{\phi}, \tilde{\sigma})$, es una realización de la caja negra $p : L \rightarrow Y$ si:

$$\tilde{\sigma}(\tilde{\phi}(\tilde{x}_0, u)) = p(u), x=[u].$$

Minimalidad

Acá se harán dos construcciones para establecer el concepto de minimalidad.

Primera construcción

Si se tiene un sistema original no accesible $(\tilde{X}, \tilde{x}_0, \tilde{\phi}, \tilde{\sigma})$ de $p: L \rightarrow Y$, cualquiera, que sea la realización. Se construye un sistema accesible que es la realización de la misma aplicación p , y el cual es más pequeña, ya que es un subconjunto de \tilde{X} .

Los pasos a seguir para construir este sistema accesible son:

1. Sea $X_1 \subset \tilde{X}$ un nuevo espacio de estado dado por:

$$X_1 = \{x : x_0 - \text{accesible}\} \subset \tilde{X}$$

$$(X_1, x_1, \phi^1, \sigma^1, Y_1) = G_1.$$

Definamos $x_1 = \tilde{x}_0 \in X_1$, BBBB. $\tilde{\phi}(\tilde{x}_0, \theta) = \tilde{x}_0$, si se restringe $\tilde{\phi}/X_1$, como

$\tilde{\phi}/X_1 : X_1 \rightarrow \tilde{X}_1$, tal que $\tilde{\phi}/X_1 = J \circ \phi^1$ con $\phi^1 : X_1 \rightarrow X_1$ y $J : X_1 \rightarrow \tilde{X}_1$

$$\begin{array}{ccc} X_1 & \xrightarrow{\phi^1} & X_1 \\ \tilde{\phi}/X_1 \downarrow & \searrow J \circ \phi^1 & \downarrow J \\ \tilde{X} & \xrightarrow{\tilde{\phi}^1} & \tilde{X} \end{array}.$$

Entonces $\tilde{\phi}(x, u) = \tilde{\phi}(\tilde{\phi}(\tilde{x}_0, v), u) = \tilde{\phi}(\tilde{x}_0, vu) \in X_1$.

Su importancia es que el estado, aplicando la dinámica, no va a salir del espacio de estado X_1 . En resumen queda:

$$x = \tilde{\phi}(\tilde{x}_0, u) = \tilde{\phi}(x_0^1, u) = \phi_1(x_0^1, u) \Rightarrow x_0^1 - \text{accesible}.$$

2. Ahora se redefine la salida σ_1 del nuevo espacio de estado X_1 . Tomemos $\sigma_1 = \tilde{\sigma}/X_1 : \tilde{X} \rightarrow Y$, y se quiere ver si ésta tiene la misma aplicación de entrada-salida. Así,

$$p(u) = \tilde{\sigma}(\tilde{\phi}(\tilde{x}_0)) = \sigma_1(\phi^1(x_1, u)) = \sigma_1(\tilde{\phi}(\tilde{x}_0, u)) = \tilde{\sigma}(\tilde{\phi}(\tilde{x}_0, u)) = p(u),$$

por lo que tenemos una nueva realización de la aplicación p . En definitiva nuestro nuevo espacio de estado tiene la notación : $(X_1, x_0^1, \phi, \sigma_1)$, esta construcción nos lleva a una nueva realización más pequeña, pero, no es una realización mínima.

Segunda construcción

Esta segunda construcción se va a desarrollar de forma similar que la primera y además se usarán los resultados obtenidos en la misma.

Dado el sistema dinámico de la primera construcción, $(X_1, x_1, \phi^1, \sigma_1) \Rightarrow x_1, x_2 \in X_1$.

Los pasos a seguir para construir este sistema accesible son:

1. Se define un nuevo espacio de estado X_2 como el conjunto de clases sobre el espacio X_1 , tal que $X_2 = \{[x] : x \in X_1\}$, con $x_2 = [x_1]$, $x_2 \in X_2$, $x_1 \in X_1$. Como es indistinguible se cumple que $L(x_2) = L([x_1]) = L(x_1)$.

Ahora se redefine la dinámica ϕ_2 como $\phi_2(x_2, u) = \phi_2([x_1], u) = [\phi_1(x_1, u)]$.

Si consideramos $\phi_2(x^2, u) = \phi_2([\bar{x}^1], u) = [\phi_1(\bar{x}^1, u)]$, entonces $[\phi_1(x^1, u)] = [\phi_1(\bar{x}^1, u)]$ y hay que comprobar que $x^1 N \bar{x}^1 \Rightarrow \phi_1(x^1, u) N \phi_1(\bar{x}^1, u)$.

Como sabemos $u \in L(x^1) = L(\bar{x}^1)$.

Si tomamos $v \in L(\phi_1(x^1, u)) \Leftrightarrow uv \in L(x^1)$, como $L(\phi_1(x^1, u)) = L(\phi_1(\bar{x}^1, u))$, $\forall v$, se tiene que las salidas son:

$$\sigma_1[\phi_1(\phi_1(x^1, u), v)] = \sigma_1(\phi_1(x^1, uv)) = \sigma_1[\phi_1(\phi_1(\bar{x}^1, u), v)] = \sigma_1(\phi_1(\bar{x}^1, uv)),$$

por lo tanto $[\phi_1(x^1, u)] = [\phi_1(\bar{x}^1, u)]$.

2. Ahora se redefine la salida σ_2 como la salida del espacio de estado X_2 . Se define

$$\sigma_2(x^2) = \sigma_2([\bar{x}^1]) = \sigma_1(x^1) = \sigma_2([\bar{x}^1]) = \sigma_1(x^1)$$

y se quiere saber si tiene la misma aplicación entrada-salida.

Entonces tomando $\sigma_2(\phi_2(x_0^2, u)) = \sigma_2([\phi_1(x_0^1)]) = \sigma_1(\phi_1(x_0^1, u)) = p(u)$, se observa que es una nueva realización con la misma aplicación p . En definitiva, el nuevo espacio de estado tiene por notación $(X^2, x_0^2, \phi_2, \sigma_2)$, éste es un nuevo sistema dinámico definido del sistema anterior en donde solo existen estados distinguibles.

Recordemos que esta realización debe contener como propiedades la accesibilidad y la observabilidad, por lo que ahora mostramos que esta nueva realización posee estas propiedades.

Accesibilidad: Considere x_0^2 -accesible. Se toma $x^2 \in X_1$, arbitrario, tal que

$$x^2 = [x^1] \exists u \Rightarrow x^1 = \phi_1(x_0^1, u). \text{ Calculando}$$

$$\phi_2(x_0^2, u) = \phi_2([\bar{x}^1], u) = [\phi_1(x_0^1, u)] = [x^1] = x^2$$

Por lo tanto es accesible.

Observabilidad: Si $x^2 \neq \bar{x}^2 \Rightarrow$ son distinguibles. Entonces hay que probar que $\exists u$, tal que:

$$L(x^2) \neq L(\bar{x}^2). \text{ Si } x^2 = [x^1], \bar{x}^2 = [\bar{x}^1], x^1 \text{ y } \bar{x}^1,$$

son distinguibles. Entonces $L(x^1) \neq L(\bar{x}^1) \Leftrightarrow L(x^2) \neq L(\bar{x}^2)$.

Existe un u tal que $\sigma_2(\phi_2(x^2, u)) = \sigma_2(\phi_2(\bar{x}^1, u))$.

Ya se había señalado que $L(x^1) \neq L(\bar{x}^1)$. No obstante se cumple que $\exists v \in$

$$\sigma_1(\phi_1(x^1, v)) \neq \sigma_1(\phi_1(\bar{x}^1, v)).$$

Así

$$\sigma_1(\phi_1(x^1, v)) = \sigma_2[\phi_1(x^1, v)] = \sigma_2(\phi_2([\bar{x}^1], v)) = \sigma_2(\phi_2(x^2, v)),$$

por otro lado $\sigma_1(\phi_1(\bar{x}^1, v)) \neq \sigma_1(\phi_1(\bar{x}^2, v))$, lo que implica que

$$\sigma_2(\phi_2(x^2, v)) \neq \sigma_2(\phi_2(\bar{x}^2, v)). \text{ Por lo tanto el sistema es observable.}$$

Realización Mínima

Para obtener la realización mínima de sistema, se mostrará un procedimiento basado en la siguiente definición y los dos teoremas. Otro procedimiento es mostrado en el anexo A.

Definición: Sean, $(X_1, x_0^1, \phi_1, \sigma_1)$, $(X_2, x_0^2, \phi_2, \sigma_2)$ dos sistemas dinámicos, sobre el mismo L y con el mismo espacio de salida Y . Si $f : X_1 \rightarrow X_2$ es una aplicación que cumple con las siguientes propiedades:

1. $f(x_0^1) = x_0^2$,

2. Sea $x \in X_1$; $L(x)=L(f(x))$, $u \in L(x)$, entonces $f(\phi_{1u}(x)) = f(\phi_{2u}(x))$,
3. $\sigma_1(x) = \sigma_2(f(x))$, entonces se dice, que f preserva la dinámica. Si f preserva la dinámica, entonces los dos sistemas realizan el mismo sistema de entrada salida:

$$p_1(u) = \sigma_1(\phi_{1u}(x_0^1)) = \sigma_2(f(\phi_{1u}(x_0^1))) = \sigma_2(f(\phi_{2u}(x_0^1))) = \sigma_2(\phi_{1u}(f(x_0^1))) = \sigma_2(\phi_{2u}(x_0^2)) = p_2(u)$$

De lo anterior podemos enunciar los siguientes teoremas

Teorema 1 : Si $P : L \rightarrow Y$, tiene dos realizaciones mínimas entonces ellas son isomorfas.

Teorema 2 : La realización $(\tilde{X}, \tilde{x}_0, \tilde{\phi}, \tilde{\sigma})$ es mínima si y solo si x_0 es accesible y observable. Esto indica que la minimalidad implica accesibilidad y observabilidad.

Procedimiento

Acá se mostrará como redefiniendo el espacio de estado podemos obtener un modelo matemático del proceso lingüístico.

Un nuevo espacio de estado

Este nuevo espacio de estado está constituido por la longitud de u . Sea $u = u_0 u_1 u_2 \dots u_{k-1}$ y su longitud como Longitud de u por $|u| = k$, con $\theta \Leftrightarrow x_0 = [\theta]$. Se define:

$$\begin{aligned} X_0 &= \{x_0\}, \text{ todas las clases de longitud 0, es decir, la palabra vacía.} \\ &\vdots \\ X_k &= \{[u] : |u| = k\}, \text{ todas las clases de las palabras de longitud k.} \end{aligned}$$

Una nueva dinámica

Si se toma: $\phi : X_k \times \Omega^* \rightarrow X_1$, como una dinámica global, donde X_1 depende de la longitud de la palabra. Entonces se define una dinámica local que corresponde a las ecuaciones en diferencia como $f_k : X_k \times \Omega \rightarrow X_{k+1}$ cuyo dominio es $\text{Dom } f_k : x \in X_k$, con $\Omega(x) =$ las letras accesibles. Si se tiene $w \in \Omega(x)$, entonces ϕ está definida como $f_k(x, w) = \phi(x, w)$. Si se tiene un estado accesible, entonces existe $u = u_0 u_1 u_2 \dots u_{k-1}$ tal que $\phi(x_0, u) = x$, con lo que se obtiene:

$$\begin{aligned} x_1 &= f_0(x_0, u_0) = [u_0] \in X_1, \\ x_2 &= f_1(x_1, u_1) = f_1([u_0], u_1) = [u_0 u_1] \in X_2 \\ &\vdots \\ x_k &= f_{k-1}(x_{k-1}, u_{k-1}) = f_{k-1}([u_0 \dots u_{k-2}], u_{k-1}) = [u_0 \dots u_{k-1}] \in X_k. \end{aligned}$$

Esta construcción nunca termina porque cada letra es accesible a la palabra anterior, además tiene similitud con un control discreto, con la diferencia que esto es un modelo lingüístico

1.1.2. Ejemplo: Simplificación Matemática por cálculo formal

Consideremos el problema de simplificación de una expresión matemática, a partir de un cierto número de identidades que serán descritas. El objetivo es obtener la simplificación de la expresión en un número finito (mínimos) de pasos, usando las identidades mencionadas.

Consideremos la expresión

$$\frac{(x^2 - y^2)}{(x + y)^2}, \quad (1.2)$$

y las identidades siguientes:

$$\begin{aligned} (x^2 - y^2) &= (x + y)(x - y), \\ x^2 &= xx \\ \frac{yx}{rx} &= \frac{y}{r} \end{aligned} \quad (1.3)$$

Una secuencia posible para obtener la simplificación esta dada por:

$$\begin{aligned} \frac{(x^2 - y^2)}{(x + y)^2} &= \frac{(x - y)(x + y)}{(x + y)^2} = \frac{(x - y)(x + y)}{(x + y)(x + y)} = \frac{(x - y)}{(x + y)}, \text{ ó} \\ \frac{(x^2 - y^2)}{(x + y)^2} &= \frac{(x^2 - y^2)}{(x + y)(x + y)} = \frac{(x - y)(x + y)}{(x + y)(x + y)} = \frac{(x - y)}{(x + y)}. \end{aligned}$$

Como vemos, las identidades actúan sobre ciertas partes de la expresión.

Con esta descripción del problema, podemos plasmar un sistema dinámico a eventos discretos definido sobre un lenguaje formal como sigue.

Un Modelo de Sistemas a Eventos Discretos

Consideremos el hecho que un ambiente de cálculo contiene las identidades anteriores con el fin de efectuar la simplificación de una expresión análoga a la descrita anteriormente.

Sea $m_i, i=1, \dots, j$, el conjunto de las j selecciones posibles para utilizar las identidades sobre la expresión matemática. Sea m_0 la condición inicial. Sean $w_l, l=1, \dots, j$, el conjunto de las j identidades y m_{j+1} la condición final. Por hipótesis solo es posible aplicar una identidad a la porción marcada (m_i) de la expresión.

Sea, $\Omega = \{m_0, m_1, \dots, m_{j+1}, w_1, \dots, w_j\}$, el alfabeto representante de las selecciones y las operaciones. En el caso particular planteado en (1.2) y (1.3), el alfabeto es el siguiente: $\Omega = \{m_0, m_1, \dots, m_4, w_1, \dots, w_3\}$. En éste caso el número de selecciones coincide con el número de operaciones, pero en general no necesariamente coinciden

El proceso comienza con una selección y consideramos la siguiente convención :

$$\begin{aligned} m_i m_i &= m_i m_i \forall i, \\ m_i m_j &= m_i m_j \forall i, j, i \neq j, \\ m_i w_i &\text{ la selección y la operación deben corresponder.} \end{aligned}$$

De esta manera tenemos, por ejemplo, que en el k -ésimo paso, tenemos una palabra admisible de longitud k , como la siguiente:

$$\overbrace{m_1 w_1 m_1 w_1 m_3 w_3 m_2 w_2 \dots m_5 w_5}^k$$

El modelo

La yuxtaposición o concatenación de k+1 letra de una palabra de longitud k es considerada como una acción sobre el estado, quiere decir que podemos interpretar el proceso de yuxtaposición de letras a las palabras como la evolución del sistema dinámico que definimos a continuación. Sea L el conjunto de todas las palabras posibles, formada por las letras de Ω. Entonces podemos definir la dinámica sobre L, como la aplicación parcial f:LxΩ→L, f(wu)=wu. Si L = U_{k=0}[∞]L_k, donde L_k es el conjunto de todas las palabras admisibles de longitud k, entonces, podemos obtener una palabras w = u₀u₁ ⋯ u_{k-1}, pudiendo definir:

$$f_k : L_k \times \Omega \rightarrow L_{k+1}, f_k(u_0 u_1 \dots u_{k-1}, u_k) = u_0 u_1 \dots u_{k-1} u_k.$$

Si la palabra inicial es x₀ = 0, y x_k = u₀u₁ ⋯ u_{k-1} ∈ L_k, entonces x_{k+1} = f(x_k,u_k), x₀ = 0 es una descripción análoga de un sistema dinámico de control discreto o de un sistema dinámico a eventos discretos.

Consideremos ahora, el siguiente SDED:

$$x(k+1) = f_k(x(k),u(k)), x(0) = x_0,$$

donde

$$f_x : X_k \times U_k \rightarrow X_{k+1}, x(k) \in X_k, u(k) \in U_k$$

$$X_{k+1} = X_k \cup (m_j, w_n)_k, \dim(x(k)) = k, \dim(u(k)) = 1.$$

De esta manera, el conjunto de controles admisibles para las palabras admisibles se presentan en la siguiente tabla

Instrucción (Selección/Operación)	Control Admisibles	Aplicación de la Identidad
m ₀		Condición Inicial
m ₁	1	(x ² - y ²) → (x ² - y ²)
w ₁	2	(x ² - y ²) → (x - y)(x + y)
m ₂	3	x ² → x ²
w ₂	4	x ² → x x
m ₃	5	y x/r x → y x/r x
w ₃	6	y x/r x → y/r
m ₄	7	Condición Final

Tabla 1.1: Selección vs. Operación

El sistema dinámico anterior tiene una estructura inhabitual ya que dimensión del vector de estado es variable. La dinámica es definida como la concatenación de instrucciones que forma la palabra; quiere decir que ella describe la secuencia de identidades utilizadas para resolver el problema de simplificación

Cada palabra de longitud k está representada por un vector de longitud k.

De esta manera, las aplicaciones f_k pueden ser calculadas. Obteniéndose las siguientes etapas.

Etapas 1, Se comienza con la selección (u operación). x₀ = 0, U₀ = {1,3}, x¹ = {1,3} y

$$\begin{aligned} f_0(0,1) &= 01 = m_0 m_1, \\ f_0(0,3) &= 03 = m_0 m_2, \end{aligned}$$

la dinámica esta dada por:

$$f_0(0, u(0)) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} x_0(0) + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u(0) = \begin{pmatrix} x_0(1) \\ x_1(1) \end{pmatrix}.$$

Etapa 2, Seleccionar y operar sobre la selección anterior. $x^{1'} = \{01,03\}$, $U_1 = \{1,2,3,4\}$ y $x^{2'} = \{011,013, 033,031,012, 034\}$

$$\begin{aligned} f_1(01,1)' &= (011)' = m_0 m_1 m_1, \\ f_1(01,2)' &= (012)' = m_0 m_1 w_1, \\ f_1(01,3)' &= (013)' = m_0 m_1 m_3, \\ f_1(01,4)' &= \text{No es permitido}, \\ f_1(03,1)' &= (031)' = m_0 m_3 m_1, \\ f_1(03,2)' &= \text{No es permitido}, \\ f_1(03,3)' &= (033)' = m_0 m_3 m_3, \\ f_1(03,4)' &= (034)' = m_0 m_2 w_2. \end{aligned}$$

La dinámica será entonces descrita por:

$$f_1(x_0(1), x_1(1), u(1)) = \begin{pmatrix} x_0(2) \\ x_1(2) \\ x_2(2) \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} x_0(1) \\ x_1(1) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} u(1)$$

Etapa 3, Seleccionar u operar sobre el paso anterior, $x^{2'} = \{011,013, 031,033,012, 034\}$, $U_2 = \{1,2,3,4\}$ et $x^{3'} = \{0111,0113,0131,0133,0311,0313,0331,0333,0123,0341\}$, La dinámica será descrita entonces por:

$$f_2(x_0(2), x_1(2), x_2(2), u(2)) = \begin{pmatrix} x_0(3) \\ x_1(3) \\ x_2(3) \\ x_3(3) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_0(2) \\ x_1(2) \\ x_2(2) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u(2)$$

Etapa 4: $x^{3'} = \{0ijl, 0ijk, 0123, 0341\}$, con $i,j=1,3$ $U_3 = \{1,2,3,4\}$, y $x^{4'} = \{0ijk1, 0ijk3, 01234, 03412\}$ avec $i,j,k=1,3$. La dinámica será entonces descrita por:

$$f_3(x_0(3), \dots, x_3(3), u(3)) = \begin{pmatrix} x_0(4) \\ x_1(4) \\ x_2(4) \\ x_3(4) \\ x_4(4) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_0(3) \\ x_1(3) \\ x_2(3) \\ x_3(3) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u(3)$$

Etapa 5: $x^{4'} = \{0ijk1l, 0ijk3, 01234, 03412\}$ con $i,j,k=1,3$, $U_4 = \{1,3,5\}$, y $x^{5'} = \{0ijk1n1, 0ijkn3, 012345, 034125\}$ con $i,j,k,n=1,3$. La dinámica será entonces descrita por:

$$f_4(x_0(4), \dots, x_4(4), u(4)) = \begin{pmatrix} x_0(5) \\ x_1(5) \\ x_2(5) \\ x_3(5) \\ x_4(5) \\ x_5(5) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_0(4) \\ x_1(4) \\ x_2(4) \\ x_3(4) \\ x_4(4) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u(4)$$

Etapa 6: $x^5 = \{0ijkn1, 0ijknq3, 012345, 034125\}$ con $i, j, k, n, q = 1, 3$, $U_4 = \{1, 3, 6\}$ y $x^6 = \{0ijknq1, 0ijknq3, 0123456, 0341256\}$ con $i, j, k, n, q = 1, 3$. La dinámica será entonces descrita:

$$f_5(x_0(5), \dots, x_5(5), u(5)) = \begin{pmatrix} x_0(6) \\ x_1(6) \\ x_2(6) \\ x_3(6) \\ x_4(6) \\ x_5(6) \\ x_6(6) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_0(5) \\ x_1(5) \\ x_2(5) \\ x_3(5) \\ x_4(5) \\ x_5(5) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u(5)$$

Etapa 7: $x^6 = \{0ijknq1, 0ijknq3, 0123456, 0341256\}$ con $i, j, k, n, q = 1, 3$, $U_4 = \{1, 3, 7\}$ y $x^7 = \{0ijknqr1, 0ijknqr3, 01234567, 03412567\}$, con $i, j, k, n, q, r = 1, 3$. La dinámica será entonces descrita por:

$$f_6(x_0(6), \dots, x_6(6), u(6)) = \begin{pmatrix} x_0(7) \\ x_1(7) \\ x_2(7) \\ x_3(7) \\ x_4(7) \\ x_5(7) \\ x_6(7) \\ x_7(7) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} x_0(6) \\ x_1(6) \\ x_2(6) \\ x_3(6) \\ x_4(6) \\ x_5(6) \\ x_6(6) \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u(6)$$

En general tenemos que la dinámica nos queda:

$$\begin{pmatrix} x_0(k+1) \\ x_1(k+1) \\ \vdots \\ x_k(k+1) \\ x_{k+1}(k+1) \end{pmatrix} = f_k((x_0(k), \dots, x_k(k)), u(k)) = \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & \ddots & \ddots & & \\ \vdots & \ddots & \ddots & 0 & 0 \\ \vdots & & \ddots & 1 & 0 \\ 0 & 0 & \dots & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix} \begin{bmatrix} x_0(k) \\ x_1(k) \\ \vdots \\ x_{k-1}(k) \\ x_k(k) \end{bmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u(k).$$

Definición de una estrategia óptima

Las palabras obtenidas en el lenguaje L, resultan de la realización de operaciones por medio de las identidades. Ellas representan un planteamiento para solucionar el problema de simplificación buscado. Estas palabras pueden dar una solución al problema; y en caso de fracaso se puede decir que todavía no se llega a una solución.

Definimos una función sobre el lenguaje L de las palabras admisibles de la siguiente manera:

$$s(u) = \begin{cases} 0 & \text{Si } u \text{ representa una solución.} \\ 1 & \text{Si } u \text{ no representa una solución.} \end{cases}$$

Por medio de la función $s(\cdot)$ se puede formalizar la estrategia óptima (posible ayuda inteligente) para encontrar una solución.

Sea $|u| = |u_1 \cdots u_k| = k_0$ la longitud de una palabra u . Definimos la función $p:L \rightarrow Z_+$ como:

$$p(u) = \{v: s(uv)=0 \text{ y } |v| \leq \min \}.$$

Entonces, dada la palabra u que representa la secuencia anterior de operaciones y selecciones, la escogencia de v (una selección u operación) es :

- una buena alternativa si $p(uv) < p(u)$
- no aporta nada a la solución si $p(uv) = p(u)$, y
- una mala alternativa $p(uv) > p(u)$.

Sea la condición inicial, $x_0 \in X_{k_0}$ correspondiente a la secuencia $u = u_1 u_2 \cdots u_{k_0}$. Una palabra v , $v = v_0 v_1 \cdots v_{K-1}$, de longitud K , tal que $uv \in L$ es admisible si $s(uv)=0$. Debemos ahora minimizar la longitud de la palabra.

Ahora el problema pasa a ser un problema de optimización: un problema de control óptimo de SED descrito como sigue.

Dado el sistema dinámico:

$$x(k+1) = f_{k_0+k}(x(k), v(k)), \quad k = 0, \dots, K-1.$$

donde $x(K)$ representa la secuencia de comandos uv de palabras y la función de costo bidimensional:

$$\Phi(x(K), K) = (s(uv), |v|).$$

El mínimo es definido a través de una relación lexicográfica dada por: $(a, b) \leq (c, d)$ sí $a < c$ ó $a = c$ y $b \leq d$.

El mínimo será entonces dado por: $\min_v \Phi(x(K), K) = (0, p(u))$ y el problema de cálculo de p en u , $p(u)$, se ha reducido a un problema de control óptimo:

$$x(k+1) = f_{k_0+k}(x(k), v(k)), \quad x(0) = x_0, \quad u \in X_{k_0},$$

$$\min_{uv \in L} \Phi(x(K), K).$$

1.1.3. Ejemplo: Supervisor para la primera etapa blanqueo de sebo

Un proceso genérico de blanqueo de sebo, consta principalmente de tres etapas. La primera etapa es el secado del material, la segunda etapa es el proceso de blanqueo y la tercera etapa es el filtrado. Acá mostraremos el funcionamiento de la primera etapa y describimos un modelo de comportamiento por medio de autómatas de estado finito y el modelo matemático formal de estos.

En la Figura 1.3 se muestra un mímico de un sistema de control y adquisición de datos (SCADA) para el proceso de secado y en la Figura 1.4 se muestra el panel de control del mismo.

La primera etapa, denominada secado, se divide en dos procesos. El primer proceso es el llenado del tanque pulmón. El segundo proceso es el proceso de secado del material a partir de dos ciclos de recirculación. El primer ciclo de recirculación se establece entre el tanque pulmón y el intercambiador, donde el sebo debe tener una temperatura de 90 grados centígrados, si es menor no es posible extraer la humedad y si es superior la grasa se puede quemar. La segunda etapa de recirculación se establece entre el tanque pulmón, el intercambiador y el extractor de humedad donde el sebo deberá tener un 2% de humedad. Al terminar este proceso, un sebo seco para el proceso de blanqueo es disponible.

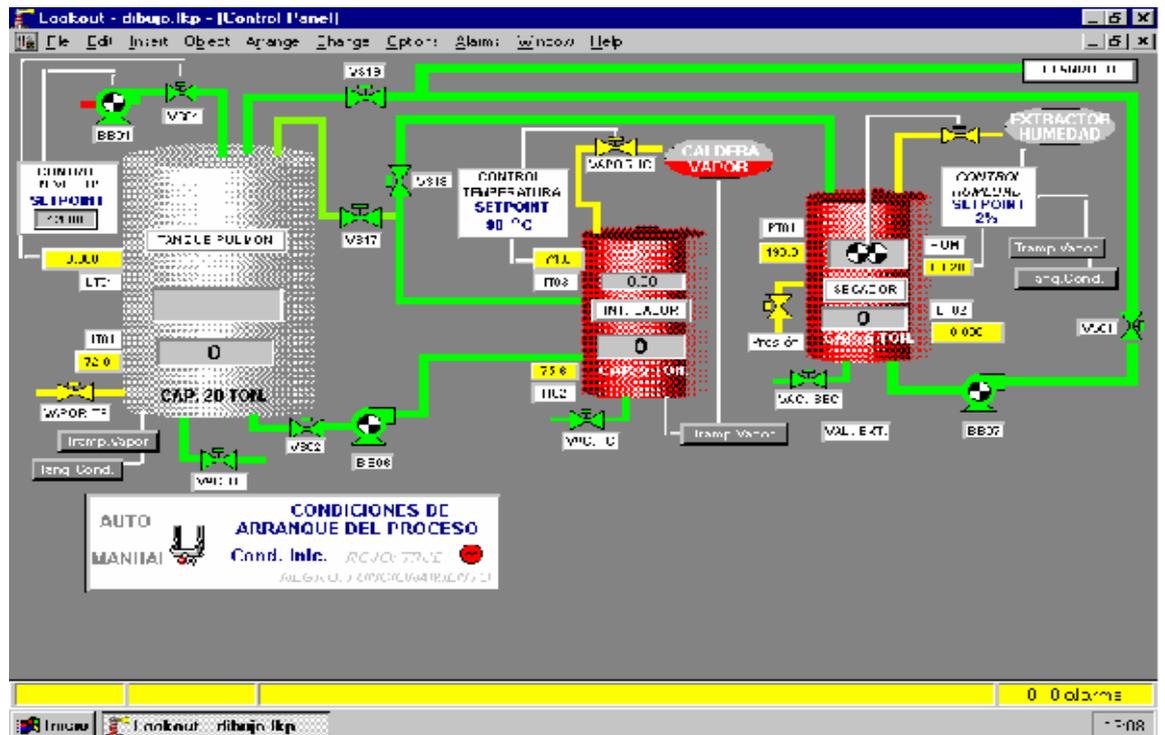


Figura 1.3: SCADA Primera Etapa Blanqueo de Sebo

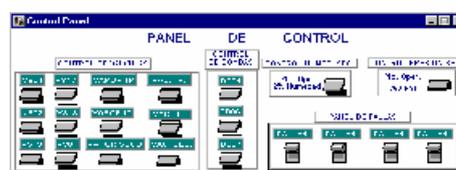


Figura 1.4: Panel de control

En la segunda etapa, denominada blanqueo, el sebo de la primera etapa es distribuido entre el tanque de blanqueo y el tanque de preparación de tierra. En el tanque de preparación de tierra, una parte de sebo seco y otra de arcilla se mezclan para preparar la tierra de blanqueo. El resto del sebo permanece en el tanque de blanqueo. Cuando la mezcla en el tanque de preparación de tierra adquiere una textura homogénea, esto es, toma un color marrón uniforme, esta mezcla es enviada al tanque de blanqueo. La mezcla de tierra de blanqueo y el sebo seco en el tanque de blanqueo se agitan hasta que las condiciones de parada son satisfechas.

La preparación de tierra para blanquear es diferente para cada tipo de jabón a fabricar. Así, diferentes cantidades de material seco con diferentes cantidades y tipos de arcilla son usados.

En la tercera etapa, denominada filtrado, una parte de la mezcla blanqueada se utiliza para preparar un aditivo que acelera el proceso de filtraje en los filtros. Una vez que este aditivo es colocado en los filtros se vierte la materia del proceso de blanqueo. El resultado es una grasa blanqueada seca que es

almacenada para la producción de jabón, quedando en el filtro la tierra de blanqueo con las impurezas de la grasa.

Proceso Llenado Tanque Pulmón:

1. Abrir válvula de descarga del tanque de almacenamiento de grasa cruda
2. Encender bomba
3. Verificar el nivel del tanque pulmón hasta el punto de operación prefijado o hasta un 75% de su capacidad máxima.
4. Apagar bomba y cerrar válvula.

Autómatas obtenidos para el proceso de llenado del tanque pulmón

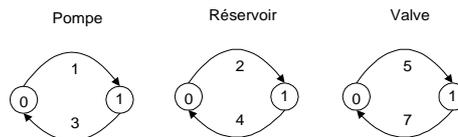


Figura 1.5: Autómata de la bomba, de la válvula y del tanque pulmón

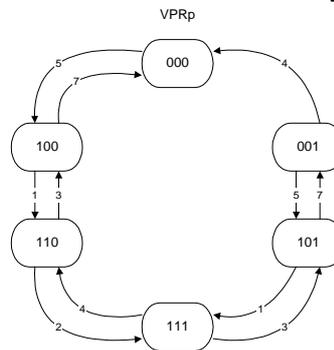


Figura 1.6: Autómata del llenado del tanque pulmón

Modelo matemático, proceso llenado tanque pulmón

Consideremos como estado inicial $x_0 = \theta = (000)'$, si $u = 5$ entonces, $f_0(x_0, u) = (100)'$.

El espacio de estado es $X_1 = (100)'$ y las letras admisibles asociadas a este espacio de estado son $u \in \{1,7\} = \Omega_1 \{(100)'\}$, la función f_1 obtenida es

$$f_1(x, u) = ((u - 1)(000)' - (u - 7)(110)') / 6.$$

El próximo espacio de estado es $X_2 = \{(000)', (110)'\}$, y las letras admisibles asociadas a este espacio de estado son: $u \in \{5\} = \Omega_2 \{(000)'\}$ ó $u \in \{3,4\} = \Omega_2 \{(110)'\}$, la función f_2 obtenida es:

$$f_2(x, u) = (1 - x_1) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + x_1 \left[(u - 2) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - (u - 3) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right].$$

A continuación se muestran los restantes espacios de estado, letras admisibles y funciones obtenidas.

$n=3$, $X_3 = \{(100)', (111)'\}$, $u \in \{1,7\} = \Omega_3 \{(100)'\}$ ó $u \in \{3,4\} = \Omega_3 \{(111)'\}$,

$$f_3(x, u) = (1 - x_2) \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + x_2 \left[(u - 3) \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} - (u - 4) \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \right].$$

$n=4$, $X_4 = \{(000)', (110)', (101)'\}$, $u \in \{5\} = \Omega_4 \{(000)'\}$ ó $u \in \{2,3\} = \Omega_4 \{(110)'\}$, ó $u \in \{1,7\} = \Omega_4 \{(101)'\}$,

$$f_4(x, u) = (1 - x_1)(1 - x_2) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + x_1 x_2 \left[(u - 2) \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} - (u - 3) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right] + x_1(1 - x_2) \left[\frac{(u - 1)}{6} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} - \frac{(u - 7)}{6} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right].$$

$n=5$, $X_5 = \{(001)', (100)', (111)'\}$, $u \in \{1,7\} = \Omega_5 \{(100)'\}$ ó $u \in \{3,4\} = \Omega_5 \{(111)'\}$, ó $u \in \{4,5\} = \Omega_5 \{(001)'\}$,

$$f_5(x, u) = (1 - x_1)(1 - x_2) \left[(u - 4) \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} - (u - 5) \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \right] + x_1 x_2 \left[(u - 3) \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} - (u - 4) \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} \right] + x_1(1 - x_2) \left[\frac{(u - 1)}{6} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} - \frac{(u - 7)}{6} \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \right].$$

$n=6$, $X_6 = \{(101)', (000)', (110)'\}$.

Se puede observar que los estados se van repitiendo de la forma $X_9 = X_7 = X_5$, $X_8 = X_6 = X_4$, etc.

De manera general se obtiene el modelado $X_k = X_{k+2}$, $k > 5$, por lo cual el periodo es 2. Si consideramos a k un número arbitrario, entonces

$$R(k) = \text{Res}(k, \text{mod}2) = \begin{cases} 1, & \text{si } k \text{ es impar} \\ 0, & \text{si } k \text{ es par} \end{cases},$$

es el residuo a la división por 2 $f_k(x, u) = f_{4+R(k)}(x, u)$.

Proceso de recirculación

1. Abrir la válvula descarga del tanque pulmón.
2. Encender bomba.
3. Activar el control de vapor para el intercambiador (abrir válvula de vapor).
4. Abrir válvula del intercambiador al tanque pulmón.
5. Se mantiene los pasos 1-4 hasta que la temperatura en el intercambiador este alrededor de los 90 °C. Al alcanzar el valor de 90 °C. Se procede con el paso 6.
6. Abrir válvula del intercambiador al extractor de humedad
7. Cerrar válvula del intercambiador al tanque pulmón.
8. Abrir válvula entre el extractor y el tanque pulmón.
9. Activar el control de humedad (calentar paredes del extractor, activar la aspersion del sebo sobre las paredes del extractor).

La recirculación se mantiene hasta obtener sebo con 2% humedad . Una vez alcanzado los valores de humedad dados, se procede con a segunda etapa, el blanqueo. Así que se transfiere el sebo seco a la segunda etapa, cerrando la válvula entre el extractor y el tanque pulmón y abriendo la válvula que va al tanque de blanqueo y preparación de tierra.

Autómatas obtenidos para el proceso de recirculación

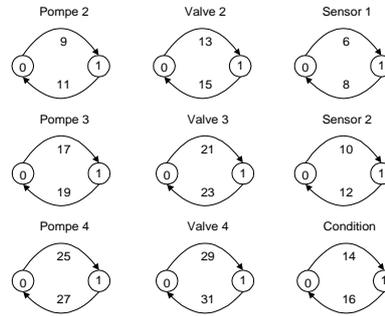


Figura 1.7: Autómatas de las bombas 2, 3, 4, de las válvulas 2, 3, 4, los sensores 1, 2, y de la condición

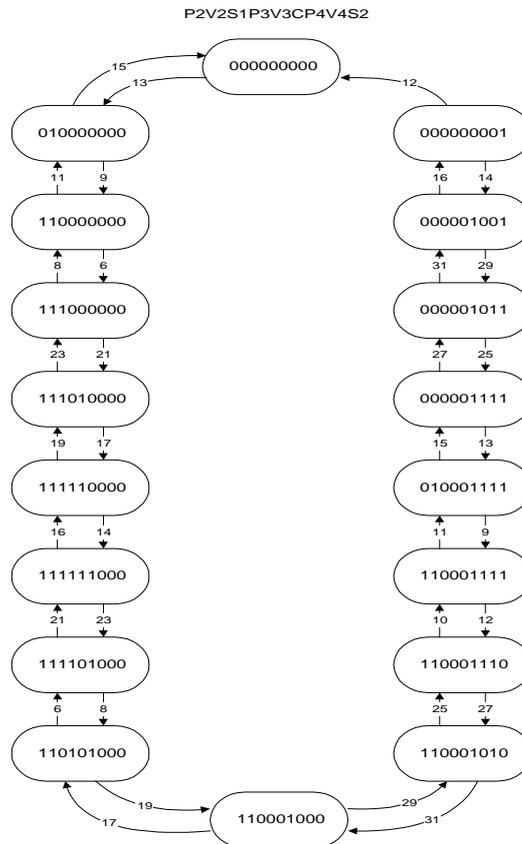


Figura 1.8: Autómata del proceso de secado

Modelo matemático, proceso de recirculación

Consideremos que el estado inicial es $x_0 = (00000000) = \theta$, entonces si $u \in \{13\} = \Omega\{(00000000)'\}$, la función f_0 es $f_0(x_0, u) = (010000000)'$.

El espacio de estado es $X_1 = (010000000)'$ y las letras admisibles asociadas a este espacio de estado son $u \in \{9, 15\} = \Omega\{(010000000)'\}$, la función f_1 obtenida es

$$f_1(x, u) = ((u - 9)(000000000)' - (u - 15)(110000000)') / 6.$$

El próximo espacio de estado es $X_2 = \{(000000000)', (110000000)'\}$, y las letras admisibles asociadas a este espacio de estado son: $u \in \{13\} = \Omega\{(000000000)'\}$ ó $u \in \{1, 6\} = \Omega\{(110000000)'\}$, la función f_2 obtenida es:

$$f_2(x, u) = (1 - x_1) \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{x_1}{5} \left[(u - 6) \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - (u - 11) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right].$$

A continuación se muestran los restantes espacios de estado, letras admisibles y funciones obtenidas.

$$n=3, \quad X_3 = \{(01000000)', (11100000)'\}, \quad u \in \{9, 15\} = \Omega\{(01000000)'\} \quad \delta \\ u \in \{8, 21\} = \Omega\{(11100000)'\},$$

$$f_3(x, u) = \frac{(1 - x_1)}{6} \left[(u - 9) \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - (u - 15) \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right] + \frac{x_1}{13} \left[(u - 8) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - (u - 21) \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right].$$

$$n=4, \quad X_4 = \{(00000000)', (11101000)', (11000000)'\}, \quad u \in \{17, 23\} = \Omega\{(11101000)'\} \quad \delta \\ u \in \{6, 11\} = \Omega\{(11000000)'\},$$

$$f_4(x, u) = (1 - x_2)(1 - x_3) \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \frac{x_2 x_3}{6} \left[(u - 17) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} - (u - 23) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right] +$$

$$x_2(1-x_3) \left[\begin{array}{c} \left(\begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right) \\ \frac{(u-6)}{5} \\ 0 \\ -\frac{(u-11)}{5} \\ \left(\begin{array}{c} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right) \end{array} \right].$$

$n=5$, $X_5 = \{(010000000)'\}, (111000000)'\}, (111110000)'\}$, $u \in \{9,15\} = \Omega\{(010000000)'\}$ ó $u \in \{8,21\} = \Omega\{(111000000)'\}$, ó $u \in \{14,19\} = \Omega\{(111110000)'\}$,

$$f_5(x, u) = \frac{(1-x_3)(1-x_4)}{6} \left[\begin{array}{c} \left(\begin{array}{c} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right) \\ (u-9) \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right] - (u-15) \left[\begin{array}{c} \left(\begin{array}{c} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right] + x_3x_4 \left[\begin{array}{c} \left(\begin{array}{c} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right) \\ \frac{(u-14)}{5} \\ 1 \\ -\frac{(u-19)}{5} \\ \left(\begin{array}{c} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right) \end{array} \right] +$$

$$\frac{x_3(1-x_4)}{13} \left[\begin{array}{c} \left(\begin{array}{c} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right) \\ (u-8) \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right] - (u-21) \left[\begin{array}{c} \left(\begin{array}{c} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} \right].$$

$n=6$, $X_6 = \{(000000000)'\}, (111010000)'\}, (110000000)'\}, (111111000)'\}$,

$$f_6(x, u) = (1 - x_2)(1 - x_3)(1 - x_4) \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \frac{x_2 x_3 (1 - x_4)}{6} (u - 17) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (u - 23) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} +$$

$$\frac{x_2 (1 - x_3)(1 - x_4)}{5} (u - 6) \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (u - 11) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \frac{x_2 x_3 x_4}{7} (u - 16) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (u - 23) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

$n=7, X_7 = \{(01000000)', (11100000)', (11111000)', (111101000)\},$

$$f_7(x, u) = \frac{(1 - x_3)(1 - x_4)(1 - x_5)}{6} (u - 9) \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (u - 15) \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} +$$

$$\frac{x_3 (1 - x_4)(1 - x_5)}{13} (u - 8) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (u - 21) \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + \frac{x_3 x_4 x_5}{5} (u - 14) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (u - 19) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} +$$

$$+ x_3 x_4 (1 - x_5) \begin{bmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} - \frac{(u-11)}{13} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{bmatrix}.$$

$n=8, X_8 = \{(00000000)', (11101000)', (11000000)', (11111100)', (11010100)\},$

$$f_8(x, u) = (1 - x_2)(1 - x_3)(1 - x_4) \begin{bmatrix} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} + x_2 x_3 (1 - x_4) \left[\begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - \frac{(u-17)}{6} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - \frac{(u-23)}{6} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right] +$$

$$\frac{x_2(1-x_3)(1-x_4)}{5} \left[\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (u-11) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right] + \frac{x_2 x_3 x_4}{7} \left[\begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (u-23) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right] +.$$

$$+ \frac{x_2(1-x_3)x_4}{13} \left[\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} - (u-19) \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right].$$

$$f_8(x, u) = f_6(x, u) + \frac{x_2(1-x_3)x_4}{13} \left[(u-6) \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} - (u-19) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right].$$

n=9, $X_9 = X_7 \cup \{(110001000)\}$,

$$f_9(x, u) = f_7(x, u) + \frac{(1-x_3)(1-x_5)x_6}{12} \left[(u-17) \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} - (u-29) \begin{pmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right]$$

n=10, $X_{10} = X_8 \cup \{(110001010)\}$,

$$f_{10}(x, u) = f_8(x, u) - x_2(1-x_3)(1-x_4)x_6 \left[\frac{(u-6)}{5} \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} - \frac{(u-11)}{5} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \right] +$$

$$+ \frac{x_2(1-x_3)x_4}{13} (u-6) \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - (u-19) \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

$$n=11, X_{11} = X_7 \cup \{(110001110)\},$$

$$f_{11}(x, u) = f_7(x, u) - \frac{(1-x_3)(1-x_5)x_6x_7}{12} (u-17) \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} - (u-29) \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$+ \frac{(1-x_3)(1-x_5)x_6x_7}{17} (u-10) \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} - (u-27) \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

$$n=12, X_{12} = X_{10} \cup \{(110001111)\},$$

$$f_{12}(x, u) = f_{10}(x, u) - x_2(1-x_3)(1-x_4)x_6x_7 \left[\frac{(u-25)}{5} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} - \frac{(u-31)}{5} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} \right] +$$

$$n=13, X_{13} = X_{11} \cup \{(010001111)\},$$

$$f_{13}(x, u) = f_{11}(x, u) - \frac{(1-x_3)(1-x_5)x_6x_7x_8}{17} (u-10) \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} - (u-27) \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \\ + \frac{(1-x_3)(1-x_5)x_6x_7x_8}{6} (u-9) \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} - (u-15) \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

El procedimiento continua hasta obtener el último estado X_{17} ,

$$X_{17} = X_{15} \cup \{(000000001)\}.$$

1.2. Conclusiones

- La principal meta en este capítulo, que es la obtención de modelos formales de SED a partir de las representaciones clásicas se logra y se muestran ejemplo de ello.
- Para el desarrollo ulterior de esta tesis los modelos formales son un prerrequisito, ya que las extensiones propuestas usan estas descripciones. Además, como ya mencionamos en la introducción, en la mayoría de los casos preferimos usar modelos matemáticos formales en donde, en principio, se tiene técnicas y herramientas para hacer análisis. Adicionalmente, estas técnicas analíticas proveen un buen punto de vista en el efecto de cambios de parámetros sobre las propiedades del sistema.
- Como podemos ver que en el caso de lenguajes formales la técnica usada para generar modelos matemáticos de sistemas dinámicos a eventos discretos posee una restricción de gran importancia, más aún si se trata de sistemas complejos, ya que en cada instante de tiempo crece el espacio de estado. Esta restricción puede complicar un poco el procedimiento que a posteriori se presenta para realizar optimización sobre sistemas dinámicos a eventos discretos.

Capítulo 1

MODELOS FORMALES DE SDED	1
1.1. MODELOS FORMALES EN LEGUAJES CERRADOS: CASOS GENERALES DE SISTEMAS LINGÜÍSTICOS	1
1.1.1. <i>Lenguajes Cerrados y Realización Mínima</i>	2
1.1.2. <i>Ejemplo: Simplificación Matemática por cálculo formal</i>	8
1.1.3. <i>Ejemplo: Supervisor para la primera etapa blanqueo de sebo</i>	12
1.2. CONCLUSIONES	23

Capítulo 1

FIGURA 1.1: CLASES DE EQUIVALENCIA	3
FIGURA 1.2: RELACIÓN ENTRE LA CAJA NEGRA Y EL SISTEMA LINGÜÍSTICO	4
FIGURA 1.3: SCADA PRIMERA ETAPA BLANQUEO DE SEBO	13
FIGURA 1.4: PANEL DE CONTROL	13
FIGURA 1.5: AUTÓMATA DE LA BOMBA, DE LA VÁLVULA Y DEL TANQUE PULMÓN	14
FIGURA 1.6: AUTÓMATA DEL LLENADO DEL TANQUE PULMÓN	14
FIGURA 1.7: AUTÓMATAS DE LAS BOMBAS 2, 3, 4, DE LAS VÁLVULAS 2, 3, 4, LOS SENSORES 1, 2, Y DE LA CONDICIÓN	16
FIGURA 1.8: AUTÓMATA DEL PROCESO DE SECADO	16

Capítulo 1

TABLA 1.1: SELECCIÓN VS. OPERACIÓN.....	9
---	---

COMPUTACIÓN SIMBÓLICA en OPTIMIZACIÓN DISCRETA

En este capítulo presentamos una solución al problema de optimización simbólica de funciones definidas sobre un conjunto discreto finito. El resultado principal en este capítulo es un algoritmo, denominado SCDO, que permite obtener los mínimos de una función paramétrica en donde las variables están definidas sobre el hiper-cubo $\{-1,1\}^k$ y los parámetros están definidos en un dominio dado. Así el dominio discreto finito de la función a minimizar es representado en los vértices de un hyper-cubo $\{-1,1\}^k$, y el algoritmo SCDO puede ser aplicado. Varias propiedades interesantes de funciones Booleanas son descubiertas aquí y son relacionadas a una particular representación, sobre los vértices de un hipercubo $\{-1,1\}^k$. El algoritmo simbólico propuesto tiene esencialmente la misma complejidad computacional que los métodos de búsqueda clásica, con la ventaja de su posible aplicación a la optimización paramétrica.

2.1. Introducción a la Optimización de Funciones sobre Dominios Discretos

La optimización de funciones definidas sobre un conjunto discreto es generalmente realizada puramente por algoritmos enumerativos numéricos, [RD84], con severas limitaciones en el número de variables. Como un enfoque alternativo, aquí proponemos el uso de la computación simbólica para obtener iterativamente la solución óptima formal de un problema de minimización sobre un conjunto finito de variables Booleanas.

Una fórmula explícita de la optimización de una variable es primero obtenida, usando, como la base del dominio Booleano de una variable, al conjunto $\{-1,1\}$ en vez del conjunto $\{0,1\}$. Entonces el cálculo simbólico provee fórmulas explícitas para el caso Booleano multivariable. Los dominios finitos de variables enteras pueden ser transformados en un dominio Booleano finito con la ayuda de una representación diádica de enteros [RD84].

Adicionalmente, el polinomio de interpolación de Lagrange provee una representación general de funciones definidas sobre conjuntos discretos [FS99]. Una combinación de estas herramientas genera una representación canónica de funciones definidas sobre conjuntos discretos. La aplicación de esta representación canónica de funciones es caracterizada por una secuencia de matrices A_k , las cuales exhiben algunas propiedades que se resaltan en este capítulo. El uso de estas propiedades reduce considerablemente la complejidad computacional del método.

Los ejemplos al final del capítulo ilustran el análisis simbólico desarrollado y como se aborda el problema cuando funciones con parámetros exógenos se consideran.

2.2. Optimización Simbólica de Funciones Booleanas

Aquí, el mínimo absoluto de funciones Booleanas será expresado por una fórmula explícita. El uso de esta fórmula es propuesta como una alternativa a los métodos de búsqueda clásicos. Una particular característica del enfoque propuesto es el uso de valores Booleanos en el conjunto $\{-1,1\}$ en vez del conjunto $\{0,1\}$ como dominio binario unidimensional básico.

Así, el dominio de la función puede escogerse como un subconjunto de los vértices del hipercubo dado por:

$$D \subset \{-1,1\}^{k+1}, f:D \rightarrow \mathfrak{R}.$$

Por conveniencia, una función definida sobre D puede ser extendida al conjunto entero del producto $\{-1,1\}^{k+1}$, de la siguiente manera. Fijando un $x_0 \in D$ arbitrario, una apropiada extensión de f sobre $\{-1,1\}^{k+1}$ es dada por:

$$\bar{f}_{x_0} : \{-1,1\}^{k+1} \rightarrow \mathfrak{R}, \bar{f}_{x_0}(x) = \begin{cases} f(x) & \text{if } x \in D, \\ f(x_0)+1 & \text{if } x \in \{-1,1\}^{k+1} \setminus D. \end{cases} \quad (2.1)$$

Es claro que el mínimo absoluto y los puntos donde es alcanzado, son los mismos para la función f y \bar{f}_{x_0} , aquí podemos suponer, sin perder generalidad, que el dominio de definición de la función Booleana es el conjunto producto: $\{-1,1\}^{k+1}$.

2.2.1. Caso monovariante

Consideremos, $f : \{-1,1\} \rightarrow \mathfrak{R}$. Entonces el lema siguiente puede ser enunciado.

Lemma 2.1:

El valor mínimo de la función Booleana f es obtenido por medio la función signo (Sign) de la siguiente manera

$$u^* = -\text{Sign}[f(1) - f(-1)] = \text{Sign}[f(-1) - f(1)], \quad (2.2)$$

donde la función Sign es usada en el sentido siguiente:

$$\text{Sign}(t) = \begin{cases} 1 & \text{if } t > 0, \\ -1 & \text{if } t < 0, \end{cases} \quad (2.3)$$

y, para $t=0$, $\text{Sign}(0) = \{-1,1\}$ es un conjunto valuado. En el último caso, la función f obtiene su óptimo (mínimo o máximo) en ambos puntos del dominio.

La prueba de este lema es obtenida de forma inmediata, considerando los dos posible valores $u^* = -1$ ó $u^* = 1$. ?

2.2.2 Caso multivariable

La aplicación del Lema 2.1 para el caso vectorial toma la forma de un algoritmo el cual contiene dos partes. La primera parte es un cálculo simbólico hacia atrás, que consiste de $k+1$ pasos como se describe abajo. En cada paso, el Lema 2.1 es usado para proveer una expresión simbólica del valor óptimo $u_i^+(u_0, \dots, u_{i-1})$. La segunda parte, comienza con el cálculo explícito del valor óptimo de u_0^* . Entonces se procede hacia delante para calcular la secuencia óptima de los u_i^* , para $i=1, \dots, k$.

Las dos partes del algoritmo serán mostradas con detalle a continuación.

Parte 1

Considere la función $f_0 : \{-1,1\}^{k+1} \rightarrow \mathfrak{R}$, $f_0(u_0, \dots, u_{k-1}, u_k) = f(u_0, \dots, u_{k-1}, u_k)$.

- Paso 1: Se define la función parcial $u_k \rightarrow f_0(u_0, \dots, u_k)$ para u_0, \dots, u_{k-1} fijos. Por el Lema 2.1, el mínimo del problema de optimización paramétrica es obtenido como

$$u_k^+(u_0, u_1, \dots, u_{k-1}) = \text{Sign}(f_0(u_0, u_1, \dots, u_{k-1}, -1) - f_0(u_0, u_1, \dots, u_{k-1}, 1)).$$

Se define la función $f_1 : \{-1,1\}^k \rightarrow \mathfrak{R}$, como:

$$f_1(u_0, u_1, \dots, u_{k-1}) = f_0(u_0, u_1, \dots, u_{k-1}, u_k^+(u_0, u_1, \dots, u_{k-1})).$$

- Paso i. Suponga que las funciones

$$f_2 : \{-1,1\}^{k-1} \rightarrow \mathfrak{R}, \dots, f_{i-1} : \times_0^{k-i+1} \{-1,1\} \rightarrow \mathfrak{R},$$

están generadas como en el paso 1 y sus óptimos simbólicos son alcanzados en

$$u_{k-1}^+(u_0, \dots, u_{k-2}), \dots, u_{k-i+2}^+(u_0, \dots, u_{k-i+1}).$$

Entonces, a partir del Lema 2.1, la función parcial

$$u_{k-i+1} \rightarrow f_{i-1}(u_0, \dots, u_{k-i}, u_{k-i+1}),$$

alcanza su mínimo en

$$u_{k-i+1}^+(u_0, u_1, \dots, u_{k-i}) = \text{Sign}(f_{i-1}(u_0, u_1, \dots, u_{k-i}, -1) - f_{i-1}(u_0, u_1, \dots, u_{k-i}, 1)),$$

y la función a ser optimizada puede ser reemplazada por la función $f_i : \{-1,1\}^{k-i} \rightarrow \mathfrak{R}$, definida por:

$$f_i(u_0, u_1, \dots, u_{k-i}) = f_{i-1}(u_0, u_1, \dots, u_{k-i}, u_{k-i}^+(u_0, u_1, \dots, u_{k-i})).$$

- Paso k. La última función $f_k : \{-1,1\} \rightarrow \mathfrak{R}$, alcanza su valor mínimo en $u_0^* = u_0^+ = \text{Sign}(f_k(-1) - f_k(1))$, el cual es un subconjunto $\{-1,1\}$. Por el Lema 2.1, nosotros conocemos que u_0^* pertenece al conjunto $\{-1,1\}$ solamente en el caso de que f_k sea constante sobre $\{-1,1\}$.

Parte 2:

El cálculo de los valores óptimos, u_i^* , donde $f_{k-i}(\cdot)$ alcanza el mínimo, es obtenida por substitución hacia delante de la siguiente manera:

- Paso 1:

$$\text{El valor inicial es : } u_0^* = \text{Sign}(f_k(-1) - f_k(1)).$$

- Paso i, para $i=1, \dots, k$. Suponga que $u_0^*, u_1^*, \dots, u_{i-1}^*$ se han calculado. Entonces u_i^* puede ser calculado por la recursión:

$$\begin{aligned} u_i^* &= u_i^+ (u_0^*, \dots, u_{i-1}^*) \\ &= \text{Sign} \left(f_{k-i} (u_0^*, u_1^*, \dots, u_{i-1}^*, -1) - f_{k-i} (u_0^*, u_1^*, \dots, u_{i-1}^*, 1) \right). \end{aligned}$$

Dos posibles casos se pueden distinguir:

Caso 1: Si $f_{k-i} (u_0^*, u_1^*, \dots, u_{i-1}^*, -1) \neq f_{k-i} (u_0^*, u_1^*, \dots, u_{i-1}^*, 1)$, entonces $u_i^* \in \{-1, 1\}$

Caso 2: Si $f_{k-i} (u_0^*, u_1^*, \dots, u_{i-1}^*, -1) = f_{k-i} (u_0^*, u_1^*, \dots, u_{i-1}^*, 1)$, entonces, los dos posibles valores, -1 o 1, son admisibles para u_{k-i}^* . Este caso genera dos secuencias óptimas las cuales son exploradas en el siguiente paso.

Así el número de secuencias óptimas se incrementan en uno cada vez que el caso 2 ocurre.

2.3. Optimización sobre Dominio Discreto Finito

Consideremos ahora una función real general sobre un dominio discreto finito $g(x) : D_g \subset \mathbb{Z}^N \rightarrow \mathfrak{R}$. El caso multidimensional ($N > 1$) puede ser transformado en el caso monovariable por enumeración.

El caso de dominio finito es trasladado en el previamente definido dominio Booleano usando el siguiente lema.

Lema 2.2:

Considere el conjunto $\{0, 1, \dots, n\}$, y $\{-1, 1\}^{k+1}$. Si k es el entero más pequeño tal que $n < 2^{k+1}$, entonces la aplicación

$$\psi : \{0, 1, \dots, n\} \rightarrow \{-1, 1\}^{k+1},$$

es unívocamente definida por su inversa:

$$u = \chi(u_0, \dots, u_k) = \frac{1}{2}(1 + u_0) + (1 + u_1) + \dots + 2^{k-1}(1 + u_k). \quad (2.4)$$

La prueba del Lema 2.2 sigue de la unicidad de la representación diádica de algún $n \in \mathbb{N}$, para $2^k - 1 < n \leq 2^{k+1} - 1$. ?

La aplicación del Lema 2.2 en el caso de $n = \text{Car}(D_g)$ permite transformar la función $g(x) : D_g \rightarrow \mathfrak{R}$ en la función $\varphi(u_0, \dots, u_k) : \{-1, 1\}^{k+1} \rightarrow \mathfrak{R}$ sobre la condición $2^k - 1 < n \leq 2^{k+1} - 1$. La función $\varphi(u_0, \dots, u_k)$ es entonces unívocamente definida por:

$$\varphi = g \circ \chi.$$

Usando la extensión descrita en la ecuación (2.1), puede suponerse, sin restricción, que el dominio de φ es $\{-1, 1\}^{k+1}$.

2.3.1. Representación polinómica canónica

Cualquier función definida sobre un dominio finito puede reemplazarse, equivalentemente, por un polinomio que toma los mismos valores que la función original en cada punto discreto.

En particular, cualquier función Booleana $\varphi(u_0, \dots, u_k)$ como la definida antes puede ser reformulada como una función polinómica $f(u_0, \dots, u_k)$ sobre un dominio $D \subset \{-1, 1\}^{k+1}$, como:

$$f(u_0, \dots, u_k) = \sum_{\underline{\alpha}} c_{\underline{\alpha}} u_0^{\alpha_0} u_k^{\alpha_k}, \quad (2.5)$$

con un número finito de secuencias $\underline{\alpha} = (\alpha_0, \dots, \alpha_k)$ que pertenezcan a un subconjunto finito de \mathbb{N}^{k+1} .

En el caso de una función φ definida sobre un dominio $D \subset \{-1, 1\}^{k+1}$, tenemos adicionalmente la siguiente propiedad:

$$\begin{aligned} u_i^p &= 1 \text{ si } p \text{ es par} \\ u_i^p &= u_i \text{ si } p \text{ es impar.} \end{aligned} \quad (2.6)$$

Por lo tanto en todos los monomios implicados en la representación (2.5), esto es: $u_0^{\alpha_0} \dots u_k^{\alpha_k}$, es suficiente considerar $(\alpha_0, \dots, \alpha_k) \in \{0, 1\}^{k+1}$.

Tal representación también es provista por la interpolación de Lagrange sobre todos los puntos del dominio.

$$\begin{aligned} f(u_0, \dots, u_k) &= \sum_{\underline{\varepsilon} \in \{-1, 1\}^{k+1}} \varphi(\varepsilon_0, \varepsilon_1, \dots, \varepsilon_k) \frac{u_0 + \varepsilon_0}{2\varepsilon_0} \frac{u_1 + \varepsilon_1}{2\varepsilon_1} \dots \frac{u_k + \varepsilon_k}{2\varepsilon_k} = \Psi, \\ f(\underline{\varepsilon}) &= \Psi(\underline{\varepsilon}). \end{aligned} \quad (2.7)$$

En esta representación es obvio que todos los monomios en (2.7) tiene la forma $u_0^{\alpha_0} u_1^{\alpha_1} \dots u_k^{\alpha_k}$, donde $(\alpha_0, \alpha_1, \dots, \alpha_k) \in \{0, 1\}^{k+1}$. Aquí, todas las funciones sobre $\{-1, 1\}^{k+1}$ pueden ser representadas como:

$$f(u_0, \dots, u_k) = \sum_{(\alpha_0, \dots, \alpha_k) \in \{0, 1\}^{k+1}} f_{\alpha_0, \dots, \alpha_k} u_0^{\alpha_0} \dots u_k^{\alpha_k}. \quad (2.8)$$

Si hacemos la correspondencia entre los valores de la función con los vértices del hipercubo $\{-1, 1\}^{k+1}$ en un vector de dimensión 2^{k+1} , denotado por \tilde{F}_k , y ordenamos los vértices del hipercubo en orden lexicográfico tenemos:

$$\tilde{F}_k = (f(-1, \dots, -1, -1), f(-1, \dots, -1, 1), \dots, f(1, \dots, 1, 1))^T.$$

Ahora, ordenamos los coeficientes de la expansión polinómica (2.8) en un vector de dimensión 2^{k+1} , en orden antilexicográfico, denotado por F_k , como:

$$F_k = (f_{111\dots 1}, \dots, f_{100\dots 0}, f_{011\dots 1}, \dots, f_{000\dots 0})^T.$$

Entonces el siguiente lema puede ser enunciado.

Lemma 2.3:

$\tilde{F}_i = A_i F_i$, para $i=0, \dots, k$ con

$$\begin{aligned} A_0 &= \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \in \mathfrak{R}^{2 \times 2}, \text{ y para } i=1, \dots, k, \\ A_i &= \begin{bmatrix} -A_{i-1} & A_{i-1} \\ A_{i-1} & A_{i-1} \end{bmatrix} \in \mathfrak{R}^{2^{i+1} \times 2^{i+1}}. \end{aligned} \tag{2.9}$$

Prueba

El Lema 2.3 puede ser probado por inducción matemática.

Para $i=0$, una función de una variable $u_0 \in \{-1, 1\}$ puede ser escrita:

$$f(u_0) = f_1 u_0 + f_0, \quad u_0 \in \{-1, 1\}.$$

La relación entre los coeficientes f_0, f_1 y los valores $f(-1), f(1)$ es dada por la siguiente ecuación lineal:

$$\begin{bmatrix} f(-1) \\ f(1) \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_0 \end{bmatrix}, \text{ esto es, } \tilde{F}_0 = A_0 F_0.$$

Ahora, suponemos que para alguna función de $i+1$ variables binarias,

$$\tilde{F}_i = A_i F_i,$$

entonces, para alguna función $f(u_0, \dots, u_i, u_{i+1})$, con $(u_0, \dots, u_i, u_{i+1}) \in \{-1, 1\}^{i+2}$,

$$f(u_0, \dots, u_i, -1) = -\sum_{\epsilon_0 \dots \epsilon_i} f_{\epsilon_0 \dots \epsilon_i} u_0^{\epsilon_0} \dots u_i^{\epsilon_i} + \sum_{\epsilon_0 \dots \epsilon_i} f_{\epsilon_0 \dots \epsilon_i} u_0^{\epsilon_0} \dots u_i^{\epsilon_i},$$

conduce a

$$\tilde{F}_{i+1,-1} = -A_i F_{i+1,1} + A_i F_{i+1,0} \tag{2.10}$$

y

$$f(u_0, \dots, u_i, 1) = \sum_{\epsilon_0 \dots \epsilon_i} f_{\epsilon_0 \dots \epsilon_i} u_0^{\epsilon_0} \dots u_i^{\epsilon_i} + \sum_{\epsilon_0 \dots \epsilon_i} f_{\epsilon_0 \dots \epsilon_i} u_0^{\epsilon_0} \dots u_i^{\epsilon_i},$$

conduce a

$$\tilde{F}_{i+1,1} = A_i F_{i+1,1} + A_i F_{i+1,0}, \tag{2.11}$$

donde $\tilde{F}_{i+1} = \begin{bmatrix} \tilde{F}_{i+1,-1} \\ \tilde{F}_{i+1,1} \end{bmatrix}$ y $F_{i+1} = \begin{bmatrix} F_{i+1,1} \\ F_{i+1,0} \end{bmatrix}$.

Así

$$\tilde{F}_{i+1} = \begin{bmatrix} -A_i & A_i \\ A_i & A_i \end{bmatrix} \begin{bmatrix} F_{i+1,1} \\ F_{i+1,0} \end{bmatrix} = A_{i+1} F_{i+1},$$

con

$$A_{i+1} = \begin{bmatrix} -A_i & A_i \\ A_i & A_i \end{bmatrix} \in \mathfrak{R}^{2^{i+2} \times 2^{i+2}} .?$$

En la práctica, los parámetros desconocidos son los componentes del vector F_i , para $i = 1, \dots, k$. Una de las principales ventajas de la representación canónica propuesta, es la simplicidad para invertir la matriz A_i , tal como lo indica el siguiente lema.

Previo al anuncio del lema daremos la fórmula de Sur para inversión de matrices particionadas. Sea $F \in M_{m+n \times m+n}$ una matriz particionada en bloques de la siguiente forma:

$$F = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

con $A \in M_{m \times m}$ y $D \in M_{n \times n}$ ambas invertibles entonces

$$F^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -A^{-1}B(D - CA^{-1}B)^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}.$$

Lemma 2.4:

La matriz A_i es invertible, y $A_i^{-1} = \frac{1}{2^{i+1}} A_i$.

Prueba

El Lema 2.4 también es probado por inducción matemática.

Para $i=0$, A_0 es dado por

$$A_0 = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix},$$

entonces, obviamente

$$A_0^{-1} = \frac{-1}{2} \begin{bmatrix} 1 & -1 \\ -1 & -1 \end{bmatrix} = \frac{1}{2} A_0.$$

Supongamos ahora que A_i satisface lo dicho en el Lema 2.4. Usando la conocida fórmula de inversión de Shur para bloques de matrices particionadas, obtenemos:

$$A_{i+1}^{-1} = \begin{bmatrix} -A_i & A_i \\ A_i & A_i \end{bmatrix}^{-1} = \begin{bmatrix} -\frac{1}{2}[A_i]^{-1} & \frac{1}{2}[A_i]^{-1} \\ \frac{1}{2}[A_i]^{-1} & \frac{1}{2}[A_i]^{-1} \end{bmatrix}.$$

Así la hipótesis inductiva muestra:

$$A_{i+1}^{-1} = \frac{1}{2^{i+2}} \begin{bmatrix} -A_i & A_i \\ A_i & A_i \end{bmatrix} = \frac{1}{2^{i+2}} A_{i+1} \text{ .?}$$

Por lo tanto, a partir de los valores del vector \tilde{F}_k , el vector de los coeficientes de la representación F_k , pueden ser calculados por una simple multiplicación matriz-vector como sigue:

$$F_k = \frac{1}{2^{k+1}} A_k \tilde{F}_k \quad (2.12)$$

2.3.2. Optimización simbólica

Usando la representación canónica, una función general $g(x)$ definida sobre un dominio discreto finito puede ser optimizada por el algoritmo presentado en la sección 2.2.

El vector de coeficientes de la representación polinomial $f_0(u_0, \dots, u_{k-1}, u_k) = f(u_0, \dots, u_{k-1}, u_k)$ puede ser calculado directamente por la fórmula (2.12), notando que:

$$\begin{aligned} \tilde{F}_k &= [f(-1, \dots, -1, -1), f(-1, \dots, -1, 1), \dots, f(1, \dots, 1, 1)]^T \\ &= [g(0), g(1), \dots, g(2^{k+1})]^T \end{aligned}$$

Entonces, u_k^+ puede ser calculado por:

$$u_k^+(u_0, u_1, \dots, u_{k-1}) = -\text{Sign}(f_0(u_0, u_1, \dots, u_{k-1}, -1) - f_0(u_0, u_1, \dots, u_{k-1}, 1))$$

La parte iterativa simbólica procede como sigue, para $i = k-1, \dots, 0$.

- Sustituir u_{i+1}^+ en $f_{k-i+1}(u_0, \dots, u_{i+1})$ y obtenemos $f_{k-i}(u_0, \dots, u_i)$.
- Evaluar $f_{k-i}(u_0, \dots, u_i)$ sobre el dominio D_i (vector \tilde{F}_i).
- Generar la matriz A_i .
- Resolver el sistema de ecuaciones lineales

$$\begin{bmatrix} -A_i & A_i \\ A_i & A_i \end{bmatrix} F_{i+1} = \tilde{F}_{i+1}.$$

- Generar la representación polinómica $f_{k-i}(u_0, \dots, u_i)$ con coeficientes en el vector \tilde{F}_i

Calcular $u_i^+(u_0, u_1, \dots, u_{i-1}) = -\text{Sign}(f_{k-i}(u_0, u_1, \dots, u_{i-1}, 1) - f_{k-i}(u_0, u_1, \dots, u_{i-1}, 0))$.

La segunda parte del algoritmo es explícito y procede hacia adelante para calcular la secuencia óptima $(u_0^*, u_1^*, \dots, u_k^*)$ como en la parte 2 de la sección 2.2.

2.3.3. Mejoras en el algoritmo simbólico

El algoritmo que acaba de describirse en 2.3.2. es algo difícil de poner en ejecución debido a la necesidad de evaluar la función $f_{k-i}(u_0, \dots, u_i)$ en todos los vértices del hipercubo $\{-1, 1\}^{i+1}$ para obtener su representación polinómica canónica. Una simplificación importante sería alcanzada si es posible calcular directamente el vector del coeficiente, F_i , para la iteración i , a partir del vector de coeficiente, F_{i+1} , en la iteración $i+1$, sin tener que calcular el vector de valores \tilde{F}_i . Tal mejora se puede alcanzar usando el lema siguiente.

Definición: $x \in M_{n \times m}$ entonces $|x| = [|x|_1 \ |x|_2 \ \dots \ |x|_n]^T$.

Lemma 2.5:

La secuencia de los vectores de coeficientes $\{F_i\}$ de las funciones polinómicas canónicas f_{k-i} para $i=k-1, \dots, 0$ satisface la siguiente expresión de recurrencia:

$$F_i = \frac{-1}{2^{i+1}} A_i |A_i F_{i+1,1}| + F_{i+1,0}. \quad (2.13)$$

Prueba:

La i -ésima fila de la matriz A_k puede ser expresada por $A_{k,i}$.

La función f_{k-i} de $i+1$ variables:

$$f_{k-i}(u_0, \dots, u_i) = \sum_{\underline{\varepsilon}} f_{k-i;\varepsilon_0, \dots, \varepsilon_i} u_0^{\varepsilon_0} \dots u_i^{\varepsilon_i},$$

con $\underline{\varepsilon} = (\varepsilon_0, \dots, \varepsilon_i) \in \{0, 1\}^{k+1}$, fue obtenido de $f_{k-i-1}(u_0, \dots, u_{i+1})$ substituyendo en u_{i+1} por

$$u_{i+1}^+ (u_0, \dots, u_i) = -\text{Sign} \left[\sum_{\underline{\varepsilon}} f_{k-i-1;\varepsilon_0, \dots, \varepsilon_i, 1} u_0^{\varepsilon_0} \dots u_i^{\varepsilon_i} \right].$$

Así, por otra parte

$$f_{k-i} = - \left(\sum_{\underline{\varepsilon}} f_{k-i-1;\varepsilon_0, \dots, \varepsilon_i, 1} u_0^{\varepsilon_0} \dots u_i^{\varepsilon_i} \right) \text{Sign} \left[\sum_{\underline{\varepsilon}} f_{k-i-1;\varepsilon_0, \dots, \varepsilon_i, 1} u_0^{\varepsilon_0} \dots u_i^{\varepsilon_i} \right] + \sum_{\underline{\varepsilon}} f_{k-i-1;\varepsilon_0, \dots, \varepsilon_i, 0} u_0^{\varepsilon_0} \dots u_i^{\varepsilon_i} \quad (2.14)$$

El correspondiente vector de valores \tilde{F}_i puede ser calculado de (2.14) por

$$\begin{aligned} \tilde{F}_i &= \begin{pmatrix} f_{k-i}(-1, \dots, -1) \\ \vdots \\ f_{k-i}(1, \dots, 1) \end{pmatrix} \\ &= \begin{pmatrix} A_{i,1} F_{i+1,1} \text{Sign} [A_{i,1} F_{i+1,1}] + A_{i,1} F_{i+1,0} \\ \vdots \\ A_{i,2^{i+1}} F_{i+1,1} \text{Sign} [A_{i,2^{i+1}} F_{i+1,1}] + A_{i,2^{i+1}} F_{i+1,0} \end{pmatrix} = A_i F_i \end{aligned}$$

$$= - \begin{pmatrix} |A_{i,1}F_{i+1,1}|_i + A_{i,1}F_{i+1,0} \\ \vdots \\ |A_{i,2^{i+1}}F_{i+1,1}|_{2^{i+1}} + A_{i,2^{i+1}}F_{i+1,0} \end{pmatrix},$$

que es,

$$\tilde{F}_i = -|A_i F_{i+1,1}| + A_i F_{i+1,0} = A_i F_i.$$

Por lo tanto, usando el Lema 2.4, la recursión del Lema 2.5 es obtenida multiplicando ambos términos de esta igualdad por la inversa de A_i .?

Usando el Lema 2.5, la secuencia de vectores de coeficientes $\{ F_i \}$ puede ser calculado desde el vector de coeficientes $\{ F_k \}$ de las funciones polinómicas canónicas $f_0(u_0, \dots, u_{k-1}, u_k) = f(u_0, \dots, u_{k-1}, u_k)$.

Entonces, las expresiones formales (óptimas) de las variables $u_i^+(u_0, \dots, u_{i-1})$ pueden ser calculadas ahora como se describe en la sección 2.3.2 con la ecuación (2.13).

La complejidad de la manipulación simbólica de la aplicación polinómica sobre el espacio producto es bajo. La substitución en los monomios $u_0 u_1 \cdots u_k$, es de complejidad $O(k)$. La substitución en $f(u_0, \dots, u_{k-1}, u_k)$ en la representación canónica es un producto escalar del vector de monomios por el vector de coeficientes, el cual significa $O(n)$ "elementales" operaciones, donde $n = 2^{k+1}$. En este contexto "elementales" significa, por ejemplo operaciones simbólicas con monomios y polinomios de la forma $\sum C_{\underline{\varepsilon}} u^{\underline{\varepsilon}}$, donde $\underline{\varepsilon}$ es un multi-índice vectorial de $0,1$. Esa complejidad está también $O(n)$, por lo tanto la complejidad completa de la manipulación simbólica es $O(n)$. La asignación del Signo es realizada por un producto matriz-vector entre escalares que también tienen complejidad $O(n)$. En relación con las complejidades mencionadas, el algoritmo para minimizar la función sobre $\{-1,1\}^{k+1}$, tiene complejidad $O(n)$.

2.4. Ejemplos Ilustrativos

Los ejemplos siguientes pretenden ilustrar el uso tanto del algoritmo básico presentado en 2.3.2 y la mejora de la técnica hecha en 2.3.3.

2.4.1. Ejemplo 1

Considere el siguiente polinomio:

$$f(u_0, u_1, u_2) = a_{111}u_0u_1u_2 + a_{110}u_0u_1 + a_{101}u_0u_2 + a_{100}u_0 + a_{011}u_1u_2 + a_{010}u_1 + a_{001}u_2 + a_{000}$$

2.4.1.1 Optimización simbólica

Usando el algoritmo en 2.2.2. y el esquema en 2.3.2., presentamos las dos partes que componen el algoritmo.

Parte 1

Paso 1

$$f_0(u_0, u_1, u_2) = a_{111}u_0u_1u_2 + a_{110}u_0u_1 + a_{101}u_0u_2 + a_{100}u_0 + a_{011}u_1u_2 + a_{010}u_1 + a_{001}u_2 + a_{000},$$

$$u_2^+(u_0, u_1) = \text{Sign}[f_0(u_0, u_1, -1) - f_0(u_0, u_1, 1)] = -\text{Sign}[a_{111}u_0u_1 + a_{101}u_0 + a_{011}u_1 + a_{001}],$$

Paso 2

$$\begin{aligned} f_1(u_0, u_1) &= (a_{111}u_0u_1 + a_{101}u_0 + a_{011}u_1 + a_{001})u_2^+(u_0, u_1) + a_{110}u_0u_1 + a_{100}u_0 + a_{010}u_1 + a_{000} \\ &= -(a_{111}u_0u_1 + a_{101}u_0 + a_{011}u_1 + a_{001})\text{Sign}[a_{111}u_0u_1 + a_{101}u_0 + a_{011}u_1 + a_{001}] + \\ &\quad + a_{110}u_0u_1 + a_{100}u_0 + a_{010}u_1 + a_{000} \\ &= b_{11}u_0u_1 + b_{10}u_0 + b_{01}u_1 + b_{00} = p_1(u_0, u_1), \end{aligned}$$

$$u_1^+(u_0) = \text{Sign}[p_1(u_0, -1) - p_1(u_0, 1)] = -\text{Sign}[b_{11}u_0 + b_{01}].$$

Los coeficientes b_{11} y b_{01} son obtenidos por la resolución del sistema de ecuaciones lineales siguiente

$$\begin{bmatrix} f_1(-1, -1) \\ f_1(-1, 1) \\ f_1(1, -1) \\ f_1(1, 1) \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{10} \\ b_{01} \\ b_{00} \end{bmatrix},$$

donde,

$$b_{11} = \frac{1}{4}(f_1(-1, -1) + f_1(1, 1) - f_1(-1, 1) - f_1(1, -1)),$$

$$b_{01} = \frac{1}{4}(f_1(-1, 1) + f_1(1, 1) - f_1(-1, -1) - f_1(1, -1)),$$

$$\begin{aligned} u_1^+(u_0) &= -\text{Sign}\left[\left[(a_{111} - a_{101} - a_{011} + a_{001})u_2^+(-1, -1) + (a_{111} + a_{101} + a_{011} + a_{001})u_2^+(1, 1) + \right. \right. \\ &\quad \left. \left. - (a_{111} - a_{101} + a_{011} + a_{001})u_2^+(-1, 1) - (a_{111} + a_{101} - a_{011} + a_{001})u_2^+(1, -1) + 4a_{0110}\right]u_0 + \right. \\ &\quad \left. + (a_{111} - a_{101} + a_{011} + a_{001})u_2^+(-1, 1) + (a_{111} + a_{101} + a_{011} + a_{001})u_2^+(1, 1) - \right. \\ &\quad \left. - (a_{111} - a_{101} - a_{011} + a_{001})u_2^+(-1, -1) - (a_{111} + a_{101} - a_{011} + a_{001})u_2^+(1, -1) + 4a_{010}\right]. \end{aligned}$$

Paso 3

$$\begin{aligned} f_2(u_0) &= \left[(a_{111}u_0 + a_{011})u_1^+(u_0) + (a_{101}u_0 + a_{001})\right]u_2^+(u_0, u_1^+(u_0)) + (a_{110}u_0 + a_{010})u_1^+(u_0) + a_{100} + a_{000} \\ &= b_1u_0 + b_0 = p_2(u_0). \end{aligned}$$

$$u_1^+(\cdot) = \text{Sign}[p_2(-1) - p_2(1)] = -\text{Sign}[b_1].$$

El coeficiente b_1 es obtenido por la resolución del siguiente sistema de ecuaciones lineales

$$\begin{bmatrix} f_2(-1) \\ f_2(1) \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix},$$

donde $b_1 = \frac{1}{2}(-f_2(-1) + f_2(1))$, y

$$\begin{aligned} u_0^+(\cdot) = & -\text{Sign} \left[\left[(a_{111} + a_{011})u_1^+(-1) - a_{101} + a_{001} \right] u_2^+(-1, u_1^+(-1)) + (-a_{110} + a_{010})u_1^+(-1) - \right. \\ & \left. - \left[(a_{111} + a_{011})u_1^+(1) + a_{101} + a_{001} \right] u_2^+(1, u_1^+(1)) - (a_{110} + a_{010})u_1^+(1) - 2a_{100} \right]. \end{aligned}$$

Parte 2

Paso 1

$$\begin{aligned} u_0^*(\cdot) = & -\text{Sign} \left[\left[(a_{111} + a_{011})u_1^+(-1) - a_{101} + a_{001} \right] u_2^+(-1, u_1^+(-1)) + (-a_{110} + a_{010})u_1^+(-1) - \right. \\ & \left. - \left[(a_{111} + a_{011})u_1^+(1) + a_{101} + a_{001} \right] u_2^+(1, u_1^+(1)) - (a_{110} + a_{010})u_1^+(1) - 2a_{100} \right]. \end{aligned}$$

Paso 2

$$\begin{aligned} u_1^* = & -\text{Sign} \left[\left[(a_{111} - a_{101} - a_{011} + a_{001})u_2^+(-1, -1) + (a_{111} + a_{101} + a_{011} + a_{001})u_2^+(1, 1) + \right. \right. \\ & \left. \left. - (a_{111} - a_{101} + a_{011} + a_{001})u_2^+(-1, 1) - (a_{111} + a_{101} - a_{011} + a_{001})u_2^+(1, -1) + 4a_{0110} \right] u_0^* + \right. \\ & \left. + (a_{111} - a_{101} + a_{011} + a_{001})u_2^+(-1, 1) + (a_{111}a_{101} + a_{011} + a_{001})u_2^+(1, 1) - \right. \\ & \left. - (a_{111} - a_{101} - a_{011} + a_{001})u_2^+(-1, -1) - (a_{111} + a_{101} - a_{011} + a_{001})u_2^+(1, -1) + 4a_{010} \right]. \end{aligned}$$

Paso 3

$$u_2^* = -\text{Sign} \left[a_{111}u_0^*u_1^* + a_{101}u_0^* + a_{011}u_1^* + a_{001} \right].$$

2.4.1.2. Mejora del algoritmo simbólico

Para aplicar el Lema 2.5, al ejemplo, debemos definir:

$$\begin{aligned} F_2 &= [a_{111}, a_{110}, a_{101}, a_{100}, a_{011}, a_{010}, a_{001}, a_{000}]^T, \\ F_{2,1} &= [a_{111}, a_{101}, a_{011}, a_{001}]^T, \\ F_{2,0} &= [a_{110}, a_{100}, a_{010}, a_{000}]^T. \end{aligned}$$

Entonces, podemos calcular $F_1 = [a_{11}, a_{10}, a_{01}, a_{00}]$ y $F_0 = [a_1, a_0]$. La expresión óptima de cada variable u_i^* , $i=0, 1, 2$, esta dada por:

$$\begin{aligned} u_0^* &= -\text{Sign}[a_1], \\ u_1^* &= -\text{Sign}[a_{11}u_0^* + a_{01}], \\ u_2^* &= -\text{Sign}[a_{111}u_0^*u_1^* + a_{101}u_0^* + a_{011}u_1^* + a_{001}], \end{aligned}$$

donde

$$\begin{aligned} F_{1,1} &= [a_{11}, a_{01}]^T, \\ F_{1,0} &= [a_{10}, a_{00}]^T, \end{aligned}$$

$$F_{0,1} = [a_1]^T,$$

$$F_{0,0} = [a_0]^T,$$

con

$$a_{11} = -\frac{1}{4} \left(|a_{111} - a_{101} - a_{011} + a_{001}| - | -a_{111} - a_{101} + a_{011} + a_{001} | - \right. \\ \left. - | -a_{111} + a_{101} - a_{011} + a_{001} | + | a_{111} + a_{101} + a_{011} + a_{001} | - 4a_{110} \right),$$

$$a_{01} = -\frac{1}{4} \left(- | a_{111} - a_{101} - a_{011} + a_{001} | + | -a_{111} - a_{101} + a_{011} + a_{001} | - \right. \\ \left. - | -a_{111} + a_{101} - a_{011} + a_{001} | + | a_{111} + a_{101} + a_{011} + a_{001} | - 4a_{110} \right),$$

$$a_{10} = -\frac{1}{4} \left(- | a_{111} - a_{101} - a_{011} + a_{001} | - | -a_{111} - a_{101} + a_{011} + a_{001} | + \right. \\ \left. + | -a_{111} + a_{101} - a_{011} + a_{001} | + | a_{111} + a_{101} + a_{011} + a_{001} | - 4a_{110} \right),$$

$$a_{00} = -\frac{1}{4} \left(| a_{111} - a_{101} - a_{011} + a_{001} | + | -a_{111} - a_{101} + a_{011} + a_{001} | + \right. \\ \left. + | -a_{111} + a_{101} - a_{011} + a_{001} | + | a_{111} + a_{101} + a_{011} + a_{001} | - 4a_{110} \right),$$

$$a_1 = -\frac{1}{2} \left(- | -a_{11} + a_{01} | + | a_{11} + a_{01} | - 4a_{10} \right).$$

2.4.2. Ejemplo 2

Considere el autómata representado en la figura 1.

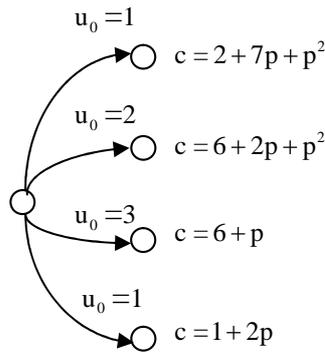


Figure 2.1: Un autómata con 4 escogencias

Una representación equivalente a este autómata es mostrado en la tabla 1.

u_0	u_0	u_1	\tilde{F}_1
1	-1	-1	$2 + 7p - p^2$
2	-1	1	$6 + 2p - p^2$
3	1	-1	$6 + p$
4	1	1	$1 + 2p$

Tabla 2.1: Tabla para el autómata de 4 escogencias

Aplicando el algoritmo descrito anteriormente, podemos obtener una expresión para determinar el camino óptimo en función del parámetro p :

Paso 1

El polinomio f de variables $[u_0, u_1]$, está definido por:

$$f_0 = a_{00} + a_{01}u_1 + a_{10}u_0 + a_{11}u_0u_1.$$

Así, el control óptimo está dado por: $u^+(1) = \text{Sign}(-2a_{01} - 2a_{11}u_0)$.

Como los valores son:

$$\tilde{F}_1 = [2 + 7p - p^2, 6 + 2p - p^2, 6 + p, 1 + 2p]^T,$$

entonces,

$$F_1 = [a_{11}, a_{10}, a_{01}, a_{00}]^T.$$

La matriz $A[1]$ es:

$$A[1] = \begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

Como la ecuación (2.12) obtenemos los coeficientes F_1 y sus expresiones son:

$$\begin{bmatrix} a_{11} \\ a_{10} \\ a_{01} \\ a_{00} \end{bmatrix} = \begin{bmatrix} \frac{-9 + 6p}{4} \\ \frac{-1 - 6p + 2p^2}{4} \\ \frac{-1 - 4p}{4} \\ \frac{15 + 12p - 2p^2}{4} \end{bmatrix}.$$

El polinomio nos queda :

$$f_0 = \frac{15 + 12p - 2p^2}{4} + \frac{-1 - 6p + 2p^2}{4}u_0 + \frac{-1 - 4p}{4}u_1 + \frac{-9 + 6p}{4}u_0u_1. \quad (2.15)$$

Entonces $u^*(1) = \text{Sign}\left[\frac{1}{2} + 2p + \frac{9}{2}u_0 - 3pu_0\right]$.

Paso 2

El polinomio f con variable u_0 , es definido por: $a_0 + a_1u_0$. El control óptimo está dado por:

$$u^+(0) = -\text{Sign}[a_1],$$

y la matriz $A[0]$ es :

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

Por el Lema 2.5, los coeficientes F_0 satisfacen la siguiente expresión:

$$F_0 = \frac{-1}{2} A_0 |A_0 F_{1,1}| + F_{1,0}. \quad (2.16)$$

De donde

$$F_0 = \begin{bmatrix} a1 \\ a0 \end{bmatrix} = -\frac{1}{2} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} \frac{3}{2}(-3+2P) \\ -\frac{1}{2}-2P \end{pmatrix} + \begin{pmatrix} \frac{1}{2}(-1-6p+2p^2) \\ \frac{15}{2} + \frac{3}{2}p - p^2 \end{pmatrix}.$$

Así, los coeficientes son:

$$\begin{aligned} a1 &= \frac{1}{2} \left| 2 - \frac{5}{2}p \right| - \frac{1}{2} \left| 2 + \frac{5}{2}p \right| + \frac{1}{4}(-1 - 6p + p^2), \\ a0 &= \frac{15}{4} + 3p - \frac{p^2}{2}, \end{aligned} \quad (2.17)$$

y

$$u^*(0) = \text{Sign} \left[\frac{1}{2} \left| 2 - \frac{5}{2}p \right| - \frac{1}{2} \left| 2 + \frac{5}{2}p \right| + \frac{1}{4}(-1 - 6p + p^2) \right]. \quad (2.18)$$

Para dos valores particulares de p , tenemos:

- Si $p=1$, la secuencia es $[u_0, u_1] = [1, 1]$, y en coordenadas originales: $u_0 = 4$.
- Si $p=5$, la secuencia es $[u_0, u_1] = [-1, 1]$, y en coordenadas originales: $u_0 = 2$.

2.5. Conclusiones

- El algoritmo de optimización simbólico propuesto es aplicable para funciones generales definidas sobre un dominio discreto finito.
- Aquí, el mínimo absoluto de funciones Booleanas es dado por una fórmula explícita. El uso de esta fórmula es propuesta como una alternativa a los métodos de búsqueda clásica.
- El método está compuesto por dos partes. La primera es un cálculo recursivo hacia atrás, en donde se obtienen las expresiones implícitas de las variables Booleanas. La segunda parte, instancia progresivamente las expresiones explícitas de los valores óptimos de las variables.
- La complejidad del algoritmo es comparable con los métodos numéricos de tipo enumerativo. Pero el algoritmo propuesto tiene la ventaja de poderse aplicar directamente a los problemas en los cuales tanto las decisiones como la función de costo asociadas dependen de parámetros exógenos.
- La complejidad de la manipulación simbólica de la aplicación polinómica sobre el espacio producto es bajo. La substitución en los monomios $u_0 u_1 \cdots u_k$, es de complejidad $O(k)$. La substitución en $f(u_0, \dots, u_{k-1}, u_k)$ en la representación canónica es un producto escalar del vector de monomios por el vector de coeficientes, el cual significa $O(n)$ "elementales" operaciones, donde $n = 2^{k+1}$. En este contexto "elementales" significa, por ejemplo operaciones simbólicas con monomios y polinomios de la forma $\sum C_{\underline{x}} u^{\underline{x}}$, donde \underline{x} es un

multi-índice vectorial de $0,1$. Esa complejidad es también $O(n)$, por lo tanto la complejidad completa de la manipulación simbólica es $O(n)$. La asignación del Signo es realizada por un producto matriz-vector entre escalares que también tienen complejidad $O(n)$. En relación con las complejidades mencionadas, el algoritmo para minimizar la función sobre $\{-1,1\}^{k+1}$, tiene complejidad $O(n)$.

- El algoritmo presentado en 2.2.2, con las mejoras en 2.3.3 conforman el algoritmo que hemos denominado SCDO.

Capítulo 2

COMPUTACIÓN SIMBÓLICA EN OPTIMIZACIÓN DISCRETA.....	1
2.1. INTRODUCCIÓN A LA OPTIMIZACIÓN DE FUNCIONES SOBRE DOMINIOS DISCRETOS	1
2.2. OPTIMIZACIÓN SIMBÓLICA DE FUNCIONES BOOLEANAS	1
2.2.1. <i>Caso monovariable</i>	2
2.2.2 <i>Caso multivariable</i>	2
2.3. OPTIMIZACIÓN SOBRE DOMINIO DISCRETO FINITO	4
2.3.1. <i>Representación polinómica canónica</i>	5
2.3.2. <i>Optimización simbólica</i>	8
2.3.3. <i>Mejoras en el algoritmo simbólico</i>	9
2.4. EJEMPLOS ILUSTRATIVOS	10
2.4.1. <i>Ejemplo 1</i>	10
2.4.2. <i>Ejemplo 2</i>	13
2.5. CONCLUSIONES	15

Capítulo 2

FIGURE 2.1: UN AUTÓMATA CON 4 ESCOGENCIAS	13
---	----

Capítulo 2

TABLA 2.1: TABLA PARA EL AUTÓMATA DE 4 ESCOGENCIAS	13
--	----

Referencias

J.J. Cardillo-Albarán: (1994) Optimización de sistemas a eventos discretos usando el principio de mínimo de Pontryagin, Tesis Msd, Facultad de Ingeniería, ULA, Mérida.

J.J. Cardillo-Albarrán and F. Szigeti: (1995) A symbolic algorithm for the optimization of integer processes over finite state space, the International Congress on Industrial and Applied Mathematics, ICIAM 95, Hamburg, Germany, 3-7, July.

F. Szigeti, G. De Sarrazin, J. Cardillo (1999) Computer aided problem solving via Optimization, Risc-Linz Report series number 99-13 May 99, Hagenberg, Austria.

Horn and Johnson, Matrix Analysis, Cambridge University Press, 1999, ISBN 0-521-30586-1.

L. E. Sigler Algebra, Springer Verlag, 1976 ISBN 0-387-90195-7.

Robert Dautray et Jacques-Louis Lions, Analyse mathématique et calcul numérique pour les sciences et les techniques, Series Scientifique, Masson, ISBN 2-225-80406-9, 1984.

ENFOQUE SIMBÓLICO DEL PRINCIPIO DE MÍNIMO

El principio de mínimo tiene en física una larga e interesante historia. La búsqueda de tales principios es predicada sobre la noción de que “la naturaleza siempre minimiza ciertas cantidades importantes cuando un proceso físico toma lugar”. El primero de tales principios de mínimo fue desarrollado en la óptica. Hero de Alejandría en el segundo siglo a.C. encontró que la ley que gobernaba la reflexión de la luz puede ser obtenida cuando un rayo de luz, viajando de un punto a otro por una reflexión desde un espejo plano, siempre toma el camino más corto posible. Una simple construcción geométrica verifica que el principio de mínimo hace la igualdad de los ángulos de incidencia y de reflexión para un rayo de luz reflejado sobre un espejo plano. El principio de Hero sobre el camino mas corto no puede, sin embargo, producir una correcta ley para la refracción. En 1657 Fermat, reformuló el principio al postular que un rayo de luz siempre viaja desde un punto a otro en un medio por un camino que requiere el menor tiempo. El principio de Fermat de menor tiempo llevó inmediatamente, no solo, a la ley correcta de reflexión, sino también a la ley de Snell de refracción.

El principio de mínimo continuó siendo tratado y en la última parte del siglo XVII, empiezan a aparecer los resultados Newton, Leibniz y los Bernoulli dando origen a lo que se conoce como cálculo de variaciones. Así, problemas tales como el Braquistocrono, la forma de una cadena de horca, entre otros problemas fueron resueltos.

La primera aplicación de un principio de mínimo en mecánica fue hecha en 1747 por Maupertuis, quien afirmó que el movimiento dinámico toma lugar con mínima acción. El principio de menor acción de Maupertuis fue basado en fundamentos teológicos (acción de minimizar a través de la sabiduría de Dios) y su concepto de acción fue bastante vago. Recordamos que la acción es una cantidad con la dimensión de longitud por momento o energía por tiempo. Solamente después, Lagrange en 1760 dio una firme fundamentación matemática al principio de menor acción. Sin embargo si esta aplicación tuvo la ventaja de hacer la transición de la mecánica clásica hacia la óptica y a la mecánica cuántica, el principio de menor acción sigue siendo menos general que el principio de Hamilton y puede derivarse de él.

En 1828 Gauss desarrollo un método de tratado mecánico por su principio de menor restricción, una modificación fue hecha más tarde por Hertz y lo personalizo en su principio de menor curvatura. Estos principios están estrechamente relacionados al principio de Hamilton y no adicionan nada al contenido de la formulación más general del Hamiltoniano, solamente resaltan el interés por un principio del mínimo en la física.

Así, el principio de Hamilton puede ser descrito como sigue:

“De todos los caminos posibles a lo largo del cual un sistema dinámico puede moverse desde un punto a otro sin un intervalo de tiempo especificado (mientras siga siendo compatible con las restricciones), el actual camino seguido es el que minimiza la integral de la diferencia entre la energía cinética y la energía potencial”.

En el cálculo de variaciones el principio de Hamilton esta dado por:

$$\delta \int_{t_1}^{t_2} (T - U) dt = 0 \quad T = T(\dot{x}), U = U(x)$$

Esta formulación establece que T-U debe ser un extremo, no necesariamente mínimo, pero en casi todas las aplicaciones importantes en dinámicas la condición de mínimo ocurre.

Si $L \cong T-U = L(x, \dot{x})$, la ecuación de Euler – Lagrange se escribe como $\frac{\partial L}{\partial x} - \frac{d}{dt} \frac{\partial L}{\partial \dot{x}} = 0$.

3.1. Ecuaciones Canónicas de Movimiento en el Formalismo Hamiltoniano

El estado de un sistema que consiste de n partículas, sujeto a m restricciones que conectan algunas de las $3n$ coordenadas rectangulares, está completamente especificado por $s=3n-m$ coordenadas generalizadas. Podemos representar el estado de tal sistema por un punto en un espacio s -dimensional llamado espacio de configuración. Cada dimensión de este espacio, corresponde a una de las coordenadas en donde podemos representar el tiempo histórico de un sistema por una curva en el espacio de configuración; cada punto especifica la configuración del sistema en un instante particular. A través de cada punto pasa una infinidad de curvas representantes del movimiento posible del sistema; cada curva corresponde a un conjunto particular de condiciones iniciales. Por lo tanto podemos hablar del camino de un sistema como el movimiento a través del espacio de configuración.

Si la energía potencial de un sistema es independiente de la velocidad, entonces los componentes del momento lineal en coordenadas rectangulares está dado por

$$p_i = \frac{\partial L}{\partial \dot{x}_i},$$

de forma equivalente podemos definir el momento generalizado por:

$$p_i = \frac{\partial L}{\partial \dot{q}_j}.$$

Nosotros sabemos que el Hamiltoniano, $H = q_k p_k$, es igual a la energía total, si la energía potencial es independiente de la velocidad y si la transformación de las ecuaciones entre coordenadas rectangulares y generalizadas no contiene el tiempo explícitamente, por lo que el Hamiltoniano es una cantidad conservativa. Bajo estas condiciones, y si $\frac{\partial H}{\partial t} = 0$, entonces $H=E=$ constante. Así, las ecuaciones de movimiento Hamiltoniano o ecuaciones canónicas del movimiento se escriben como sigue:

$$\dot{q}_k = \frac{\partial H}{\partial p_k}, \quad \dot{p}_k = -\frac{\partial H}{\partial q_k} \quad \text{y} \quad -\frac{\partial L}{\partial t} = \frac{\partial H}{\partial t}$$

3.2 El Principio de Mínimo para Procesos Continuos

3.2.1. Preliminares

El objeto básico de la teoría clásica de sistemas de control es el sistema lineal descrito por la ecuación diferencial

$$\frac{dy}{dt} = A y(t) + B u(t), \quad y(0) = x \in \mathfrak{R}^n \tag{3.1}$$

y una relación de observación o salida

$$w(t) = C y(t), \quad t \geq 0. \tag{3.2}$$

Las transformaciones lineales $A: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$, $B: \mathfrak{R}^m \rightarrow \mathfrak{R}^n$, $C: \mathfrak{R}^n \rightarrow \mathfrak{R}^k$ en (3.1) y (3.2) están identificadas con representaciones matriciales, $M(n,n)$, $M(m,n)$, $M(m,k)$ respectivamente y elementos en \mathfrak{R}^n , \mathfrak{R}^m , \mathfrak{R}^k .

Considere la siguiente ecuación

$$\frac{dq}{dt} = A(t)q(t) + a(t), \quad q(t_0) = q_0 \in \mathfrak{R}^n, \quad (3.3)$$

sobre un intervalo fijo $[0, T]$; $t_0 \in [0, T]$, donde $A(t) \in M(n, n)$, $A(t) = [a_{ij}(t); i, j = 1, \dots, n]$, $a(t) \in \mathfrak{R}^n$, $a(t) = [a_i; i = 1, \dots, n]$, $t \in [0, T]$.

Consideremos el siguiente sistema no lineal

$$\dot{y} = f(y, u), \quad y(0) = x \in \mathfrak{R}^n, \quad (3.4)$$

$$w = h(y). \quad (3.5)$$

Las funciones $f(\cdot)$ y $h(\cdot)$ están definidas sobre $\mathfrak{R}^n \times \mathfrak{R}^m$ y \mathfrak{R}^n y toman valores en \mathfrak{R}^n y \mathfrak{R}^k respectivamente.

Consideremos

$$\dot{z} = f(z(t), t), \quad z(t_0) = x \in \mathfrak{R}^n, \quad (3.6)$$

donde t_0 es un número no negativo y z una aplicación de $\mathfrak{R}^n \times \mathfrak{R}$ en \mathfrak{R}^n . Una solución $z(t)$, $t \in [0, T]$, de la ecuación (3.6) sobre el intervalo $[0, T]$, $t_0 \leq T$, es una función arbitraria absolutamente continua $z(\cdot): [0, T] \rightarrow \mathfrak{R}^n$ que satisface (3.6), para todo $t \in [0, T]$. Una solución local de (3.6) es una función absolutamente continua definida sobre el intervalo $[t_0, \tau)$, $\tau > t_0$ que satisface (3.6) para casi todo $[t_0, \tau)$. Si una solución local definida sobre $[t_0, \tau)$, no puede ser extendida a una solución sobre un intervalo mas grande $[t_0, \tau_1)$, $\tau_1 > \tau$, este es un intervalo de existencia máximo. Así, una solución local arbitraria tiene una extensión a una máxima.

Teorema 3.1 Asuma que los elementos de la función $A(\cdot)$ son localmente integrables. Entonces existe una función $S(t)$, $t \in [0, T]$, con valores en $M(n, n)$ y con elementos absolutamente continuos tales que

$$\frac{d}{dt} S(t) = A(t)S(t) \quad \text{para todo } t \in [0, T], \quad (3.7)$$

$$S(0) = I. \quad (3.8)$$

Además, la matriz $S(t)$ es invertible para un $t \in [0, T]$ arbitrario y la única solución de la ecuación (3.3) es de la forma

$$q(t) = S(t)S^{-1}(t_0)q_0 + \int_{t_0}^t S(t)S^{-1}(s)a(s)ds, \quad t \in [0, T]. \quad (3.9)$$

(Prueba, [JZ95], Parte I, página 11.)

Teorema 3.2. Asuma que para $x \in \mathfrak{R}^n$, la función $f(x, \cdot): [0, T] \rightarrow \mathfrak{R}^n$, es Borel medible y para una función $c(\cdot)$, no negativa, integrable en $[0, T]$, se tiene que

$$|f(x, t)| \leq c(t)(|x| + 1), \quad (3.10)$$

$$|f(x, t) - f(y, t)| \leq c(t)|x - y|, \quad x, y \in \mathfrak{R}^n, \quad t \in [0, T], \quad (3.11)$$

entonces la ecuación (3.6) tiene exactamente una solución $z(\cdot, x)$. Además, para $t \in [0, T]$ arbitrario, la aplicación $x \rightarrow z(t, x)$ es un homeomorfismo de \mathfrak{R}^n en \mathfrak{R}^n . (Prueba, [JZ95], Parte II, página 74.)

Teorema 3.3. Asuma que la condición del Teorema 3.2 se satisface para $t \in [0, T]$ arbitrario, y además la función $f(.,t)$ tiene una derivada continua $f_x(.,t)$ y que las funciones $f(.,t)$ y $f_x(.,t)$ son acotadas sobre subconjuntos acotados de $[0,T] \times \mathfrak{R}^n$. Entonces la aplicación $x \rightarrow z(.,x)$, que actúa desde \mathfrak{R}^n al espacio de funciones continuas $C(0,T; \mathfrak{R}^n)$, es Fréchet diferenciable en un punto arbitrario x_0 y la derivada direccional $v \in \mathfrak{R}^n$ es una solución $\xi(.)$ de la ecuación lineal

$$\dot{\xi} = f_x(z(t, x_0), t)\xi, \quad \xi(t_0) = v \quad (3.12)$$

En particular, la función $t \rightarrow z_x(t, x_0)$ es absolutamente continua y satisface la ecuación lineal

$$\frac{d}{dt} z_x(t, x_0) = f_x(z(t, x_0), t) z_x(t, x_0) \quad \text{para todo } t \in [0, T], \quad (3.13)$$

$$z_x(t_0, x_0) = I. \quad (3.14)$$

(Prueba [JZ95], Parte II, página 75.)

Los resultados anteriores son los resultados clásicos de existencia, unicidad de soluciones, así como los de la dependencia continua respecto a los parámetros y las condiciones iniciales para (3.3) y (3.6) respectivamente.

3.2.2. El Principio de Máximo: El problema de control con tiempo final fijo.

Consideremos un sistema de control

$$\dot{y} = f(y, u), \quad y(0) = x \in \mathfrak{R}^n, \quad (3.15)$$

y una funcional de costo

$$J_T(x, u(.)) = \int_0^T g(y(t), u(t)) dt + G(y(T)), \quad (3.16)$$

con $T > 0$, un número finito fijo. El conjunto de parámetros será denotado por $U \subset \mathfrak{R}^m$, la derivada de f con respecto a las variables de estado, por f_x , y las derivadas de g y G , por g_x y G_x , respectivamente.

Teorema 3.4 Asuma que las funciones f , g , G y f_x , g_x , G_x son continuas y el par de, un control acotado $\hat{u}(.)$ y la solución absolutamente continua correspondiente a (3.15) maximiza la funcional (3.16) para $t \in (0, T)$, arbitrarios, tal que la derivada por la izquierda de y ($\frac{d^-}{dt} \hat{y}(t)$), existe y es igual a $f(\hat{y}(t), \hat{u}(t))$, entonces la siguiente igualdad se satisface

$$\langle p(t), f(\hat{y}(t), \hat{u}(t)) \rangle + g(\hat{y}(t), \hat{u}(t)) \geq \max_{u \in U} \langle p(t), f(\hat{y}(t), u(t)) \rangle. \quad (3.17)$$

En (3.17), $p(t)$, simboliza la solución absolutamente continua de la ecuación

$$\dot{p} = -f_x^*(\hat{y}, \hat{u})p - g_x^*(\hat{y}, \hat{u}), \quad (3.18)$$

con la condición final

$$P(T) = G_x(\hat{y}(T)). \quad (3.19)$$

Nota: ya que una función arbitraria absolutamente continua es diferenciable en casi todas partes, la desigualdad (3.17) se satisface para casi todo $t \in (0, T)$.

Prueba

Asuma primero que $g(x,u)=0$ para $x \in E$, $y \in U$. Sea, para algún $t_0 \in (0, T)$,

$$\frac{d^-}{dt} \hat{y}(t_0) = f(\hat{y}(t_0), \hat{u}(t_0)).$$

Para el control arbitrario $v \in U$ y un $h \geq 0$ suficientemente pequeño definimos la variación de \hat{u} de la siguiente manera

$$u(t, h) = \begin{cases} \hat{u}(t), & \text{if } t \in (0, t_0 - h), \\ v, & \text{if } t \in (t_0 - h, t_0), \\ \hat{u}(t), & \text{if } t \in (t_0, T). \end{cases}$$

Sea $y(., h)$ la salida correspondiente a $u(., h)$, es claro que

$$\frac{d^+}{dh} G(y(T, 0)) \leq 0, \quad (3.20)$$

con tal que la derivada por la derecha exista en (3.20). Para probar que la derivada existe y encontrar este valor, recordemos que:

$$y(t_0, h) = \hat{y}(t_0, h) + \int_{t_0-h}^{t_0} f(y(s, h), v) ds,$$

$$\hat{y}(t_0) = \hat{y}(t_0 - h) + \int_{t_0-h}^{t_0} f(\hat{y}(s), \hat{u}(s)) ds.$$

Tomando esto en cuenta, nosotros obtenemos:

$$\begin{aligned} \frac{d^+}{dt} y(t_0, 0) &= \lim_{h \rightarrow 0} \frac{y(t_0, h) - \hat{y}(t_0)}{h} = \lim_{h \rightarrow 0} \frac{y(t_0, h) - \hat{y}(t_0 - h)}{h} - \lim_{h \rightarrow 0} \frac{\hat{y}(t_0) - \hat{y}(t_0 - h)}{h} = \\ &= f(\hat{y}(t_0), v) - f(\hat{y}(t_0), \hat{u}(t_0)). \end{aligned} \quad (3.21)$$

Además, $\frac{d}{dt} y(t, h) = f(y(t, h), \hat{u}(t))$ se satisface para casi todo $t \in [t_0, T]$. Por el Teorema 3.3, el Teorema 3.1 y por la ecuación (3.21) tenemos que:

$$\frac{d^+}{dt} y(T, 0) = S(T)S^{-1}(t_0)(f(\hat{y}(t_0), v) - f(\hat{y}(t_0), \hat{u}(t_0))),$$

donde $S(.)$ es la solución fundamental de

$$\dot{q} = f_x(\hat{y}, \hat{u})q.$$

Por lo tanto, la derivada derecha en (3.20) existe y se obtiene

$$\begin{aligned} \frac{d^+}{dh} G(y(T, 0)) &= \left\langle G_x(\hat{y}(T)), \frac{d^+}{dh} y(T, 0) \right\rangle = \left\langle G_x(\hat{y}(T)), S(T)S^{-1}(t_0)(f(\hat{y}(t_0), v) - f(\hat{y}(t_0), \hat{u}(t_0))) \right\rangle \\ &= \left\langle S^T(t_0)^{-1}S(T)G_x(\hat{y}(T)), (f(\hat{y}(t_0), v) - f(\hat{y}(t_0), \hat{u}(t_0))) \right\rangle. \end{aligned}$$

Ya que la solución fundamental de la ecuación $\dot{p} = -f_x^T(\hat{y}, \hat{u})p$ es igual a $(S^T(t))^{-1}$, para $t \in [0, T]$. Con la formula (3.9) y la desigualdad (3.20) termina la prueba del teorema en el caso cuando $g(x,u)=0$, $x \in \mathfrak{R}^n$, $u \in U$.

El caso general con la función $g(.)$ puede ser reducido al considerado anteriormente. Para ello introduciremos una variable adicional $w \in \mathfrak{R}$ la cual definimos como sigue:

$$\tilde{f}\left(\begin{bmatrix} x \\ w \end{bmatrix}, u\right) = \begin{bmatrix} f(x, u) \\ g(x, u) \end{bmatrix}, \quad \tilde{G}\begin{bmatrix} x \\ w \end{bmatrix} = G(x) + w, \quad \begin{bmatrix} x \\ w \end{bmatrix} \in \mathfrak{R}^{n+1}, u \in U.$$

Es fácil ver que el problema de maximizar la función (3.16) para el sistema (3.15) es equivalente a el problema de maximizar

$$\tilde{G}\begin{bmatrix} y(T) \\ w(T) \end{bmatrix},$$

con el sistema de control sobre \mathfrak{R}^{n+1} , dado por:

$$\frac{d}{dt}\begin{bmatrix} y \\ w \end{bmatrix} = \tilde{f}\left(\begin{bmatrix} x \\ w \end{bmatrix}, u\right) = \begin{bmatrix} f(x, u) \\ g(x, u) \end{bmatrix},$$

y condición inicial

$$\begin{bmatrix} y(0) \\ w(0) \end{bmatrix} = \begin{bmatrix} x \\ 0 \end{bmatrix} \in \mathfrak{R}^{n+1}.$$

Aplicando la versión simple del teorema, que ha sido probado antes, obtenemos el teorema en el caso general.

El resultado obtenido puede ser reformulado en términos de la función Hamiltoniana como sigue:

$$H(y, p, u) = \langle p, f(y, u) \rangle + g(y, u), \quad y \in \mathfrak{R}^n, u \in U, p \in \mathfrak{R}^n. \quad (3.22)$$

Las ecuaciones (3.15) y (3.18) obtienen la siguiente forma

$$\dot{y} = \frac{\partial H}{\partial p}(y, p, u), \quad y(0) = x, \quad (3.23)$$

$$\dot{p} = -\frac{\partial H}{\partial y}(y, p, u), \quad P(T) = G_x(y(T)), \quad (3.24)$$

y la condición (3.17) es equivalente a

$$\max_{u \in U} H(\hat{y}, \hat{p}, u) = H(\hat{y}, \hat{p}, \hat{u}) \quad (3.25)$$

en apropiados puntos de $t \in [t_0, T]$.

3.3. El principio de Mínimo para Procesos a Eventos Discretos sobre Dominios Finitos

3.3.1 Preliminares

Residuo de Taylor: Del cálculo es conocido que si $f : D \rightarrow \mathbb{R}^m, D \subset \mathbb{R}^m$, es una función suave entonces la fórmula de Taylor se define como:

$$f(y) = f(x) + \int_0^1 f'(x + t(y-x)) dt \quad (y-x) = f(x) + R f(x;y)(y-x),$$

donde el $Rf(x;y)$ es el Residuo de Taylor en forma integral y es análogo a la matriz Jacobiana de f .

Si se sustituye al gradiente por el residuo de Taylor formal algebraico, entonces el principio de mínimo se puede formular análogamente.

Lema 3.1: Si $D \subset Z^n$ es un conjunto finito y $f : D \rightarrow Z^k$ una aplicación arbitraria, entonces existe un polinomio vectorial, entero $p(\xi)$ tal que $p_D = f$.

Sea, $Z = \{\dots -2, -1, 0, 1, \dots\}$ el anillo de los enteros. La n -upla (x_1, x_2, \dots, x_n) , $x_i \in Z$ forma el Z -módulo de enteros que juega el papel de espacio vectorial n -dimensional (R^n). El Z -módulo de polinomios vectoriales enteros, p , de dimensión k , con indeterminada $\xi_1, \xi_2, \dots, \xi_n$, tiene la forma

$$p(\xi) = \begin{pmatrix} p_1(\xi_1, \xi_2, \dots, \xi_n) \\ p_2(\xi_1, \xi_2, \dots, \xi_n) \\ \vdots \\ p_k(\xi_1, \xi_2, \dots, \xi_n) \end{pmatrix},$$

donde

$$p_i(\xi_1, \xi_2, \dots, \xi_n) = \sum_{\underline{j}} p_{ij} \xi^{\underline{j}} = \sum_{\underline{j}} p_{ij} \xi^{j_1} \xi^{j_2} \dots \xi^{j_n},$$

$$\underline{j} = (j_1, \dots, j_n) \in Z_+ = \{0, 1, 2, \dots\}, p_{ij} \in Z.$$

Si $D \subset Z^n$ es un conjunto arbitrario (dominio) entonces para un polinomio vectorial $p(x)$, se puede definir una aplicación $p_D : D \rightarrow Z^m$, asociada a cada vector $x \in D$. Así

$$p_D(x) = \begin{pmatrix} \sum_{\underline{j}} p_{1j} x_1^{j_1} \dots x_n^{j_n} \\ \vdots \\ \sum_{\underline{j}} p_{mj} x_1^{j_1} \dots x_n^{j_n} \end{pmatrix} \in Z^m.$$

Como ambos, los coeficientes de p_{ij} y los de x_1, \dots, x_m son enteros, implica que la imagen $p_D(x)$ también es un vector de enteros.

Si $D \subset Z^n$ es un conjunto finito (un sistema de eventos discretos, el espacio de estado de un autómata, lenguaje formal, etc.) entonces la aplicación $f : D \rightarrow Z^m$ $f : D \rightarrow Z^k$ puede ser representada como una función polinómica entera, $p_D = f$.

Además como el espacio de estado es finito, las funciones pueden ser consideradas polinómicas, ya que por interpolación en todos los puntos del dominio de f se obtiene una representación polinómica.

Nota: Una forma directa de obtener la representación polinómica es usar la interpolación de Lagrange tal como se muestra en el capítulo 1, ejemplo2 y en el capítulo 2, sección 2.3.1.

Lema 3.2. Sea, $p(\xi)$ un polinomio vectorial, entero. Entonces existe una matriz polinomial $Rf(\xi; \eta)$ denominada residuo de Taylor, con indeterminadas $\xi_1, \dots, \xi_n, \eta_1, \dots, \eta_n$, tal que $p(\eta) - p(\xi) = Rp(x : y)(y - x)$.

También, si $f(\xi) = \sum_{\underline{j} \in Z_+^n} p_{\underline{j}} \xi^{\underline{j}}$ es una serie formal, se prueba la existencia del residuo de Taylor, $Rf(\xi; \eta)$.

Así el residuo de Taylor, en este caso es también una serie formal, pero matricial. Si $D \subset Z^n$ es un

conjunto arbitrario y la serie $f(\xi) = \sum_{j \in \mathbb{Z}_+^n} p_j \xi^j$ converge para cada $x \in D$, entonces este es un caso similar al anterior, que induce una aplicación, $p_D : D \rightarrow \mathbb{Z}^k$, por $x \mapsto f_D(x)$.

3.3.2. Procesos discretos enteros finitos

Sean $X_k \subset X \subset \mathbb{Z}^n$, $U_k \subset U$, conjuntos arbitrarios. Sea $f_k : X_k \times U_k \rightarrow X_{k+1}$ una aplicación, que satisface para cada $u(k) \in U_k$ la aplicación $x(k+1) \mapsto f_k(x(k), u(k))$ con $x(k) \in X_k$ polinomial o de serie formal. Sea $\Phi_K : X_K \rightarrow \mathbb{Z}^m$ una función polinomial o de serie formal.

Sea $K \in \mathbb{Z}, K > 0$, $x_0 \in X_0$. A $u = (u(0), u(1), \dots, u(K-1)) \in U$ es llamado control, a $x = (x(0), x(1), \dots, x(K)) \in X$, se le llama la trayectoria (estado) correspondiente al control u y a x_0 la condición (estado) inicial, si,

$$x(k+1) = f_k(x(k), u(k)), \quad x(0) = x_0, \quad (3.26)$$

entonces al par (x, u) se llama proceso discreto, correspondiente al estado inicial x_0 .

Definimos una función de costo multiobjetiva, J_1 sobre el conjunto de los procesos discretos, por

$$J_1(x, u) = \Phi_K(x(K)). \quad (3.27)$$

Supongamos que \mathbb{Z}^m esté equipada con una relación de orden $y_1 \leq y_2$. Entonces, definimos el problema de optimización como sigue.

Decimos que el proceso (x^*, u^*) , $u^* = u^*(1), u^*(2), \dots, u^*(K-1)$, $x^*(k+1) = f_k(x^*(k), u^*(k))$, $x_0^* = \xi \in X$, es óptimo (mínimo) si para todo proceso admisible (x, u) , la siguiente desigualdad se cumple

$$J_1(x^*, u^*) \leq J_1(x, u) \Leftrightarrow \Phi(x^*(K)) \leq \Phi(x(K)). \quad (3.28)$$

A este proceso lo llamamos de tipo uno.

Supongamos ahora que tenemos la función $g_k : X_k \times U_k \rightarrow \mathbb{Z}^m$. Entonces, la función objetivo puede ser definida por

$$J_2(x, u) = \Phi(x(K)) + \sum_{i=0}^{K-1} g_i(x(i), u(i)).$$

Aquí la definición del proceso óptimo respecto a J_2 es similar al anterior si definimos al sistema (3.26) como

$$\begin{aligned} \tilde{X} &= X \times X_K \subset \mathbb{Z}^{n+m}, \\ \tilde{x} &= (x, x_{n+1}, \dots, x_m)^T = (x^T, \bar{x}^T)^T \in \mathbb{Z}^{n+m}, \\ \tilde{f}(\tilde{x}, u) &= (f(x, u)^T, \bar{x}^T + g(x, u)^T)^T \end{aligned}$$

y f, g cumplen las hipótesis respecto a ser polinomiales o series formales, entonces \tilde{f} también lo cumple, y nos queda

$$\tilde{x}(t+1) = \tilde{f}(\tilde{x}(t), u(t)), \quad \tilde{x}_0 = (x_0^T, 0)^T. \quad (3.29)$$

Realizando la minimización sobre

$$J_3(\tilde{x}, u) = \tilde{p}(\tilde{x}(K)) = \tilde{x}(K) + Q(x(K)). \quad (3.30)$$

Así, el problema planteado en (3.29) y (3.30) es un problema de optimización de tipo uno.

Observación: Si el conjunto $X \times U$ es finito entonces por el Lema 1, f , g y Φ , son polinomiales.

3.3.3. Principio de Mínimo de Pontryaguin sobre dominios finitos (PMPDF)

Acá, probamos una condición necesaria para un proceso óptimo (x^*, u^*) , respecto al problema de optimización del tipo uno planteado anteriormente en (3.28), de forma análoga al Principio de Mínimo.

Dado $x(k+1) = f_k(x(k), u(k))$, $x(0) = x_0$, con $X_k \subset X \subset Z^n$, $U_k \subset U$ y una función de costo multiobjetiva, $J_1(x, u) = \Phi(x(K))$, con $\Phi: X_K \rightarrow Z^m$, bajo el supuesto que Z^m está equipada con una relación de orden $y_1 \leq y_2$. Decimos que el proceso (x^*, u^*) , $u^* = u^*(1), u^*(2), \dots, u^*(K-1)$, $x^*(i+1) = f_i(x^*(i), u^*(i))$, $x_{i+1}^* = f(x_i^*, u_i^*)$, $x_0^* = \xi \in X$, es óptimo (mínimo) si para todo proceso admisible (x, u) , la siguiente desigualdad se cumple

$$J_1(x^*, u^*) \leq J_1(x, u) \Leftrightarrow \Phi(x^*(K)) \leq \Phi(x(K)). \quad (3.31)$$

La ecuación adjunta, se define en relación a dos procesos (x^*, u^*) y (x, u) , de la misma longitud (K) , entonces, la ecuación adjunta (trayectoria dual, ó adjunta) para el estado dual $p=(p(0), p(1), \dots, p(K)) \in \mathfrak{R}^{n \times n \times k}$, se define por:

$$p(k) = p(k+1)Rf_k(x^*(k); x(k), u(k)), \quad p(K) = R\Phi(x^*(K); x(K)).$$

En términos de la trayectoria adjunta definida por los procesos (x^*, u^*) y (x, u) , los estados duales son:

$$p(k+1)x(k+1) = p(k+1)f_k(x(k), u(k)), \quad p(k+1)x^*(k+1) = p(k+1)f_k(x^*(k), u^*(k)).$$

Restando de la primera ecuación la segunda, se obtiene:

$$\begin{aligned} p(k+1)[x(k+1) - x^*(k+1)] &= p(k+1)[f_k(x(k), u(k)) - f_k(x^*(k), u^*(k))] = \\ &= p(k+1)[f_k(x(k), u(k)) - f_k(x^*(k), u(k))] + p(k+1)[f_k(x^*(k), u(k)) - f_k(x^*(k), u^*(k))] = \\ &= p(k+1)R_x f_k(x^*(k), x(k), u(k))(x(k) - x^*(k)) + p(k+1)[f_k(x^*(k), u(k)) - f_k(x^*(k), u^*(k))] = \\ &= p(k)(x(k) - x^*(k)) + p(k+1)f_k(x^*(k), u(k)) - p(k+1)f_k(x^*(k), u^*(k)). \end{aligned}$$

Así

$$p(k+1)[x(k+1) - x^*(k+1)] = p(k)(x(k) - x^*(k)) + p(k+1)f_k(x^*(k), u(k)) - p(k+1)f_k(x^*(k), u^*(k)). \quad (3.32)$$

Sabemos que

$$\Phi(x(K)) - \Phi(x^*(K)) = R_x \Phi(x^*(K); x(K))(x(K) - x^*(K)) = p(K)(x(K) - x^*(K)).$$

Como $p(K) = R\Phi(x^*(K); x(K))$, $x(0) = x^*(0)$, sumando la igualdad obtenida desde $k=0, \dots, K-1$, en (3.32) tenemos:

$$\sum_{k=0}^K p(k+1) [x(k+1) - x^*(k+1)] = \sum_{k=0}^{K-1} p(k) [x(k) - x^*(k)] + \sum_{k=0}^{K-1} p(k+1) [f_k(x^*(k), u(k)) - f_k(x^*(k), u^*(k))].$$

Extrayendo el último término de la expresión del lado izquierdo obtenemos:

$$p(K) [x(K) - x^*(K)] = \sum_{k=0}^{K-1} p(k) [x(k) - x^*(k)] + \sum_{k=0}^{K-2} p(k+1) [x(k+1) - x^*(k+1)] + \sum_{k=0}^{K-1} p(k+1) [f_k(x^*(k), u(k)) - f_k(x^*(k), u^*(k))] =$$

$$p(K) [x(K) - x^*(K)] = \sum_{k=0}^{K-1} p(k+1) [f_k(x^*(k), u(k)) - f_k(x^*(k), u^*(k))]$$

entonces

$$\Phi(x(K)) - \Phi(x^*(K)) = \sum_{k=0}^{K-1} p(k+1) [f_k(x^*(k), u(k)) - f_k(x^*(k), u^*(k))]. \quad (3.33)$$

Ahora definimos los controles (variados) $u_{i,v} = (u^*(0), \dots, u^*(i-1), v, u^*(i+1), \dots, u^*(K-1))$ para $i = 0, 1, \dots, K-1$ y $v \in U$. El proceso correspondiente a $u_{i,v}$ es $(x_{i,v}, u_{i,v})$, $x_{i,v}(k+1) = f_k(x_{i,v}(k), u_{i,v}(k))$, donde $x_{i,v} = (x^*(0), \dots, x^*_{i,v}(i+1), \dots, x^*_{i,v}(k-1))$ y las ecuaciones para los estados duales correspondientes son:

$$p_{i,v}(k) = p_{i,v}(k+1) R_x f_k(x^*(k), x_{i,v}(k), u_{i,v}(k)), \quad k > i, \\ p_{i,v}(K) = R\Phi(x^*(K), x_{i,v}(K))$$

En este caso la fórmula (3.33) se reduce a la forma

$$\Phi(x(K)) - \Phi(x^*(K)) = \sum_{k=0}^{K-1} p(k+1) [f_k(x^*(k), u(k)) - f_k(x^*(k), u^*(k))], \\ = p_{i,v}(k+1) [f_k(x^*(k), v) - f_k(x^*(k), u^*(k))].$$

Lo que se quiere es un proceso óptimo (mínimo) para J_1 , entonces:

$$\Phi(x_{i,v}(K)) - \Phi(x^*(K)) = p_{i,v}(i+1) f_k(x^*(i), v) - p_{i,v}(i+1) f_k(x^*(i), u^*(i)) = 0,$$

ó mejor escribir

$$p_{i,v}(i+1) f_k(x^*(i), v) = p_{i,v}(i+1) f_k(x^*(i), u^*(i)),$$

cuando $v = u^*(i)$ se cumple la igualdad.

Definamos la función de Hamilton o Hamiltoniano como: $H: X \times U \times Z^{m \times n} \rightarrow Z^m$, por $H(k, x, u, p) = p f_k(x, u)$, entonces el Principio de Mínimo de Pontriaguin se puede expresar por medio de la ecuación:

$$0 \leq H(i, x^*(i), u^*(i), p_{i,v}(i+1)) - H(i, x^*(i), v, p_{i,v}(i+1)),$$

ó mejor escribir

$$H(i, x^*(i), u^*(i), p_{i,v}(i+1)) \leq H(i, x^*(i), v, p_{i,v}(i+1)),$$

la cual es una desigualdad variacional en $v, u^*(i)$.

Teorema 3.5. PMPDF:

Dado $x(k+1) = f_k(x(k), u(k))$, $x(0) = x_0$, con $X_i \subset X \subset Z^n$ $U_i \subset U$ y una función de costo multiobjetiva, $J_1(x, u) = \Phi(x(K))$, con $\Phi: X_K \rightarrow Z^m$ definida sobre el conjunto los procesos discretos, bajo el supuesto que Z^m está equipada con una relación de orden dada por: $y_1 \leq y_2$.

Suponga que (x^*, u^*) es un proceso óptimo de longitud K . Si $u^*(0) = u(0)$, $u^*(1) = u(1)$, ..., $u^*(i-1) = u(i-1)$, $u(i) = v$, $u^*(i+1) = u(i+1)$, ..., $u^*(K-1) = u(K-1)$, es el control y $x_{i,v}$ es la trayectoria correspondiente, $p_{i,v}$ es la solución de la ecuación adjunta correspondiente al par de procesos, (x^*, u^*) , (x, u) , entonces se cumple que :

$$H(i, x^*(i), u^*(i), p_{i,v}(i+1)) \leq H(i, x^*(i), v, p_{i,v}(i+1))$$

Podemos reescribir la ecuación en diferencia del sistema y la ecuación adjunta en términos del Hamiltoniano como :

$$\begin{aligned} x^*(i+1) &= R_p H(i, x^*(i), u^*(i), p_{i,v}(i+1)), \\ p_{i,v}(i+1) &= R_x H(i, x^*(i), u^*(i), p_{i,v}(i+1)). \end{aligned}$$

Así, el dualismo Hamiltoniano aquí encontrado es equivalente al del cálculo de variaciones dadas por las ecuaciones (3.23) y (3.24).

3.3.4. Algoritmo simbólico para aplicar PMPDF

Ahora, mostramos el algoritmo simbólico para aplicar el Principio de Mínimo de Pontriaguin sobre Dominios Finitos. En cada etapa es necesario usar dos rutinas, las cuales contemplan:

- 1.- Simplificar la expresión simbólica obtenida.
- 2.- Resolver la desigualdad variacionales de la forma $0 \leq p(\xi, u^*, u) [f(\xi, u) - f(\xi, u^*)]$,

donde $\xi \in X$ es el estado dado. Decimos que $u^* \in \Omega(\xi)$ es una solución (\exists) de la desigualdad, si esta se satisface para todo (\forall) $u \in \Omega(\xi)$. En general supongamos que $\Psi(\xi)$ es una solución de la desigualdad variacional; entonces podemos definir la aplicación $\Psi: X \rightarrow U$. Por supuesto, Ψ puede ser multifunción.

$$u^*(\xi) = \Psi(\xi)$$

3.3.4.1. Algoritmo simbólico

En término de la solución Ψ_i de la desigualdad variacional, mostramos un algoritmo para aplicar PMPDF.

Paso 1:

- i) Calcular el estado final: $x^*(K) = f_{K-1}(x^*(K-1), u^*(K-1))$.
- ii) Calcular $x_{K-1,v}(K) = f_{K-1}(x^*(K-1), v)$.

iii) Calcular la ecuación adjunta:

$$p_{K-1,v}(K) = R_x \Phi(x^*(K), x_{K-1,v}(K)) = R_x \Phi(f_{K-1}(x^*(K-1), u^*(K-1)), f_{K-1}(x^*(K-1), v)).$$

iv) Calcular el Hamiltoniano

$$0 \leq \Phi(x_{K-1,v}(K)) - \Phi(x^*(K)) = p_{K-1,v}(K) [f_{K-1}(x^*(K-1), v) - f_{K-1}(x^*(K-1), u^*(K-1))].$$

v) Aplicar rutina 1, simplificar la expresión.

vi) Aplicar rutina 2, donde $\Psi = \Psi(K-1, \xi)$, es obtenida tal que $u^*(K-1) \in \Psi(K-1, x^*(K-1))$ satisface la desigualdad.

⋮

Paso K-i

Suponga que para i , $\Psi = \Psi(K-1, \xi), \dots, \Psi(i+1, \xi)$ se tienen. Entonces por inducción puede demostrarse que las siguientes expresiones :

$$(1) \quad x^*(i+1), x^*(i+2), \dots, x^*(K),$$

$$(2) \quad x_{i,v}(i+1), x_{i,v}(i+2), \dots, x_{i,v}(K),$$

(1) son polinomios de $x^*(i)$ y $u^*(i)$ y (2) son polinomios de $x^*(i)$, $u^*(i)$, y v . Por consiguiente $p_{i,v}(i+1), p_{i,v}(i+2), \dots, p_{i,v}(K)$, también son polinomios de $x^*(i)$, $u^*(i)$, y v . Aplicando la rutina 1, para simplificar la expresión simbólica obtenida y aplicando la rutina 2, se obtiene la función (posiblemente vectorial) $\Psi = \Psi(i, \xi)$ tal que $u^*(i) \in \Psi(i, x^*(i))$ que satisface la desigualdad

$$\Phi(x_{i,v}(i+1)) - \Phi(x^*(i+1)) \geq 0.$$

Es fácil ver que el Hamiltoniano nos queda:

$$0 \leq p_{i,v}(i+1) [f_i(x^*(i), v) - f_i(x^*(i), u^*(i))] = p(x^*(i), u^*(i), v) [f_i(x^*(i), v) - f_i(x^*(i), u^*(i))].$$

La construcción del proceso óptimo en términos de las funciones $\Psi(0, \xi), \Psi(1, \xi), \dots, \Psi(T-1, \xi)$ esta dado por:

$$\begin{aligned} x^*(0) &= \xi, \quad u^*(0) = \Psi(0, \xi), \\ x^*(1) &= f(x^*(0), u^*(0)), \quad u^*(1) = \Psi(1, x^*(1)), \\ &\vdots \\ x^*(i+1) &= f(x^*(i), u^*(i)), \quad u^*(i+1) = \Psi(i+1, x^*(i+1)), \\ &\vdots \end{aligned}$$

Así, tanto el control óptimo y la trayectoria óptima son obtenidas para cada etapa. La construcción anterior muestra que puede ser considerada como la generalización de la Ecuación de Riccati para procesos discretos.

Obviamente el algoritmo mostrado esta incompleto ya que no muestra la forma de obtener la expresión del control óptimo, esta responsabilidad recae en la aplicación de la rutina 2 que se desarrollara en la sección 3.5.

3.3.4.2. Ejemplo Ilustrativo

Dado el siguiente sistema

$$x(k+1) = f_k(x(k), u(k)), \quad x(0) = \xi,$$

y la funcional de costo

$$\Phi(x(K)).$$

Por definición tenemos

$$\begin{aligned} x^*(k+1) &= f_k(x^*(k), u^*(k)), & x^*(0) &= \xi, \\ x_{i,v}(k+1) &= f_k(x_{i,v}(k), u_{i,v}(k)), & x_{i,v}(0) &= \xi, \\ p_{i,v}(k) &= p_{i,v}(k+1) R_x f_k(x^*(k), x_{i,v}(k), u^*(k)), \\ p_{i,v}(K) &= R\Phi(x^*(K); x_{i,v}(K)), \end{aligned}$$

entonces el Hamiltoniano está dado por:

$$H(K-1, x^*(K-1), u^*(K-1), p_{K-1,v}(K)) \leq H(K-1, x^*(K-1), v, p_{K-1,v}(K)).$$

Sea $i=K-1$ y $k \leq K-1$ entonces

$$x_{K-1,v}(k) = x^*(k), \quad x_{K-1,v}(K) = f_{K-1}(x^*(K-1), v).$$

La desigualdad nos queda en función de $u^*(K-1)$, v nos queda:

$$\begin{aligned} 0 &\leq \Phi(x_{K-1,v}(K)) - \Phi(x^*(K)) = R\Phi(x^*(K), x_{K-1,v}(K)) (x_{K-1,v}(K) - x^*(K)) = \\ &= p_{K-1,v}(K) [f_{K-1}(x_{K-1,v}(K-1), v) - f_{K-1}(x^*(K-1), u^*(K-1))], \\ p_{K-1,v}(K) f_{K-1}(x^*(K-1), u^*(K-1)) &\leq p_{K-1,v}(K) f_{K-1}(x^*(K-1), v). \end{aligned}$$

Podemos escribir también:

$$H(K-1, x^*(K-1), u^*(K-1), p_{K-1,v}(K)) \leq H(K-1, x^*(K-1), v, p_{K-1,v}(K)).$$

La función a minimizar es $v \rightarrow \Phi(f_{K-1}(x^*(K-1), v))$, y alcanza el mínimo en $u^*(K-1) = \Psi_{K-1}(x^*(K-1))$.

Por el paso anterior obtenemos:

$$\begin{aligned} x_{K-2,v}(K-2) &= x^*(K-2), & x_{K-2,v}(K-1) &= f_{K-2}(x^*(K-2), v), \\ x_{K-2,v}(K) &= f_{K-1}(x^*(K-1), v) = f_{K-1}(f_{K-2}(x^*(K-2), v), u^*(K-1)) \\ &= f_{K-1}(f_{K-2}(x^*(K-2), v), \Psi_{K-1}(x^*(K-1))) = f_{K-1}(f_{K-2}(x^*(K-2), v), \Psi_{K-1}(x^*(K-1))) \\ &= f_{K-1}(f_{K-2}(x^*(K-2), v), \Psi_{K-1}(f_{K-2}(x^*(K-2), u^*(K-2))), \end{aligned}$$

$$\begin{aligned} x^*(K) &= f_{K-1}(x^*(K-1), u^*(K-1)) = f_{K-1}(f_{K-2}(x^*(K-2), u^*(K-2)), \Psi_{K-1}(x^*(K-1))) \\ &= f_{K-1}(f_{K-2}(x^*(K-2), u^*(K-2)), \Psi_{K-1}(f_{K-2}(x^*(K-2), u^*(K-2)))). \end{aligned}$$

Los estados duales son:

$$\begin{aligned} p_{K-2,v}(K) &= R\Phi(x^*(K), x_{K-2,v}(K)) \\ p_{K-2,v}(K-1) &= p_{K-2,v}(K) R f_{K-1}(x^*(K-1), x_{K-2,v}(K-1), u^*(K-1)). \end{aligned}$$

De esta forma,

$$H(K-2, x^*(K-2), u^*(K-2), p_{K-2,v}(K-1)) \leq H(K-2, x^*(K-2), v, p_{K-2,v}(K-1)),$$

es una desigualdad en $u^*(K-2)$ y v . La solución depende de $x^*(K-2)$, y el óptimo queda expresado como

$$u^*(K-2) = \Psi(x^*(K-2)).$$

Supongamos que las funciones $\Psi_{K-1}(x^*(K-1))$, $\Psi_{K-2}(x^*(K-2))$, ..., $\Psi_{k+1}(x^*(k+1))$ fueron obtenidas. Así por inducción podemos probar que

- (i) $x^*(k+1)$, $x^*(k+2)$, ..., $x^*(K)$, son funciones de $x^*(k)$ y $u^*(k)$.
- (ii) $x_{k,v}(k+1)$, $x_{k,v}(k+2)$, ..., $x_{k,v}(K)$, son funciones de $x^*(k)$, $u^*(k)$ y v .

(iii) $p_{k,v}(k+1), p_{k,v}(k+2), \dots, p_{k,v}(K)$ son funciones de $x^*(k), u^*(k)$ y v .

Si continuamos la substitución de $u^*(K-i)$ hasta $i=k$, la ecuación $u^*(k) = \Psi_k(x^*(k))$ que depende solamente de $x^*(k)$, dependerá de la $x^*(0)=\xi$. Esto es

$$\begin{aligned} u^*(K-i) &= \Psi_{K-i}(x^*(K-i)), \\ &\vdots \\ u^*(1) &= \Psi_1(x^*(1)), \\ u^*(0) &= \Psi_0(x^*(0)) = \Psi_0(\xi). \end{aligned}$$

La construcción del proceso óptimo en términos de $\Psi_0(x^*(0)), \Psi_1(x^*(1)), \dots, \Psi_{K-1}(x^*(K-1))$, esta dado por:

$$\begin{aligned} x^*(0) &= \xi, & u^*(0) &= \Psi_0(\xi), \\ x^*(1) &= f_0(x^*(0), u^*(0)), & u^*(1) &= \Psi_1(x^*(1)) \\ x^*(2) &= f_1(x^*(1), u^*(1)), & u^*(2) &= \Psi_2(x^*(2)) \\ &\vdots \\ x^*(i+1) &= f_i(x^*(i), u^*(i)), & u^*(i+1) &= \Psi_{i+1}(x^*(i+1)). \end{aligned}$$

3.3.4.3. Algoritmo Simbólico detallado para PMPDF

A continuación mostramos los detalles del algoritmo dado en 3.3.4.1.

- **Construcción de la trayectoria del estado**

- Construcción del sistema dinámico, esto es:

$$x(k+1) = f_k(x(k), u(k)), \quad x(0) = x_0, \text{ para todo } k \in [0, K].$$

- Construir los sistemas dinámicos que corresponde al control variado, esto es los $X_{i,v}(k+1)$ donde $u(k) = v$, para cada i , con $i=0, \dots, K$. Donde i es el índice que corresponde al control variado. Esto nos queda como:

$$X_{i,v}(k+1) = f_k(x_{i,v}(k), u_{i,v}(k)), \quad x_{i,v}(0) = \xi,$$

- **Construcción del sistema adjunto**

- Calcular el residuo de la función de costo

$$R_x \Phi(x^*(K); x(K)) = p_{K-1,v}(K) = R_x \Phi(x^*(K), x(K-1))$$

- Calcular el residuo del sistema dinámico para todo k , nos queda

$$R_x f_k(x^*(k), x(k)).$$

- Construir el sistema adjunto a tiempo reversa, esto es:

$$p(k) = p(k+1) R f_k(x^*(k); x(k), u(k)), \quad p(K) = R \Phi(x^*(K); x(K)).$$

- **Construcción del Hamiltoniano y obtener las expresiones de la secuencia de control óptimo por resolución de las desigualdades**

- $k=K$

- Determinar la desigualdad variacional, esto es:

$$0 \leq H(k-1, x^*(k-1), u^*(k-1), p_{k-1,v}(k)) - H(k-1, x^*(k-1), v, p_{k-1,v}(k)).$$

- Resolver la desigualdad variacional, el resultado es una expresión de la forma

$$u^*(k-1) \in \Psi(k-1, x^*(k-1)).$$

- para $k=K-1, \dots, 1$

- Determinar la desigualdad variacional, por

$$0 \leq p_{k,v}(k+1) [f(x^*(k),v) - f(x^*(k),u^*(k))] \\ 0 \leq p(x^*(k),u^*(k),v) [f(x^*(k),v) - f(x^*(k),u^*(k))]$$

- Resolver la desigualdad variacional, el resultado es una expresión de la forma $u^*(k-1) \in \Psi(k-1, x^*(k-1))$

○ Para $u^*(0) \in \Psi(0, x^*(0))$, toma el valor $u^*(0) = \Psi(0, \xi)$

• **Calcular la(s) trayectoria(s) y la(s) secuencia(s) de control óptimo**

○ Dado $u^*(0)$ y $x^*(0) = \xi$

○ Para $k=1, \dots, T$

Calcular:

$$x^*(k+1) = f_k(x^*(k), u^*(k)), \quad u^*(k) = \Psi(k, x^*(k)) .$$

Observación: Considere el hecho donde la secuencia puede tener mas de K etapas. Así,

$$\Phi_K(x^*(K)) \leq \Phi_{K-1}(x^*(K-1))$$

por lo que:

$$\Phi_K(x^*(K)) - \Phi_K(x^*(K-1)) \leq \Phi_{K-1}(x^*(K-1)) - \Phi_K(x^*(K-1)) .$$

Si, $\Phi_k = \Phi, \forall k=0, \dots, K$. Independiente de la etapa, entonces

$$\Phi_K(x^*(K)) - \Phi_K(x^*(K-1)) \leq 0 \\ R\Phi_K(x^*(K); x^*(K-1))(x^*(K) - x^*(K-1)) \leq 0 .$$

3.4. Ejemplo:

Aplicación del Principio de Mínimo para SDED al cálculo formal

Construcción del Modelo polinomial

Tomemos el ejemplo 1 del capítulo 1. Vamos a aplicar el PMPE para obtener la secuencia de instrucciones necesarias para realizar la simplificación de una expresión matemática en un número mínimo de etapas

Sea, el sistema dinámico

$$\begin{pmatrix} x_0(k) \\ \vdots \\ \vdots \\ x_{k+1}(k+1) \end{pmatrix} = \begin{pmatrix} I_{k \times k} \\ 0 \end{pmatrix} \begin{pmatrix} x_0(k) \\ \vdots \\ \vdots \\ x_k(k) \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u(k),$$

y la función de costo $\Phi(x(K), K) = (s(uv), |v|) = (7 - x_K(K)) + \sum_{i=0}^{K-1} (1 - p_i(x_i(i))p_i(u(i)))$,

donde

$$p_k(x_k(k)) = \frac{x_k(k)(x_k(k)-2)(x_k(k)-3)(x_k(k)-4)(x_k(k)-5)(x_k(k)-6)(x_k(k)-7)}{720} + \\ + \frac{x_k(k)(x_k(k)-2)(x_k(k)-3)(x_k(k)-4)(x_k(k)-5)(x_k(k)-6)(x_k(k)-7)}{144} ,$$

$$p_k(u(k)) = \frac{u(k)(u(k)-2)(u(k)-3)(u(k)-4)(u(k)-5)(u(k)-6)(u(k)-7)}{720} + \frac{u(k)(u(k)-2)(u(k)-3)(u(k)-4)(u(k)-5)(u(k)-6)(u(k)-7)}{144}$$

Los valores de los polinomios $p_k(x_k(k))$ y $p_k(u(k))$ para los $x_k(k)$ y $u(k)$ admisibles son presentados en la tabla 1.

$x_k(k)$	$u(k)$	$p_i(x_i(i))p_i(u(i))$	$1-p_i(x_i(i))p_i(u(i))$
0	1	0	1
0	3	0	1
1	1	1	0
1	2	0	1
1	3	1	0
2	3	0	1
3	1	1	0
3	3	1	0
3	4	0	1
4	5	0	1
5	6	0	1
6	7	0	1

Tabla 4.1: Valores de los polinomios

Introducimos una variable suplementaria \tilde{x} y redefinimos el sistema dinámico como:

$$\begin{pmatrix} x_0(k+1) \\ \vdots \\ x_{k+1}(k+1) \\ \tilde{x}(k+1) \end{pmatrix} = \begin{pmatrix} I_{k \times k} & \bar{0} \\ 0 \dots 0 & \\ 0 \dots 1 & \end{pmatrix} \begin{pmatrix} x_0(k) \\ \vdots \\ x_k(k) \\ \tilde{x}(k) \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \end{pmatrix} u(k) + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ 1-p(x,u) \end{pmatrix}$$

con $x_0(0) = 0$, $\tilde{x}(0) = 0$, y $\Phi(x(K), K) = (s(uv), |v|) = \begin{pmatrix} 7 - x_K(K) \\ \tilde{x}(K) \end{pmatrix}$

Aplicando el algoritmo 3.3.4.3.

Si suponemos que la séptima etapa es la última etapa, (T=7) obtenemos:

Construcción de la trayectoria de estado

Evolución del sistema dinámico, para $k = 0, \dots, 6$

$$x_0(k+1) = x_0(k) = x_0(0)$$

\vdots

$$x_k(k+1) = x_k(k) = u(k-1)$$

$$x_{k+1}(k+1) = u(k)$$

$$\tilde{x}(k+1) = \tilde{x}(k) + 1 - p_k(x_k(k))p_k(u(k)) = \tilde{x}(0) + k + 1 - \sum_{i=0}^k p_i(x_i(i))p_i(u(i))$$

Construcción del sistema dinámico con control variado, esto es $x_{i,k}^*(t) = f_{i,k}(x^*(t), v(k))$, para $i = k$, y $k = 0, \dots, 6$

$$\begin{aligned}
 x_{0_{k,v}}^* (k+1) &= x_{0_{k,v}}^* (k) = x_0^* (0) \\
 x_{1_{k,v}}^* (k+1) &= x_{1_{k,v}}^* (k) = u^* (0) \\
 &\vdots \\
 x_{k_{k,v}}^* (k+1) &= x_{k_{k,v}}^* (k) = u^* (t-1) \\
 x_{k+1_{k,v}}^* (k+1) &= v(k) \\
 \tilde{x}^* (k+1) &= \tilde{x}^* (k) + 1 - p_k(x_k^*(k))p_k(v(k))
 \end{aligned}$$

Contrucción del sistema adjunto

Residuo de la función de costo en T=7

$$\begin{aligned}
 \Phi(x(7) - x^*(7)) &= R\Phi(x^*(7), x(7))(x(7) - x^*(7)) = \\
 &= \begin{pmatrix} 7 - x_7(7) - (7 - x_7^*(7)) \\ \tilde{x}(7) - \tilde{x}^*(7) \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0(7) - x_0^*(7) \\ x_1(7) - x_1^*(7) \\ x_2(7) - x_2^*(7) \\ x_3(7) - x_3^*(7) \\ x_4(7) - x_4^*(7) \\ x_5(7) - x_5^*(7) \\ x_6(7) - x_6^*(7) \\ x_7(7) - x_7^*(7) \\ \tilde{x}(7) - \tilde{x}^*(7) \end{pmatrix}
 \end{aligned}$$

Residuo del sistema dinámico

Consideraciones: $p_i(x_i(i)) - p_i(x_i^*(i)) = \tilde{p}_i(x^*, x)(x_i(i) - x_i^*(i))$ y

$$\begin{pmatrix} x(k+1) - x^*(k+1) \\ \tilde{x}(k+1) - \tilde{x}^*(k+1) \end{pmatrix} = R_x f_k \begin{pmatrix} x^*(k), x(k) \\ \tilde{x}^*(k), \tilde{x}(k) \end{pmatrix} \begin{pmatrix} x(k) - x^*(k) \\ \tilde{x}(k) - \tilde{x}^*(k) \end{pmatrix}.$$

Para $k = 0, \dots, 6$

$$R_x f_k \begin{pmatrix} x^*, x \\ \tilde{x}^*, \tilde{x} \end{pmatrix} = \begin{pmatrix} I_{k \times k} & 0 \\ 0 \dots 0 - p_i(u(i))\tilde{p}_i(x_i^*(i), x(i)) & 1 \end{pmatrix}$$

Calculo de la ecuación adjunta

$$k=7, p(7) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$k=6, p(6) = p(7)R_x f \begin{pmatrix} x^*(6), u(6), x(6) \end{pmatrix} =$$

$$\begin{aligned}
 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -p_6(u(6))\tilde{p}_6(x^*, x) & 1 \end{pmatrix} = \\
 &= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -p_6(u(6))\tilde{p}_6(x^*, x) & 1 \end{pmatrix}
 \end{aligned}$$

$$k=5, \quad p(5) = p(6)R_x f(x^*(5), u(5), x(5)) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -p_5(u(5))\tilde{p}_5(x^*, x) & 1 \end{pmatrix}$$

$$k=4, \quad p(4) = p(5)R_x f(x^*(4), u(4), x(4)) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -p_5(u(5))\tilde{p}_5(x^*, x) & 1 \end{pmatrix}$$

$$k=3, \quad p(3) = p(4)R_x f(x^*(3), u(3), x(3)) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -p_4(u(4))\tilde{p}_4(x^*, x) & 1 \end{pmatrix}$$

$$k=2, \quad p(2) = p(3)R_x f(x^*(2), u(2), x(2)) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -p_3(u(3))\tilde{p}_3(x^*, x) & 1 \end{pmatrix}$$

$$k=1, \quad p(1) = p(2)R_x f(x^*(1), u(1), x(1)) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -p_2(u(2))\tilde{p}_2(x^*, x) & 1 \end{pmatrix}$$

$$k=0, \quad p(0) = p(1)R_x f(x^*(0), u(0), x(0)) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -p_1(u(1))\tilde{p}_1(x^*, x) & 1 \end{pmatrix}$$

Construcción del Hamiltoniano

k=7

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0^*(6) \\ x_1^*(6) \\ x_2^*(6) \\ x_3^*(6) \\ x_4^*(6) \\ x_5^*(6) \\ x_6^*(6) \\ u^*(6) \\ \tilde{x}^*(6) + 1 - p_6(x_6^*(6))p_6(u^*(6)) \end{pmatrix} \leq$$

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0^*(6) \\ x_1^*(6) \\ x_2^*(6) \\ x_3^*(6) \\ x_4^*(6) \\ x_5^*(6) \\ x_6^*(6) \\ v(6) \\ \tilde{x}^*(6) + 1 - p_6(x_6^*(6))p_6(v(6)) \end{pmatrix}$$

$$\begin{pmatrix} -u^*(6) \\ \tilde{x}^*(6) + 1 - p_6(x_6^*(6))p_6(u^*(6)) \end{pmatrix} \leq \begin{pmatrix} -v(6) \\ \tilde{x}^*(6) + 1 - p_6(x_6^*(6))p_6(v(6)) \end{pmatrix},$$

k=6

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -p_6(u^*(6))\tilde{p}_6(x^*, x^*) & 1 & 0 \end{pmatrix} \begin{pmatrix} x_0^*(5) \\ x_1^*(5) \\ x_2^*(5) \\ x_3^*(5) \\ x_4^*(5) \\ x_5^*(5) \\ u^*(5) \\ \tilde{x}^*(5) + 1 - p_5(x_5^*(5))p_5(u^*(5)) \end{pmatrix} \leq \begin{pmatrix} x_0^*(5) \\ x_1^*(5) \\ x_2^*(5) \\ x_3^*(5) \\ x_4^*(5) \\ x_5^*(5) \\ v(5) \\ \tilde{x}^*(5) + 1 - p_5(x_5^*(5))p_5(v(5)) \end{pmatrix}$$

$$\begin{pmatrix} 0 \\ -p_6(u^*(6))\tilde{p}_6(x^*, x^*)u^*(5) - p_5(x_5^*(5))p_5(u^*(5)) \end{pmatrix} \leq \begin{pmatrix} 0 \\ -p_6(u^*(6))\tilde{p}_6(x^*, x^*)v(5) - p_5(x_5^*(5))p_5(v(5)) \end{pmatrix},$$

k=5

$$\begin{aligned} & \left(\begin{array}{c} 0 \\ -p_5(u^*(5))\tilde{p}_5(x^*, x^*)u^*(4) - p_4(x_4^*(4))p_4(u^*(4)) \end{array} \right) \leq \left(\begin{array}{c} 0 \\ -p_5(u^*(5))\tilde{p}_5(x^*, x^*)v(4) - p_4(x_4^*(4))p_4(v(4)) \end{array} \right) \\ & \quad \quad \quad k=4 \\ & \left(\begin{array}{c} 0 \\ -p_4(u^*(4))\tilde{p}_4(x^*, x^*)u^*(3) - p_3(x_3^*(3))p_3(u^*(3)) \end{array} \right) \leq \left(\begin{array}{c} 0 \\ -p_4(u^*(4))\tilde{p}_4(x^*, x^*)v(3) - p_3(x_3^*(3))p_3(v(3)) \end{array} \right) \\ & \quad \quad \quad k=3 \\ & \left(\begin{array}{c} 0 \\ -p_3(u^*(3))\tilde{p}_3(x^*, x^*)u^*(2) - p_2(x_2^*(2))p_2(u^*(2)) \end{array} \right) \leq \left(\begin{array}{c} 0 \\ -p_3(u^*(3))\tilde{p}_3(x^*, x^*)v(2) - p_2(x_2^*(2))p_2(v(2)) \end{array} \right) \\ & \quad \quad \quad k=2 \\ & \left(\begin{array}{c} 0 \\ -p_2(u^*(2))\tilde{p}_2(x^*, x^*)u^*(1) - p_1(x_1^*(1))p_1(u^*(1)) \end{array} \right) \leq \left(\begin{array}{c} 0 \\ -p_2(u^*(2))\tilde{p}_2(x^*, x^*)v(1) - p_1(x_1^*(1))p_1(v(1)) \end{array} \right) \\ & \quad \quad \quad k=1 \\ & \left(\begin{array}{c} 0 \\ -p_1(u^*(1))\tilde{p}_1(x^*, x^*)u^*(0) - p_0(x_0^*(0))p_0(u^*(0)) \end{array} \right) \leq \left(\begin{array}{c} 0 \\ -p_1(u^*(1))\tilde{p}_1(x^*, x^*)v(0) - p_0(x_0^*(0))p_0(v(0)) \end{array} \right) \end{aligned}$$

Resolución de las desigualdades variacionales

Utilizando la tabla 3.1, por sustitución se resuelven las desigualdades variacionales

$$k=7, \quad \left(\begin{array}{c} -u^*(6) \\ \tilde{x}^*(6) + 1 - p_6(x_6^*(6))p_6(u^*(6)) \end{array} \right) \leq \left(\begin{array}{c} -v(6) \\ \tilde{x}^*(6) + 1 - p_6(x_6^*(6))p_6(v(6)) \end{array} \right),$$

como $u \in \{1,3,7\} \Rightarrow u^*(6)=7$.

$$k=6, \quad -p_6(u^*(6))\tilde{p}_6(x^*, x^*)u^*(5) - p_5(x_5^*(5))p_5(u^*(5)) \leq -p_6(u^*(6))\tilde{p}_6(x^*, x^*)v(5) - p_5(x_5^*(5))p_5(v(5)),$$

como $u \in \{1,3,6\} \Rightarrow u^*(5)=6$.

$$k=5, \quad -p_5(u^*(5))\tilde{p}_5(x^*, x^*)u^*(4) - p_4(x_4^*(4))p_4(u^*(4)) \leq -p_5(u^*(5))\tilde{p}_5(x^*, x^*)v(4) - p_4(x_4^*(4))p_4(v(4))$$

como $u \in \{1,3,5\} \Rightarrow u^*(4)=5$

$$k=4, \quad -p_4(u^*(4))\tilde{p}_4(x^*, x^*)u^*(3) - p_3(x_3^*(3))p_3(u^*(3)) \leq -p_4(u^*(4))\tilde{p}_4(x^*, x^*)v(3) - p_3(x_3^*(3))p_3(v(3))$$

como $u \in \{1,2,3,4\} \Rightarrow u^*(3) \in \{2,4\}$.

$$k=3, \quad -p_3(u^*(3))\tilde{p}_3(x^*, x^*)u^*(2) - p_2(x_2^*(2))p_2(u^*(2)) \leq -p_3(u^*(3))\tilde{p}_3(x^*, x^*)v(2) - p_2(x_2^*(2))p_2(v(2))$$

como $u \in \{1,2,3,4\} \Rightarrow u^*(2) \in \{1,3\}$.

$$k=2, -p_2(u^*(2))\tilde{p}_2(x^*, x^*)u^*(1) - p_1(x_1^*(1))p_1(u^*(1)) \leq -p_2(u^*(2))\tilde{p}_2(x^*, x^*)v(1) - p_1(x_1^*(1))p_1(v(1))$$

como $u \in \{1,2,3,4\} \Rightarrow u^*(1) \in \{2,4\}$.

$$k=1, -p_1(u^*(1))\tilde{p}_1(x^*, x^*)u^*(0) - p_0(x_0^*(0))p_0(u^*(0)) \leq -p_1(u^*(1))\tilde{p}_1(x^*, x^*)v(0) - p_0(x_0^*(0))p_0(v(0))$$

como $u \in \{1,3\} \Rightarrow u^*(1) \in \{1,3\}$.

Dado los valores obtenidos en cada etapa y por las restricciones descritas en el modelado (ejemplo1, capítulo 1), vemos que podemos componer dos secuencias óptimas, así las dos soluciones son mostradas:

$$01234567 \Rightarrow m_0 m_1 w_1 m_2 w_2 m_3 w_3 m_4$$

$$03412567 \Rightarrow m_0 m_2 w_2 m_1 w_1 m_3 w_3 m_4$$

3.5. Resolución de Desigualdades Variacionales

La(s) desigualdad(es) variacional(es) obtenida(s) en cada etapa, cuando aplicamos PMPDF contienen dos indeterminadas (u^*, v) y un método único de resolución de estas desigualdades no existe. Dado que si para todo v existe u^* tal que:

$$\Phi(x_{i,v}(i+1)) - \Phi(x^*(i+1)) = p_{i,v}(i+1) [f_i(x^*(i), v) - f_i(x^*(i), u^*(i))] \geq 0,$$

se satisface, entonces u^* es el control óptimo en esta etapa, por ser solución de la desigualdad variacional. Como podemos apreciar la solución de la desigualdad variacional, esta prescrita a poder eliminar el cuantificador universal, \forall y el cuantificador de existencial, \exists . Así que, bien podríamos haber formulado un procedimiento vía las técnicas de eliminación de cuantificadores que permite obtener la solución a este tipo de problemas, pero como podemos apreciar en el desarrollo del algoritmo descriptivo en la sección 3.3.4.3., los resultados parciales son expresiones simbólicas en función del estado y la mayoría de las técnicas desarrolladas en este sentido no permiten tener una forma explícita a una solución simbólica.

Por lo que, aquí presentaremos un método que, en principio, da solución a la desigualdad variacional (solución al problema de síntesis).

Por lo tanto, si u^* y $v \in Z$ podemos, en principio, transformar este problema en un problema min-max sobre el dominio $\{-1,1\}$. Las solución se realiza en dos fases, en la primera fase un mínimo es calculado para v y en la segunda fase un máximo es calculado para u^* . En ambos cálculos el algoritmo SCDO detallado en el capítulo 2 es usado.

3.5.1. Solución de las desigualdades variacionales vía solución Min-Max.

La desigualdad variacional a resolver tiene la forma $(p(v,u,x^*) - p(u^*,u,x^*)) \geq 0$. Así, sin perder generalidad, podemos suponer que el dominio está compuesto por 2^{k+1} elementos, esto es que u^* y $v \in \{0,1,2, \dots, 2^{k+1} - 1\}$, en caso contrario es posible hacer una extensión como en 2.2. del capítulo 2.

Entonces la solución de la desigualdad variacional es dada por el siguiente algoritmo.

1. Transformar la desigualdad sobre el dominio $\{-1,1\}$, una forma posible es usar la representación diádica siguiente (lema2, capítulo 2):

$$v=1/2((1+v_0)+2(1+v_1)+ \dots +2^k(1+v_k)) \text{ y } u^*=1/2((1+u^*_0)+2(1+u^*_1)+ \dots +2^k(1+u^*_k)).$$

2. Aplicar el algoritmo SCDO sobre las variables v . Aquí el resultado es una función con indeterminada u^* y parámetro x^* .

3. A la función obtenida en 2 se le debe calcular el máximo. Para ello se cambia el signo a la función y podemos aplicar el algoritmo SCDO de nuevo.

4. Aplicar el algoritmo SCDO sobre las variables u^* , y se obtiene una expresión paramétrica en x^* .

- 5 Invertir la transformación diádica, para obtener el resultado en coordenadas originales parametrizada x^* , como $u^*(.) \in \Psi(., x^*(.))$.

Observaciones:

- a) Independientemente de la cardinalidad de u^* y de v , solo el dominio original debe ser transformado en el dominio $\{-1,1\}$ de tal manera que los resultados son expresados en una secuencia de 1 y -1 , parametrizadas por x^* .
- b) En principio $k+1$ valores mínimos tiene que ser obtenidos y que representan una expresión o valor en coordenadas originales de la solución de la desigualdad variacional.
- c) Tal como se describe en 2.2.2., más de un valor óptimo se obtiene, si el caso 2 ocurre.

3.5.2. Ejemplo: Optimización paramétrica usando PMPDF

Este ejemplo ilustra la potencialidad de la técnica desarrollada en esta tesis, para hacer optimización paramétrica via PMPDF.

Considere el siguiente sistema dinámico a eventos discreto paramétrico

$$x(k+1) = f_k(x(k), u(k), p), \quad x(0) = \xi,$$

con $p \in \mathfrak{R}^+$, y

$$\begin{aligned} X_0 &= \{-1,1\}, X_1 = \{-1,1\}, X_2 = \{-1,1\} \times \{-1,1\}, U = \{-1,1\}, \\ f_0(x, u, p) &= \text{Sign}((p+1)xu + px + 3u + p^2 - 1), \\ f_1(x, u, p) &= \begin{pmatrix} \text{Sign}((0.5p^2 + p - 0.5)xu + (p^2 + 2p - 1)x + \\ + (p^2 + 2p + 2)u + 2p^2 + 4p + 4) \\ \text{Sign}((0.5p^2)xu + (0.25p^2)x + \\ + (p^2 + p + 2)u + 0.5p^2 + 0.5p + 1) \end{pmatrix} \end{aligned}$$

La función de costo está dada por

$$J = G(x_1(2), x_2(2), p) + g_1(x(1), u(1), p) + g_0(x(0), u(0), p)$$

El problema de control óptimo es definido como:

$$J^* = \min_{u(0), u(1)} \{g_0(x(0), u(0), p) + g_1(x(1), u(1), p) + G(x_1(2), x_2(2), p)\},$$

donde

$$\begin{aligned} g_0(x, u, p) &= (1+p)xu + 0.5x + (p+0.5p^2)u + p, \\ g_1(x, u, p) &= (p^2-0.5)xu + (p^2+p)x + (p^2+p+0.5)u + 1, \\ G(x_1, x_2, p) &= (1+p+2p^2)x_1 + (p^2+1)x_2 + p^2. \end{aligned}$$

Se puede verificar que para $p \in \mathfrak{R}^+$

- $\text{Sign}[(0.5p^2 + p - 0.5)x(1)u(1) + (p^2 + 2p - 1)x(1) + (p^2 + 2p + 2)u(1) + 2p^2 + 4p + 4] = 1$
- $\text{Sign}[(0.5p^2)x(1)u(1) + (0.25p^2)x(1) + (p^2 + p + 2)u(1) + 0.5p^2 + 0.5p + 1] = u(1)$,

entonces

$$f_1(x, u, p) = \begin{pmatrix} 1 \\ u(1) \end{pmatrix}.$$

Siguiendo el algoritmo de 3.3.4.3., tenemos:

Construcción de las trayectorias de estado

Construcción del sistema dinámico y extensión del sistema

$$\begin{pmatrix} x_1(0) \\ x_2(0) \\ \bar{x}_1(0) \end{pmatrix} = \begin{pmatrix} x(0) \\ 0 \\ 0 \end{pmatrix},$$

$$\begin{pmatrix} x_1(1) \\ x_2(1) \\ \bar{x}_1(1) \end{pmatrix} = \begin{pmatrix} \text{Sign}((p+1)x_1(0)u(0) + px_1(0) + 3u(0) + p^2 - 1) \\ 0 \\ (1+p)x_1(0)u(0) + 0.5x_1(0) + (p+0.5p^2)u(0) + p + \bar{x}_1(0) \end{pmatrix},$$

$$\begin{pmatrix} x_1(2) \\ x_2(2) \\ \bar{x}_1(2) \end{pmatrix} = \begin{pmatrix} \text{Sign}((0.5p^2 + p - 0.5)x_1(1)u(1) + (p^2 + 2p - 1)x_1(1) + (p^2 + 2p + 2)u(1) + 2p^2 + 4p + 4) \\ \text{Sign}((0.5p^2)x_1(1)u(1) + (0.25p^2)x_1(1) + (p^2 + p + 2)u(1) + 0.5p^2 + 0.5p + 1) \\ (p^2 - 0.5)x_1(1)u(1) + (p^2 + p)x_1(1) + (p^2 + p + 0.5)u(1) + 1 + \bar{x}_1(1) \end{pmatrix}$$

ó

$$\begin{pmatrix} x_1(2) \\ x_2(2) \\ \bar{x}_1(2) \end{pmatrix} = \begin{pmatrix} 1 \\ u(1) \\ (p^2 - 0.5)x_1(1)u(1) + (p^2 + p)x_1(1) + (p^2 + p + 0.5)u(1) + 1 + \bar{x}_1(1) \end{pmatrix}$$

La función de costo extendida en la etapa final es

$$\tilde{G}(x_1(2), x_2(2), \bar{x}_1(2), p) = (1+p+2p^2)x_1(2) + (p^2+1)x_2(2) + p^2 + \bar{x}_1(2)$$

Construcción del sistema dinámico con control variado

Para $i=0$, tenemos

$$\begin{pmatrix} x_{0v}(1,1) \\ x_{0v}(2,1) \\ \bar{x}_{0v}(1,1) \end{pmatrix} = \begin{pmatrix} x_1^*(1) \\ x_1^*(1) \\ \bar{x}_1^*(1) \end{pmatrix}_0 = \begin{pmatrix} \text{Sign}((p+1)x_1^*(0)v(0) + px_1^*(0) + 3v(0) + p^2 - 1) \\ 0 \\ (1+p)x_1^*(0)v(0) + 0.5x_1^*(0) + (p+0.5p^2)v(0) + p \end{pmatrix},$$

$$\begin{pmatrix} x_{ov}(1,2) \\ x_{ov}(2,2) \\ \bar{x}_{ov}(1,2) \end{pmatrix} = \begin{pmatrix} x_1^*(2) \\ x_2^*(2) \\ \bar{x}_1^*(2) \end{pmatrix}_0 = \begin{pmatrix} \text{Sign}((0.5p^2 + p - 0.5)x_1^*(1)u^*(1) + (p^2 + 2p - 1)x_1^*(1) + (p^2 + 2p + 2)u^*(1) + 2p^2 + 4p + 4) \\ \text{Sign}((0.5p^2)x_1^*(1)u^*(1) + (0.25p^2)x_1^*(1) + (p^2 + p + 2)u^*(1) + 0.5p^2 + 0.5p + 1) \\ (p^2 - 0.5)x_1^*(1)u^*(1) + (p^2 + p)x_1^*(1) + (p^2 + p + 0.5)u^*(1) + 1 + \bar{x}_1^*(1) \end{pmatrix}$$

ó

$$\begin{pmatrix} x_{ov}(1,2) \\ x_{ov}(2,2) \\ \bar{x}_{ov}(1,2) \end{pmatrix} = \begin{pmatrix} x_1^*(2) \\ x_2^*(2) \\ \bar{x}_1^*(2) \end{pmatrix}_0 = \begin{pmatrix} 1 \\ u^*(1) \\ (p^2 - 0.5)x_1^*(1)u^*(1) + (p^2 + p)x_1^*(1) + (p^2 + p + 0.5)u^*(1) + 1 + \bar{x}_1^*(1) \end{pmatrix}$$

Para $i=1$, tenemos

$$\begin{pmatrix} x_{1v}(1,1) \\ x_{1v}(2,1) \\ \bar{x}_{1v}(1,1) \end{pmatrix} = \begin{pmatrix} x_1^*(1) \\ x_2^*(1) \\ \bar{x}_1^*(1) \end{pmatrix}_1 = \begin{pmatrix} \text{Sign}((p+1)x_1^*(0)u^*(0) + px_1^*(0) + 3u^*(0) + p^2 - 1) \\ 0 \\ (1+p)x_1^*(0)u^*(0) + 0.5x_1^*(0) + (p+0.5p^2)u^*(0) + p \end{pmatrix},$$

$$\begin{pmatrix} x_{1v}(1,2) \\ x_{1v}(2,2) \\ \bar{x}_{1v}(1,2) \end{pmatrix} = \begin{pmatrix} x_1^*(2) \\ x_2^*(2) \\ \bar{x}_1^*(2) \end{pmatrix}_1 = \begin{pmatrix} \text{Sign}((0.5p^2 + p - 0.5)x_1^*(1)v(1) + (p^2 + 2p - 1)x_1^*(1) + (p^2 + 2p + 2)v(1) + 2p^2 + 4p + 4) \\ \text{Sign}((0.5p^2)x_1^*(1)v(1) + (0.25p^2)x_1^*(1) + (p^2 + p + 2)v(1) + 0.5p^2 + 0.5p + 1) \\ (p^2 - 0.5)x_1^*(1)v(1) + (p^2 + p)x_1^*(1) + (p^2 + p + 0.5)v(1) + 1 + \bar{x}_1^*(1) \end{pmatrix}$$

ó

$$\begin{pmatrix} x_{1v}(1,2) \\ x_{1v}(2,2) \\ \bar{x}_{1v}(1,2) \end{pmatrix} = \begin{pmatrix} x_1^*(2) \\ x_2^*(2) \\ \bar{x}_1^*(2) \end{pmatrix}_1 = \begin{pmatrix} 1 \\ v(1) \\ (p^2 - 0.5)x_1^*(1)v(1) + (p^2 + p)x_1^*(1) + (p^2 + p + 0.5)v(1) + 1 + \bar{x}_1^*(1) \end{pmatrix}$$

Construcción del Sistema Adjunto

Cálculo del residuo de la función de costo

La condición final de la ecuación adjunta es el residuo de la función de costo extendida en la etapa final.

$$\begin{aligned} R\tilde{G}(x_1(2), x_2(2), \bar{x}_1(2), p) &= G(x_1(2), x_2(2), \bar{x}_1(2), p) - G(x_1^*(2), x_2^*(2), \bar{x}_1^*(2), p) \\ &= \begin{pmatrix} 1 + p + 2p^2 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(2) - x_1^*(2) \\ x_1(2) - x_2^*(2) \\ \bar{x}_1(2) - \bar{x}_1^*(2) \end{pmatrix} \end{aligned}$$

Cálculo del residuo del sistema dinámico

El residuo del sistema extendido es

$$R_x f_0(x^*, x, u) = f_0(x(0), u(0)) - f_0(x^*(0), u(0)) =$$

$$= \begin{pmatrix} \text{Sign}((p+1)x_1(0)u(0) + px_1(0) + 3u(0) + p^2 - 1) - \\ \text{Sign}((p+1)x_1^*(0)u(0) + px_1^*(0) + 3u(0) + p^2 - 1) / (x_1 - x_1^*) & 0 & 0 \\ 0 & 0 & 0 \\ (1+p)u(0) + 0.5 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(0) - x_1^*(0) \\ x_1(0) - x_1^*(0) \\ \bar{x}_1(0) - \bar{x}_1^*(0) \end{pmatrix}$$

$$R_x f_1(x^*, x, u) = f_1(x(1), u(1)) - f_1(x^*(1), u(1)) =$$

$$\begin{pmatrix} \text{Sign}((0.5p^2 + p - 0.5)x_1(1)u(1) + (p^2 + 2p - 1)x_1(1) \\ + (p^2 + 2p + 2)u(1) + 2p^2 + 4p + 4) - \text{Sign}((0.5p^2 + p - 0.5)x_1^*(1)u(1) + \\ (p^2 + 2p - 1)x_1^*(1) + (p^2 + 2p + 2)u(1) + 2p^2 + 4p + 4) / (x_1(1) - x_1^*(1)) & 0 & 0 \\ \text{Sign}(0.5p^2 x_1(1)u(1) + 0.25p^2 x_1(1) + (p^2 + p + 2)u(1) \\ + 0.5p^2 + 0.5p + 1) - \text{Sign}(0.5p^2 x_1^*(1)u(1) + 0.25p^2 x_1^*(1) + \\ (p^2 + p + 2)u(1) + 0.5p^2 + 0.5p + 1) / (x_1(1) - x_1^*(1)) & 0 & 0 \\ (p^2 - 0.5)u(1) + (p^2 + p) & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(2) - x_1^*(2) \\ x_2(2) - x_2^*(2) \\ \bar{x}_1(2) - \bar{x}_1^*(2) \end{pmatrix}$$

ó

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ (p^2 - 0.5)u(1) + (p^2 + p) & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1(2) - x_1^*(2) \\ x_2(2) - x_2^*(2) \\ \bar{x}_1(2) - \bar{x}_1^*(2) \end{pmatrix}$$

Construcción del sistema adjunto

Para $i=1$

$$p_{1v}(2) = \begin{pmatrix} 1 + p + 2p^2 & p^2 + 1 & 1 \end{pmatrix}$$

$$p_{1v}(1) = p_{1v}(2) R_x f_1(x^*(1), x_{iv}(1), u(1), p)$$

$$= \begin{pmatrix} 1 + p + 2p^2 & p^2 + 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ (p^2 - 0.5)v(1) + (p^2 + p) & 0 & 1 \end{pmatrix} =$$

$$p_{1v}(1) = \begin{pmatrix} (p^2 - 0.5)v(1) + (p^2 + p) & 0 & 1 \end{pmatrix}$$

Para $i=0$

$$p_{0v}(2) = \begin{pmatrix} 1 + p + 2p^2 & p^2 + 1 & 1 \end{pmatrix}$$

$$p_{0v}(1) = p_{0v}(2) R_x f_1(x^*(1), x_{iv}(1), u(1), p)$$

$$p_{0v}(1) = \begin{pmatrix} 1+p+2p^2 & p^2+1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ (p^2-0.5)u^*(1) + (p^2+p) & 0 & 1 \end{pmatrix} =$$

$$p_{0v}(1) = \begin{pmatrix} (p^2-0.5)u^*(1) + (p^2+p) & 0 & 1 \end{pmatrix}$$

Contrucción del Hamiltoniano

Determinar la desigualdad variacional

$$0 \leq p_{iv}(i+1) \left(f_i(x^*(i), v(i)) - f_i(x^*(i), u^*(i)) \right)$$

Para $i=1$

$$0 \leq p_{1v}(2) \left(f_1(x^*(1), v(1)) - f_1(x^*(1), u^*(1)) \right)$$

$$0 \leq \begin{pmatrix} 1+p+2p^2 & p^2+1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ v(1) - u^*(1) \\ [(p^2-0.5)x_1^*(1) + (p^2+p+0.5)] [v(1) - u^*(1)] \end{pmatrix}$$

Así,

$$0 \leq [(p^2-0.5)x_1^*(1) + (2p^2+p+1.5)] [v(1) - u^*(1)]$$

Para $i=0$

$$0 \leq p_{0v}(1) \left(f_0(x^*(0), v(0)) - f_0(x^*(0), u^*(0)) \right)$$

$$0 \leq \begin{pmatrix} 1+p+2p^2 & p^2+1 & 1 \end{pmatrix} = \begin{pmatrix} \text{Sign}((p+1)x_1^*(0)v(0) + px_1^*(0) + 3v(0) + p^2 - 1) - \\ \text{Sign}((p+1)x_1^*(0)u^*(0) + px_1^*(0) + 3u^*(0) + p^2 - 1) \\ 0 \\ [(1+p)x_1^*(0) + (p+0.5p^2)] [v(0) - u^*(0)] \end{pmatrix}$$

Así,

$$0 \leq [(p^2-0.5)u^*(1) + (p^2-p)] \left[\text{Sign}((p+1)x_1^*(0)v(0) + px_1^*(0) + 3v(0) + p^2 - 1) - \right. \\ \left. \text{Sign}((p+1)x_1^*(0)u^*(0) + px_1^*(0) + 3u^*(0) + p^2 - 1) \right] + [(1+p)x_1^*(0) + (p+0.5p^2)] [v(0) - u^*(0)]$$

Resolución de las desigualdades variacionales

Siguiendo el algoritmo de 3.5.1., tenemos

Parte 1

Considere las desigualdades variacionales,

Para $i=1$

$$0 \leq [(p^2-0.5)x_1^*(1) + (2p^2+p+1.5)] [v(1) - u^*(1)]$$

La aplicación del algoritmo SCDO sobre v en la desigualdad nos da

$$\text{pol}(v(1), u^*(1)) = [(p^2-0.5)x_1^*(1) + (2p^2+p+1.5)] [v(1) - u^*(1)]$$

$$v^+(1) = -\text{Sign}[(p^2-0.5)x_1^*(1) + (2p^2+p+1.5)]$$

Substituyendo y cambiando el signo nos queda

$$-\text{pol}(v^+(1), u^*(1)) = -\text{pol}(u^*(1)) = [(p^2 - 0.5)x_1^*(1) + (2p^2 + p + 1.5)] \text{Sign}[(p^2 - 0.5)x_1^*(1) + (2p^2 + p + 1.5)] + u^*(1)$$

Aplicando el algoritmo SCDO sobre u^* a la expresión anterior tenemos :

$$u^*(1) = -\text{Sign}[(p^2 - 0.5)x_1^*(1) + (2p^2 + p + 1.5)]$$

Para $i=0$

$$0 \leq [(p^2 - 0.5)u^*(1) + (p^2 - p)] \text{Sign}((p+1)x_1^*(0)v(0) + px_1^*(0) + 3v(0) + p^2 - 1) - \text{Sign}((p+1)x_1^*(0)u^*(0) + px_1^*(0) + 3u^*(0) + p^2 - 1)) + [(1+p)x_1^*(0) + (p+0.5p^2)] [v(0) - u^*(0)]$$

La aplicación del algoritmo SCDO sobre v a la desigualdad nos da

$$\text{pol}(v(0), u^*(0)) = [(p^2 - 0.5)u^*(1) + (p^2 - p)] \text{Sign}((p+1)x_1^*(0)v(0) + px_1^*(0) + 3v(0) + p^2 - 1) - \text{Sign}((p+1)x_1^*(0)u^*(0) + px_1^*(0) + 3u^*(0) + p^2 - 1)) + [(1+p)x_1^*(0) + (p+0.5p^2)] [v(0) - u^*(0)]$$

$$v^+(0) = \text{Sign}([(p^2 - 0.5)u^*(1) + (p^2 - p)] \text{Sign}(-x_1^*(0) + p^2 - 4) - \text{Sign}((2p+1)x_1^*(0) + p^2 + 2)) - 2[(1+p)x_1^*(0) + (p+0.5p^2)]$$

Aplicando el algoritmo SCDO sobre u^* a la expresión anterior tenemos:

$$u^*(0) = -\text{Sign}([(p^2 - 0.5)u^*(1) + (p^2 - p)] \text{Sign}((2p+1)x_1^*(0) + p^2 + 2) - \text{Sign}(-x_1^*(0) + p^2 - 4)) + 2[(1+p)x_1^*(0) + (p+0.5p^2)]$$

Parte 2

Calculo de las trayectorias y del control óptimo

Los resultados para dos valores de p son mostrados a continuación.

Para $p=5$

$i=1$

$$u^*(1)_5 = -\text{Sign}[24.5x_1^*(1) + 56.5], \text{ if } x_1^*(1) = \pm 1, \text{ entonces } u^*(1)_5 = -1$$

$i=0$

$$u^*(0)_5 = -\text{Sign}[[24.5u^*(1) + 30] \text{Sign}(11x_1^*(0) + 27) - \text{Sign}(-x_1^*(0) + 21)] + 6x_1^*(0) + 17.5],$$

$$\text{Si } x_1^*(1) = \pm 1, \text{ entonces } u^*(1)_5 = -1$$

Para $p=0$

$i=1$

$$u^*(1)_0 = -\text{Sign}[-0.5x_1^*(1) + 1.5], \text{ Si } x_1^*(1) = \pm 1, \text{ entonces } u^*(1)_0 = -1$$

$i=0$

$$u^*(0)_0 = -\text{Sign}[-u^*(1) + 2x_1^*(0)],$$

$$\text{Si } x_1^*(0) = -1, \text{ entonces } u^*(0)_0 = 1, \text{ y si } x_1^*(0) = 1, \text{ entonces } u^*(0)_0 = -1,$$

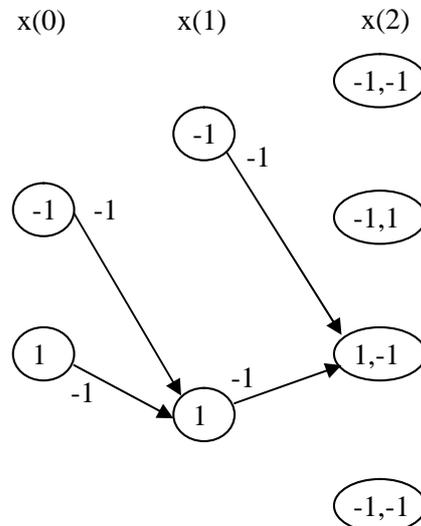


Figura 3.1: Grafo con $p=5$

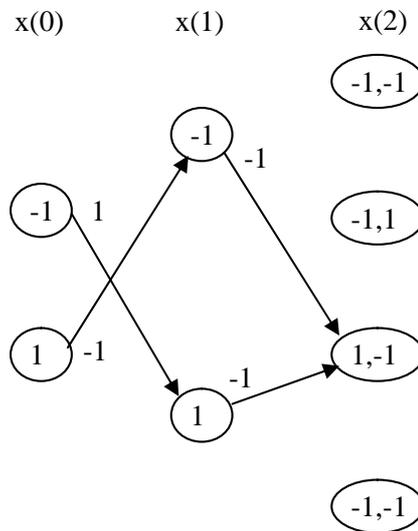


Figura 3.2: Grafo con $p=0$

3.6. El Método de Relajación Aplicado a Sistemas Discretos

La técnica de relajación se puede describir de manera sencilla para la minimización de una función sobre un dominio finito: $f : \{1, 2, \dots, n\} \rightarrow \mathfrak{R}$. A cada función sobre el dominio $Z_n = \{1, 2, \dots, n\}$, se puede asociar otra función sobre el conjunto de medidas probabilísticas

$$\hat{P} = \left\{ P = (p(i)) : 0 \leq p(i), i = 1, 2, \dots, n; \sum_{i=1}^n p(i) = 1 \right\},$$

de la siguiente manera: $i \mapsto f(i) \Leftrightarrow P \rightarrow E_p(f) = \sum_{i=1}^n p(i)f(i)$. Además, \hat{P} , el dominio de la esperanza matemática, es una extensión natural del dominio Z_n . En efecto, a cada $i \in Z_n$ se puede asociar la medida probabilística definida por:

$$P_i = (0, \dots, 0, \underset{i}{1}, 0, \dots, 0), \text{ es decir, } p_i(j) = \begin{cases} 1, & j=i, \\ 0, & j \neq i. \end{cases}$$

Es fácil ver que el mínimo global del problema no lineal $\min_{i \in Z_n} f(i)$ y el mínimo de la función lineal coinciden en $i_0 \leftrightarrow P_{i_0}$.

A la extensión de una función no lineal a una función lineal utilizando la técnica presentada se le denomina el método de relajación. Para sistemas de control clásico, ya en los años 60 J. Warga y otros utilizaron la técnica de relajación. Curiosamente en los últimos años ha resurgido el uso de la técnica, pero en otro contexto. Así, en vez del uso de la optimización combinatoria se propone resolver los problemas relajados, con el fin de disminuir la complejidad computacional. En [JW62], [JW162] se analiza la equivalencia entre el problema original y el problema relajado.

En este trabajo, planteamos el problema de relajación en el contexto de la optimización de una clase de procesos a eventos discretos (sistemas a eventos discretos) equipados con funciones de costo cóncavas y convexas. El primero, genéricamente, siempre garantiza que la equivalencia entre el problema original y el problema relajado es dada esencialmente con una condición de principio de mínimo. Mientras que el segundo, es consecuencia del teorema de separación para conjuntos convexos; garantizando la equivalencia entre el problema original y el relajado por una condición de principio de mínimo.

3.6.1. El problema de optimización de procesos a eventos discretos

Consideremos el siguiente problema de optimización

$$x(k+1) = A(u(k))x(k) + b(u(k)), \quad x(0) = 0 \quad (3.34)$$

donde $A: \{-1,1\}^m \rightarrow \mathbb{R}^{n \times n}$, $b: \{-1,1\}^m \rightarrow \mathbb{R}^n$, son funciones de dominio finito, definido sobre el conjunto $\{-1,1\}^m$ de los vértices del hipercubo de dimensión m . La función de costo es definida por $\Phi: \mathbb{R}^n \rightarrow \mathbb{R}$, como $J(x, u) = \Phi(x(K))$.

Ahora, el objetivo es hallar una secuencia $u^*(0), u^*(1), \dots, u^*(K-1)$, tal que, para la trayectoria correspondiente $x^*(0), x^*(1), \dots, x^*(K)$, se cumpla:

$$J(x^*, u^*) = \Phi(x^*(K)) \leq \Phi(x(K)) = J(x, u),$$

para todo $u(0), u(1), \dots, u(K-1)$.

La de trayectoria variada $x_{i,u}(0), x_{i,u}(1), \dots, x_{i,u}(K)$ y la trayectoria dual $\varphi_{i,u}(i+1), \varphi_{i,u}(i+2), \dots, \varphi_{i,u}(K)$ correspondientes están dadas por

$$\begin{aligned} x_{i,u}(k) &= x^*(k), \quad k \leq i, \\ x_{i,u}(i+1) &= A(u)x^*(i) + b(u), \\ x_{i,u}(k+1) &= A(u^*(k))x_{i,u}(k) + b(u^*(k)), \quad k > i \text{ y} \\ \varphi_{i,u}(k) &= A(u^*(k))^T \varphi_{i,u}(k+1), \quad \varphi_{i,u}(K) = R\Phi(x^*(K); x_{i,u}(K)), \quad k > i, \end{aligned}$$

donde

$$R\Phi(x^*(K); x(K)) = \int_0^1 \text{grad}\Phi(x^*(K) + t(x_{i,u}(K) - x^*(K))) dt.$$

Entonces el criterio de optimalidad es la desigualdad:

$$\left\langle \varphi_{i,u}(i+1), \left(A(u^*(i))x^*(i) + b(u^*(i)) \right) \right\rangle \leq \left\langle \varphi_{i,u}(i+1), A(u)x^*(i) + b(u) \right\rangle,$$

para todo i y u .

Aquí la relajación se define en la dinámica para i , y medida de probabilidad $P = (p(u))$, $u \in \{-1,1\}^m$; de la siguiente manera. La trayectoria $x_{i,P}(0), x_{i,P}(1), \dots, x_{i,P}(K)$, se definen por:

$$\begin{aligned} x_{i,P}(k) &= x^*(k), \quad k \leq i, \\ x_{i,P}(i+1) &= \sum_u p(u) (A(u)x^*(k) + b(u)), \\ x_{i,P}(k+1) &= A(u^*(k))x_{i,P}(k) + b(u^*(k)), \quad k > i. \end{aligned}$$

El problema de relajación consiste en poder obtener una condición tal que:

$$\Phi(x^*(K)) = \min_{u(k) \in \{-1,1\}^m} \Phi(x(K))_{\forall k=0,1,\dots,K} = \min_{P \in \mathcal{P}} \Phi(x_{i,P}(K))_{\forall i}$$

se cumpla.

3.6.2 Resultados

Empecemos esta sección dando algunas observaciones y lemas necesarios para los principales resultados que obtuvimos.

Observemos, que si $P = P_v$, es decir

$$p_v(u) = \begin{cases} 1, & u=v, \\ 0, & u \neq v, \end{cases}$$

entonces $x_{i,P}(k) = x_{i,u}(k)$.

La trayectoria dual correspondiente, en este caso, se define como:

$$\varphi_{i,P}(k) = A(u^*(k))^T \varphi_{i,P}(k+1), \quad \varphi_{i,P}(K) = R\Phi(x^*(K); x_{i,P}(K))$$

Lema 3.3: Si $0 \leq \sum c_i x(i)$, con $\sum x(i) = 1$, $x(i) \geq 0$, entonces $0 \leq c_i$.

Prueba

Por el absurdo, supongamos que $c_i < 0$, entonces, dado que

$$x_0(i) = \begin{cases} 1 & i=i_0, \\ 0 & i \neq i_0, \end{cases}$$

entonces $0 \leq \sum c_i x_0(i) = c_{i_0} < 0$, es una contradicción.

Lema 3.4: Si $x_{i,P}(k)$ es la trayectoria del i -ésimo problema relajado por $P = (p(u))$ y $x_{i,u}(k)$ es la solución del problema perturbado en el paso i por u , entonces

$$x_{i,P}(k) = \sum p(u) x_{i,u}(k), \quad k=i+1, \dots, K. \quad (3.35)$$

Prueba

Por definición

$$x_{i,u}(i+1) = A(u)x^*(i) + b(u),$$

y

$$x_{i,P}(i+1) = \sum p(u) (A(u)x^*(i) + b(u)) = \sum p(u) x_{i,u}(i+1),$$

entonces (3.35) es válido para $i+1$. Suponiendo por inducción matemática que (3.35) es válido para k , entonces para $k+1$,

$$\begin{aligned} x_{i,p}(k+1) &= A(u^*(k))x_p(k) + b(u^*(k)) = \\ &= A(u^*(k))\left(\sum_u p(u)x_{i,v}(k)\right) + b(u^*(k)) = \\ &= \sum_u p(u)\left(A(u^*(k))x_{i,v}(k) + b(u^*(k))\right) = \sum_u p(u)x_{i,v}(k+1). \square \end{aligned}$$

Teorema 3.6: Si Φ es cóncava, es decir, si para todo $\lambda, \mu \geq 0$, $\lambda + \mu = 1$, con $x, y \in \mathfrak{R}^n$, se cumple que

$$\lambda\Phi(x) + \mu\Phi(y) \leq \Phi(\lambda x + \mu y),$$

entonces el problema relajado y el problema original tienen el mismo mínimo. Es decir, si $u^*(0), u^*(1), \dots, u^*(K-1)$, es el control óptimo, entonces, para cada i fijo, el i -ésimo problema relajado

$$\min_{P \in \tilde{P}} \Phi(x_{i,p}(K)),$$

tiene el mínimo en $P_{u^*(i)}$, definido por

$$P_{u^*(i)}(u) = \begin{cases} 1 & u = u^*(i), \\ 0 & u \neq u^*(i), \end{cases}$$

con $\varphi_{i,p}$ el adjunto de $x_{i,p}$.

Prueba

Primero demostraremos la igualdad

$$\begin{aligned} \Phi(x_{i,p}(K)) - \Phi(x^*(K)) &= \sum_u p(u) \left\langle \varphi_{i,p}(i+1), (A(u)x^*(i) + b(u)) - (A(u^*(i))x^*(i) + b(u^*(i))) \right\rangle \\ \Phi(x_{i,p}(K)) - \Phi(x^*(K)) &= R\Phi(x^*(K); x_{i,p}(K)) (x_{i,p}(K) - x^*(K)) = \left\langle \varphi_{i,p}(K), x_{i,p}(K) - x^*(K) \right\rangle \\ &= \left\langle \varphi_{i,p}(K), A(u^*(K-1))x_{i,p}(K-1) + b(u^*(K-1)) - A(u^*(K-1))x^*(K-1) - b(u^*(K-1)) \right\rangle = \\ &= \left\langle A(u^*(K-1))^T \varphi_{i,p}(K), x_{i,p}(K-1) - x^*(K-1) \right\rangle = \left\langle \varphi_{i,p}(K-1), x_{i,p}(K-1) - x^*(K-1) \right\rangle = \dots = \\ &= \left\langle \varphi_{i,p}(i+1), x_{i,p}(i+1) - x^*(i+1) \right\rangle = \\ &= \left\langle \varphi_{i,p}(i+1), \sum_u p(u) (A(u)x^*(i) + b(u)) - (A(u^*(i))x^*(i) + b(u^*(i))) \right\rangle = \\ &= \sum_u p(u) \left\langle \varphi_{i,p}(i+1), (A(u)x^*(i) + b(u)) - (A(u^*(i))x^*(i) + b(u^*(i))) \right\rangle. \end{aligned} \quad (3.36)$$

Ahora supongamos que el i -ésimo problema relajado tiene óptimo en $P_{u^*(i)}$. Entonces

$$0 \leq \Phi(x_{i,p}(K)) - \Phi(x^*(K)),$$

y por la fórmula demostrada anteriormente

$$0 \leq \sum_u p(u) \left\langle \varphi_{i,p}(i+1), (A(u)x^*(i) + b(u)) - (A(u^*(i))x^*(i) + b(u^*(i))) \right\rangle$$

para todo P , así como también para $P = P_v$ y $\varphi_{P_v, i}(i+1) = \varphi_{i, v}(i+1)$.

Por el Lema 3.3, se tiene

$$\langle \varphi_{i, v}(i+1), A(u^*(i))x^*(i) + b(u^*(i)) \rangle \leq \langle \varphi_{i, v}(i+1), A(u)x^*(i) + b(u) \rangle \quad (3.37)$$

siendo (3.37) el principio de mínimo, sin necesidad de usar la concavidad o convexidad de Φ .

Ahora, supongamos que el principio de mínimo se cumple.

Para una medida de probabilidad $P = (p(u))$, multipliquemos la desigualdad (3.37) por $p(u)$ y sumemos para todo $v \in \{-1, 1\}^m$. Entonces

$$\begin{aligned} 0 &\leq \sum_v p(v) \langle \varphi_{i, v}(i+1), (A(u)x^*(i) + b(u)) - (A(u^*(i))x^*(i) + b(u^*(i))) \rangle = \\ &= \sum_v p(v) \langle \varphi_{i, v}(i+1), x_{i, v}(i+1) - x^*(i+1) \rangle = \dots = \\ &= \sum_v p(v) \langle \varphi_{i, v}(K), x_{i, v}(K) - x^*(K) \rangle = \sum_v p(v) (\Phi(x_{i, v}(K)) - \Phi(x^*(K))) \leq \\ &\leq \Phi\left(\sum_v p(v)x_{i, v}(K)\right) - \sum_v p(v)\Phi(x^*(K)) = \Phi(x_{i, P}(K)) - \Phi(x^*(K)), \end{aligned}$$

por lo que el mínimo del problema relajado es alcanzado en $P_{u^*(i)}$. Quedando demostrada la equivalencia entre el problema original y el relajado. \square

3.6.3. Ejemplos

3.6.3.1. Ejemplo 1

Considere el siguiente sistema lineal en $x(k)$,

$$x(k+1) = A(u(k))x(k) + b(u(k)), \quad x(0) = \xi$$

con función de costo lineal

$$\Phi(x(K)) = \langle \varphi, x(K) \rangle.$$

La aplicación lineal $\langle \varphi, x(K) \rangle$ es cóncava (también puede ser convexa) por lo cual con el teorema anterior el problema original y el problema relajado tienen el mismo mínimo. Además las ecuaciones duales son las mismas ya que la condición inicial $\varphi(K) = \varphi$, es independiente de la perturbación o de la relajación. Por lo cual en este caso el principio de mínimo es de la forma clásica

$$\langle \varphi(i+1), A(u^*(i))x^*(i) + b(u^*(i)) \rangle = \min_u \langle \varphi(i+1), A(u)x^*(i) + b(u) \rangle,$$

por lo que en la forma Hamiltoniana nos da

$$H(x^*(i), u^*(i), \varphi(i+1)) \leq H(x^*(i), u, \varphi(i+1)).$$

Nota: el Hamiltoniano es definido como $H(x, u, \varphi) = \langle \varphi, A(u)x + b(u) \rangle$.

3.6.3.2. Ejemplo 2

Considere el mismo sistema lineal anterior

$$x(k+1) = A(u(k))x(k) + b(u(k)), \quad x(0) = \xi,$$

con función de costo cuadrática de la forma

$$\Phi(x(K)) = \langle Qx(K), x(K) \rangle,$$

y las mismas consideraciones tanto para el sistema perturbado como para el relajado.

Entonces

$$\Phi(x_{i,p}(K)) - \Phi(x^*(K)) = \sum_u p(u) \langle \varphi_{i,p}(i+1), A(u)x^*(i) + b(u) - A(u^*(i))x^*(i) + b(u^*(i)) \rangle. \quad (3.38)$$

Supongamos que v es tal que

$$p_{i,v}(u) = \begin{cases} 1, & u=v, \\ 0, & u \neq v. \end{cases}$$

Si $0 \leq \Phi(x_{i,v}(K)) - \Phi(x^*(K))$, (principio de mínimo), Φ es suave en $x^*(K)$ y es convexo, entonces el hiperplano tangente que separa en $x^*(K)$ el cuerpo convexo $\{x : \Phi(x) \leq \Phi(x^*(K))\}$ de los puntos $x_{i,v}(K)$, también separa de la cápsula convexa los puntos $\{x^*(K), x_{i,v}(K), v \in \{-1, 1\}^m\}$, es decir el punto $x_{i,p}(K) = \sum_u p(u) x_{i,v}(K)$.

Por lo cual, $\Phi(x_{i,p}(K)) \geq \Phi(x^*(K))$, es un valor separador. Así

$$0 \leq \Phi(x_{i,p}(K)) - \Phi(x^*(K)) = \Phi\left(\sum_v p(v) x_{i,v}(K)\right) - \Phi(x^*(K)),$$

el valor de los dos problemas es equivalente.

El desarrollo anterior se describe gráficamente en la siguiente figura.

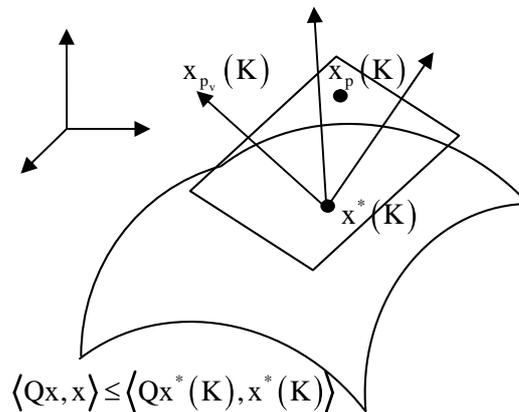


Figura 3.3: Significado geométrico de la relajación

3.7. Conclusiones

- El principio de mínimo descompone el problema de optimización global en sub-problemas de optimización caracterizados por un conjunto de desigualdades variacionales (condición necesaria).
- Para obtener la solución del conjunto de desigualdades variacionales, es necesario aplicar un algoritmo basado en el cálculo simbólico. Este puede ser considerado como una generalización de la ecuación de Ricatti.
- Como nuestro método es una extensión de las técnicas analíticas aplicadas a sistemas continuos, ella puede ser aplicada a los sistemas continuos e híbridos sin ninguna hipótesis adicional de convexidad sobre la parte discreta del modelo.
- La mayoría de los problemas de optimización discretos, no escapan al uso de métodos de optimización combinatoria, que en principio son sensibles a la complejidad computacional. Así, en este trabajo retomamos la idea del uso del método relajación sobre dominios finitos conducente a la transformación de problema discretos en la formulación clásica, la cual provee una vía para resolver el problema de optimización. Así, aquí para una familia particular de procesos discretos como lo son los procesos lineales en el estado equipados con funciones de costo cóncavas o convexas, obtenemos una condición de equivalencia entre el óptimo relajado y el óptimo del problema original en la forma de principio de mínimo. Para el caso con funciones de costo lineal la condición obtenida es una condición de principio de mínimo clásica.

Capítulo 3

ENFOQUE SIMBÓLICO DEL PRINCIPIO DE MÍNIMO	1
3.1. ECUACIONES CANÓNICAS DE MOVIMIENTO EN EL FORMALISMO HAMILTONIANO	2
3.2 EL PRINCIPIO DE MÍNIMO PARA PROCESOS CONTINUOS	2
3.2.1. Preliminares.....	2
3.2.2. <i>El Principio de Máximo: El problema de control con tiempo final fijo</i>	4
3.3. EL PRINCIPIO DE MÍNIMO PARA PROCESOS A EVENTOS DISCRETOS SOBRE DOMINIOS FINITOS.....	6
3.3.1 Preliminares.....	6
3.3.2. <i>Procesos discretos enteros finitos</i>	8
3.3.3. <i>Principio de Mínimo de Pontryaguin sobre dominios finitos (PMPDF)</i>	9
3.3.4. <i>Algoritmo simbólico para aplicar PMPDF</i>	11
3.3.4.1. Algoritmo simbólico	11
3.3.4.2. Ejemplo Ilustrativo.....	12
3.3.4.3. Algoritmo Simbólico detallado para PMPDF	14
3.4. EJEMPLO:.....	15
3.5. RESOLUCIÓN DE DESIGUALDADES VARIACIONALES	21
3.5.1. <i>Solución de las desigualdades variacionales vía solución Min-Max</i>	21
3.5.2. <i>Ejemplo: Optimización paramétrica usando PMPDF</i>	22
3.6. EL MÉTODO DE RELAJACIÓN APLICADO A SISTEMAS DISCRETOS	28
3.6.1. <i>El problema de optimización de procesos a eventos discretos</i>	29
3.6.2 <i>Resultados</i>	30
3.6.3. <i>Ejemplos</i>	32
3.6.3.1. Ejemplo 1	32
3.6.3.2. Ejemplo 2	33
3.7. CONCLUSIONES	34

Capítulo 3

FIGURA 3.1: GRAFO CON $P=5$	28
FIGURA 3.2: GRAFO CON $P=0$	28
FIGURA 3.3: SIGNIFICADO GEOMÉTRICO DE LA RELAJACIÓN	33

Capítulo 3

TABLA 4.1: VALORES DE LOS POLINOMIOS	16
--	----

PROGRAMACION DINAMICA PARAMETRICA

En este capítulo tomaremos el enfoque de programación dinámica (PD), presentaremos el caso continuo de la ecuación de Bellman y el caso discreto. Siendo este último de nuestro particular interés ya que se propone y desarrolla un algoritmo en donde, una explícita expresión paramétrica del control óptimo en forma de lazo cerrado es obtenida.

Así la descomposición del problema de optimización original en subproblemas que es la característica más resaltante del método de PD será aquí explotada. En el caso discreto la solución de cada uno de los subproblemas corresponde a la solución de la ecuación de Bellman en cada una de las etapas, donde un control óptimo es requerido.

Aquí mostraremos como el algoritmo SCDO, presentado en el capítulo 2, permite obtener la solución explícita en cada etapa, cuando el método de PD Discreta (PDD) es usado. El hecho de que el algoritmo SCDO este basado en el cálculo formal, da la potencialidad de introducir parámetros exógenos tanto en las restricciones (sistema o proceso) como en la funcional de costo del problema de optimización planteado. Por lo que una expresión explícita, en los parámetros exógenos y en el estado del sistema, del control es obtenida. Así el problema de síntesis es resuelto.

El uso del algoritmo SCDO, para la resolución de cada etapa cuando el método de PDD es usado, conforma un algoritmo al cual se ha denominado PDPA.

4.1. La Ecuación de Bellman y la función valuada

Nuestras consideraciones serán principalmente dedicadas a los sistemas de control descritos por

$$\dot{x} = f(x, u), \quad x(0) = \xi \quad (4.1)$$

y el criterio o funcional de costo dado por

$$J_T(\xi, u(\cdot)) = \int_0^T g(x(t), u(t)) dt + G(x(T)) \quad (4.2)$$

cuando $T < +\infty$. Si el intervalo de control es $[0, +\infty]$, entonces la función de costo cambia a

$$J(\xi, u(\cdot)) = \int_0^{+\infty} g(x(t), u(t)) dt. \quad (4.3)$$

Nuestro propósito será encontrar un control $u^*(\cdot)$ tal que para todos los controles admisibles $u(\cdot)$ se satisfaga

$$J_T(\xi, u^*(\cdot)) \leq J_T(\xi, u(\cdot)) \quad (4.4)$$

ó

$$J(\xi, u^*(\cdot)) \leq J(\xi, u(\cdot)). \quad (4.5)$$

Hay básicamente dos métodos para encontrar controles que minimizan la funcional de costo (4.2) y (4.3). Uno de ellos encaja en un problema de minimización dado en una familia parametrizada de problemas similares. La inclusión podría ser tal que el valor mínimo, como una función de los parámetros, satisfaga una relación analítica. Si el parámetro seleccionado es la condición inicial y la

longitud del intervalo de control es constante, entonces el mínimo valor de la funcional de costo es llamada la función valuada y la relación analítica la ecuación de Bellman. Conociendo la solución de la ecuación de Bellman uno puede encontrar una estrategia óptima en forma de lazo cerrado de control.

El otro método lleva a las condiciones necesarias sobre el óptimo, lazo abierto, estrategia formulada en la forma así llamado Principio de Máximo, descubierto por L. Pontryaguin y sus colaboradores. Ellas pueden ser obtenidas (para el caso simple) considerando una familia parametrizada de controles y el correspondiente valor de la función de costo (4.2). Este método fue tratado en detalle en el capítulo anterior.

Asuma que el espacio de estado X de un sistema de control es un subconjunto abierto de \mathfrak{R}^n y sea U el subconjunto de parámetros de control incluidos en \mathfrak{R}^m . Asuma que las funciones f , g y G son continuas en $X \times U$ y X respectivamente.

Teorema 4.1 Asuma que una función real $W(\cdot, \cdot)$, definida y continua sobre $[0, T] \times X$, es de clase C^1 sobre $[0, T] \times X$ y satisface la ecuación

$$\frac{\partial W}{\partial t}(t, \xi) = \inf_{u \in U} \left(g(x, u) + \left\langle W_{\xi}(t, x), f(x, u) \right\rangle \right), \quad (t, \xi) \in (0, T) \times X, \quad (4.6)$$

con la condición de borde

$$W(0, \xi) = G(\xi), \quad \xi \in X \quad (4.7)$$

- (i) Si $u(\cdot)$ es un control y $x(\cdot)$ la solución X -valuada absolutamente continua de (4.1), tenemos que

$$J_T(\xi, u(\cdot)) \geq W(T, \xi). \quad (4.8)$$

- (ii) Asuma que para una cierta función $\hat{v}: [0, T] \times X \rightarrow U$:

$$g(\xi, \hat{v}(t, \xi)) + \left\langle W_{\xi}(t, \xi), f(\xi, \hat{v}(t, \xi)) \right\rangle \leq g(\xi, u) + \left\langle W_{\xi}(t, \xi), f(\xi, u) \right\rangle, \quad (4.9)$$

y que \hat{x} es una solución X -valuada absolutamente continua de la ecuación

$$\begin{aligned} \frac{d}{dt} \hat{x}(t) &= f(\hat{x}(t), \hat{v}(T-t, \hat{x}(t))), \quad t \in [0, T], \\ \hat{x}(0) &= \xi. \end{aligned} \quad (4.10)$$

entonces, el control $\hat{u}(t) = \hat{v}(T-t, \hat{x}(t))$, con $t \in [0, T]$, satisface

$$J_T(\xi, \hat{u}(\cdot)) = W(T, \xi).$$

Prueba. (i) Sea $w(t) = W(T-t, x(t))$, $t \in [0, T]$. Entonces $w(\cdot)$ es una función absolutamente continua sobre un intervalo arbitrario $[\alpha, \beta] \subset (0, T)$ y

$$\begin{aligned} \frac{dw}{dt}(t) &= -\frac{\partial W}{\partial t}(T-t, x(t)) + \left\langle W_{\xi}(T-t, x(t), \frac{dx}{dt}(t)) \right\rangle \\ &= -\frac{\partial W}{\partial t}(T-t, x(t)) + \left\langle W_{\xi}(T-t, x(t), f(x(t), u(t))) \right\rangle \end{aligned} \quad (4.11)$$

se cumple para al menos todo $t \in [0, T]$. Por lo tanto, de (4.6) y (4.7) se tiene que:

$$\begin{aligned} W(T-\beta, x(\beta)) - W(T-\alpha, x(\alpha)) &= w(\beta) - w(\alpha) = \int_{\alpha}^{\beta} \frac{dw}{dt}(t) dt \\ &= \int_{\alpha}^{\beta} \left[-\frac{\partial W}{\partial t}(T-t, x(t)) + \left\langle W_{\xi}(T-t, x(t), f(x(t), u(t))) \right\rangle \right] dt \\ &\geq \int_{\alpha}^{\beta} g(x(t), u(t)) dt. \end{aligned}$$

Si α y β tienden a 0 y T respectivamente, obtenemos

$$G(x(T)) - W(T, \xi) \geq - \int_0^T g(x(t), u(t)) dt.$$

Esto prueba (i).

(ii) En forma similar tomamos en cuenta (4.9), para el control \hat{u} y la salida \hat{x} , entonces

$$\begin{aligned} G(\hat{x}(T)) - W(T, \xi) &= \int_0^T \left[-\frac{\partial W}{\partial t}(T-t, \hat{x}(t)) + \left\langle W_{\xi}(T-t, \hat{x}(t), f(\hat{x}(t), \hat{u}(t))) \right\rangle \right] dt \\ &= \int_0^T g(\hat{x}(t), \hat{u}(t)) dt. \end{aligned}$$

Por lo tanto

$$G(x(T)) + \int_0^T g(\hat{x}(t), \hat{u}(t)) dt = W(T, \xi),$$

es la identidad requerida. ?

Observaciones

1. La ecuación (4.6) es llamada la ecuación diferencial en derivadas parciales de Belman ó ecuación funcional de la programación dinámica. Del teorema 4.1 sigue, que sobre apropiadas condiciones, $W(T, \xi)$ es un valor mínimo de la funcional $J_T(\xi, \cdot)$. Aquí W es la función valuada para el problema de minimización (4.2).
2. Sea $U(t, \xi)$ el conjunto de todos los parámetros de control $u \in U$ para el cual el ínfimo sobre el lado derecho de (4.6) se logra. La función $\hat{v}(\cdot, \cdot)$ de la parte (ii) del teorema es un selector de la función multivaluada $U(\cdot, \cdot)$ en el sentido siguiente

$$\hat{v}(t, \xi) \in U(t, \xi), \quad (t, \xi) \in [0, T] \times X.$$

Por lo tanto, para que la condición del teorema sea satisfecha, tal selector no solo debe existir, sino que además el lazo cerrado de la ecuación (4.10) debe tener una solución, absolutamente continua, bien definida.

3. Un resultado similar se mantiene para la funcional de costo más general, siguiente

$$J_T(\xi, u(\cdot)) = \int_0^T e^{-\alpha t} g(x(t), u(t)) dt + e^{-\alpha T} G(x(T)). \quad (4.12)$$

4.2 Programación Dinámica Discreta

La principal motivación es encontrar una trayectoria del conjunto de puntos para una particular familia de Sistema Dinámicos a Eventos Discretos sujeto a entradas de control y parámetros exógenos de evolución.

El enfoque propuesto está basado en el cálculo formal. Este usa la técnica de programación dinámica junto con la adecuación del algoritmo SCDO para generar las expresiones explícitas (simbólicas paramétricas) de la secuencia de control óptimo.

El algoritmo SCDO obtiene iterativamente la solución óptima formal de un problema de minimización sobre un conjunto finito de variables Booleanas. En cada etapa del método de programación dinámica, el algoritmo SCDO es usado para obtener la expresión explícita (simbólica) de la secuencia de control óptimo y consecuentemente, la trayectoria que describe la sucesión óptima de estados discretos desde el estado actualmente evaluado. La secuencia óptima de control es obtenida en la etapa final del algoritmo el cual corresponde a la sustitución del estado inicial del sistema en las expresiones obtenidas.

Como el método de Programación Dinámica es un método de descomposición directa en donde se reemplaza la optimización de una función de varias variables por la resolución recursiva del problema de optimización de una sola variable entonces, la programación dinámica provee un método de optimización multietapa a través de la resolución de la ecuación funcional:

$$J_k(x_k, p) = \min_{u_k} \{c_k(x_k, u_k, p) + J_{k+1}(x_{k+1}, p)\} = \min_{u_k} \{c_k(x_k, u_k, p) + J_{k+1}(f_k(x_k, u_k, p), p)\}, \quad (4.13)$$

para $k=K$ hasta 0, con la condición terminal:

$$J_K(x_K, p) = c_K(x_K, p). \quad (4.14)$$

Clásicamente, el control óptimo de la última etapa y la función de costo dada por (4.14) son evaluados primero y el algoritmo de programación dinámica es aplicado hacia atrás, usando la ecuación de optimalidad (4.13). Por construcción $J_{k+1}(x_{k+1}, p)$, es el costo óptimo “para-ir-a” desde el estado x_{k+1} en periodo $k+1$.

En el enfoque paramétrico propuesto, la evaluación se realiza en cada etapa ya el método proporciona las expresiones formales del control óptimo actual, u_k^* , y por ende la del costo óptimo actual, $J_k(x_k, p)$, que dependen del estado actual y de los parámetros exógenos. Así, el método produce un control realimentado óptimo: $u_k^*(x_k, p)$, y el valor de la función objetivo $J_0(x_0, p)$.

Debido a la característica discreta del conjunto de los estados y de las decisiones, la resolución de la ecuación funcional (4.13) contiene una enumeración y una evaluación implícitas de todos los caminos asociados a posibles trayectorias. Sin embargo, no hay necesidad de enumerar todos los valores posibles de los parámetros. Ésta es la principal ventaja de usar expresiones simbólicas en forma cerrada del control óptimo y el valor óptimo de la función en cada etapa.

4.3. Un enfoque simbólico

El objetivo es poder obtener una expresión formal de la secuencia de control óptimo. La principal motivación para tal enfoque es que muchos problemas de optimización no pueden ser representados por una única formulación numérica. En tales problemas, el modelo, la secuencia de control y la funcional de costo contienen parámetros exógenos en sus expresiones. Estos parámetros exógenos gobiernan el comportamiento del sistema y la solución del problema. A pesar de sus ventajas en los términos de la información contenida en su expresión, los modelos paramétricos no son muy usados en

optimización, porque el cálculo clásico no puede fácilmente extraer información de las expresiones paramétricas.

4.3.1. Modelo de optimización dinámica

Considere el gráfico de la Figura 4.1. Este describe una secuencia de estados discretos, x_k el cual pertenecen al conjunto X_k , sobre la acción de control u_k en U_k . Adicionalmente, los costos asociados con la transición están definido por $c_{kij} = c_k(x_{ki}, u_{kj})$.

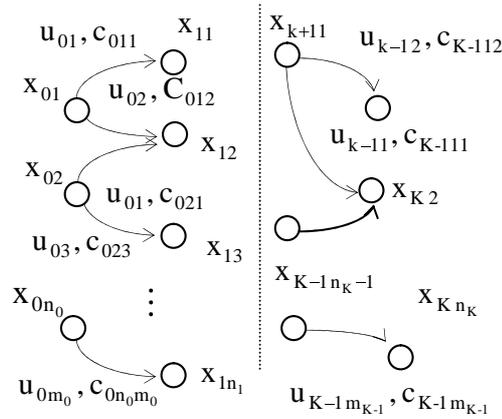


Figura 4.1: Trayectorias descritas por una secuencia de estados discretos en etapas

Una de las características básicas de los grafos considerados es que el conjunto de transiciones pueden ser perfectamente particionados en etapas. Esta propiedad permite descomponer los estados en etapas con sus controles admisibles y sus estados futuros. Muchos sistemas a eventos discretos pueden ser descritos por gráficos multietapas directamente o por la aplicación de técnicas tales como la presentada en [JG89].

Puede observarse que, en el caso paramétrico, la programación dinámica estándar no puede ser aplicada sin evaluación ó enumeración de los todos valores posibles de los parámetros. Además, la secuencia de control no se puede obtener directamente desde el cálculo recursivo de la ecuación funcional.

Si, nosotros consideramos las ecuaciones (4.13) y (4.14) el problema de programación dinámica puede ser reformulado a través de la resolución de la ecuación funcional:

$$J_k(x_k, p) = \min_{u_1, \dots, u_{m_k} \in \{-1, 1\}} \{c_k(x_k, u_1, \dots, u_{m_k}, p) + J_{k+1}(f_k(x_k, u_1, \dots, u_{m_k}, p), p)\} \quad (4.15)$$

para $k=K-1$ hasta 0, con condición terminal:

$$J_K(x_K, p) = C_K(x_K, p), \quad (4.16)$$

ya que usamos el Lema 2.1 y 2.2, del Capítulo 2, de tal manera que se pueda construir un modelo de optimización dinámica.

Así, la secuencia de control será entonces calculada formalmente de la ecuación funcional paramétrica asociada.

El sistema dinámico definido por el grafo de la Figura 4.1, es especificado por la quintupla

$$G(U, X, P, F(.,.)), \quad (4.17)$$

donde, dado que el Lema 2.1 es aplicado, $U = \cup U_k$, $U_k \subset \{-1,1\}^{m_k}$, $2^{m_k-1} < \text{Card}(U_k) \leq 2^{m_k}$, es el conjunto no vacío de entradas. Una entrada o una secuencia de entradas representan la acción que se toma para aplicar una decisión basada en una evaluación, $X = \cup X_k$, $X_k \subset \{-1,1\}^{n_k}$, $2^{n_k-1} < \text{Card}(X_k) \leq 2^{n_k}$, es el conjunto no vacío de estados posibles del sistema, y P es el conjunto de posibles valores para el vector de parámetros p .

Sin perder generalidad, podemos suponer que $u_k \in \{-1,1\}^{m_k}$, y $x_k \in \{-1,1\}^{n_k}$. En caso contrario una extensión como la descrita en el lema 2.2 se da a continuación. Consideremos el control $u_0 \in \{-1,1\}^{m_0}$, ..., $u_{K-1} \in \{-1,1\}^{m_{K-1}}$, con los correspondientes estados $x_0 \in \{-1,1\}^{n_0}$, ..., $x_{K-1} \in \{-1,1\}^{n_{K-1}}$. Definimos

$$x_{k+1}(p) = \tilde{f}_k(x, u, p) = \begin{cases} f_k(x, u, p) & \text{si } x \in X_k \text{ y } u \in U_k, \\ \bar{f}_k(x, u, p) & \text{si } x \notin X_k \text{ o } u \notin U_k. \end{cases}$$

La función de transición de estado $f_k : \{-1,1\}^{n_k} \times \{-1,1\}^{m_k} \times P \rightarrow \{-1,1\}^{n_{k+1}}$, $k=0,1,\dots,K-1$, tiene componentes

$$f_{k,i}(x, u, p) = \text{Sign} \left[\sum_{\alpha \in \{0,1\}^{n_k}} \sum_{\beta \in \{0,1\}^{m_k}} f_{k,i,\alpha,\beta}(p) x_1^{\alpha_1} \cdots x_{n_k}^{\alpha_{n_k}} u_1^{\beta_1} \cdots x_{m_k}^{\beta_{m_k}} \right], \quad (4.18)$$

para $i=1,2,\dots,n_{k+1}$ y

$$c_k : \{-1,1\}^{n_k} \times \{-1,1\}^{m_k} \times P \rightarrow \mathfrak{R}, \quad k=0,1,\dots,K-1,$$

$$c_k(x, u, p) = \sum_{\alpha \in \{0,1\}^{n_k}} \sum_{\beta \in \{0,1\}^{m_k}} c_{k,\alpha,\beta}(p) x_1^{\alpha_1} \cdots x_{n_k}^{\alpha_{n_k}} u_1^{\beta_1} \cdots x_{m_k}^{\beta_{m_k}},$$

con la función de costo $c_K : X_K \times P \rightarrow \mathfrak{R}$,

$$c_K(x, p) = \sum_{\alpha \in \{0,1\}^{n_k}} c_{K,\alpha}(p) x_1^{\alpha_1} \cdots x_{n_k}^{\alpha_{n_k}}.$$

Definimos la función de costo extendida

$$\tilde{c}_k(x, u, p) = \begin{cases} c_k(x, u, p) & \text{si } x \in X_k \text{ y } u \in U_k, \\ \infty & \text{si } x \notin X_k \text{ o } u \notin U_k. \end{cases}$$

Entonces las soluciones mínimas del problema extendido son obviamente iguales que las del original. El problema de la optimización se puede entonces formular como sigue:

$$\text{Minimize } J(x_0, u_0, \dots, u_{K-1}, p), \quad (4.19)$$

sujeto a:

$$x_{k+1} = f_k(x_k, u_k, p), \quad \text{for } k = 0,1,\dots,K. \quad (4.20)$$

con $x_k \in \{-1,1\}^{n_k}$, $u_k \in \{-1,1\}^{m_k}$.

Así, una adecuación del algoritmo SCDO, descrito en el capítulo 2, será desarrollado para obtener el óptimo en cada etapa, esto es, obtener las expresiones simbólicas del control óptimo en cada etapa óptima. La programación dinámica paramétrica se puede entonces aplicar a los sistemas descritos por transiciones como en (4.20).

4.3.2. Optimización simbólica en el paso k

Como hemos dicho, el enfoque simbólico propuesto debe hacer la evaluación en cada etapa del proceso de optimización, por lo que aquí presentamos el procedimiento a seguir para hacer dicha evaluación. La evaluación está dada por la toma de decisiones de forma binaria o múltiple. Ambas maneras de toma de decisión se desarrollan. Se resalta el hecho de que la forma “múltiple decisión” se puede efectuar por dos enfoques. El segundo enfoque de “escogencia múltiple” permite usar el algoritmo SCDO, así la usando SCDO conlleva al algoritmo propuesto, denominado PDPA.

4.3.2.1. Escogencia binaria

Aquí, describimos el método básico de la optimización de una etapa para el caso de una decisión simple binaria. El modelo esta representado por el siguiente gráfico.

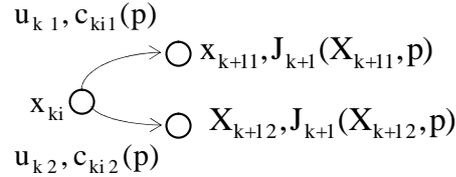


Figura 4.2: Grafo para el caso de una simple decisión binaria

Usando el lema 2.2, $u_k \in \{u_{k1}, u_{k2}\}$ es reemplazada por $\tilde{u}_k \in \{-1, 1\}$. Entonces, consideremos

$$J_k(x_{ki}, p) = \min_{\tilde{u}_k} (J(x_{ki}, -1, p), J(x_{ki}, 1, p)), \quad (4.21)$$

como la representación formal de la toma de decisión en el paso k, con

$$J(x_{ki}, -1, p) = c_{k1}(p) + J_{k+1}(x_{k+11}, p), \quad J(x_{ki}, 1, p) = c_{k2}(p) + J_{k+1}(x_{k+12}, p). \quad (4.22)$$

Así, obtenemos la siguiente tabla de transición para el estado x_{ki} .

x_{k+1}	u_k	\tilde{u}_k	$J_k(x_k, p)$
x_{k+11}	u_{k1}	-1	$J(x_{ki}, -1, p)$
x_{k+12}	u_{k2}	1	$J(x_{ki}, 1, p)$

Tabla 4.1: Tabla de transición para el estado x_{ki}

Aplicando el lema 2.1, la expresión simbólica del control en coordenadas $\{-1, 1\}$ es

$$\tilde{u}_k^+(x_k, p) = \text{Sign}(c_k(x_{k1}, -1, p) - c_k(x_{k1}, 1, p) + J_{k+1}(x_{k+11}, p) - J_{k+1}(x_{k+12}, p)). \quad (4.23)$$

Por lo tanto en coordenadas originales la expresión del control óptimo es dado por:

$$u_k^+ = \frac{u_{k1}}{2}(1 - \tilde{u}_k^+) + \frac{u_{k2}}{2}(1 + \tilde{u}_k^+). \quad (4.24)$$

4.3.2.2. Escogencia múltiple

Consideremos el caso cuando una escogencia múltiple es dada, como lo muestra en la figura 4.3 con $J_{kj}(p) = c_{kij} + J_{k+1}(x_{k+1j}, p)$, $j=1, \dots, M_k$.

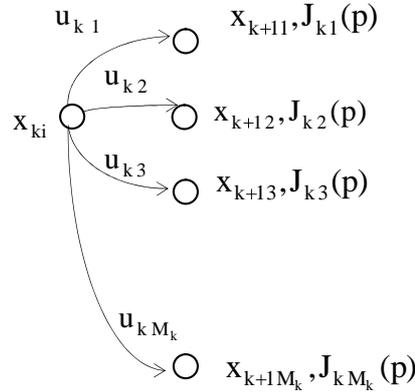


Figura 4.3: Grafo del caso general

Primer enfoque: Descomposición en escalera

La descomposición en escalera del grafo de la figura 4.3 es representado en la figura 4.4.

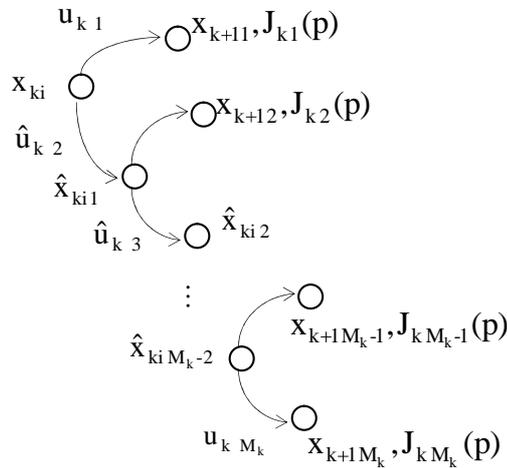


Figura 4.4: Descomposición en escalera

Usando esta descomposición, el caso de escogencia múltiple se puede resolver como una secuencia de casos de escogencia binaria.

Para ilustrar la técnica un ejemplo será desarrollado. Consideremos el grafo de la figura 4.5. La descomposición en escalera de este grafo es presentado en la figura 4.6.

Para simplificar los cálculos, la técnica solo se aplica a x_{k_1} , el mismo proceso se aplica a x_{k_2} , y en general se puede aplicar simultáneamente a ambos x_{k_1} , x_{k_2} .

La aplicación del lema 2.1 a x_{k_1} , es descrito ahora en detalle. Las transiciones asociadas al estado x_{k_1} en el grafo de la figura 4.6 están dadas en la siguiente tabla.

x_{k+1}	u_k	\tilde{u}_k	$J_k(x_k, p)$
x_{k+1_1}	u_{k_1}	-1	$J(x_{k_1}, -1, p)$
\tilde{x}_{k_1}	\tilde{u}_{k_2}	1	$J(x_{k_1}, 1, p)$

Tabla 4.2: Tabla de transición desde el estado x_{k_1}

El correspondiente control óptimo está dado por:

$$\tilde{u}_k^* = \text{Sign}(J(x_{k1}, -1, p) - J(x_{k1}, 1, p)), \tag{4.25}$$

donde,

$$J(x_{k1}, -1, p) = c_{k11}(p) + J_{k+1}(x_{k+11}, p),$$

$$J(x_{k1}, 1, p) = \min(c_{k13}(p) + J_{k+1}(x_{k+13}, p), c_{k14}(p) + J_{k+1}(x_{k+14}, p)).$$

Las transiciones asociada al estado \tilde{x}_{k1} , se muestran en la siguiente tabla.

\tilde{x}_{k1}	u_k	\tilde{u}_k	$J_k(x_k, p)$
x_{k+13}	u_{k3}	-1	$J(x_{k1}, -1, p)$
x_{k+14}	u_{k4}	1	$J(x_{k1}, 1, p)$

Tabla 4.3: Tabla de transición desde el estado \tilde{x}_{k1} .

El control óptimo asociado a la tabla 4.3 es:

$$\tilde{u}_k^* = \text{Sign}(J(x_{k1}, -1, p) - J(x_{k1}, 1, p)), \tag{4.26}$$

donde, $J(x_{k1}, -1, p) = c_{k13}(p) + J_{k+1}(x_{k+13}, p)$, $J(x_{k1}, 1, p) = c_{k14}(p) + J_{k+1}(x_{k+14}, p)$.

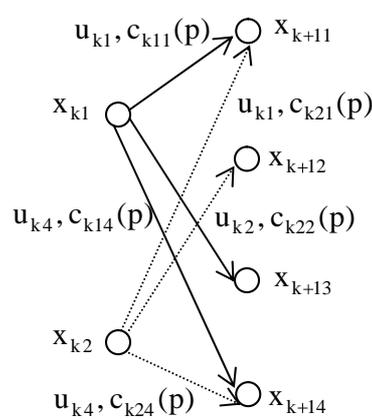


Figura 4.5: Ejemplo ilustrativo

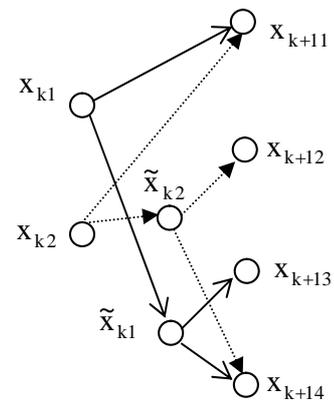


Figura 4.6: Grafo en escalera

El control óptimo en el estado x_k para un valor particular del parámetro p es entonces obtenido desde la evaluación de la expresión de \tilde{u}_k en cada etapa

Así:

- (i). Si el valor de \tilde{u}_{ki} es 1, se procede a evaluar la próxima etapa.
- (ii). Si el valor de \tilde{u}_{ki} es -1 el control óptimo es el correspondiente al valor $\tilde{u}_{ki} = 1$ en la tabla de transición.
- (iii). Si el valor de \tilde{u}_{ki} es cero ambos valores -1 y 1 son controles admisibles. En este caso dos posibles trayectorias óptimas se generan y la evaluación continua

De esta manera, $M_k - 1$ expresiones (paramétricas) simbólicas explícitas son obtenidas para construir el control óptimo de este paso.

Aunque esta descomposición aumenta el tamaño del problema, proporciona una técnica simple para obtener el control óptimo en el paso k .

Segundo enfoque: Forma vectorial, adecuación del algoritmo SCDO

En este enfoque todas las variables de decisión binaria del paso k son consideradas simultáneamente. Para $M_k = 2^{m_k}$ posibles escogencias en el estado, $x_k \in \{-1, 1\}^{m_k}$, las transiciones son presentadas en la tabla 4.4.

La aplicación iterativa del Lema 2.1 (para cada $u_{ki}, i=1, \dots, m_k$) en el caso de decisión binaria donde todas las variables son consideradas simultáneamente es equivalente al problema descrito en 2.2 (Optimización de funciones Booleas). Por lo que el algoritmo SCDO puede ser aplicado como se describe en 2.2.2. Así las dos partes del algoritmo SCDO se detallan a continuación.

La primera parte, correspondiente al cálculo simbólico hacia atrás, consiste de m_k pasos. En cada uno de los m_k pasos, el Lema 2.1. es usado para proveer la expresión simbólica del valor óptimo de $u_{k,i}^+(u_{k1}, \dots, u_{k,i-1}, p)$. La segunda parte comienza cuando el valor de x_k es conocido. Entonces procedemos hacia delante para calcular la secuencia óptima ($u_{k,i}^*(p)$) para $i=1, \dots, m_k$.

x_{k+1}	u_k	$u_{k1} \dots u_{km_k}$	$J_k(x_k, p)$
x_{k+11}	u_{k1}	-1 -1 ... -1 -1	$J_{k1}(p)$
x_{k+12}	u_{k2}	-1 -1 ... -1 1	$J_{k2}(p)$
\vdots	\vdots	\vdots	\vdots
$x_{k+12^{m_k}}$	$u_{k2^{m_k}}$	1 1 ... 1 1	$J_{k,2^{m_k}}(p)$

Tabla 4.4: Tabla de transición del estado x_k , del grafo de la figura 4.3

Aplicando el algoritmo SCDO obtenemos:

Parte 1:

La función de costo para el paso k :

$$g_{k,0}(x_k, p, u_1, \dots, u_{m_k}) = c_k(x_k, u_1, \dots, u_{m_k}, p) + J_{k+1}(f_k(x_k, u_1, \dots, u_{m_k}, p), p).$$

Calcular para $i=0, \dots, m_k - 2$

$$u_{k,m_k-i}^+(x_k, p, u_1, \dots, u_{m_k-i-1}) = \text{Sign}(g_{k,i}(x_k, p, u_1, \dots, u_{m_k-i-1}, -1) - g_{k,i}(x_k, p, u_1, \dots, u_{m_k-i-1}, 1)),$$

$$g_{k,i+1}(x_k, p, u_1, \dots, u_{m_k-i-1}) = g_{k,i}(x_k, p, u_1, \dots, u_{m_k-i-1}, u_{k,m_k-i}^+(x_k, p, u_1, \dots, u_{m_k-i-1})).$$

Así

$$u_{k,1}^+(x_k, p) = \text{Sign}(g_{k,m_k}(x_k, p, -1) - g_{k,m_k}(x_k, p, 1)).$$

Parte 2

Asuma que el valor de x_k^* es conocido. Entonces el control óptimo es:

$$u_{k1}^*(x_k^*, p) = u_{k1}^+(x_k^*, p),$$

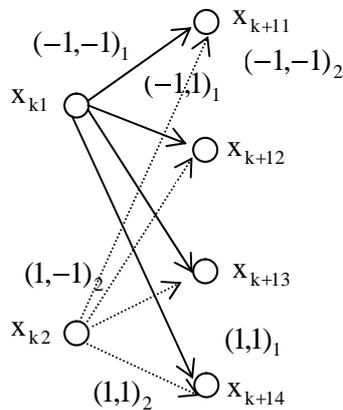
y para $i=2, \dots, m_k$, tenemos que

$$u_{ki}^*(x_k, u_1^*, \dots, u_{i-1}^*, p) = u_{ki}^+(x_k, u_1^*, \dots, u_{i-1}^*, p).$$

Obteniéndose m_k expresiones (paramétricas) simbólicas. Ellas pueden perfectamente corresponder a la forma explícita del control óptimo en la etapa k cuando el método de programación dinámica es usado. Con la expresión obtenida en la primera parte, el control óptimo en el paso k toma la forma $u_k^+(x_k, p)$ para $x_k = x_k^*$.

Para ilustrar la técnica consideremos un ejemplo dado por el grafo de la figura 4.5. Al aplicar el procedimiento a x_{k1} es necesario realizar la extensión de los $J_k(x_k, p)$ al producto $\{-1, 1\}^2$, como se describe en la sección 2,2 del Capítulo 2. Así un valor $J_{k2}(x_k, p)$ es asignado para x_{k+12} .

El grafo extendido de la Figura 4.3 es presentado en la Figura 4.7. Las correspondientes decisiones son mostradas en la Tabla 4.5. y como puede verse el número controles binarios es $m_k=2$.



	u_{k1}	u_{k2}	$J_k(x_k, p)$
x_{k+11}	-1	-1	$J_{k1}(p)$
x_{k+12}	-1	1	$J_{k2}(p)$
x_{k+13}	1	-1	$J_{k3}(p)$
x_{k+14}	1	1	$J_{k4}(p)$

Figura 4.7: Grafo de transiciones modificado.

Tabla 4.5: Decisiones binarias extendidas

El procedimiento sigue, usando el algoritmo SCDO.

La decisión a tomar entre cual de los dos enfoques usar, dependerá de la complejidad de las expresiones de la función a considerar. El primer enfoque envuelve expresiones simples, pero el número de pasos, es al menos $2^{m_k} - 1$, y puede llevar a realizar extensos cálculos. Por el contrario en el segundo enfoque solamente m_k expresiones son necesarias para determinar el actual control óptimo, pero el cálculo puede ser complejo.

En el segundo enfoque, el algoritmo SCDO puede aplicarse directamente, permitiendo obtener la expresión simbólica explícita paramétrica del control óptimo en cada etapa cuando el método de programación dinámica es usado.

4.4. Programación Dinámica Paramétrica

Acá el problema de síntesis a lazo cerrado del control en el caso de parámetros exógenos de evolución, para cada etapa, cuando el método de programación dinámica se utiliza es resuelto. La solución esta dada por el uso del algoritmo SCDO. Así un algoritmo detallado a implantar esta técnica se presenta a continuación

4.4.1. Algoritmo programación dinámica paramétrica (PDPA)

El algoritmo para el método de programación dinámica paramétrica (PDPA), basado en optimización simbólica, es como sigue.

Considere $J_K^+(x_K, p) = c_K(x_K, p)$.

Para $k=K-1$ to 0, calcular

$$J_k(x_k, u_k, p) = \text{Min}_{u_k} [c_k(x_k, p) + J_{k+1}^+(f_k(x_k, u_k, p), p)] \quad (4.27)$$

Como, $g_k(x_k, u_k, p) = c_k(x_k, u_k, p) + J_{k+1}^+(f_k(x_k, u_k, p), p)$, se aplica la transformación diádica (TD), lema 2.1, esto nos da:

$$g_k(x_k, u_{k1}, \dots, u_{kq}, p) = c_k(x_k, u_{k1}, \dots, u_{kq}, p) + J_{k+1}^+(f_k(x_k, u_{k1}, \dots, u_{kq}, p), p).$$

Se aplica el algoritmo de optimización simbólica en el paso k, (4.3.2.), y se obtiene

$$(u_{k1}^+(x_k, p), \dots, u_{kq}^+(x_k, p)) \xrightarrow{\text{TD}^{-1}} u_k^+(x_k, p)$$

donde

$$u_{ki}^+(x_k, p) = u_{ki}^+(x_k, u_{k1}, \dots, u_{ki-1}, p).$$

Nota: Usando el procedimiento de escogencia múltiple, $q = M_k - 1$ pasos son necesarios si el primer enfoque (representación en escalera) es aplicado y $q = m_k$ pasos si el segundo enfoque (algoritmo SCDO) es aplicado. Con $M_k = 2^{m_k}$.

Así, obtenemos

$$J_k^+(x_k, u_k^+, p) = c_k(x_k, p) + J_{k+1}^+(f_k(x_k, u_k^+, p), p) = c_k(x_k, p) + I_k(x_k, p).$$

Si x_0^* es conocido entonces, podemos calcular $k=0$ hasta $K-1$

$$u_k^*(x_k^*, p) = u_k^+(x_k^*, p), \quad J_k^*(x_k^*, u_k^*, p) = J_k^+(x_k^*, u_k^*, p), \quad x_{k+1}^* = f_k(x_k^*, u_k^*, p),$$

y finalmente obtenemos $J_k^*(x_k^*, p) = c_k(x_k^*, p)$ $J_K^*(x_K^*, p) = c_K(x_K^*, p)$

4.5. Ejemplo

Considere el sistema dinámico paramétrico discreto, $x(k+1) = f_k(x(k), u(k), p)$, $x(0) = \xi$, con $p \in \mathfrak{R}^+$, y $X_0 = \{-1, 1\}$, $X_1 = \{-1, 1\}$, $X_2 = \{-1, 1\} \times \{-1, 1\}$, $U = \{-1, 1\}$,

$$f_0(x, u, p) = \text{Sign}((p+1)xu + px + 3u + p^2 - 1),$$

$$f_1(x, u, p) = \begin{pmatrix} \text{Sign}((0.5p^2 + p - 0.5)xu + (p^2 + 2p - 1)x + \\ + (p^2 + 2p + 2)u + 2p^2 + 4p + 4) \\ \text{Sign}((0.5p^2)xu + (0.25p^2)x + \\ + (p^2 + p + 2)u + 0.5p^2 + 0.5p + 1) \end{pmatrix}.$$

El problema de control óptimo es definido por:

$$\min_{u(0), u(1)} \{g_0(x(0), u(0), p) + g_1(x(1), u(1), p) + G(x_1(2), x_2(2), p)\},$$

donde

$$\begin{aligned} g_0(x, u, p) &= (1+p)xu + 0.5x + (p+0.5p^2)u + p, \\ g_1(x, u, p) &= (p^2 - 0.5)xu + (p^2 + p)x + (p^2 + p + 0.5)u + 1, \\ G(x_1, x_2, p) &= (1+p+2p^2)x_1 + (p^2 + 1)x_2 + p^2. \end{aligned}$$

Aplicando PDPA, obtenemos

$$J_1(x(1), p) = \min_{u(1)} \{g_1(x(1), u(1), p) + G(x(2), p)\}, \quad J_0(x(0), p) = \min_{u(0)} \{g_0(x(0), u(0), p) + J_1(x(1), p)\}.$$

Por substitución

$$\begin{aligned} J_1(x(1), p) &= \min_{u(1)} \left\{ (p^2 - 0.5)x(1)u(1) + (p^2 + p)x(1) + (p^2 + p + 0.5)u(1) + 1 + (1+p+2p)^* \right. \\ &\quad \text{Sign} \left[(0.5p^2 + p - 0.5)x(1)u(1) + (p^2 + 2p - 1)x(1) + (p^2 + 2p + 2)u(1) + 2p^2 + 4p + 4 \right] + \\ &\quad \left. + (p^2 + 1) * \text{Sign} \left[(0.5p^2)x(1)u(1) + (0.25p^2)x(1) + (p^2 + p + 2)u(1) + 0.5p^2 + 0.5p + 1 \right] + p^2 \right\} \end{aligned}$$

se puede verificar que para $p \in \mathfrak{R}^+$

- $\text{Sign} \left[(0.5p^2 + p - 0.5)x(1)u(1) + (p^2 + 2p - 1)x(1) + (p^2 + 2p + 2)u(1) + 2p^2 + 4p + 4 \right] = 1$
- $\text{Sign} \left[(0.5p^2)x(1)u(1) + (0.25p^2)x(1) + (p^2 + p + 2)u(1) + 0.5p^2 + 0.5p + 1 \right] = u(1)$

En este caso, J_1 puede ser simplificado como sigue:

$$J_1(x(1), p) = \min_{u(1)} \left\{ (p^2 - 0.5)x(1)u(1) + (p^2 + p)x(1) + (2p^2 + p + 1.5)u(1) + (2 + p + 3p^2) \right\}$$

Así, el control óptimo paramétrico es $u_1^+(x(1), p) = -\text{Sign} \left[(p^2 - 0.5)x(1) + (2p^2 + p + 1.5) \right]$.

El valor óptimo de J_1 es

$$J_1^+(x(1), p) = \left\{ - \left[(p^2 - 0.5)x(1) + (2p^2 + p + 1.5) \right] + (p^2 + p)x(1) + (2 + p + 3p^2) \right\}.$$

Por substitución, el valor óptimo de J_0 es:

$$\begin{aligned} J_0^+(x(0), p) &= \min_u \left\{ (1+p)x(0)u + 0.5x(0) + (p+0.5p^2)u + p + \left\{ - \left[(p^2 - 0.5)\text{Sign} \left((p+1)x(0)u + \right. \right. \right. \right. \\ &\quad \left. \left. \left. px(0) + 3u + p^2 - 1 \right) + 2p^2 + p + 1.5 \right] \right\} + (p^2 + p) \\ &\quad \left. \text{Sign} \left[(p+1)x(0)u + px(0) + 3u + p^2 - 1 \right] + 3p^2 + p + 2 \right\}, \end{aligned}$$

Así, el control óptimo paramétrico es

$$\begin{aligned} u_0^+(x(0), p) &= \text{Sign} \left[-2(1+p)x(0) - 2(p+0.5p^2) + (p^2 + p) \left\{ \text{Sign} \left[-x(0) + p^2 - 4 \right] - \text{Sign} \left[(2p+1)x + p^2 + 2 \right] \right\} + \right. \\ &\quad \left. + \left[(p^2 - 0.5)\text{Sign} \left[(2p+1)x(0) + p^2 + 2 \right] + 2p^2 + p + 1.5 \right] - \left[(p^2 - 0.5)\text{Sign} \left[-x(0) + p^2 - 4 \right] + 2p^2 + p + 1.5 \right] \right]. \end{aligned}$$

Entonces, para $x^*(0)$ dado

$$u^*(0) = u_0^+(x^*(0), p), \quad x^*(1) = \text{Sign} \left[(p+1)x(0)u^*(0) + px(0) + 3u^*(0) + p^2 - 1 \right],$$

$$J_1^*(x^*(1), p) = J_1^+(x^*(1), p),$$

$$u_1^*(x^*(1), p) = -\text{Sign} \left[(p^2 - 0.5)x^*(1) + (2p^2 + p + 1.5) \right]$$

$$x^*(2) = \begin{pmatrix} \text{Sign}((0.5p^2 + p - 0.5)x^*(1)u^*(1) + \\ + (p^2 + 2p - 1)x^*(1) \\ + (p^2 + 2p + 2)u^*(1) + 2p^2 + 4p + 4) \\ \text{Sign}((0.5p^2)x^*(1)u^*(1) + (0.25p^2)x^*(1) + \\ + (p^2 + p + 2)u^*(1) + 0.5p^2 + 0.5p + 1) \end{pmatrix}$$

$$J_2^*(x^*(2), p) = G(x^*(1), x^*(2), p).$$

El control óptimo y la trayectoria óptima para $p=0$ es como sigue

- Si $x^*(1) = \pm 1$, entonces $u^*(1)_0 = -1$.
- Si $x_1^*(0) = -1$, entonces $u^*(0)_0 = 1$, y si $x_1^*(0) = 1$, entonces $u^*(0)_0 = -1$.

El control óptimo y la trayectoria óptima para $p=5$ es como sigue

- Si $x^*(1) = \pm 1$, entonces $u^*(1)_5 = -1$.
- Si $x^*(0) = \pm 1$, entonces $u^*(0)_5 = -1$.

Las representación gráfica para $p=0$ y $p=5$ en las Figuras 8 y 9 respectivamente.

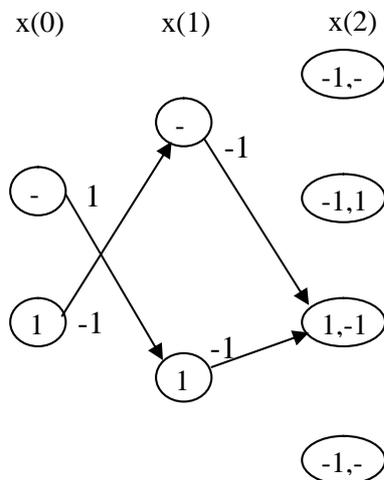


Figura 4.8: Grafo con $p=0$

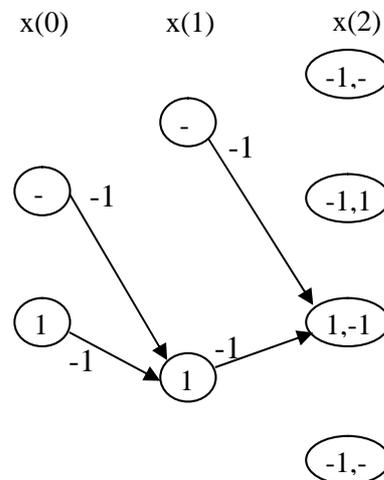


Figura 4.9: Grafo con $p=5$

Es claro que la estructura del grafo puede cambiar con cambio en el parámetro p , en cual es una gran dificultad en los procedimientos clásicos.

4.6. Conclusiones

- La combinación de la programación dinámica con cálculo formal es un enfoque prometedor en la caso de sistemas dinámicos conducidos por parámetros exógenos de evolución. El hecho de que pocos autores han explorado este enfoque es probablemente debido a la complejidad combinatoria de la programación dinámica, junto con la dificultad para proporcionar las expresiones explícitas para el control óptimo y para la función valuada. Sin embargo, estas limitaciones se pueden superar para algunas clases de los sistemas discretos y sistema a eventos discretos. Adicionalmente presenta una sustancial ventaja sobre el procedimiento clásico.
- En detalle se ha mostrado, que para los sistemas de decisión multi-etapa, el costo óptimo y los controles óptimos se pueden obtener en forma cerrada simbólica, permitiendo ser fácilmente implantables en muchas situaciones en las cuales se consideran parámetros exógenos.

Capítulo 4

PROGRAMACION DINAMICA PARAMETRICA.....	1
4.1. LA ECUACIÓN DE BELLMAN Y LA FUNCIÓN VALUADA	1
4.2 PROGRAMACIÓN DINÁMICA DISCRETA	4
4.3. UN ENFOQUE SIMBÓLICO	4
4.3.1. <i>Modelo de optimización dinámica</i>	5
4.3.2. <i>Optimización simbólica en el paso k</i>	7
4.4. PROGRAMACIÓN DINÁMICA PARAMÉTRICA	11
4.4.1. <i>Algoritmo programación dinámica paramétrica (PDPA)</i>	11
4.5. EJEMPLO.....	12
4.6. CONCLUSIONES	14

Capítulo 4

FIGURA 4.1: TRAYECTORIAS DESCRITAS POR UNA SECUENCIA DE ESTADOS DISCRETOS EN ETAPAS.....	5
FIGURA 4.2: GRAFO PARA EL CASO DE UNA SIMPLE DECISIÓN BINARIA.....	7
FIGURA 4.3: GRAFO DEL CASO GENERAL	8
FIGURA 4.4: DESCOMPOSICIÓN EN ESCALERA	8
FIGURA 4.5: EJEMPLO ILUSTRATIVO	9
FIGURA 4.6: GRAFO EN ESCALERA.....	9
FIGURA 4.7: GRAFO DE TRANSICIONES MODIFICADO.	11
FIGURA 4.8: GRAFO CON P=0	14
FIGURA 4.9: GRAFO CON P=5	14

Capítulo 4

TABLA 4.1: TABLA DE TRANSICIÓN PARA EL ESTADO x_{ki}	7
TABLA 4.2: TABLA DE TRANSICIÓN DESDE EL ESTADO x_{k1}	8
TABLA 4.3: TABLA DE TRANSICIÓN DESDE EL ESTADO \tilde{x}_{k1}	9
TABLA 4.4: TABLA DE TRANSICIÓN DEL ESTADO x_k , DEL GRAFO DE LA FIGURA 4.3	10
TABLA 4.5: DECISIONES BINARIAS EXTENDIDAS	11

CONCLUSIONES GENERALES

El desarrollo de esta tesis se centra en la optimización de Sistemas Dinámicos a Eventos Discretos sobre dominios finitos. Para tal fin, se abordaron los dos principios básicos de la optimización: “El Principio de Mínimo” y “la Programación Dinámica”. Sobre cada uno de ellos, una extensión basado en el cálculo formal se realizó considerando dominios de definición finitos y parámetros exógenos de evolución. El resultado de estas extensiones es la secuencia explícita de controles óptimos en función del estado y parámetros exógenos de evolución.

Para ambas extensiones modelos formales del proceso y de la función de costos se requieren, para ello, en esta tesis se muestra como sobre dos familias particulares de Sistema a Eventos Discretos: “Lenguajes formales” y “Autómatas de estado finito”, obtenemos modelos formales. En principio, las técnicas mostradas aquí para obtener modelos formales, pueden aplicarse sobre las representaciones naturales ya conocidas de Sistemas Dinámicos a Eventos Discretos. Así, sobre lenguajes formales, el modelo formal se obtuvo a través de sistemas dinámicos inducidos por concatenación. Además, se muestran como obtener una realización mínima del modelo. Sobre los procesos descritos por autómatas de estado finito, el modelo formal se dio a través de una descripción vía interpolación (interpolación de Lagrange), sobre la representación del autómata (conexión de nodos por arcos), por lo que un sistema dinámico con lado derecho polinómico se obtuvo.

La extensión que realizamos al Principio de Mínimo es para procesos a eventos discretos sobre dominios finitos, para ello el gradiente es sustituido por el residuo de Taylor formal algebraico, de tal manera que nuestra extensión se reformula de forma análoga como el Principio de Mínimo. La extensión no cuenta con ningún criterio de convexidad pero, esto conlleva a que el control variado aparezca en la ecuación dual. El resultado de la extensión la presentamos en la forma de un teorema (Teorema 3.5) en donde la condición de optimalidad (condición necesaria), es una desigualdad variacional en términos del control óptimo $u^*(.)$, el control variado v y el estado aparece como parámetro. Como hecho curioso para el proceso a eventos discretos y para la ecuación dual el formalismo Hamiltoniano se cumple como en la física.

La solución de la desigualdad variacional anterior conduce al valor de control óptimo requerido, pero como es sabido no existe un único método para resolver tal desigualdad y en nuestro caso como la desigualdad está parametrizada por el estado no se conocen métodos que permitan obtener una solución, sin antes realizar una evaluación del parámetros. Es por ello que desarrollamos un algoritmo que permite obtener de forma simbólica (paramétrica) los mínimos de una función cuyo dominio de definición es el hypercubo con vértices $\{-1,1\}^n$. A este algoritmo lo hemos denominado SCDO. Así, para la resolución de la desigualdad variacional parametrizada en el estado, construimos un procedimiento para calcular un mínimo sobre el control variado y un máximo sobre el control óptimo de tal manera que el resultado es la expresión simbólica del control óptimo en función del estado. A este procedimiento lo hemos denominado Min-Max. Para el cálculo del máximo y el mínimo se debe realizar una transformación sobre el dominio de definición del control variado y del control óptimo (como por ejemplo la transformación diádica) con el fin de utilizar el algoritmo SCDO.

La extensión realizada en el método de programación dinámica, consistió en utilizar en cada etapa del proceso de optimización definido sobre conjuntos finitos, el algoritmo SCDO, para obtener la expresión explícita del control óptimo en cada etapa.

Debido a la naturaleza simbólica del algoritmo SCDO, es posible incorporar parámetros exógenos de evolución tanto en la definición del proceso a eventos discretos como a la función de costo.

Como la mayoría de los problemas de optimización discretos, no escapan al uso de métodos de optimización combinatoria, que en principio son sensibles a la complejidad computacional. En esta tesis, como tópico adicional retomamos la idea del uso del método relajación sobre dominios finitos conducente a la transformación de problema discretos en la formulación clásica, la cual provee una vía para resolver el problema de optimización. Así, aquí para una familia particular de procesos discretos como lo son los procesos lineales en el estado equipados con funciones de costo cóncavas o convexas, obtenemos una condición de equivalencia entre el óptimo relajado y el óptimo del problema original en la forma de principio de mínimo. Para el caso con funciones de costo lineal la condición que obtuvimos es una condición de principio de mínimo clásica.

En cuanto a las perspectivas podemos decir que un campo que explorar es ampliar a otras clases de familias de sistemas a eventos discretos las técnicas de optimización paramétrica aquí desarrolladas. Una posible aplicación directa es que estas técnicas sean usadas en los procesos de ayuda a la toma de decisiones en sistemas de control supervisorio. Otro campo abierto, es la extensión del método de relajación a una amplia gamma de procesos y funciones de costo.

Conclusiones

CONCLUSIONES GENERALES..... 1

ANEXO A

Fundamentos sobre Leguajes Formales y Realización Mínima

Definiciones

Alfabeto: Conjunto finito de letras o caracteres que pertenecen a un espacio particular $\Omega = \{a, b, c, \dots\}$.

Palabra: Se define como una secuencia finita de letras o caracteres $w_{i1} w_{i2} \dots w_{ik}$, pertenecientes a un alfabeto.

Palabra vacía: Se define como aquella palabra que no contiene ninguna letra y se representa como θ .

Ω^* : Se define como el conjunto de todas las palabras de longitud finita incluida la palabra vacía, $\Omega^* = \{w_{i1} w_{i2} \dots w_{ik} \in \Omega, \theta\}$.

Lenguajes: constituye el conjunto no vacío de palabras de Ω^* , entonces L es un lenguaje del alfabeto Ω , si $L \subset \Omega^*$. Por ejemplo, $L = \{p, xy, yx, \dots\} \subset \Omega^*$.

Operaciones:

Concatenación: Es la unión o enlace de palabras de un lenguaje, $\Omega^* \times \Omega^* \rightarrow \Omega^*$. Por ejemplo suponga que $u, v \in \Omega^*$ con $u = u_1 u_2 \dots u_k$, $v = v_1 v_2 \dots v_j$, se define $uv = u_1 u_2 \dots u_k v_1 v_2 \dots v_j$.

Asociatividad: Es la propiedad de agrupar palabras sin alterar el resultado. Por ejemplo suponga que $u, v, w \in \Omega^*$ con $u = u_1 u_2 \dots u_k$, $v = v_1 v_2 \dots v_j$, $w = w_1 w_2 \dots w_l$ se define $(uv)w = u(vw)$.

Conmutatividad: Se define cuando el orden de las palabras no afecta el significado y solo se cumple entre palabras específicas. Por ejemplo, la palabra vacía conmuta con cualquier palabra, así, si $u = u_1 \dots u_k \in \Omega^*$, entonces $u\theta = \theta u$, aquí θ se comporta como unidad o elemento neutro. Generalmente $uv \neq vu$, no cumple la conmutatividad, como caso particular si $u=v$, entonces si se cumple.

Monoide: Es el conjunto Ω^* de todas las palabras formadas de las letras de Ω , respecto a la concatenación y el elemento neutro θ . Definido como $(\Omega^*, \text{concatenación}, \theta)$.

Propiedades del Alfabeto

Las propiedades del alfabeto se definen a partir del proceso P, siendo éste el que transforma a todas las palabras al mismo conjunto de salida Y. Esto es, $P(u) = y \in Y$. Por ejemplo, sea, si $P: \Omega^* \rightarrow Y$, se define por $P_o(u) = y_o \forall u$. Las palabras $u, v \in \Omega^*$, cumplen que $P_o(uw) = P_o(vw) \forall w$.

Distinguibilidad: Sean $u, v \in \Omega^*$ dos palabras. Si existe al menos un $w \in \Omega^*$, tal que $p(uw) \neq p(vw)$, entonces se dice que u, v son distinguibles.

Indistinguibilidad: Si se tienen dos palabras u y v tal que $\forall w \in \Omega^*$, $p(uw) = p(vw)$, entonces u y v son indistinguibles y se denota uNv .

Simetría: Si se tiene dos palabras tal que $uNv \Rightarrow vNu$, entonces se dice que u y v es simétrica.

Transitividad: Si se tienen tres palabras u, v y z tal que $uNv, vNz \Rightarrow uNz$.

De o anterior se deduce que uNv , es una relación de equivalencia denominada la equivalencia de Nerode.

Clase de Equivalencia

Se define al conjunto de las palabras que son indistinguibles de $u \in \Omega^*$, denotándose como $[u] \subset \Omega^*$ y se dice que es la clase definida por la palabra u . Así $v \in [u] \Leftrightarrow uNv, [u] \exists u$. Por lo tanto debe cumplirse los siguientes casos:

$$\begin{aligned} [u] &= [v] \\ [u] \cap [v] &= \theta. \end{aligned}$$

Espacio de Estado: Se dice que X , el conjunto de las clases de equivalencia es el espacio de estado y está constituido por el conjunto de todas las clases de equivalencia de la relación N , como sigue $X = \{ x: x=[u]: u \in \Omega^* \}$.

El sistema dinámico inducido por la concatenación:

Se define

$$\begin{aligned} \phi: X \times \Omega^* &\rightarrow X, \\ x \in X, \phi(x, v) &= [uv], \end{aligned}$$

donde $x = [u]$.

Se define $[u]$ como $[u] = \{u, u^*, u^{**}, \dots\}$, que significa el conjunto de todos los u que definen la misma clase. Siendo:

1. $\phi(x, v) = [uv]$,
2. $\phi(x, v) = [u^*v]$.

Si se logra comprobar que 1 y 2 son iguales desaparece la ambigüedad. De lo contrario, la función está mal definida.

Si u y u^* son equivalentes entonces uNu^* . ¿Serán entonces uv, u^*v indistinguibles?, para ello hay que concatenar por un w y comprobar que se cumple la siguiente relación:

$$p((uv)w) = p((u^*v)w) \quad \forall w \Rightarrow uvNu^*v.$$

Si desarrollamos la relación se tiene que $p((uv)w) = p(u(vw)) = p(u^*(vw)) = p((u^*v)w)$, por lo que $uvNu^*v$ se cumple.

Propiedades de la dinámica

- **Identidad**, Si $\phi(x, \theta) = \phi([u], \theta) = [u\theta] = [u] = x$, así $\phi(., \theta) = id_x$.
 - $\phi(x, vw) = \phi([u], vw) = [u(vw)] = [(uv)w] = \phi([uv], w) = \phi(\phi(x, v)w)$.
- Si un sistema cumple estas dos propiedades se dice que es un sistema dinámico.

Transformación de estado inducida

$$\phi_u : X \rightarrow X, \phi_u(x) = \phi(x, u)$$

Función de salida

Se define como $\sigma : X \rightarrow Y, \sigma(x) = \sigma([u]) = p(u)$.

Si se cumple que uNu^* entonces se debe comprobar que $\sigma(x) = \sigma([u]) = p(u) = p(u^*)$. Para esto, hay que concatenar por w , tal que: $p(uw) = p(u^*w)$. Se puede observar que

$$p(u) = p(uq) = p(u^*q) = p(u^*)$$

En la Figura A.1 se presenta un diagrama resumen de las funciones anteriores descritas

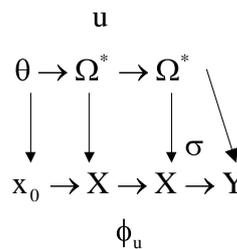


Figura A.1: Diagrama Resumen

Esto es $\phi_u : X \rightarrow X, \sigma : X \rightarrow Y, u \in \Omega^*, x_0 = [\theta]$. Si consideramos, una relación entrada salida P entre el alfabeto y la función Y como muestra la Figura A.2.

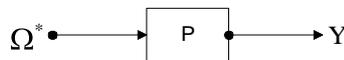


Figura A.2 :Función salida

Ahora mostramos la correspondencia entre la caja negra de la Figura A.2 y en conjunto de ecuaciones mostrado en la Figura 1. Calculemos la correspondencia,

$$u \rightarrow \sigma(\phi(x_0, u)) \in Y$$

$$\sigma(\phi(x_0, u)) = \sigma(\phi([\theta], u)) = \sigma([\theta u]) = \sigma([u]) = p(u),$$

hemos obtenido una representación de la caja negra en espacio de estado, es decir se ha obtenido una **realización** de la aplicación entrada salida. En la Figura A.3 se representa gráficamente lo anterior.

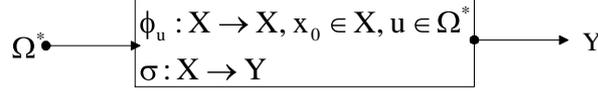


Figura A.3: Sistema dinámico controlado por u

Así, $(X, \phi, x_0 = [\theta], \sigma, Y)$ es el sistema dinámico que realiza p.

Recordemos que esta realización debe contener como propiedades la accesibilidad y la observabilidad, por lo que ahora mostramos que esta nueva realización posee estas propiedades.

- **Accesibilidad:** Considere x_0^2 -accesible. Se toma $x^2 \in X_1$, arbitrario, tal que $x^2 = [x^1] \exists u \Rightarrow x^1 = \phi_1(x_0^1, u)$. Calculando

$$\phi_2(x_0^2, u) = \phi_2([x_0^1], u) = [\phi_1(x_0^1, u)] = [x^1] = x^2$$

Por lo tanto es accesible.

- **Observabilidad:** Si $x^2 \neq \bar{x}^2 \Rightarrow$ son distinguibles, Entonces hay que probar que $\exists u$, tal que:
 1. $L(x^2) \neq L(\bar{x}^2)$. Si $x^2 = [x^1]$, $\bar{x}^2 = [\bar{x}^1]$ x^1 y \bar{x}^1 , son distinguibles. Entonces $L(x^1) \neq L(\bar{x}^1) \Leftrightarrow L(x^2) \neq L(\bar{x}^2)$.
 2. Existe un u tal que $\sigma_2(\phi_2(x^2, u)) = \sigma_2(\phi_2(\bar{x}^1, u))$. Ya se habia señalado que $L(x^1) \neq L(\bar{x}^1)$. No obstante se cumple que $\exists v \in \sigma_1(\phi_1(x^1, v)) \neq \sigma_1(\phi_1(\bar{x}^1, v))$. Así $\sigma_1(\phi_1(x^1, v)) = \sigma_2[\phi_1(x^1, v)] = \sigma_2(\phi_2([x^1], v)) = \sigma_2(\phi_2(x^2, v))$, por otro lado $\sigma_1(\phi_1(\bar{x}^1, v)) \neq \sigma_1(\phi_1(\bar{x}^2, v))$, lo que implica que $\sigma_2(\phi_2(x^2, v)) \neq \sigma_2(\phi_2(\bar{x}^2, v))$. Por lo tanto el sistema es observable.

Realización mínima

Para obtener la realización mínima de sistema, se mostrarán dos procedimientos basados en la siguiente definición.

Definición: Sean, $(X_1, x_0^1, \phi_1, \sigma_1)$, $(X_2, x_0^2, \phi_2, \sigma_2)$ dos sistemas dinámicos, sobre el mismo L y con el mismo espacio de salida Y. Si $f : X_1 \rightarrow X_2$ es una alicación que cumple con las siguientes propiedades:

1. $f(x_0^1) = x_0^2$,
2. Sea $x \in X_1$; $L(x) = L(f(x))$, $u \in L(x)$, entonces $f(\phi_{1u}(x)) = \phi_{2u}(f(x))$,
3. $\sigma_1(x) = \sigma_2(f(x))$, entonces se dice, que f preserva la dinámica. Si f preserva la dinámica, entonces los dos sistemas realizan el mismo sistema de entrada salida:

$$p_1(u) = \sigma_1(\phi_{1u}(x_0^1)) = \sigma_2(f(\phi_{1u}(x_0^1))) = \sigma_2(\phi_{2u}(f(x_0^1))) = \sigma_2(\phi_{2u}(x_0^2)) = p_2(u).$$

Procedimiento 1

Asuma que se tienen los siguientes tres sistemas dinámicos: $(\tilde{X}, \tilde{x}_0, \tilde{\phi}, \tilde{\sigma})$, $(X_1, x_0^1, \phi_1, \sigma_1)$, $(X_2, x_0^2, \phi_2, \sigma_2)$, donde se cumple que $\tilde{\phi}_u(x) = \phi(x, u)$ y $\phi_{1u}(x) = \phi_1(x, u)$.

Sobre los tres sistemas dinámicos anteriores definimos dos tipos de funciones de costo.

Función de costo tipo 1

Sea j una aplicación que preserva la dinámica entre $(X_1, x_0^1, \phi_1, \sigma_1)$, $(\tilde{X}, \tilde{x}_0, \tilde{\phi}, \tilde{\sigma})$, sea $j: X_1 \rightarrow \tilde{X}$ inmersión, es decir, si $x, \bar{x} \in X_1$, $x \neq \bar{x} \Rightarrow j(x) \neq j(\bar{x})$.

Ejemplo: Si $X_1 \subset X$, entonces $x \in X_1 \subset \tilde{X}$, $x = j(x) \in \tilde{X}$, se tiene $j(\phi_1(x, u)) = \tilde{\phi}(j(x), u)$, $x_0^1 = \tilde{x}_0^1 = j(x_0^1)$. Como se puede observar se tiene una realización mínima.

Función de costo tipo 2

Sea g una aplicación que preserva la dinámica entre $(\tilde{X}, \tilde{x}_0, \tilde{\phi}, \tilde{\sigma})$, $(X_2, x_0^2, \phi_2, \sigma_2)$, además $g: \tilde{X} \rightarrow X_2$ sobreyectiva, es decir si $x^2 \in X_2$ arbitraria, entonces existe $x \in \tilde{X}$ con $g(x) = x^2$. Por lo tanto $g(x) = [x]$.

Ejemplo: Como se sabe $\phi_2([x], u) = [\tilde{\phi}(x, u)]$, se tiene $\phi_2(g(x), u) = g(\tilde{\phi}(x, u))$.

Propiedades de g

1. Preserva la dinámica.
2. Preserva la unidad, es decir el estado inicial $x_0^2 = [x_0] = g(\tilde{x}_0)$.
3. Preserva la salida, es decir $\tilde{\sigma}(x) = \sigma_2(g(x))$.

Se dice que $(\tilde{X}, \tilde{x}_0, \tilde{\phi}, \tilde{\sigma})$, es una realización (sistema dinámico) mínima si para cada inmersión j y sobreyección g , que preserva la dinámica, las dos son biyectivas.

Teorema 1: Si $P: L \rightarrow Y$, se tiene dos realizaciones mínimas entonces ellas son isomorfas.

Prueba: Si se tiene $(X_1, x_0^1, \phi_1, \sigma_1)$, $(X_2, x_0^2, \phi_2, \sigma_2)$. Se define una función $f: X_1 \rightarrow X_2$, que es biyectiva de tal manera que todo conmuta.

Teorema 2: La realización $(\tilde{X}, \tilde{x}_0, \tilde{\phi}, \tilde{\sigma})$ es mínima si y solo si x_0 es accesible y observable. Esto indica que la minimalidad implica accesibilidad y observabilidad.

Prueba: Considerando

$$X_1 = \{x : \text{accesible}\},$$

$$X_2 = \{[x] : x \in \tilde{X}\}, \text{ conjunto de las clases indistinguibles,}$$

$$j: X_1 \rightarrow \tilde{X}, j(x) = x. \text{ Inyectiva,}$$

$$g: \tilde{X} \rightarrow X_2, g(x) = [x]. \text{ Sobreyectiva.}$$

Hay que probar que j y g son isomorfismos. Se sabe que \tilde{X} es accesible lo que significa que para todo $x \in \tilde{X}$ existe un u tal que $x = \phi(\tilde{x}_0, u) = \tilde{\phi}(j(x_0^1), u) = j(\phi_1(x_0^1, u)) = j(x)$. La ecuación nos sugiere que para cada \tilde{X} hay una imagen en X por lo tanto j es biyectiva.

Por otro lado tenemos que $x, \bar{x} \in \tilde{X}$, lo que implica que $L(x) = L(\bar{x})$ ó $L(x) \neq L(\bar{x})$. Entonces $\exists v \in L(x) = L(\bar{x})$, con $\sigma(\phi(x, v)) \neq \tilde{\sigma}(\tilde{\phi}(\tilde{x}, v))$, esto implica que $L(x) = L(\bar{x})$ ó $L(x) \neq L(\bar{x})$. Vemos que :

$$\sigma_2(\phi_2(g(x), u)) = \sigma_2(g(\phi_2(x, u))) = \tilde{\sigma}(\tilde{\phi}(x, v)) \neq \sigma_2(\phi_2(g(\bar{x}), u)) = \sigma_2(g(\phi_2(\bar{x}, u))) = \tilde{\sigma}(\tilde{\phi}(\bar{x}, v))$$

lo que nos dice que $g(x) \neq g(\bar{x})$.

Procedimiento 2

Acá se mostrará como redefiniendo el espacio de estado podemos obtener un modelo matemático del proceso lingüístico.

Un nuevo espacio de estado

Este nuevo espacio de estado está constituido por la longitud de u . Sea $u = u_0 u_1 u_2 \dots u_{k-1}$ y su longitud como Longitud de $u = |u| = k$, con $\theta \Leftrightarrow x_0 = [\theta]$. Se define:

$$\begin{aligned} X_0 &= \{x_0\}, \text{ todas las clases de longitud } 0, \text{ es decir, la palabra vacía.} \\ X_1 &= \{[u] : |u| = 1\}, \text{ todas las clases de las palabras de longitud } 1. \\ &\vdots \\ X_k &= \{[u] : |u| = k\}, \text{ todas las clases de las palabras de longitud } k. \end{aligned}$$

Una nueva dinámica

Consideremos $x \in X_k$, $\phi(x, w) = \phi([u], w) = [uw] \in X_{k+1}$. Si se tiene una palabra v de longitud k como v , $|v| = k$ y $x \in X_k$, $|u| = K$, entonces $\phi(x, v) = \phi([u], v) = [uv] \in X_{k+K}$. Como se ve con ϕ se pueden definir muchas aplicaciones de transición.

Si se toma: $\phi : X_K \times \Omega^* \rightarrow X_1$, como una dinámica global, donde X_1 depende de la longitud de la palabra. Entonces se define una dinámica local que corresponde a las ecuaciones en diferencia como $f_K : X_K \times \Omega \rightarrow X_{k+1}$ cuyo dominio es $\text{Dom } f_K : x \in X_K$, con $\Omega(x) =$ las letras accesibles. Si se tiene $w \in \Omega(x)$, entonces ϕ esta definida como $f_K(x, w) = \phi(x, w)$. Si se tiene un estado accesible, entonces existe $u = u_0 u_1 u_2 \dots u_{k-1}$ tal que $\phi(x_0, u) = x$, con lo que se obtiene:

$$\begin{aligned} X_1 &= f_0(x_0, u_0) = [u_0] \in X_1, \\ X_2 &= f_1(x_1, u_1) = f_1([u_0], u_1) = [u_1 u_0] \in X_2 \\ &\vdots \\ X_k &= f_{k-1}(x_k, u_k) = f_k([u_0 u_1 \dots u_{k-2}], u_{k-1}) = [u_0 u_1 \dots u_{k-1}] \in X_k \end{aligned}$$

Esta construcción nunca termina porque cada letra es accesible a la palabra anterior, además tiene similitud con un control discreto, con la diferencia que esto es un modelo lingüístico.

Una nueva función de costo

Consideremos $\Psi(x_k) \rightarrow$ mínima, definida como $\Psi : X_k \rightarrow \mathfrak{R}, \mathfrak{N}, \mathfrak{Z}$, función sobre un conjunto finito de los reales o los enteros.

Se presentará la minimización de funciones vectoriales como $\Psi = (\Psi_1, \Psi_2, \dots, \Psi_n)$. Para ello es necesario definir una relación de orden .

Relación de orden

Consideremos la siguiente relación de orden ($x \leq$), cuyas propiedades son :

1. $x, y \in X$, arbitrario con $x \leq y$ ó $y \leq x$.
2. Si $x \leq y$ y $y \leq x$ entonces $x=y$. Antisimétrica.
3. Si $x \leq y$ y $y \leq z$ entonces $x \leq z$. Transitiva.
4. Relación de orden lineal para los elementos positivos $p = \{x : 0 \leq x\}$. La relación de orden es lineal si p es un semigrupo. Así, $x + y \leq y + z$ si y solo si $x \leq y$.

La relación de orden entre vectores, que poseen notación $(x_1, x_2, \dots, x_n)^* \in \mathfrak{R}^n$, es de dos formas :

- **Relación de orden parcial (\leq) :**

$$\begin{aligned} x_1 &\leq y_1 \\ x_2 &\leq y_2 \\ &\vdots \\ x_n &\leq y_n \end{aligned}$$

- **Relación de orden completa (\leq):**

$$\begin{aligned} x_1 &< y_1 \text{ ó} \\ x_1 &= y_1, x_2 < y_2 \text{ ó} \\ x_1 &= y_1, x_2 = y_2, x_3 < y_3 \text{ ó} \\ &\vdots \\ x_1 &= y_1, \dots, x_{n-1} = y_{n-1}, x_n \leq y_n \end{aligned}$$

Este es conocido como orden Lexicográfico.

Para que $\Psi : D \rightarrow \mathfrak{R}^n$, $x_0 \in D$, alcance un mínimo se necesita una relación de orden. Esto es $(\mathfrak{R}^n, \leq) \forall x \in D$, tal que $\Psi(x), \Psi(x_0)$ son comparables, entonces $\Psi(x) \leq \Psi(x_0)$.

ANEXO A

FUNDAMENTOS SOBRE LEGUAJES FORMALES Y REALIZACIÓN MÍNIMA.....	1
DEFINICIONES	1
OPERACIONES:	1
PROPIEDADES DEL ALFABETO	1
CLASE DE EQUIVALENCIA	2
EL SISTEMA DINÁMICO INDUCIDO POR LA CONCATENACIÓN:	2
<i>Propiedades de la dinámica</i>	3
<i>Transformación de estado inducida</i>	3
<i>Función de salida</i>	3
REALIZACIÓN MÍNIMA.....	4
<i>Procedimiento 1</i>	5
<i>Procedimiento 2</i>	6

REFERENCIAS

- [AB00] Amir Beck and Marc Teboulle, Global Optimality Condition for Quadratic Optimization Problem with Binari Constraints, *SIAM Journal Optimization*, vol. 11, n° 1, pp. 179-188, 2000.
- [AT51] A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. University of California Press, Berkeley, 1951. second ed., rev. Reprinted in [CJ98].
- [AT95] Antonio Tornambé, “Discrete-Event System Theory, An Introduction ”, Copiright@1995,World Scientific Publishers Co. Pte Ltd, ISBN 9810223617.
- [AS54] A. Seidenberg. A new decision method for elementary algebra. *Annals of Math.*, 60(2):-374, 1954.
- [AS00] A. Strzebonski. Solving systems of strict polynomial inequalities. *Journal of Symbolic Computation*, 29:-480, 2000.
- [BS96] [B. De Schutter](#), Max-Algebraic System Theory for Discrete Event Systems PhD thesis, Faculty of Applied Sciences, K.U.Leuven, Leuven, Belgium, Feb. 1996, ISBN 90-5682-016-8.
- [CB99] C. W. Brown. Guaranteed solution formula construction. In *Proc. International Symposium on Symbolic and Algebraic Computation*, pages 137-144, 1999.
- [CB01] C. W. Brown. Improved projection for cylindrical algebraic decomposition. Accepted for publication in the *Journal of Symbolic Computation*, April 2001.
- [CB99-1] C. W. Brown. *Solution Formula Construction for Truth Invariant CAD's*. PhD thesis, University of Delaware, 99.
- [CC95] C. G. Cassandras, S. Lafortune, and G. L. Olsder, “ Introduction to the modelling, control and optimization of discrete event system,” in *Trends in Control: A European Perspective* (A. Isidori, ed.), pp. 217-291, Berlin Germany: Springer-Verlag, 1995.
- [DK95] Dean Kelley, “Teoría de Automatas y Lenguajes Formales”, Prentice-Hall 1995, ISBN 0-13-497777-7.
- [CJ98] B.F. Caviness and J. R. Johnson, editors. *Quantifier Elimination and Cylindrical Algebraic Decomposition*. Texts and Monographs in Symbolic Computation. Springer-Verlag, 1998.
- [DG88] D. Yu. Grigor'ev. The complexity of deciding tarski algebra. *Journal of Symbolic Computation*, 5:-108, 1988.
- [DH97] J. H. Davenport and J. Heintz. Real quantifier elimination is doubly exponential. *Journal of Symbolic Computation*, 5:-35, 1997.
- [DL69] David G. Luenberger “Optimization by Vector Methods”, Copyright@1969 by John Wiley & Sons, Inc.,SBN 47155359x.
- [ES98] Eduardo Sontag, “Mathematical Control Theory”, Springer-Verlag, ISBN 0-387-98489-5, 1998.
- [FB92] F. Bocelli, G. Cohen, G. L. Olsder, and J. P. Quadrat, synchronization and linearity, New York: John Wiley & Sons, 1992.
- [FR74] M. J. Fischer and M. O. Rabin. Super-exponential complexity of Pressburger arithmetic. *Complexity of Computation (AMS-SIAM Proceedings)*, 7:-41, 1974.
- [FS97] Ferenc Szigeti, Juan Cardillo, Secuencia optima de instrucciones en Lenguajes Formales usando el principio de mínimo de Pontriaguin. *Memorias del III Coloquio de Ecuaciones Diferenciales y sus aplicaciones*, 1997.
- [FS99] F. Szigeti, G. De Sarrazin, J. Cardillo (1999) Computer aided problem solving via optimization, Risc-Linz Report series number 99-13 May 99, Hagenberg, Austria.
- [GC85] G. Cohen, D. Dubois, J. P. Quadrat, and M. Viot, “A linear-system-theoretic view of discrete-event process and its use for performance evaluation in manufacturing,” *IEEE Transactions on Automatic Control*, vol. 30, n° 3, pp. 210-220, Mar. 1985..
- [GC75] G. E. Collins. Quantifier elimination for the elementary theory of real closed fields by cylindrical algebraic decomposition. In *Lecture Notes In Computer Science*, volume Vol. 33, pages 134-183. Springer-Verlag, Berlin, 1975. Reprinted in [[Caviness and Johnson\(1998\)](#)].
- [GC94] G. Cohen, “Diodes and discrete event systems”, in proceeding of the 11th International Conference on Analysis and Optimization of Systems, Sophia-Antipolis, France, vol. 199 of *Lecture Notes in Control and Information Sciences*, pp. 223-236, Springer-Verlag, 1994.
- [GC94] G. Cohen and J. P. Quadrat, eds., Proceeding of the 11th International Conference on Analysis and Optimization of Systems, Sophia-Antipolis, France, vol. 19 of *Lecture Notes in Control and Information Sciences*, Berlin, Germany, Springer-Verlag, June 1994.

- [GS01] Gisela de Sarrazin, J. Cardillo, Ferenc Szigeti, Optimal Solution of a Computer Task, Non Linear Análisis, 47 (2001) 1549-1560.
- [HJ99] Horn and Johnson, Matrix Analysis, Cambridge University Press, 1999, ISBN 0-521-30586
- [HH90] H. Hong. An improvement of the projection operator in cylindrical algebraic decomposition. In *Proc. International Symposium on Symbolic and Algebraic Computation*, pages 261-264, 1990.
- [HH92] H. Hong. Simple solution formula construction in cylindrical algebraic decomposition based quantifier elimination. In *Proc. International Symposium on Symbolic and Algebraic Computation*, pages 177-188, 1992.
- [HH93] H. Hong. Quantifier elimination for formulas constrained by quadratic equations via slope resultants. *Computer J.*, 36(5):-449, 1993.
- [HH97] H. Hong, R. Liska, and S. Steinberg. Testing stability by quantifier elimination. *Journal of Symbolic Computation*, 24(2):-187, Aug. 1997. Special Issue on Applications of Quantifier Elimination.
- [JC94] J.J. Cardillo-Albarrán: (1994) Optimización de sistemas a Eventos discretos usando el principio de mínimo de Pontryagin, Thesis Msd, Fac. Engineering, ULA, Mérida.
- [JC95] J.J. Cardillo-Albarrán and F. Szigeti: (1995) A symbolic algorithm for the optimization of integer processes over finite state space, International Congress on Industrial and Applied Mathematics, ICIAM 95, Hamburg, Germany, 3-7, July.
- [JC97] Juan Cardillo, Edgar Chacón, Un modelo de automatización integrada orientado a objeto, IV Jornadas de Control, 17 Octubre 1997. Mérida, Venezuela.
- [JC971] Juan Cardillo, Ferenc Szigeti, Inteligencia Artificial : Enfoque Linguístico usando el Principio de Mínimo de Pontriaguin, IV Jornadas de Control. 17 de Octubre de 1997. Mérida, Venezuela.
- [JC98] Juan Cardillo, Ferenc Szigeti, Edgar Chacón, Automatización Integrada de Procesos de Producción Continua vía Optimización SDED., XI Jornadas de Matemáticas, 31 de marzo al 03 de Abril de 1998. Cumana, Venezuela.
- [JC981] Cardillo J., Chacon E., Un modelo de automatización integrada en procesos continuos, IV Jornadas Científico técnicas Facultad de Ingeniería, Universidad de Los Andes, pp 468-473, Noviembre 1998, Mérida – Venezuela.
- [JC982] Miranda Y., Cardillo J., Evaluación de metodología para automatización integrada en la primera etapa de blanqueo de sebo, IV Jornadas Científico técnicas Facultad de Ingeniería, Universidad de Los Andes, pp 316-322, Noviembre 1998, Mérida – Venezuela.
- [JC983] Juan Cardillo. Edgar Chacón, Un Modelo de Automatización Integrada de Procesos Continuos, IV Jornadas de Sistemas de Instrumentación y Control, III Jornadas Panamericanas de Automatización, Mayo 1998.
- [JC984] José Yepez, Jesús Rodríguez y Juan Cardillo, Implementación Computacional Simbólica del Método de Linealización Extendida para el diseño de Compensadores y Observadores no Lineales, IV Jornadas de Sistemas de Instrumentación y Control, III Jornadas Panamericanas de Automatización, Mayo 1998.
- [JC99] Juan Cardillo, Ferenc Szigeti, Optimización de sistemas a eventos discretos por desigualdades variacionales. I Coloquio de Automatización y Control, Noviembre 1999, Mérida-Venezuela.
- [JC991] Juan Cardillo, Ferenc Szigeti, Discrete event optimization by inequalities, COMPUTERS AND COMPUTATIONAL ENGINEERING IN CONTROL, pp.150-157, Sept. 1999.
- [JC992] J. Calderón J., E. Chacón E., L. Becerra L “Gas process automation in petroleum production”. *Proc. 3rd. International Conference on Industrial Automation* Montreal Canada, pp. 2.5-2.7., (1999).
- [JC00] Juan Cardillo, “OPTIMISATION DE SYSTEMES A EVENEMENTS DISCRETS”, DEA SYSTEMES AUTOMATIQUES, Ecole Doctorale Systemes, 1999-2000, CNRS, LAAS.
- [JC001] Cardillo J, Ferenc Szigeti, Discrete event optimisation by inequalities., IMACS/IEEE CSCC’99 International multiconference, Atenas – Grecia, del 04 al 08 de julio de 1999.
- [JC002] Juan Cardillo, Edgar Chacón, Rafeel Rivas, Coordinación / Optimización de Complejos de Producción Continua mediante el uso de Sistemas Híbridos Jerárquicos. V Jornadas de Sistemas de Instrumentación y Control, IV Jornadas Panamericanas de Automatización, Mayo 2000
- [JC003] Gisela De Sarrazin, Juan Cardillo, Ferenc Szigeti, Optimal solution task, World Congress of Nonlinear Analysts, Julio 19-26-2000, Catania Italy.
- [JC004] Juan Cardillo, Ferenc Szigeti, Symbolic Computation in Discrete Event Dynamical System Optimization, World Congress of Nonlinear Analysts, Julio 19-26-2000, Catania Italy.

- [JC005] Juan Cardillo, Ferenc Szigeti, Optimización de Sistemas a Eventos Discretos, Charla Plenaria en las I Jornadas de Investigación de Operaciones, Anzoátegui 2000, Pto. LA Cruz del 23 al 25 de Noviembre del 2000.
- [JC006] Juan Cardillo, Edgar Chacón, Rafael Rivas y Flor Narciso, Diseño de un Ambiente SCADA para la Hidrológica Aguas de Mérida, II Coloquio de Automatización y Control, CAC'00, 27 de Noviembre al 01 de Diciembre del 2000, pág 123- 129, Pto. La Cruz - Venezuela.
- [JC007] Juan Cardillo, Edgar Chacón, Rafael Rivas, Autómatas Jerárquicos Comunicantes en Complejos de Producción Continua, II Coloquio de Automatización y Control, CAC'00, 27 de Noviembre al 01 de Diciembre del 2000, pág. 135-143, Pto. La Cruz Venezuela.
- [JC01] J. Cardillo, F. Szigeti, J.C. Hennet, J-L Calvet "Symbolic Computation in Discrete Optimization", LAAS report, submitted to *Journal of Symbolic Computation* (2001).
- [JC011] Juan Cardillo, Ferenc Szigeti, Jean Claude Hennet, Jean Louis Calvet, Programación Dinámica y cálculo formal, propuesta paramétrica para sistemas a eventos discretos, III Coloquio de Automatización y Control CAC 2001.
- [JC012] Juan Cardillo, Edgar Chacón, Carlos Briceno, SIOp, Propuesta de sistemas de información operaciones para procesos de producción continua. III Coloquio de Automatización y Control CAC 2001.
- [JC02] Juan Cardillo, Ferenc Szigeti, Jean Louis Calvet, Jean Claude Hennet, Parametric Dynamic Programming for discrete Event System, Submitted in IEEE Transaction on SMC, part A.
- [JC021] Juan Cardillo, Ferenc Szigeti, J-L Calvet, J-C Hennet, Parametric Dynamic Programming for discrete event systems, 10th IEEE Mediterranean Conference on Control and Automation, MED 2002.
- [JC03] Juan Cardillo, Ferenc Szigeti, Jean Claude Hennet, Jean Louis Calvet, Modélisation algébrique, pour l'optimisation de systèmes a événements discrets, 4^e Conférence Francophone de Modélisation et SIMulation, Organisation et Conduite d'Activités dans l'Industrie et les Services, MOSIM'03. Toulouse -France Avril 2003.
- [JC031] F. Szigeti, J. Cardillo, J. C. Hennet y J. L. Calvet, El Método de Relajación Aplicado a Sistemas Discretos, IV Coloquio de Ecuaciones Diferenciales y Aplicaciones, Maracaibo, Noviembre 2003
- [JC032] Juan Cardillo, Ferenc Szigeti, Jean-Louis Calvet, Jean-Claude Hennet, ALGORTIMO SCDO EN EL PRINCIPIO DE MINIMO Y LA PROGRAMACION DINAMICA DEFINIDA SOBRE DOMINIOS FINITOS, IV Coloquio de Automatización y Control, CAC'03, Mérida Venezuela, Noviembre 2003.
- [JC033] Ing. María Alejandra SANGUINETTI_AVENDAÑO, Prof. Juan CARDILLO ESTUDIO E IMPLEMENTACIÓN DE MODELOS DE SIMULACIÓN AL SISTEMA DE ENTRENAMIENTO DE OPERADORES (OTS) DEL SISTEMA DE CONTROL POR COMPUTADOR GURI (SCCG), IV Coloquio de Automatización y Control, CAC'03, Mérida Venezuela, Noviembre 2003.
- [JG89] J.L. Gimenez "Contribution a la decomposition de systemes interconnectes par programmation dynamique non serielle. Application a des systemes de puissance". *Doctorate thesis*, Toulouse, France (1989).
- [JH90] J. Heintz, M. Roy, and P. Solernó. Sur la complexité du principe de Tarski-Seidenberg. *Bull. Soc. Math. France*, 118:-126, 1990.
- [JL01] [J.B.LASSERRE](#), An explicit exact SDP relaxation for nonlinear 0-1 programs, Rapport LAAS No00475 8th International Conference on Integer Programming and Combinatorial Optimization (IPCO'VIII), Utrecht (Pays-Bas), 13-15 Juin 2001, Lecture Notes in Computer Sciences 2081, Eds. K.Aardal, B.Gerards, 2001, Springer, ISBN 3-540-42225-0, pp. 293-303
- [JL03] [J.B.LASSERRE](#) , [T.PRIETO RUMEAU](#) SDP VS. LP Relaxations for the moment approach in some performance evaluation problems, Rapport LAAS N°03125, Mars 2003, 16p.
- [JM98] John. O. Moody, Panos J. Antsaklis, "Supervisory Control of Discrete Event Systems Using", Lumer Academic Publishers, Copyright@1998 by Klumer Academic Publishers.
- [JP77] J. L. Peterson, "Petri Nets," Computing Surveys, vol. 9, n° 3, pp.223-252, Sept. 1977.
- [JP81] J. L. Peterson, Petri Net Theory and the Modeling of Systems. Englewood Cliffs, New Jersey: Prentice-Hall, 1981.
- [JR92] J. Renegar. On the computational complexity and geometry of the first-order theory of the reals, parts I-III. *Journal of Symbolic Computation*, 13:-352, 1992.

- [JW62] J. Warga, Relaxed Variational Problem, *Journal of Mathematical Analysis and Applications*, 4, 111-128, 1962.
- [JW162] J. Warga, Necessary Condition for Minimum in Relaxed Variational Problem, *Journal of Mathematical Analysis and Applications* 4, 129-145, 1962.
- [JZ92] Jerzy Zabczyk, *Mathematical Control Theory, An Introduction*, Birkhäuser Boston, ISBN 0-8176-3645-5, 1992.
- [KM84] K.V. Mital, "Métodos de Optimización", Editorial LIMUSA, ISBN 968-16145, 1984.
- [KP98] Kevin M. Passino, Kevin L. Burgess, "Stability Analysis of Discrete Event Systems", Copyright@1998 by John Wiley and Sons. Inc., ISBN 0-471-24185-7.
- [LB95] L. Ben-Naoum, R. Boel, L. Bongaerts, B. de Shutter, Y. Peng, P. Valckenaers, J. Vandewalle, and V. Wertz, "Methodologies form discrete event dynamical systems: A survey", *Journal A*, vol. 36, n° 4, pp. 3-14, Dec. 1995.
- [LS76] L. E. Sigler *Algebra*, Springer Verlag, 1976 ISBN 0-387-90195-7.
- [MB99] M. Boulehmi M. "Mise en oeuvre et évaluation d'algorithmes d'optimisation et de commande optimale sur un système de calcul formel". *Doctorate thesis*, (1999).
- [MG79] Michael R. Garey, David S. Johnson, "Computer and Intractability: A Guide to the Theory of NP Completeness", Copyright@1979 Bell Telephone Laboratories, Incorporated, W.H. Freeman and Company.
- [ML93] M. Le Borgue, B. Dutertre, A. Benveniste, and P. Le Guernic, "Dynamical Systems over Galois Fields", in *Proceeding of the 2nd European Control Conference*, Groningen, The Netherlands, pp.2181-284, June 1993.
- [MZ91] M. C. Zhon and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Boston: Kluwer Academic Publishers, 1991.
- [NL02] Ngo Van Long and Kigi Shimoruma, *A New Proof Of The Maximun Principle*, Scientific Series, CIRANO, Montreal Janvier 2002.
- [OT89] Optimization Tutorial, optimization with an equality constraint: The envelope theorem, <http://www.chas.utoronto.ca/~osborne/MATHTutorial/MEE.HTM>.
- [PC69] P. J. Cohen. Decision procedures for real and p-adic fields. *Comm. Pure and Applied Math.*, 22(2):-151, March 1969.
- [PI89] *Proceeding of the IEEE*, vol. 77, n° 1, Jan. 1989. Special issue on Dynamics of Discrete Systems.
- [PK93] P. Kozák, S. Balemi, and R. Smedinga, eds. *Discrete, Event Systems: Modeling and Control*, vol. 13 of *Progress in Systems and Control Theory*, Birkhäuser Verlag Basel, Switzerland, 1993 (*Proceeding of the Joint Workshop on Discrete Event Systems (WODES'92)*, August 26-28, 1992, Prague Czechoslovakia).
- [PP93] P. Pedersen, M. Roy, and A. Szpirglas. Counting real zeros in the multivariate case. In F. Eyssette and A. Galligo, editors, *Proc. MEGA'92*. Birkhäuser Boston, 1993.
- [RB75] R. Bellman *Dynamic Programming*, Princeton University Press, N.Y., (1957).
- [RD84] Robert Dautray et Jacques-Loius Lions, *Analyse mathématique et calcul numérique pour les sciences et les techniques*, Series Scientifique, Masson, ISBN 2-225-80406-9, 1984.
- [RD84] Robert Dautray et Jacques-Louis Lions, "Analyse Mathématique et calcul numérique pour les sciences et les techniques", Series Scientifique, Masson, ISBN 2-225-80406-9, 1984.
- [RISK] :http://www.risc.uni-linz.ac.at/cain/virtual_library/QE/main/main.html
- [RK69] R. E. Kalman, P.L. Falb, M.A. Arbib, "Topics in Mathemaical System Theory", Copyright@1969 by McGraw-Hill, Inc., 07-033255-X.
- [RL78] R. Larson R and J. Casti. *Principle of Dynamic Programming, Control and System Theory Part I and II*. Marcel Dekker ISBN 0-8247-6589-3, (1978).
- [RW02] Richard Weber, Lent, *Optimization and Control*, 2002, notes courses, 2002, <http://www.statslab.am.ac.uk>.
- [SM98] S. McCallum. An improved projection operator for cylindrical algebraic decomposition. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation. Springer-Verlag, Vienna, 1998.
- [SS81] J. Schwartz and M. Sharir. On the 'piano movers' problem. i. the case of a two-dimensional rigid polygonal body moving amidst polygonal barriers. Technical Report 39, Department of Computer Science, Courant Institute of Mathematical Sciences, 1981.
- [TM89] T. Murata, "Petrit Nets: Proprieties, analysis and applications", *Proceedings of the IEEE*, vol. 77, n° 4, pp. 541-580, Apr. 1989.

- [VW88]V. Weispfenning. The complexity of linear problems in fields. *Journal of Symbolic Computation*, 5:-27, 1988.
- [VW94] V. Weispfenning. Quantifier elimination for real algebra -- the cubic case. In *Proc. International Symposium on Symbolic and Algebraic Computation*, pages 258-263, 1994.
- [VW97]V. Weispfenning. Quantifier elimination for real algebra -- the quadratic case and beyond. *AAECC*, 8:-101, 1997.
- [VW98]V. Weispfenning. A new approach to quantifier elimination for real algebra. In B. Caviness and J. Johnson, editors, *Quantifier Elimination and Cylindrical Algebraic Decomposition*, Texts and Monographs in Symbolic Computation. Springer-Verlag, Vienna, 1998.
- [YH92] Y. C. Ho, ed., *Discrete Event Dynamic Systems: Analyzing Complexity and Performance in the Modern World* Piscataway, New Jersey: IEEE Press, 1992.

Web

<http://mathworld.wolfram.com/Quantifier.html>

http://www.risc.uni-linz.ac.at/cain/virtual_library/OE/

<http://www.sil.org/linguistics/GlossaryOfLinguisticTerms/WhatIsAQuantifier.htm>

REFERENCIAS 1