



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA

CONTROL PREDICTIVO HÍBRIDO BASADO EN SENSORES VISUALES: TEORÍA Y APLICACIÓN A FLOTACIÓN DE MINERALES

FELIPE EDUARDO NÚÑEZ RETAMAL

Tesis presentada a la Dirección de Investigación y Postgrado
como parte de los requisitos para optar al grado de
Magister en Ciencias de la Ingeniería

Profesor Supervisor:
ALDO CIPRIANO ZAMORANO

Santiago de Chile, Mayo 2008

© MMVIII, FELIPE EDUARDO NÚÑEZ RETAMAL



PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE
ESCUELA DE INGENIERIA

CONTROL PREDICTIVO HÍBRIDO BASADO EN SENSORES VISUALES: TEORÍA Y APLICACIÓN A FLOTACIÓN DE MINERALES

FELIPE EDUARDO NÚÑEZ RETAMAL

Miembros del Comité:

ALDO CIPRIANO ZAMORANO

MIGUEL TORRES TORRITI

JUAN YIANATOS BERNARDINO

JAIME NAVÓN COHEN

Tesis presentada a la Dirección de Investigación y Postgrado
como parte de los requisitos para optar al grado de
Magister en Ciencias de la Ingeniería

Santiago de Chile, Mayo 2008

Por sustento y arte...

AGRADECIMIENTOS

Quiero agradecer primeramente a mi profesor supervisor Don Aldo Cipriano por su apoyo y confianza durante todo el tiempo que me ha tocado trabajar con él. También agradezco a mi familia, mis padres y mi hermana, por su cariño y comprensión durante toda mi vida. A mis amigos Carlos y José Luis que me han acompañado durante estos años en la universidad y a Rodrigo y Pamela quienes este último año han sido mis compañeros en la oficina y se han convertido en grandes amigos. Finalmente quiero agradecer a Fernanda y Philippe por su gran ayuda con el inglés y el francés en los artículos, abstract y en la vida en general.

INDICE GENERAL

AGRADECIMIENTOS	iv
INDICE DE FIGURAS	ix
INDICE DE TABLAS	xi
RESUMEN	xii
ABSTRACT	xiii
1. INTRODUCCIÓN	1
1.1. Motivación	1
1.2. Objetivos de la Tesis	3
1.3. Organización de la Tesis	3
2. ELEMENTOS DE CONTROL PREDICTIVO	5
2.1. Modelación de Sistemas	5
2.1.1. Modelos Lineales	5
2.1.2. Modelos No Lineales	7
2.2. Control Predictivo Basado en Modelos	9
2.2.1. Descripción General	9
2.2.2. Formulación Lineal	11
2.2.3. Formulación No Lineal	12
2.3. Ejemplo de Control Predictivo	12
2.3.1. Descripción del Sistema en Estudio	12
2.3.2. Control Predictivo Lineal	13
2.3.3. Control Predictivo No Lineal	15
2.3.4. Resultados Experimentales	16
3. CONTROL PREDICTIVO HÍBRIDO	17
3.1. Sistemas Híbridos	17

3.1.1.	Descripción General	17
3.1.2.	Ejemplo de Sistemas Híbridos	17
3.2.	Modelos Híbridos	21
3.2.1.	Modelos MLD	21
3.2.2.	Modelos Switched Affine	25
3.2.3.	Estimación de Estado	27
3.2.4.	Estabilidad	29
3.3.	Identificación de Sistemas Híbridos	30
3.3.1.	Estrategia Basada en Clustering	31
3.3.2.	Estrategia Bayesiana	32
3.3.3.	Estrategia Algebraica	34
3.4.	Control Predictivo de Sistemas Híbridos	36
3.4.1.	Descripción General	36
3.4.2.	Formulación Equivalente de Modelos Switched Affine	37
3.4.3.	Estabilidad en Lazo Cerrado	39
3.5.	Ejemplo de Control Predictivo Híbrido	40
3.5.1.	Sistema en Estudio	40
3.5.2.	Análisis de Estabilidad en Lazo Abierto	41
3.5.3.	Control Predictivo Híbrido	41
4.	MODELOS DINÁMICOS DE PROCESOS VISUALES	45
4.1.	Sensores Visuales	45
4.1.1.	Sensores de Espectro Visible	45
4.1.2.	Otros Sensores	46
4.2.	Texturas Dinámicas	47
4.2.1.	Modelación de Texturas Dinámicas	47
4.2.2.	Identificación de Texturas Dinámicas	49
4.2.3.	Clasificación de Texturas Dinámicas	50
4.2.4.	Modelación de Texturas Dinámicas como Sistemas Híbridos	52
4.2.5.	Segmentación Espacial de Texturas Híbridadas	53

4.2.6.	Segmentación Temporal de Texturas Híbridas	55
4.2.7.	Segmentación Espacio–Temporal de Texturas Híbridas	56
4.3.	Ejemplo de Modelación y Clasificación de Procesos Visuales	56
4.3.1.	Descripción del Sistema en Estudio	56
4.3.2.	Caracterización por Estado de Operación	57
4.3.3.	Resultados Experimentales	58
5.	CONTROL PREDICTIVO DE PROCESOS VISUALES	60
5.1.	Descripción General	60
5.2.	Aplicaciones Desarrolladas	60
5.3.	Ejemplo de Control Predictivo de Procesos Visuales	62
5.3.1.	Descripción del Sistema en Estudio	62
5.3.2.	Modelación del Proceso Visual	62
5.3.3.	Diseño del Controlador Predictivo	63
5.3.4.	Resultados Experimentales	65
5.4.	Control Predictivo Híbrido Basado en Sensores Visuales	67
6.	CONTROL PREDICTIVO HÍBRIDO BASADO EN SENSORES VISUALES APLICADO A FLOTACIÓN DE MINERALES	70
6.1.	El Proceso de Flotación de Minerales	70
6.1.1.	Descripción del Proceso	70
6.1.2.	Objetivos del Control en el Proceso de Flotación	72
6.1.3.	Estrategias de Control Usadas en Flotación	72
6.1.4.	Variables Relevantes en el Proceso	73
6.1.5.	Sensores Visuales Comerciales Especializados en Flotación	75
6.2.	Caracterización Híbrida–Experta de la Apariencia Superficial	76
6.2.1.	Adquisición de Imágenes de Proceso	76
6.2.2.	Definición de los Estados de Operación	76
6.2.3.	Modelación del Proceso Visual	77
6.2.4.	Clasificación del Estado de Operación	85

7. CONCLUSIONES Y TRABAJO FUTURO	86
BIBLIOGRAFIA	88
ANEXO A. STATE ESTIMATION BASED MODEL PREDICTIVE CONTROL FOR LHD VEHICLES	91
ANEXO B. HYBRID MODELING OF FROTH FLOTATION SUPERFICIAL APPEARANCE APPLYING DYNAMIC TEXTURES ANALYSIS	97
ANEXO C. SUPERFICIAL FROTH CHARACTERIZATION IN A FLOTATION CELL APPLYING DYNAMIC TEXTURES ANALYSIS AND ON-LINE TEXTURE CLASSIFICATION	102
ANEXO D. VISUAL INFORMATION PREDICTIVE MODELING FOR ADVANCED FLOTATION CONTROL	112

INDICE DE FIGURAS

2.1 Modelos en bloque (a): Modelo de Hammerstein; (b): Modelo de Wiener	9
2.2 Esquema general de un controlador predictivo basado en modelos.	10
2.3 Resultados del seguimiento de trayectoria del LHD	16
3.1 Sistema de tanques del ejemplo	18
3.2 Esquema de control tolerante a fallas.	20
3.3 Esquema de un modelo <i>switched affine</i>	37
3.4 Sistema <i>Switched Affine</i>	38
3.5 Resultados del control predictivo híbrido para el modelo MLD	42
3.6 Resultados del control predictivo híbrido para el modelo MLD fuera de equilibrio	44
4.1 Esquema general de un sensor visual CCD.	46
4.2 Esquema de una imagen digital de NxM pixeles	47
4.3 Estados operacionales definidos para el estudio	57
5.1 Cuadros obtenidos como salida del modelo para $k = 20, 40, 60, 80$	64
5.2 Cuadro de la secuencia real y su respectiva referencia.	64
5.3 Esquema del sistema en lazo cerrado	64
5.4 Resultados del control del proceso visual	66
5.5 Error de control para una simulación de largo $k = 200$	67
5.6 Lazo de control predictivo híbrido basado en sensores visuales	68
6.1 Esquema general de una celda de flotación	70
6.2 Esquema de proceso de flotación rougher y limpieza	71
6.3 Estados operacionales definidos para el estudio	77

6.4 Cuadros seleccionados de las secuencias reales y los cuadros correspondientes obtenidos a partir de los modelos ARMA para $t = 10, 20, 30, 40, 50, 60$	78
6.5 Error de síntesis para el estado 1	79
6.6 Error de síntesis para el estado 2	80
6.7 Error de síntesis para el estado 3	80
6.8 Resultados del test de predicción para los cuadros $\tau + 1$ y $\tau + 2$	81
6.9 Resultados del test de predicción para los cuadros $\tau + 10$ y $\tau + 15$	82
6.10 Error de predicción para el estado 1	83
6.11 Error de predicción para el estado 2	84
6.12 Error de predicción para el estado 3	84

INDICE DE TABLAS

3.1 Equivalencias para relaciones lógicas	22
4.1 Resultados del test de clasificación de estados operacionales.	59
6.1 Resultados del test de clasificación de estados operacionales.	85

RESUMEN

En el último tiempo el control basado en modelos ha cobrado gran popularidad debido a la posibilidad de controlar eficientemente diversos procesos. Sin embargo, existe una complicación extra en la obtención de los modelos involucrados. Por otra parte existen procesos en donde la apariencia representa una importante fuente de información acerca del desempeño del mismo. Es por esto que se hace preponderante contar con técnicas que permitan incorporar información visual al modelo del proceso.

En este trabajo se estudian métodos de modelación de procesos visuales y se propone un esquema de control predictivo híbrido que incorpora la información visual en la forma de estados de operación discreto. Por esto se hace necesario estudiar técnicas de control predictivo, modelación híbrida y control predictivo híbrido; incluyendo identificación, estimación de estado y estabilidad. Se presentan ejemplos prácticos de control predictivo basado en estimación de estado para vehículos mineros LHD, control predictivo híbrido para un sistema MLD, control predictivo de un proceso visual generado artificialmente y modelación de información visual para un sistema de transportes. Los resultados obtenidos muestran que es posible extender los conceptos de control predictivo y modelo predictor a procesos visuales modelados según una estructura ARMAX. Finalmente se probó que es posible aplicar las técnicas de modelación de procesos visuales a un proceso de alta complejidad como la flotación de minerales, obteniéndose modelos que caracterizan la apariencia superficial de la celda de flotación en diferentes estados de operación.

Palabras Claves: Sistemas Híbridos, Texturas Dinámicas, Flotación de Minerales, Control Predictivo, Control Predictivo Híbrido.

ABSTRACT

In recent times model based control has arisen as an important technique mainly due to the good results in process control industry. However, there exists an extra complication in the obtaining of the process models. In the process hand, there exists several processes where the appearance represent an important source of information. This fact make necessary to count with a technique that permit the incorporation of visual information to process model.

In this work different visual processes modeling techniques are studied, also an hybrid predictive control scheme that incorporate visual information as discrete state are presented. This make necessary to study several predictive control, hybrid modeling and hybrid predictive control strategies; including identification, state estimation and stability. Practical examples of state estimation based model predictive control for LHD vehicles, hybrid predictive control for MLD systems, predictive control of visual processes and visual information modeling for transportation systems are presented. Finally the possibility of apply visual processes modeling techniques to a high complexity process as froth flotation process was probe. Obtaining several models that describes the superficial appearance of the cell in different operational states.

Keywords: Hybrid Systems, Dynamic Textures, Froth Flotation, Model Predictive Control, Hybrid Model Predictive Control.

1. INTRODUCCIÓN

En esta tesis se trata el problema de la incorporación de información visual en lazos de control avanzado basado en modelos. El problema a resolver es diseñar un sistema de modelación y reconocimiento de información visual de un proceso altamente complejo como es la flotación de minerales, e incorporar la información en una estrategia de monitoreo.

En el presente capítulo se realiza una descripción general del tema a tratar en esta tesis, junto con presentar los objetivos que se persigue lograr y los contenidos tratados en cada capítulo.

1.1. Motivación

El control automático es una herramienta esencial en la industria de procesos modernos debido a los grandes beneficios en estabilidad, seguridad y economía que brinda. Es así como las industrias han ido automatizando gradualmente sus procesos partiendo desde controles locales del tipo PID hasta controles optimizantes de toda la planta utilizando sistemas de control avanzado.

Sin embargo, existen procesos altamente complejos donde aún no se alcanza un desempeño óptimo. Es en estos procesos donde el control avanzado ha encontrado gran potencial de aplicación, especialmente las estrategias basadas en modelos del sistema en cuestión.

El control basado en modelos trae consigo grandes ventajas en relación al control del tipo PID, como por ejemplo la posibilidad de compensar retardos y perturbaciones medidas. Sin embargo, posee la complicación extra de la obtención del modelo del sistema. Las técnicas de modelación fenomenológica, o basada en primeros principios, tienen limitaciones en procesos de alta complejidad donde las relaciones entre las variables involucradas no se conocen cabalmente. Por otra parte, la modelación usando datos de proceso, conocida

como identificación de modelos, no siempre da buenos resultados debido principalmente a su dependencia de las características de los datos utilizados en la identificación y la necesidad de utilizar una estructura predefinida que muchas veces ignora las no-linealidades del sistema.

En procesos en los cuales existen estados de operación claramente definidos de los que se tiene un conocimiento acabado, es posible modelar el proceso como un sistema híbrido caracterizado por un número de estados de operación, con un modelo simple asociado, los cuales interactúan siguiendo una secuencia ya sea determinística o estocástica. Esta estrategia facilita las labores de modelación ya que, por ejemplo, se pueden obtener modelos fenomenológicos simples para estados de operación conocidos completamente y modelos identificados a partir de datos para estados de operación total o parcialmente desconocidos.

Por otra parte, existen procesos en los cuales la apariencia visual entrega información acerca del estado de operación del mismo; un paso lógico a seguir en estos procesos es la incorporación de la información visual en el modelo que describe el sistema. Lo anterior se logra usando sensores visuales que entregan medidas cuantitativas de la apariencia del proceso, a partir de la cual se puede identificar el estado de operación que determinará el modelo activo del sistema híbrido.

Es así como surge el concepto de control híbrido basado en sensores visuales, en donde la apariencia determina el estado de operación del sistema que se modela como un sistema híbrido utilizando modelos independientes para cada estado de operación. En esta tesis se profundizará en un enfoque predictivo dando origen al concepto de control predictivo híbrido basado en sensores visuales VHMPC (Visual Hybrid Model Predictive Control).

1.2. Objetivos de la Tesis

El objetivo principal de esta tesis es desarrollar un procedimiento para diseñar lazos de control predictivo híbrido basado en mediciones entregadas por un sensor visual, en forma de secuencias de imágenes, y aplicarlo en procesos donde los esquemas de control clásico no logren resultados óptimos o la modelación global no sea viable. Específicamente se realizará una modelación híbrida basada en visión para el proceso de flotación de minerales.

La consecución del objetivo principal presenta desafíos intermedios que se deben resolver en etapas previas, y que se pueden resumir en los siguientes puntos:

- Adquisición de datos e imágenes de proceso en distintos estados de operación.
- Obtención del modelo híbrido que describe el proceso.
- Determinación de las técnicas de procesamiento de imágenes, de ser necesarias, para la síntesis de la información visual.
- Obtención del modelo que incorpora la información visual.
- Diseño de la función objetivo para el control predictivo.

1.3. Organización de la Tesis

El capítulo 2 presenta una introducción a elementos propios del control predictivo, tales como: modelación de sistemas, teoría de control predictivo de sistemas lineales y no lineales, para finalizar con un ejemplo práctico de control predictivo basado en estimación de estado aplicado a vehículos LHD.

En el capítulo 3 se trata en detalle el control predictivo de sistemas híbridos, comenzando con una introducción a sistemas y modelos híbridos, para luego presentar técnicas de identificación de modelos híbridos y el marco teórico del control predictivo híbrido. Finalmente se presenta un ejemplo de control predictivo híbrido para un sistema descrito por

un modelo MLD.

El capítulo 4 trata de modelación dinámica de procesos visuales. Comienza con una revisión teórica de sensores visuales, posteriormente se detallan las técnicas de modelación de secuencias de imágenes utilizando texturas dinámicas. Finalmente se presenta un ejemplo de modelación y clasificación de información visual en un sistema de transportes.

El capítulo 5 trata de control predictivo de procesos visuales. Se presenta una revisión de aplicaciones desarrolladas por otros autores para posteriormente presentar un sistema de control predictivo de color para un proceso visual. Finalmente se propone una estrategia de control predictivo híbrido utilizando información visual.

El capítulo 6 consiste en la aplicación del método desarrollado a una planta de flotación de minerales. Se realiza una descripción detallada del proceso, la caracterización híbrida-experta de la apariencia superficial y se presentan resultados experimentales obtenidos con imágenes reales de proceso.

Finalmente se presentan las conclusiones que se desprenden del trabajo realizado y las propuestas de trabajo futuro en esta línea de investigación.

2. ELEMENTOS DE CONTROL PREDICTIVO

A continuación se presenta un preámbulo teórico y un ejemplo práctico de los elementos involucrados en un lazo de control predictivo.

2.1. Modelación de Sistemas

Se entiende por sistema al proceso bajo análisis. Las características principales de un sistema se capturan en un modelo asociado, el cual debe ser capaz de reproducir el comportamiento real del sistema con el grado de precisión necesario para realizar el análisis deseado. El grado de precisión especificado repercutirá en la complejidad y estructura del modelo del sistema.

La modelación lleva implícita una etapa de ajuste o estimación de parámetros, que tiene la tarea de particularizar el modelo al sistema bajo estudio. Para realizar una estimación de parámetros, se debe contar con datos reales de proceso que sean representativos de los distintos estados de operación del sistema.

Según la complejidad, estructura y obtención del modelo, se pueden establecer diversas divisiones en clases (Ljung, 1999) destacando la distinción entre modelos lineales y no lineales.

2.1.1. Modelos Lineales

Un modelo $\mathcal{M} : u(t) \rightarrow y(t)$ se dice lineal si y sólo si cumple con el principio de superposición:

$$\mathcal{M}(\alpha_1 u_1(t) + \alpha_2 u_2(t)) = \alpha_1 \mathcal{M}(u_1(t)) + \alpha_2 \mathcal{M}(u_2(t)) \quad (2.1)$$

Si un modelo $\mathcal{M} : u(t) \rightarrow y(t)$ no cumple con el principio de superposición se dirá que es no lineal. En general un modelo lineal se puede escribir como (Ljung, 1999):

$$y_k = G(q)u_k + H(q)e_k \quad (2.2)$$

donde $G(q)$ y $H(q)$ son funciones polinomiales en el operador q de adelanto en el tiempo, y e_k una variable aleatoria con distribución $f_e(\cdot)$.

Al representar las funciones $G(q)$ y $H(q)$ como funciones racionales de q , y considerando la variable aleatoria $e_k \sim N(0, \sigma^2)$, se obtienen los tipos de modelos conocidos como modelos de entrada-salida o modelos en funciones de transferencia.

La terminología general para esta clase de modelos es la siguiente (Araya, 2004)

- Los modelos se denominan modelos auto regresivos (Auto Regressive) cuando los denominadores de $G(q)$ y $H(q)$ son polinomios de orden mayor o igual que uno.
- Si $G(q)$ es distinto de cero, se denominan modelos con entradas externas (eXogenous inputs)
- Si el numerador de $H(q)$ es un polinomio de orden mayor que uno, se denominan modelos de media móvil (Moving Average)
- Si $H(q)$ incluye un integrador, se denominan modelos con característica integral (Integrated Models).

Según sus características los modelos son llamados por siglas que las resumen, por ejemplo existen los modelos ARX (Auto Regressive with eXogenous inputs), ARMAX (Auto Regressive with Moving Average and eXogenous inputs), etc. Aprovechando la estructura polinomial, los modelos en funciones de transferencia pueden representarse de la siguiente forma:

$$A(q)y_k = \frac{B(q)}{F(q)}u_k + \frac{C(q)}{D(q)}e_k \quad (2.3)$$

donde $A(q)$, $B(q)$, $C(q)$, $D(q)$ y $F(q)$ son polinomios en el operador q^{-1} .

Una representación alternativa a los modelos en funciones de transferencia son los cada vez mas populares modelos en variables de estado que presentan las siguientes características (Soliman & Srinath, 1997):

- Toda la información necesaria del sistema se encuentra en el vector de estados x_k .
- Proporcionan interpretaciones sobre el comportamiento del sistema que no pueden entregar ni el método de la respuesta al impulso ni la descripción por ecuaciones diferenciales.
- Son especialmente aptos para modelación fenomenológica.
- Se pueden extender a sistemas no lineales o variantes en el tiempo.
- Presentan ventajas en el manejo de sistemas con múltiples entradas y salidas (MIMO).

La gran ventaja de los modelos en variables de estado es la posibilidad de determinar la respuesta del sistema para el instante k conociendo las entradas y el valor del vector de estados para un instante $k_0 < k$.

Un modelo lineal en variables de estado tiene la siguiente forma general:

$$x_{k+1} = Ax_k + Bu_k + w_k \quad (2.4)$$

$$y_k = Cx_k + Du_k + v_k \quad (2.5)$$

donde A , B , C y D son matrices constantes en el tiempo, usualmente se considera $D = 0$, y w_k , v_k son perturbaciones estocásticas generalmente consideradas con distribución Gaussiana. A diferencia de los modelos en funciones de transferencia, los modelos en variables de estado tienen la particularidad de que el estado x_k no siempre puede determinarse a partir de las mediciones de las variables de entrada y salida, a menos que el modelo cumpla la propiedad de observabilidad. De existir estados que no son directamente observables, se hace necesario el uso de los denominados estimadores de estado como los llamados Filtros de Kalman (Welch & Bishop, 2004).

2.1.2. Modelos No Lineales

Pese a que la mayoría de los sistemas reales tienen un comportamiento no lineal es natural utilizar modelos lineales para representarlos, debido a los grandes esfuerzos de cálculo

y complejidad matemática que representan los modelos no lineales. El gran aumento en la capacidad de procesamiento de las computadoras modernas ha reducido la dificultad para manejar modelos no lineales favoreciendo su uso en la industria y el mundo académico. Los modelos no lineales surgen como una extensión de los modelos lineales, impulsados por la necesidad de obtener una mayor exactitud en la modelación de sistemas no lineales. Es así como aparecen los denominados modelos NARMAX (Non-linear ARMAX), NARX, NMAX, etc. cuya expresión general es:

$$y_k = f(y_{k-1}, \dots, y_{k-n_y}, u_k, \dots, u_{k-n_u}, e_{k-1}, \dots, e_{k-n_e}) + e_k \quad (2.6)$$

donde $f(\cdot)$ es una función no lineal.

Asimismo, se puede considerar un modelo no lineal en variables de estado dado por:

$$x_{k+1} = f(x_k, u_k, w_k) \quad (2.7)$$

$$y_k = h(x_k, v_k) \quad (2.8)$$

donde $f(\cdot)$ y $h(\cdot)$ son funciones no lineales. Al igual que en el caso lineal, el vector de estado se puede determinar usando estimadores de estado para modelos no lineales como los llamados Filtros de Kalman Extendidos (Welch & Bishop, 2004).

Un caso especial de modelos no lineales son los llamados modelos en bloques, los que se componen de modelos independientes conectados en cascada: modelos lineales dinámicos y modelos no lineales estáticos. La figura 2.1 muestra un esquema de los modelos de bloques más comunes: los de Hammerstein y los de Wiener.

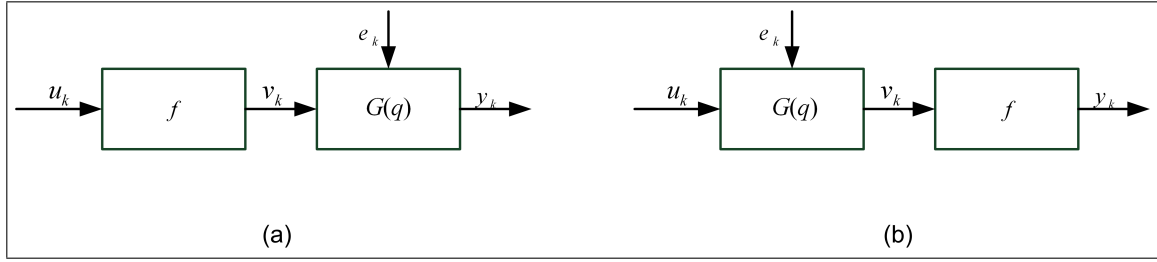


FIGURA 2.1. Modelos en bloque (a): Modelo de Hammerstein; (b): Modelo de Wiener

Los modelos de Hammerstein tienen la ventaja que bajo cierta estructura de f y $G(q)$ son lineales en los parámetros, lo que facilita la tarea de ajuste de parámetros del modelo (Araya, 2004).

Otros modelos no lineales son las redes neuronales, los modelos basados en lógica difusa y los modelos neuro-difusos (Araya, 2004).

2.2. Control Predictivo Basado en Modelos

2.2.1. Descripción General

El objetivo del control predictivo es optimizar, sobre una secuencia de controles $\{u_{k+j}\}_{j=1}^{N_u}$, la respuesta de un proceso usando como criterio el valor de una función objetivo. La optimización se realiza usando un modelo que provee información del comportamiento futuro del proceso, lineal o no lineal. La formulación del problema de control como un problema de optimización permite la inclusión de restricciones explícitas sobre las variables involucradas lo que representa una gran ventaja sobre los métodos de control clásicos en procesos de alta complejidad. La realimentación se logra aplicando el primer valor del control u_{k+1} y resolviendo nuevamente el problema de optimización asociado introduciendo el efecto de éste control, reflejado en las mediciones en el instante $k + 1$, en una estrategia conocida como *Receding Horizon Control (RHC)*. Un esquema general de un controlador predictivo se presenta en la figura 2.2 (Camacho & Bordons, 2004).

Las principales características del control predictivo se listan a continuación (Camacho & Bordons, 2004):

- Uso explícito de un modelo para el cálculo de las predicciones.
- Cálculo de la ley de control mediante la minimización de una función objetivo.
- Aplicación al sistema del primer elemento de control en cada iteración haciendo uso del principio de Receding Horizon Control.
- El caso multivariable es fácilmente manejable representando sólo un aumento de los cálculos involucrados, más no un aumento en la complejidad de éstos.
- La formulación permite el uso de modelos lineales o no-lineales.
- Posibilidad de incluir restricciones explícitas sobre las variables de proceso.
- Compensación intrínseca de los retardos involucrados.
- Compensación intrínseca de las perturbaciones medidas que afectan el proceso.
- Grandes ventajas sobre el control tradicional cuando se cuenta con los valores futuros de la referencia.
- Posibilidad de controlar sistemas con fase no-mínima e incluso inestables.

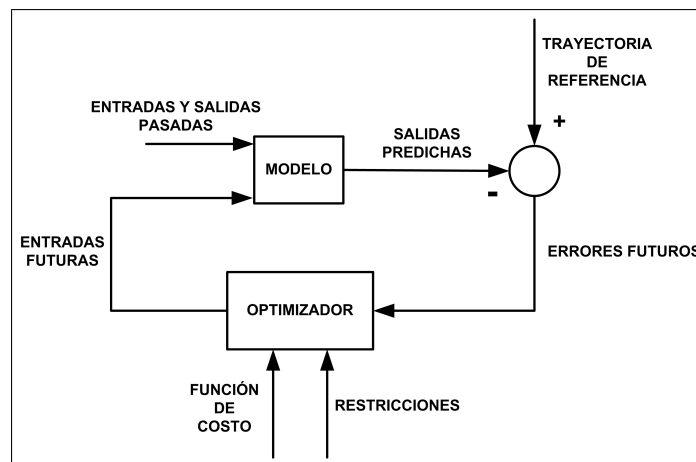


FIGURA 2.2. Esquema general de un controlador predictivo basado en modelos.

Como se dijo, el control predictivo se basa en el uso de un modelo del proceso, el cual puede obtenerse por modelación fenomenológica o por identificación del proceso dando origen a diversos algoritmos de control predictivo que difieren en el método de cálculo de las predicciones, pero que tienen en común el uso de un modelo, una función objetivo y el uso del principio de RHC. A continuación se describe brevemente las formulaciones lineal

y no-lineal del problema de control predictivo usando un modelo en variables de estado, pero fácilmente generalizables para otro tipo de modelo.

2.2.2. Formulación Lineal

El proceso se describe por un modelo lineal en tiempo discreto con m entradas, n variables de estado y q salidas del tipo:

$$x_{k+1} = Ax_k + Bu_k \quad (2.9)$$

$$y_k = Cx_k \quad (2.10)$$

donde A , B y C son matrices de $n \times n$, $n \times m$ y $q \times n$ respectivamente.

El control predictivo busca la secuencia de controles que son solución del siguiente problema de optimización:

$$\begin{aligned} \min_{\{u_{k+j}\}_{j=1}^{N_u}} J(N_1, N_2, N_u) = & \sum_{j=N_1}^{N_2} \delta_j [\hat{y}_{k+j} - yr_{k+j}]^2 + \\ & \sum_{j=1}^{N_u} \lambda_j [\Delta u_{k+j-1}]^2 + \sum_{j=1}^{N_u} \gamma_j [u_{k+j-1} - ur_{k+j-1}]^2 \end{aligned} \quad (2.11)$$

donde N_u es el horizonte de control, N_1 y N_2 son los horizontes de predicción, δ_j es el peso asociado a la desviación de la referencia en el instante $k + j$, γ_j es el peso asociado a la desviación de la referencia del control en el instante $k + j - 1$, λ_j es el peso asociado a los incrementos en el control en el instante $k + j - 1$, yr_{k+j} es la referencia para las variables controladas y ur_{k+j-1} es la referencia para las variables manipuladas. El control óptimo se determina sujeto a las restricciones del modelo dado por (2.9) y (2.10), y a restricciones de dominio del tipo:

$$\begin{aligned} x & \in \mathbb{X} \\ u & \in \mathbb{U} \\ y & \in \mathbb{Y} \end{aligned} \quad (2.12)$$

donde $\mathbb{X} \subseteq \mathbb{R}^n$, $\mathbb{U} \subseteq \mathbb{R}^m$ y $\mathbb{Y} \subseteq \mathbb{R}^q$.

En ausencia de restricciones de dominio, se puede encontrar una solución explícita al problema de control predictivo (Camacho & Bordons, 2004). Si existen restricciones de dominio se debe recurrir a programación cuadrática para encontrar la solución en tiempo real.

2.2.3. Formulación No Lineal

La formulación no lineal difiere del caso lineal en la estructura del modelo utilizado, el cual viene ahora dado generalmente por la ecuación de estado:

$$x_{k+1} = f(x_k, u_k) \quad (2.13)$$

$$y_k = h(x_k, u_k) \quad (2.14)$$

donde $f(\cdot)$ y $h(\cdot)$ son funciones no-lineales. Las restricciones son las mismas que para el caso lineal y vienen dadas por (2.13) y (2.14) y restricciones de dominio.

Una estrategia para encontrar la solución del problema de control predictivo no lineal es linealizar en torno a la trayectoria de referencia resultando un modelo lineal variante en el tiempo, el cual en ausencia de restricciones resulta en una solución explícita.

De no aplicar linealización explícita, la solución al problema de control predictivo no-lineal con restricciones se resuelve utilizando métodos numéricos de optimización no-lineal, por ejemplo del tipo cuasi-Newton o algoritmos genéticos, que minimizan la función objetivo general dada por (2.11).

2.3. Ejemplo de Control Predictivo

2.3.1. Descripción del Sistema en Estudio

El sistema en estudio corresponde a un vehículo no tripulado utilizado en faenas mineras, conocido como LHD (Load-Haul-Dump). Un LHD es una suerte de grúa automatizada, ampliamente usada en la extracción, sobre todo en túneles de difícil o peligroso acceso para

un operario. Se puede ver como la unión de una parte frontal, equipada de un eje, unida a una parte trasera (similar a la delantera) por medio de una unión de otro eje vertical. El movimiento del vehículo se produce aplicando una propulsión en las ruedas y un torque en la unión vertical. Al LHD se le asigna un camino predeterminado, el que debe seguir con la mayor precisión posible.

Un modelo en tiempo continuo del LHD, no lineal en el espacio de los estados, viene dado por (Nuñez, Navarro, Aguado, & Cipriano, 2008):

$$\begin{aligned} \dot{x}_1 &= u_1 \cos(\theta_1) \\ \dot{y}_1 &= u_1 \sin(\theta_1) \\ \dot{\theta}_1 &= \frac{u_1 \sin(\gamma) - L_0 \Gamma}{L_0 + L_1 \cos(\gamma)} \\ \dot{\gamma} &= \Gamma \end{aligned} \tag{2.15}$$

donde u_1 y Γ son las variables manipuladas con Γ proporcional al torque aplicado. Las variables medidas son los ángulos θ_1 y γ , por lo que los otros estados del sistema deben ser estimados.

Si se discretiza el modelo del LHD y se añaden perturbaciones estocásticas, se obtiene un modelo en tiempo discreto en el espacio de los estados de la forma (Nuñez et al., 2008):

$$x_{k+1} = f(x_k, u_k, w_k) \tag{2.16}$$

$$y_k = Cx_k + v_k \tag{2.17}$$

donde $f(\cdot)$ es una función no lineal.

2.3.2. Control Predictivo Lineal

Si se considera el modelo no lineal (2.16), (2.17) y la trayectoria de referencia:

$$x_{k+1}^* = f(x_k^*, u_k^*) \tag{2.18}$$

$$y_{k+1}^* = Cx_k^* \quad (2.19)$$

x^* , u^* son el estado y control óptimos conocidos que generan la trayectoria (2.18), (2.19).

Se tiene:

$$x = x^* + x^1 \quad y \quad u = u^* + u^1 \quad (2.20)$$

donde x^1 y u^1 representan las desviaciones con respecto a los valores óptimos x^* y u^* .

Luego tomando la aproximación de Taylor de primer orden de (2.18) se obtiene el modelo lineal variable en el tiempo:

$$x_{k+1}^1 = A_k x_k^1 + B_k u_k^1 \quad (2.21)$$

$$y_k = Cx_k^1 \quad (2.22)$$

Para el diseño del controlador predictivo lineal, se propone la siguiente función objetivo:

$$\begin{aligned} J(N_2, N_u) = & \sum_{j=1}^{N_2} \delta [\hat{y}_{k+j} - yr_{k+j}]^2 \\ & + \sum_{j=1}^{N_u} \lambda [\Delta u_{k+j-1}]^2 \end{aligned} \quad (2.23)$$

Las salidas del controlador predictivos se obtienen minimizando las desviaciones x^1 con el mínimo esfuerzo de control extra u^1 .

Las salidas del modelo se pueden predecir como (Nuñez et al., 2008):

$$\begin{aligned} y_{k+1} &= C(A_k x_k + B_k u_{k-1} + B_k \Delta u_k) \\ y_{k+2} &= C(A_{k+1} A_k x_k + (A_{k+1} B_k + B_{k+1}) u_{k-1} \\ &\quad + (A_{k+1} B_k + B_{k+1}) \Delta u_k + B_{k+1} \Delta u_{k+1}) \\ &\vdots \end{aligned} \quad (2.24)$$

Lo que en forma matricial queda:

$$Y_{k+1} = \Psi x_k + \Phi u_{k-1} + \Theta \Delta u_k \quad (2.25)$$

donde:

$$\Theta = \begin{pmatrix} CB_k & \cdots \\ C(A_k B_k + B_{k+1}) & \cdots \\ \vdots & \ddots \\ \sum_{i=k}^{k+N_2-1} C \prod_{j=k+1}^i A_j B_{2k+N_2-1-i} & \cdots \\ \cdots & 0 \\ \cdots & 0 \\ \ddots & \vdots \\ \cdots & \sum_{i=k}^{k+N_2-N_u} C \prod_{j=k+1}^{k+i} A_j B_{2k+N_2-1-i} \end{pmatrix} \quad (2.26)$$

$$\Psi = \begin{pmatrix} CA_k \\ C \prod_{j=k}^{k+1} A_j \\ \vdots \\ C \prod_{j=k}^{k+N_2-1} A_j \end{pmatrix} \quad (2.27)$$

$$\Phi = \begin{pmatrix} CB_k \\ \sum_{i=k}^{k+1} C \prod_{j=k+1}^i A_j B_{2k+1-i} \\ \vdots \\ \sum_{i=k}^{k+N_2-1} C \prod_{j=k+1}^i A_j B_{2k+N_2-1-i} \end{pmatrix} \quad (2.28)$$

En ausencia de restricciones se puede usar la ecuación de Maciejowsky (Camacho & Bordons, 2004), para encontrar la solución cerrada del problema de control predictivo:

$$\Delta u_k = (\Theta^T \Theta + \lambda I)^{-1} \Theta^T (y r_k - \Psi \hat{x}_k - \Phi u_{k-1}) \quad (2.29)$$

2.3.3. Control Predictivo No Lineal

El problema de control predictivo no lineal consiste en resolver en línea el problema de minimización dado por (2.11), (2.16) y (2.17).

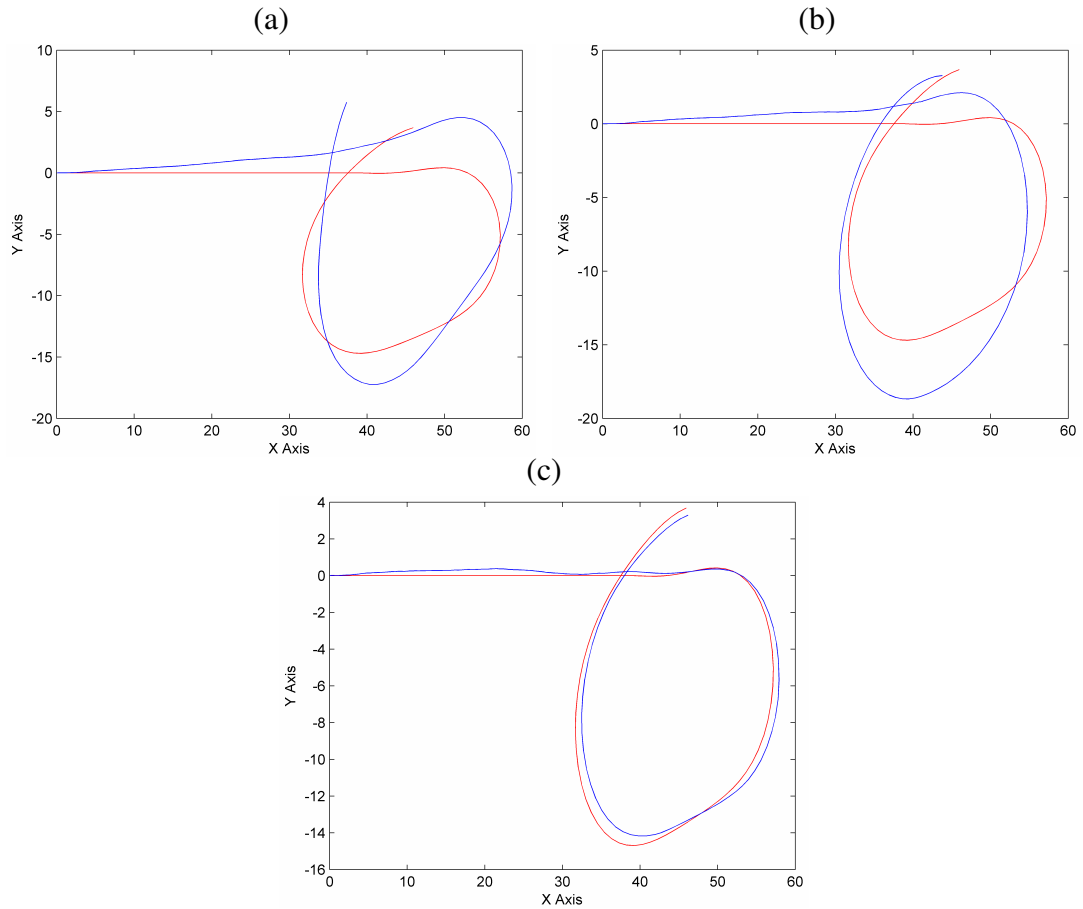


FIGURA 2.3. Resultados del seguimiento de trayectoria del LHD; Rojo: Trayectoria de referencia; Azul: Trayectoria real; (a): Respuesta en lazo abierto; (b): Respuesta obtenida con el controlador predictivo lineal; (c): Respuesta obtenida con el controlador predictivo no lineal.

2.3.4. Resultados Experimentales

Se programó en Matlab el modelo del LHD y los controladores predictivos, utilizando un filtro de Kalman Extendido como estimador de estado (Welch & Bishop, 2004) y un periodo de muestreo $T = 0.2$ s, que mostró ser suficientemente grande para poder controlar y mantener los requerimientos computacionales dentro de un rango que permitiera simular en tiempo real. Los resultados obtenidos se presentan en la figura 2.3. Se aprecia que los controladores predictivos mejoran la respuesta del sistema en lazo abierto, especialmente el controlador predictivo no lineal que presenta un desempeño casi perfecto. Sin embargo, el controlador predictivo no lineal es computacionalmente más demandante.

3. CONTROL PREDICTIVO HÍBRIDO

3.1. Sistemas Híbridos

3.1.1. Descripción General

Se llama sistemas híbridos a aquellos sistemas cuyas propiedades dinámicas dependen de la interacción entre variables con un dominio continuo y variables con un dominio finito, llamadas variables discretas. Usualmente el comportamiento continuo surge de las relaciones físico-químicas que gobiernan el proceso, mientras que el comportamiento discreto se puede originar por diversas causas, entre ellas (Christofides & El-Farra, 2005):

- Discontinuidades en las características físico-químicas del proceso, originadas por cambios de fase, reacciones químicas, etc.
- Uso de sensores y actuadores con posiciones discretas, como son sensores binarios, válvulas on-off, etc.
- Uso de sistemas supervisores basados en lógica digital.

Independientemente de la fuente del comportamiento híbrido (continuo-discreto), el proceso se puede caracterizar por estructuras dinámicas diferentes en diferentes situaciones (regímenes de operación), y es por esto que es intuitivamente simple ver el proceso como intervalos continuos relacionados por transiciones discretas regidas por un sistema de decisión superior.

3.1.2. Ejemplo de Sistemas Híbridos

Para clarificar el concepto de sistemas híbridos, y la fuente de comportamiento híbrido, a continuación se presentan 2 ejemplos concretos de sistemas híbridos: un sistema de tanques (Cruz, 2006) y un sistema de control tolerante a fallas (Christofides & El-Farra, 2005).

Sistema de Tanques.: Considere el sistema de tanques de la figura 3.1, en el cual se asume que los dos componentes de flujos, el componente A del tanque 1 y el componente B del tanque 2, se mezclan en el tanque 3. Adicionalmente se asume que cuando los flujos de entrada q_A y q_B , están activados, son aproximadamente

constantes. La razón para esta suposición es que el flujo q_A del tanque 1 se controlan por una bomba de encendido-apagado que da un flujo casi constante. En el tanque 2 el nivel se controla por una válvula de entrada por medio de sensores binarios. Esto implica que el flujo q_B es aproximadamente constante cuando la válvula de salida está abierta. Se requiere efectuar el mezclado de los dos componentes cuidando los efectos de la reacción; después del mezclado se deposita la mezcla en un recipiente.

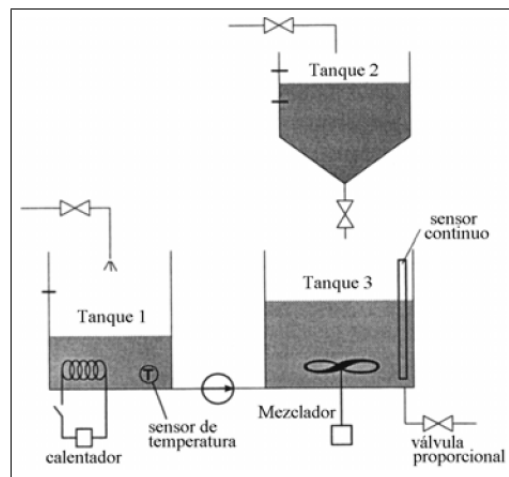


FIGURA 3.1. Sistema de tanques del ejemplo

A partir de la descripción del sistema, se pueden diferenciar claramente las componentes continuas y discretas del sistema (Cruz, 2006):

- Variables continuas:
 - (i) El flujo q_A del tanque 1
 - (ii) El flujo q_B del tanque 2
 - (iii) El volumen del tanque 3
 - (iv) Calentador (comportamiento de la temperatura)
- Variables discretas:
 - (i) Estado de la bomba del tanque 1
 - (ii) Estado de la válvula de salida del tanque 2
 - (iii) Estado de la válvula de salida del tanque 3

(iv) Agitador (encendido-apagado)

(v) Sensor binario de temperatura

Una vez identificadas las variables continuas y discretas que interactúan en el sistema en estudio, se procede a obtener un modelo híbrido que describa el comportamiento del sistema.

Sistema de Control Tolerante a Fallas.: Algunos sistemas de control con la habilidad para manejar situaciones de falla, requieren múltiples configuraciones de control y un adecuado dispositivo de selección entre ellos para preservar la estabilidad y control en lazo cerrado cuando el controlador activo falla. La ocurrencia de fallas y el switching entre las diferentes estrategias de control, resultan en un comportamiento dinámico del tipo híbrido para el sistema en lazo cerrado, a pesar de que el sistema propiamente tal pueda no tener variables discretas en su dinámica. Como ejemplo se considera un reactor de mezcla continua no isotérmico, donde ocurren tres reacciones exotérmicas irreversibles paralelamente de la forma $A \xrightarrow{k_1} B$, $A \xrightarrow{k_2} U$ y $A \xrightarrow{k_3} R$, donde A es el reactante, B es el producto y U y R son subproductos no deseados. La alimentación al reactor consiste en un flujo puro de reactante A a tasa F , concentración molar C_{A0} y temperatura T_{A0} . Debido a la naturaleza no isotérmica del proceso, se utiliza una chaqueta para extraer o adicionar calor al reactor. El modelo matemático que describe el proceso viene dado por (Christofides & El-Farra, 2005):

$$\begin{aligned}\frac{dT}{dt} &= \frac{F}{V}(T_{A0} - T) + \sum_{i=1}^3 \frac{-\Delta H_i}{\rho c_p} k_{io} \exp\left(\frac{-E_i}{RT}\right) C_A + \frac{Q}{\rho c_p V} \\ \frac{dC_A}{dt} &= \frac{F}{V}(C_{A0} - C_A) - \sum_{i=1}^3 k_{io} \exp\left(\frac{-E_i}{RT}\right) C_A \\ \frac{dC_B}{dt} &= \frac{F}{V}C_B + k_{1o} \exp\left(\frac{-E_1}{RT}\right) C_A\end{aligned}\tag{3.1}$$

donde C_A y C_B denotan las concentraciones de las especies A y B , T denota la temperatura del reactor, Q denota la tasa de calor que entra al reactor, V denota el volumen del reactor, ΔH_i , k_i , E_i , $i = 1, 2, 3$ denotan las entalpías, constantes

preexponenciales y energía de activación de las tres reacciones respectivamente, c_p y ρ son el calor específico y la densidad del fluido dentro del reactor. El objetivo del control es llevar al proceso a un punto de equilibrio inestable en lazo abierto. Para esto se cuenta con tres candidatos a variables manipuladas siguiendo los esquemas que se presentan en la figura 3.2. Cada lazo cerrado puede llevar el proceso al punto de equilibrio deseado, sin embargo, debido a fallas comunes se idea una estrategia de switching entre los lazos de control donde el controlador nominal corresponde a la configuración (1) de la figura 3.2.

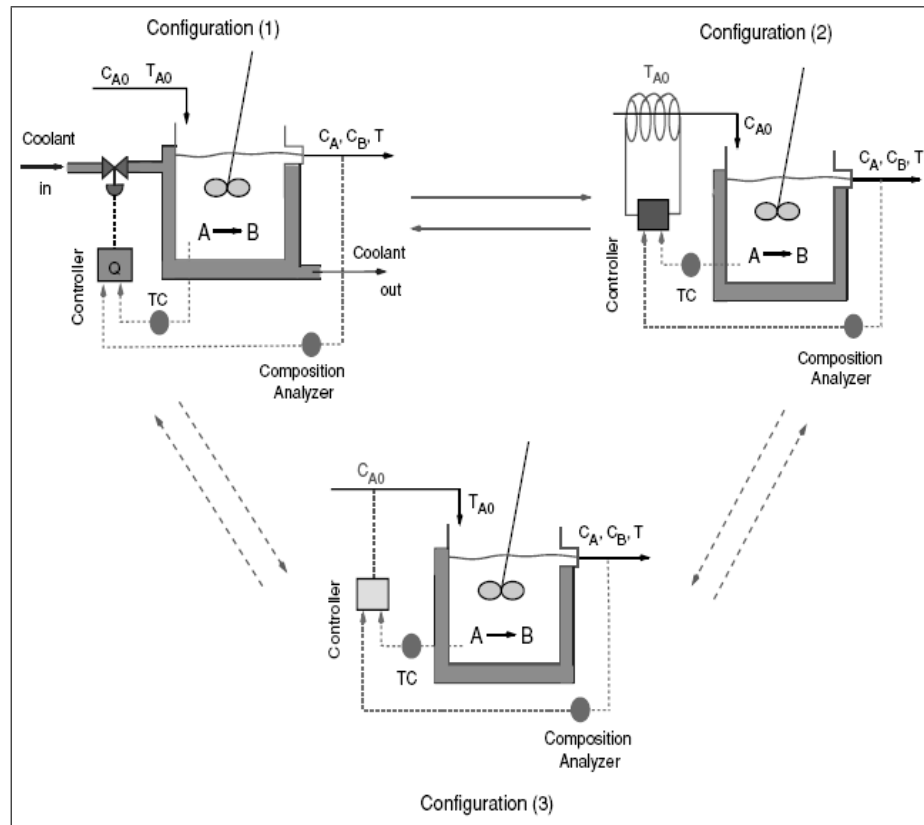


FIGURA 3.2. Esquema de control tolerante a fallas.

El problema de control consiste en diseñar un sistema supervisor que estime el estado de operación del sistema en lazo cerrado y decida que estrategia de control es la más adecuada.

3.2. Modelos Híbridos

Por modelo híbrido se entiende a la representación de un sistema mediante interacciones de variables continuas y discretas. Según la estructura en que se modelen las interacciones, se pueden distinguir clases de modelos híbridos, (Paoletti, Julosky, Ferrari-Trecate, & Vidal, 2007): modelos del tipo *Mixed Logical Dynamics* (MLD), del tipo *PieceWise Affine* (PWA), del tipo *Linear Complementarity* (LC), del tipo *Extended Linear Complementarity* (ELC) y del tipo *Max-Min-Plus-scaling* (MMPS). Todas estas clases cumplen con el principio de equivalencia, es decir, un sistema se puede describir equivalentemente por un modelo enunciado siguiendo cualquiera de las estructuras mencionadas anteriormente, con diferencias en la complejidad y cálculos asociados (Heemels, Schumacher, & Weiland, 2001), lo que permite aprovechar el conocimiento del sistema facilitando las tareas de modelación híbrida. En esta tesis se entrará en detalle en las dos clases mas numerosas en aplicaciones: MLD y PWA.

3.2.1. Modelos MLD

Los modelos MLD surgieron como alternativa para modelar sistemas híbridos en que no es posible, o es contraproducente, realizar una estructuración jerárquica por capas, donde la capa superior corresponde a componentes lógicas y la capa inferior a componentes dinámicas continuas. El esquema MLD modela las interacciones discretas como desigualdades lineales que involucran variables enteras y continuas (Bemporad & Morari, 1999).

Los modelos MLD describen las relaciones lógicas siguiendo las reglas de la tabla 3.1.

Si existen relaciones modeladas como productos de variables lógicas, en este caso se procede a definir una variable lógica auxiliar y reemplazar el producto por las siguientes

TABLA 3.1. Equivalencias para relaciones lógicas

Proposición Lógica	Desigualdades equivalentes
$x_1 \vee x_2$	$\delta_1 + \delta_2 \geq 1$
$x_1 \wedge x_2$	$\delta_1 = 1, \delta_2 = 1$
$\sim x_1$	$\delta_1 = 0$
$x_1 \rightarrow x_2$	$\delta_1 - \delta_2 \leq 0$
$x_1 \leftrightarrow x_2$	$\delta_1 - \delta_2 = 0$
$x_1 \oplus x_2$	$\delta_1 + \delta_2 = 1$

desigualdades que involucran a la variable auxiliar:

$$\delta_3 = \delta_1 \delta_2 \equiv \begin{cases} -\delta_1 + \delta_3 & \leq 0 \\ -\delta_2 + \delta_3 & \leq 0 \\ \delta_1 + \delta_2 - \delta_3 & \leq 1 \end{cases}$$

Frecuentemente los sistemas en estudio presentan comportamientos discontinuos que se modelan como productos entre variables lógicas y variables continuas. Para transformar estos casos en desigualdades que involucren variables lógicas y continuas, se procede de la siguiente manera:

Sea $z(t) = \delta(t)f(x)$, se define $M = \max f(x)$ y $m = \min f(x)$. Luego las desigualdades equivalentes son:

$$z(t) \leq M\delta(t)$$

$$z(t) \geq m\delta(t)$$

$$z(t) \leq f(x) - m(1 - \delta(t))$$

$$z(t) \geq f(x) - M(1 - \delta(t))$$

Los modelos MLD pueden ser descritos en forma general por:

$$x(t+1) = A_t x(t) + B_{1t} u(t) + B_{2t} \delta(t) + B_{3t} z(t) \quad (3.2)$$

$$y(t) = C_t x(t) + D_{1t} u(t) + D_{2t} \delta(t) + D_{3t} z(t) \quad (3.3)$$

$$E_{2t}\delta(t) + E_{3t}z(t) \leq E_{1t}u(t) + E_{4t}x(t) + E_{5t} \quad (3.4)$$

donde:

$$x = [x_c \ x_l]'$$

es el vector de estados en el cual se distinguen estados continuos x_c y binarios x_l , e

$$y = [y_c \ y_l]'$$

es el vector de salida.

$$u = [u_c \ u_l]'$$

es el vector de entradas. $z(t) = \delta(t)x(t)$ es una variable auxiliar que modela no linealidades del tipo productos de variables continuas con discretas (binarias).

Las expresiones anteriores describen un sistema lineal en tiempo discreto. Pero, es fácilmente extensible al tiempo continuo cambiando $x(t+1)$ por $\dot{x}(t)$ o a una versión no lineal cambiando las desigualdades y ecuaciones lineales por funciones no lineales más generales.

En principio, las ecuaciones (3.2),(3.3),(3.4) pueden ser satisfechas por muchos valores de $\delta(t)$ y $z(t)$. Sin embargo, es deseable que $x(t+1)$ y $y(t)$ queden determinadas únicamente por $x(t)$ y $u(t)$. Para lograrlo, es necesario considerar las siguientes definiciones (Bemporad & Morari, 1999):

Sea $\mathcal{F}_{Bt}$ el conjunto de todos los índices $i \in \{1, \dots, r_l\}$, tal que $[B_{2t}]^i \neq 0$, donde $[B_{2t}]^i$ corresponde a la columna i de B_{2t} . Análogamente se definen $\mathcal{F}_{Dt}$, $\mathcal{G}_{Bt}$ y $\mathcal{G}_{Dt}$, para D_{2t} , B_{3t} y D_{3t} respectivamente. También se definen $\mathcal{F}_t \equiv \mathcal{F}_{Bt} \cup \mathcal{F}_{Dt}$ y $\mathcal{G}_t \equiv \mathcal{G}_{Bt} \cup \mathcal{G}_{Dt}$.

Un modelo MLD se dirá *bien definido* si, $\forall t$:

- i $x(t)$ y $u(t)$ satisfacen (3.4) para algún $\delta(t) \in \{0, 1\}^{r_l}$, $z(t) \in \mathbb{R}^{r_c}$ y $x_l(t+1) \in \{0, 1\}^{r_l}$.
- ii $\forall i \in \mathcal{F}_t$ existe una función $\mathcal{D}_{it} : \mathbb{R}^{n+m} \mapsto \{0, 1\}$ tal que la i -ésima componente $\delta_i(t) = \mathcal{D}_{it}(x(t), u(t))$ y, equivalentemente $\forall j \in \mathcal{G}_t$ existe una función $\mathcal{L}_{jt} : \mathbb{R}^{n+m} \mapsto \mathbb{R}$ tal que $z_j(t) = \mathcal{L}_{jt}(x(t), u(t))$

En adelante se considerará que los modelos MLD tratados cumplen con ser *bien definidos*, lo que no es una pérdida de generalidad ya que los modelos derivados de sistemas reales son *bien definidos* y esta tesis se enfoca en procesos reales.

A modo de ejemplo se considera un sistema con el siguiente modelo asociado (Bemporad & Morari, 1999):

$$x(t+1) = \begin{cases} 0.8x(t) + u(t) & \text{si } x(t) \geq 0 \\ -0.8x(t) + u(t) & \text{si } x(t) < 0 \end{cases} \quad (3.5)$$

donde $x(t) \in [-10, 10]$, y $u(t) \in [-1, 1]$. Es claro que el modelo presenta una discontinuidad que puede ser modelada con una variable lógica asociada $\delta_1(t)$ definida como $\delta_1(t) = 0$ si $x(t) < 0$; $\delta_1(t) = 1$ si $x(t) \geq 0$. Luego el modelo híbrido queda:

$$x(t+1) = 1.6\delta_1(t)x(t) - 0.8x(t) + u(t) \quad (3.6)$$

Transformando el producto $\delta_1(t)x(t)$ en desigualdades, finalmente el modelo MLD queda:

$$\begin{aligned} x(t+1) &= 1.6z(t) - 0.8x(t) + u(t) \\ z(t) &\leq 10\delta_1(t) \\ z(t) &\geq -10\delta_1(t) \\ z(t) &\leq x(t) + 10(1 - \delta_1(t)) \\ z(t) &\geq x(t) - 10(1 - \delta_1(t)) \end{aligned} \quad (3.7)$$

Es importante notar que los modelos MLD definidos según (3.2),(3.3),(3.4) incluyen las siguientes clases de sistemas (Bemporad & Morari, 1999)

- Sistemas lineales.
- Sistemas lineales con restricciones.
- Sistemas lógicos secuenciales (autómatas).
- Sistemas híbridos lineales.
- Algunos sistemas de eventos discretos.

- Sistemas dinámicos no lineales, donde la no linealidad se puede expresar como lógica combinatorial.

3.2.2. Modelos Switched Affine

Los modelos *Switched Affine* se definen como una colección de modelos lineales conectados por switches que son indexados por una variable discreta adicional, llamada el estado discreto del sistema. Los modelos en los cuales el estado discreto es determinado por una partición poliédrica del espacio del estado son llamados modelos *Piecewise affine* PWA, los que son especialmente aptos para aproximar dinámicas no lineales vía linealización en múltiples puntos de operación.

La propiedad de equivalencia entre modelos híbridos, permite que los modelos PWA sean usados para describir un gran número de sistemas que presentan estructuras híbridas.

Un modelo *Switched Affine* estocástico en tiempo discreto puede representarse en el espacio de estados como:

$$x_{k+1} = A_{\sigma(k)}x_k + B_{\sigma(k)}u_k + f_{\sigma(k)} + w_k \quad (3.8)$$

$$y_k = C_{\sigma(k)}x_k + D_{\sigma(k)}u_k + g_{\sigma(k)} + v_k \quad (3.9)$$

donde $x_k \in \mathbb{R}^n$, $u_k \in \mathbb{R}^p$, y $y_k \in \mathbb{R}^q$ son el estado continuo, la entrada y la salida respectivamente, mientras que w_k y v_k son los ruidos de modelación y sensor respectivamente. El estado discreto $\sigma(k)$ determina en que estado se encuentra el sistema y se asume que puede tomar un conjunto finito de valores $\sigma(k) \in \{1, \dots, s\}$, donde s representa el número de modelos *affine*. En general, el estado discreto $\sigma(k)$ es función del tiempo k , de las entradas u_k , de las salidas y_k y del estado x_k . Las matrices A_i , B_i , C_i , D_i , f_i y g_i . $i = 1, \dots, s$ representan las dinámicas *affine* de cada submodelo, por lo que el modelo dado por (3.8) y (3.9) se puede considerar como un conjunto de modelos *affine* conectados por switches indexados por $\sigma(k)$.

La evolución del estado discreto $\sigma(k)$ puede describirse de diversas maneras (Paoletti et al., 2007). En los modelos *Jump Linear* (JL) el estado discreto es desconocido, determinístico y toma valores finitos. En modelos *Jump-Markov Linear* (JML) el estado discreto está determinado por una cadena de Markov irreducible gobernada por probabilidades de transición. En modelos *PieceWise Affine* PWA, el estado discreto viene dado por la regla:

$$\sigma(k) = i \leftrightarrow [x_k \ u_k]' \in \Omega_i, i = 1, \dots, s \quad (3.10)$$

donde $\{\Omega_i\}_{i=1}^s$ es una partición completa del dominio.

Para efectos de identificación de modelos, es conveniente escribir el modelo en estructura entrada-salida. Para órdenes fijos n_a y n_b se definen los modelos *Switched AutoRegressive eXogenous* (SARX) usando el regresor r_k definido como:

$$r_k = [y_{k-1}^T \cdots y_{k-n_a}^T \ u_k^T \cdots u_{k-n_b}^T]^T \quad (3.11)$$

Expresando la salida y_k como una función de r_k , se obtiene:

$$y_k = \theta_{\sigma(k)}^T \begin{bmatrix} r_k \\ 1 \end{bmatrix} + e_k \quad (3.12)$$

donde θ_i son los vectores de parámetros que determinan a cada submodelo y $e_k \in \mathbb{R}^q$ es termino correspondiente al ruido del proceso. En lo que sigue se llamará a $\varphi_k = \begin{bmatrix} r_k \\ 1 \end{bmatrix}$ vector regresor extendido (Paoletti et al., 2007).

Al igual que para los modelos expresados en el espacio de los estados, existen diversas formas para describir la evolución del estado discreto. En los modelos *PieceWise Auto Regressive eXogenous* (PWARX) el mecanismo de switching se determina por una partición poliédrica del dominio de los regresores $\mathcal{R} \subseteq \mathbb{R}^d$, donde $d = qn_a + p(n_b + 1)$. Es decir, para los modelos PWARX el estado discreto viene dado por:

$$\sigma(k) = i \leftrightarrow r_k \in \mathcal{R}_i, i = 1, \dots, s \quad (3.13)$$

donde $\{\mathcal{R}\}_{i=1}^s$ es una partición completa del dominio $\mathcal{R}$.

3.2.3. Estimación de Estado

Cuando se utilizan modelos en variables de estado, se hace necesario contar con una estrategia de estimación de estado. A continuación se detalla un procedimiento para la estimación de estado de modelos MLD y PWA basada en horizontes móviles o MHE (Moving Horizon Estimation) (Ferrari-Trecate, Mignone, & Morari, 2002).

Sea un modelo MLD dado por (3.2), (3.3) y (3.4) que cumple con ser *bien definido*. En presencia de perturbaciones estocásticas, es posible obtener un modelo PWA equivalente de la forma (Ferrari-Trecate et al., 2002): (3.8), (3.9) sujeto a las restricciones de dominio:

$$\begin{aligned} x &\in \mathbb{X} \\ u &\in \mathbb{U} \\ w &\in \mathbb{W} \end{aligned} \tag{3.14}$$

donde $\mathbb{W}$ es un poliedro acotado que contiene el origen y representa restricciones explícitas sobre la incertidumbre del modelo.

Se introduce la siguiente función de costo:

$$J(\tau, t, w, v, x_\tau, \Gamma_\tau) = \sum_{k=\tau}^{t-1} \|v_k\|_R^2 + \|w_k\|_Q^2 + \Gamma_\tau(x_\tau) \tag{3.15}$$

donde $\tau, t \in \mathbb{N}$, $\tau < t$, Γ_τ es una función continua y Q y R son matrices positivas definidas de dimensión apropiada, y típicamente corresponden a las inversas de las matrices de covarianza de los ruidos w y v .

El observador de información completa en el instante T se define como la estimación de la trayectoria del estado x_k de 0 a T a partir de las mediciones y_k . La estimación se obtiene a partir de la solución del siguiente problema de optimización con restricciones

(Ferrari-Trecate et al., 2002):

$$\min_{x_0, w} J(0, T, w, v, x_0, \Gamma_0) \quad (3.16)$$

sueto a (3.8), (3.9), (3.14).

El observador de información completa utiliza todas las mediciones disponibles para realizar la estimación, lo que es deseable pero a medida que T crece se hace impracticable su implementación computacional. Si se considera un horizonte fijo M , tal que $T > M$ el problema de optimización se puede reescribir como:

$$\min_{x_{T-M}, w} J(T - M, T, w, v, x_{T-M}, \bar{\Gamma}_{T-M}) \quad (3.17)$$

sueto a (3.8), (3.9), (3.14). Donde:

$$\bar{\Gamma}_{T-M}(z) = \min_{x_0, w, x_{T-M}=z} J(0, T - M, w, v, x_0, \bar{\Gamma}_0) \quad (3.18)$$

sueto a (3.8), (3.9), (3.14). La formulación 3.17 es ahora acotada por el horizonte fijo M . La función $\bar{\Gamma}_{T-M}$ se conoce como costo terminal o *arrival cost* y resume el efecto de la información obtenida en las mediciones $y_k, k = 0, \dots, T - M - 1$.

El estimador de horizonte móvil viene dado por resolver en cada paso de tiempo el problema de optimización dado por:

$$\min_{x_{T-M}, w} J(T - M, T, w, v, x_{T-M}, \Gamma_{T-M}) \quad (3.19)$$

sueto a (3.8), (3.9), (3.14). Donde Γ_{T-M} es una secuencia de penalizaciones que aproxima la función de costo terminal. Métodos para obtener la secuencia de penalizaciones se encuentran en (Ferrari-Trecate et al., 2002).

3.2.4. Estabilidad

La estabilidad del sistema realimentado es primordial en aplicaciones de control, por esto se debe contar con herramientas teóricas que permitan saber si un sistema es o no estable. En el caso de los sistemas híbridos los conceptos de estabilidad son equivalentes a aquellos aplicables a sistemas descritos únicamente por variables cuyo dominio es continuo.

Si se considera un sistema MLD dado por (3.2), (3.3) y (3.4), se definen las siguientes propiedades de estabilidad (Bemporad & Morari, 1999):

- Un vector $x_e \in \mathbb{R}^{n_c} \times \{0, 1\}^{n_l}$ se dice un estado de equilibrio para el sistema y la entrada $u_e \in \mathbb{R}^{m_c} \times \{0, 1\}^{m_l}$ si $[x'_e \ u'_e]'$ pertenece al dominio y $x(t, t_0, x_e, u_e) = x_e, \forall t > t_0$. El par (x_e, u_e) se dice un par de equilibrio.
- Dado un par de equilibrio (x_e, u_e) se dice un par estable si, dado $t_0 \in \mathbb{Z}, \forall \epsilon > 0 \exists \delta(\epsilon, t_0)$ tal que $\|x_0 - x_e\| \leq \delta \Rightarrow \|x(t, t_0, x_0, u_e) - x_e\| \leq \epsilon, \forall t > t_0$.
- Dado un par de equilibrio (x_e, u_e) se dice un par asintóticamente estable si es estable y además $\exists r > 0$ tal que $\forall x_0$ en una vecindad de x_e de radio r : $B(x_e, r)$ y $\forall \epsilon > 0, \exists T(\epsilon, t_0)$ tal que $\|x(t, t_0, x_0, u_e) - x_e\| \leq \epsilon, \forall t \geq T$.
- Dado un par de equilibrio (x_e, u_e) se dice un par exponencialmente estable si es asintóticamente estable y además $\exists \delta > 0, \alpha > 0, 0 < \beta < 1$ tal que $\forall x_0 \in B(x_e, \delta)$ se cumple $\|x(t, t_0, x_0, u_e) - x_e\| \leq \alpha \beta^{t-t_0} \|x_0 - x_e\|$.

Cabe notar que la convergencia asintótica del vector de componentes lógicas $x_l(t)$ a x_{l_e} es equivalente a la existencia de un tiempo t_e para el cual se cumpla $x_l(t) = x_{l_e}, \forall t > t_e$. Consecuentemente, las propiedades de estabilidad pueden ser verificadas solo para la parte continua $x_c(t)$ siempre que exista una vecindad alrededor del vector de equilibrio de la parte continua x_{c_e} tal que si se perturba $x_c(t)$ se mantenga la condición $x_l(t) = x_{l_e}$.

Para un par de equilibrio (x_e, u_e) en sistemas *bien definidos*, se puede establecer un valor de equilibrio para las variables auxiliares usando las funciones $\mathcal{D}$ y $\mathcal{L}$ definidas anteriormente. Además, para componentes indefinidas, se puede relajar el concepto de equilibrio utilizando la siguiente definición:

- Sea (x_e, u_e) un par de equilibrio para un sistema *bien definido*. Se asume que $\mathcal{F} = \lim_{t \rightarrow \infty} \mathcal{F}_t$ y $\mathcal{G} = \lim_{t \rightarrow \infty} \mathcal{G}_t$ existen. Para $i \in \mathcal{F}$, $j \in \mathcal{G}$ sean $\delta_{e,i}$ y $z_{e,j}$ los valores de equilibrio. Un vector auxiliar δ (o z) se dirá admisible si $\delta_i = \delta_{e,i}, \forall i \in \mathcal{F}, (z_j = z_{e,j}, \forall j \in \mathcal{G})$ y $\exists t_e$ tal que se satisface (3.4) $\forall t > t_e$.

3.3. Identificación de Sistemas Híbridos

El problema de identificación para modelos SARX se formula como sigue:

Dada una serie de N pares de datos entrada-salida (y_k, u_k) , $k = 1, \dots, N$, se desea estimar los ordenes del modelo n_a, n_b , el número de submodelos s y los vectores de parámetros asociados a cada submodelo θ_i , $i = 1, \dots, s$. Mas aún se desea estimar la función que determina el estado discreto $\sigma(k)$.

El problema general de identificación híbrida se puede particularizar tomando como conocida alguna de las incógnitas n_a, n_b o s , resultando un problema de menor complejidad (Paoletti et al., 2007), (Vidal, 2007).

Para modelos del tipo PWARX, el problema de identificación híbrida se formula como sigue:

Dada una serie de N pares de datos entrada-salida (y_k, u_k) , $k = 1, \dots, N$, estimar los ordenes del modelo n_a, n_b , el número de submodelos s , los vectores de parámetros asociados a cada submodelo θ_i , $i = 1, \dots, s$ y las regiones que conforman la partición del dominio $\mathcal{R}_i$, $i = 1, \dots, s$. Cabe notar que en el caso de modelos PWARX, la estimación de la partición implica la estimación de la función que determina el valor del estado discreto $\sigma(k)$ de acuerdo a (3.13).

Nuevamente el problema general se puede particularizar dando como conocidas algunas de las incógnitas, resultando un problema de menor complejidad (Paoletti et al., 2007).

Para modelos SARX/PWARX formulados en el espacio de estado según (3.8) y (3.9), el problema general de identificación híbrida se formula como:

Dada una serie de N pares de datos entrada-salida (y_k, u_k) , $k = 1, \dots, N$, estimar el orden del modelo n , el número de submodelos s , y las 6-tuplas $(A_i, B_i, C_i, D_i, f_i, g_i)$, $i = 1, \dots, s$. Mas aún, estimar la función que determina el valor de $\sigma(k)$, lo que en el caso de modelos PWARX corresponde a estimar las regiones Ω_i , $i = 1, \dots, s$.

A continuación se describen 3 procedimientos para la identificación de sistemas híbridos con modelos del tipo SARX/PWARX, conocidos como la estrategia basada en clustering, la estrategia bayesiana y la estrategia algebraica. Cabe destacar que existen otros procedimientos para la identificación de sistemas híbridos con modelos SARX/PWARX y de otras clases de modelos (Paoletti et al., 2007), pero el análisis se centrará en los mencionados anteriormente, lo que no significa pérdida de generalidad debido a la propiedad de equivalencia entre modelos híbridos.

3.3.1. Estrategia Basada en Clustering

La estrategia basada en *clustering* aprovecha el hecho de que el mapa PWA es localmente lineal. Si los datos fueron generados por un modelo del tipo (3.12), existen grupos de vectores regresores vecinos que pertenecen a la misma región. Los vectores de parámetros que se calculan para estos pequeños grupos de datos locales deben contener el vector de parámetros del correspondiente submodelo. Luego, la información acerca de los submodelos puede obtenerse mediante *clustering* de los vectores de parámetros locales.

La estrategia basada en *clustering* procede como sigue, considerando c como un parámetro fijo del método (Paoletti et al., 2007).

- Regresión local: Para $k = \bar{n}, \dots, N$, se construye un conjunto $\mathcal{C}_k$ considerando (y_k, r_k) y los $c - 1$ vectores regresores r_j mas cercanos a r_k , según norma euclidiana. Posteriormente, mediante mínimos cuadrados se calcula el vector local de parámetros θ_k^{LS} para cada set $\mathcal{C}_k$
- Construcción de los vectores característicos: Cada dato (y_k, r_k) se mapea en el vector característico:

$$\xi_k = [(\theta_k^{LS})^T m_k^T]^T \quad (3.20)$$

donde $m_k = \frac{1}{c} \sum_{(y,r) \in \mathcal{C}_k} r$ es el centro de $\mathcal{C}_k$.

- Clustering: Los vectores característicos se agrupan en s grupos $\{\mathcal{F}_i\}_{i=1}^s$ aplicando el algoritmo *K-means* (Paoletti et al., 2007) y asignando valores de confianza apropiados a cada vector característico.
- Estimación de parámetros: Como el mapeo de los datos en los vectores característicos es una función biyectiva, los vectores de datos están clasificados en *clusters* $\{\mathcal{D}_i\}_{i=1}^s$ según la regla:

$$(y_k, r_k) \in \mathcal{D}_i \leftrightarrow \xi_k \in \mathcal{F}_i \quad (3.21)$$

luego el vector de parámetros θ_i para cada *cluster* $\mathcal{D}_i$ se obtiene mediante mínimos cuadrados.

La estrategia basada en *clustering* necesita que los ordenes del modelo n_a y n_b , además el número de submodelos s sean fijos. El parámetro c , que define la cardinalidad de los subespacios, es el principal parámetro de sintonía del método.

3.3.2. Estrategia Bayesiana

La estrategia bayesiana se basa en la idea de aprovechar conocimiento *a-priori* acerca de los modos de operación y los parámetros del sistema híbrido. Los vectores de parámetros θ_i se consideran variables aleatorias, y se describen mediante su función de densidad $p\theta_i(\cdot)$.

El conocimiento previo se incorpora mediante la elección apropiada de las funciones de densidad $p\theta_i(\cdot)$ de los vectores de parámetros. Las propiedades de los parámetros, como la esperanza o la máxima probabilidad *a-posteriori*, se obtienen a partir de la función de densidad.

El problema de clasificación de datos (estimación de los modelos) y estimación de parámetros, se resuelve procesando secuencialmente los datos de proceso obtenidos. En cada iteración se actualiza la función de densidad de uno de los vectores de parámetros, usando un algoritmo iterativo sub-óptimo que asume que la distribución $p_e(\cdot)$ del término estocástico aditivo e_k es conocida (Paoletti et al., 2007).

Sea $p\theta_i(\cdot, k)$ la función de densidad del vector θ_i en la iteración k . Cuando se considera el vector de datos (y_k, r_k) , la función de densidad condicional viene dada por:

$$p((y_k, r_k) | \sigma(k) = i) = \int_{\Theta_i} p((y_k, r_k) | \tilde{\theta}) p\theta_i(\tilde{\theta}, k-1) d\tilde{\theta} \quad (3.22)$$

donde Θ_i es el dominio de θ_i y $p((y_k, r_k) | \theta) = p_e(y_k - \theta^T \varphi_k)$. El estado discreto correspondiente a (y_k, r_k) se estima como $\sigma(k) = i^*$. Donde:

$$i^* = \arg \max_{i=1, \dots, s} p((y_k, r_k) | \sigma(k) = i) \quad (3.23)$$

Luego, la pertenencia de (y_k, r_k) al modo i^* se usa para actualizar la densidad de probabilidad de θ_{i^*} empleando la regla de Bayes:

$$p\theta_{i^*}(\theta, k) = \frac{p((y_k, r_k) | \theta) p\theta_{i^*}(\theta, k-1)}{\int_{\Theta_{i^*}} p((y_k, r_k) | \tilde{\theta}) p\theta_{i^*}(\tilde{\theta}, k-1) d\tilde{\theta}} \quad (3.24)$$

Las densidades de los otros vectores de parámetros quedan inalteradas, es decir:

$$p\theta_i(\cdot, k) = p\theta_i(\cdot, k-1) \quad \forall i \neq i^* \quad (3.25)$$

La implementación numérica del algoritmo, requiere la utilización de filtros de partículas (Paoletti et al., 2007).

La estrategia bayesiana requiere que los órdenes de los modelos n_a y n_b además del número de submodelos s sean dados. Los parámetros de sintonía del método son las densidades *a priori* de los vectores de parámetros $p_{\theta_i}(\cdot, 0)$ y la densidad $p_e(\cdot)$ del término estocástico.

3.3.3. Estrategia Algebraica

La estrategia algebraica resuelve el problema de identificación de modelos SARX formulándolo como un problema algebraico geomético. Resulta en una solución cerrada que es exacta en ausencia de ruido.

La idea principal detrás de esta estrategia, es considerar la identificación de múltiples modelos ARX como la identificación de un único modelo ARX *levantado* (“lifted”) que codifica simultáneamente a todos los submodelos y no depende de la secuencia de switching. Los parámetros de este modelo pueden obtenerse mediante técnicas clásicas de identificación de modelos lineales luego de aplicar una codificación polinomial a los vectores regresores. Los parámetros de los submodelos originales se obtienen a partir de las derivadas de éste polinomio codificador.

Asumiendo que el número de submodelos s y los ordenes n_a y n_b son conocidos (luego se levantará esta condición), la estrategia algebraica procede como sigue (Paoletti et al., 2007):

Si los datos fueron generados por un modelo del tipo (3.12) con $e_k = 0$, cada par (y_k, r_k) satisface:

$$y_k - \theta_i^T \varphi_k = 0 \quad (3.26)$$

para algún θ_i , $i = 1, \dots, s$. Luego, la siguiente igualdad se cumple para todo k

$$\prod_{i=1}^s (b_i^T z_k) = 0 \quad (3.27)$$

donde $b_i = [1 \ \theta_i^T]^T$ y $z_k = [-y_k \ \varphi_k^T]^T$. La ecuación (3.27) se conoce como la *restricción de desacoplamiento híbrido* (Paoletti et al., 2007), (Vidal, Soatto, & Chiuso, 2007) y es independiente de la secuencia de switching y el mecanismo de transición del modelo. Tomando

en cuenta (3.27), se define el *polinomio de desacoplamiento híbrido* como:

$$p_s(z) = \prod_{i=1}^s (b_i^T z) = h^T v_s(z) \quad (3.28)$$

Éste es un polinomio homogéneo de grado s en $z = [z_1 \dots z_K]^T$, $K = n_a + n_b + 3$. Cabe notar que (3.28) se puede escribir como una combinación lineal de todos los monomios de grado s en K variables. Tales monomios están contenidos en el vector $v_s(z)$. El vector h contiene los llamados *parámetros del modelo híbrido* y codifica los parámetros de los s submodelos. Considerando que (3.27) se cumple para todo k , el vector h se puede estimar resolviendo el siguiente sistema lineal en el caso sin ruido:

$$L_s(K)h = 0 \quad (3.29)$$

donde $L_s(K) = [v_s(z_{\bar{n}})v_s(z_{\bar{n}+1}) \dots v_s(z_N)]^T$. En el caso con ruido, $e_k \neq 0$, h debe estimarse usando mínimos cuadrados sobre el sistema (3.29). Una vez que el vector h ha sido determinado, los vectores b_i pueden obtenerse como:

$$b_i = \frac{Dp_s(z_{k_i})}{[1 \ 0 \dots 0]Dp_s(z_{k_i})} \quad (3.30)$$

donde $Dp_s(z) = \frac{\partial p_s(z)}{\partial z}$ y z_{k_i} es un dato generado por el i -ésimo submodelo ARX, el cual puede escogerse arbitrariamente una vez que $p_s(\cdot)$ es conocido.

El sistema lineal (3.29) tiene una solución única (imponiendo que el primer elemento de h sea 1) cuando la excitación de los datos es suficiente (Paoletti et al., 2007) y s , n_a y n_b son conocidos. Si s es desconocido, se puede estimar según (Vidal, 2007), (Paoletti et al., 2007):

$$s = \arg \min \{i : \text{rank}(L_i(K)) = M_i(K) - 1\} \quad (3.31)$$

donde $M_i(K) = \binom{s + K - 1}{s}$. En el caso ruidoso, es necesario incluir un umbral para estimar el rango de $L_i(K)$.

3.4. Control Predictivo de Sistemas Híbridos

3.4.1. Descripción General

El control predictivo de sistemas híbridos, tiene como principal característica el uso de un modelo híbrido del sistema para el cálculo de las predicciones. Aprovecha las ventajas de la modelación híbrida combinándolas con las potencialidades del control predictivo clásico.

A continuación se presenta la formulación del problema de control predictivo híbrido para un sistema modelado con un modelo del tipo *Switched Affine*:

Consideremos un sistema cuyo modelo viene dado por:

$$x_{k+1} = A_{\sigma(k)}x_k + B_{\sigma(k)}u_k + f_{\sigma(k)} + w_k \quad (3.32)$$

$$y_k = C_{\sigma(k)}x_k + D_{\sigma(k)}u_k + g_{\sigma(k)} + v_k \quad (3.33)$$

donde x_k , u_k , y_k y $\sigma(k)$ son el estado continuo, la entrada, la salida y el estado discreto respectivamente.

Se busca la secuencia de controles $\{u_{k+j}\}_{j=1}^{N_u}$, que es solución del siguiente problema de optimización:

$$\begin{aligned} \min_{\{u_{k+j}\}_{j=1}^{N_u}} J(N_1, N_2, N_u) = & \sum_{j=N_1}^{N_2} \delta_j [\hat{y}_{k+j} - yr_{k+j}]^2 + \\ & \sum_{j=1}^{N_u} \lambda_j [\Delta u_{k+j-1}]^2 + \sum_{j=1}^{N_u} \gamma_j [u_{k+j-1} - ur_{k+j-1}]^2 \end{aligned} \quad (3.34)$$

sujeto a restricciones de dominio y al cumplimiento del modelo híbrido.

El problema enunciado tiene una estructura discontinua debido al estado discreto, que dificulta la implementación computacional. Por esto se debe utilizar una estructura equivalente del modelo *Switched Affine* basada en desigualdades que incorporan variables binarias, resultando un problema del tipo *mixed-integer programming* que es tratable computacionalmente.

3.4.2. Formulación Equivalente de Modelos Switched Affine

Un modelo *Switched Affine* se puede considerar como la interconexión de sistemas continuos y discretos según muestra la figura 3.3. En esta figura δ_k es el evento discreto, $\sigma(k)$ es el estado discreto, x_k^c es el vector de estados continuos, x_k^l es el vector de estados discretos, u_k^c es el vector de entradas continuas y u_k^l es el vector de entradas discretas.

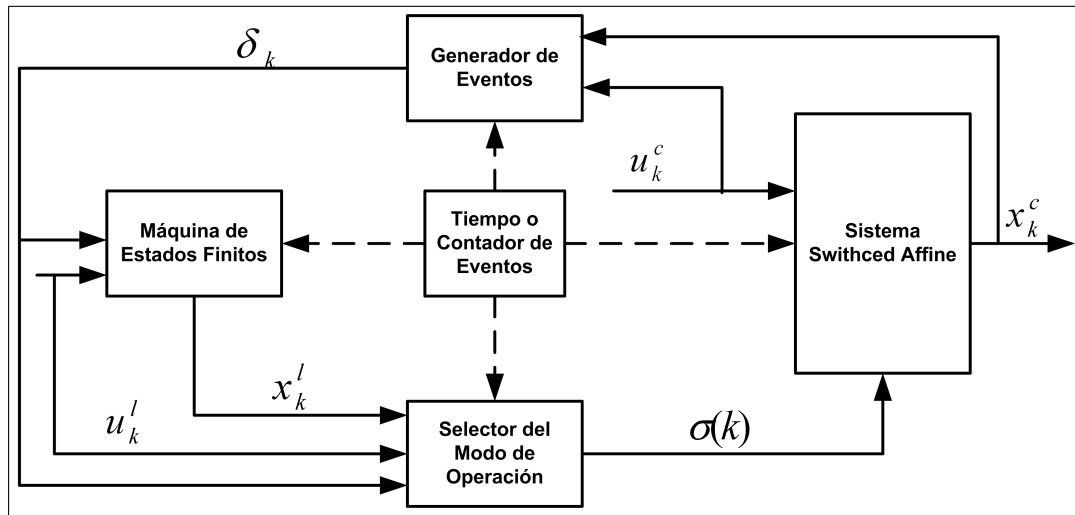


FIGURA 3.3. Esquema de un modelo *switched affine*

La interconexión de bloques de la figura 3.3 permite separar el efecto de las componentes continuas y discretas, donde cada bloque tiene una función específica que se detalla a continuación.

Sistema Switched Affine.: El sistema Switched Affine se puede considerar como un conjunto de modelos continuos indexados por switches dependientes del estado discreto $\sigma(k)$ según muestra la figura 3.4.

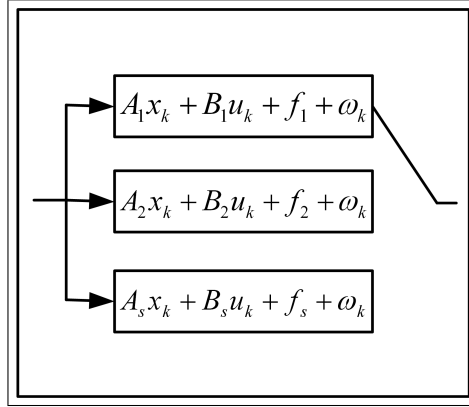


FIGURA 3.4. Sistema *Switched Affine*.

Luego la dinámica del sistema se puede modelar como:

$$x_{k+1}^c = \sum_{i=1}^s z_k^i \quad (3.35)$$

donde $z_k^i = A_i x_k + B_i u_k + f_i + w_k$ si $\sigma(k) = i$ y $z_k^i = 0$ si $\sigma(k) \neq i$.

Es decir el sistema Switched Affine integra un sistema continuo y reglas del tipo *if-then* dependientes del estado discreto $\sigma(k)$.

Generador de Eventos: El generador de eventos es el encargado de generar señales discretas que inducen cambios en la máquina de estados asociada. Los eventos discretos son una función de las entradas y salidas continuas del sistema, generalmente dadas por particiones de los dominios asociados:

$$[\delta_k^i = 1] \leftrightarrow [H^i x_k^c + K^i u_k^i \leq W^i] \quad (3.36)$$

Máquina de Estados Finitos: La máquina de estados finitos es la encargada de determinar el estado de operación del sistema, en la forma de un vector de estados discretos x_k^l , basándose en las salidas del generador de eventos y las entradas discretas. El estado evoluciona según una ecuación lógica de transición de estado, o bien según una probabilidad de transición en el caso estocástico. La naturaleza de la ecuación de evolución del estado repercutirá en el objetivo del

control predictivo, ya que de no ser forzable se realizará un control subóptimo en el estado actual, en lugar de un control óptimo forzando el estado óptimo.

Selector del Estado de Operación: El selector del estado de operación es el encargado de indexar los modelos continuos del sistema, basándose en el estado actual de la máquina de estados, las entradas discretas y los eventos generados. La salida corresponde al estado discreto del sistema $\sigma(k)$.

Si el modelo *Switched Affine* se considera como la interconexión de bloques propuesta, surge naturalmente la formulación equivalente como un modelo *MLD* (Bemporad & Morari, 1999), lo que permite formular el problema de control predictivo híbrido como un problema de optimización del tipo *mixed-integer programming* (Bemporad & Morari, 1999).

Las principales características del control predictivo híbrido se resumen a continuación:

- Uso de un modelo híbrido para el cálculo de las predicciones.
- Determinación de la ley de control a partir de un problema de optimización.
- Problema de optimización resultante del tipo *mixed-integer programming*.
- Aplicación del principio *receding horizon control*.
- Control óptimo o subóptimo en el estado dependiendo de las características de la máquina de estados asociada.

3.4.3. Estabilidad en Lazo Cerrado

La estabilidad en lazo cerrado tiene relación con la capacidad del controlador predictivo de llevar y mantener al sistema a la referencia, o a una vecindad acotada arbitrariamente pequeña de ésta.

Consideremos el problema de control predictivo para sistemas modelados por una estructura MLD dado por:

$$\min_{\{v\}_0^{T-1}} J(\{v\}_0^{T-1}, x_t) = \sum \|v_k - u_e\|_{Q_1}^2 + \|\delta_k - \delta_e\|_{Q_2}^2 + \|z_k - z_e\|_{Q_3}^2 + \quad (3.37)$$

$$\|x_k - x_e\|_{Q_4}^2 + \|y_k - y_e\|_{Q_5}^2$$

sujeto a $x_T = x_e$, (3.2), (3.3) y (3.4). De existir la solución, la ley de control sigue el *receding horizon principle* $u_t = v_0^*$, donde v_0^* es el primer elemento de la secuencia solución del problema de control predictivo.

Sea (x_e, u_e) un par de equilibrio (estable o inestable) y (δ_e, z_e) admisibles. Asumiendo que el estado inicial x_0 es tal que existe una solución para el problema de control predictivo, entonces $\forall Q_i$ simétrica y positiva definida la ley de control predictivo estabiliza el sistema según:

$$\begin{aligned} \lim_{t \rightarrow \infty} x_t &= x_e \\ \lim_{t \rightarrow \infty} u_t &= u_e \\ \lim_{t \rightarrow \infty} \|\delta_t - \delta_e\|_{Q_2} &= 0 \\ \lim_{t \rightarrow \infty} \|z_t - z_e\|_{Q_3} &= 0 \\ \lim_{t \rightarrow \infty} \|y_t - y_e\|_{Q_5} &= 0 \end{aligned} \tag{3.38}$$

cumpliendo con las restricciones $x_T = x_e$, (3.2), (3.3) y (3.4) $\forall t$. La demostración de la proposición anterior se deriva de los argumentos de Lyapunov y puede ser revisada en (Bemporad & Morari, 1999).

3.5. Ejemplo de Control Predictivo Híbrido

3.5.1. Sistema en Estudio

El sistema en estudio corresponde al modelo dado por (3.39), el cual puede formularse equivalentemente según una estructura MLD o PWA. Para el estudio se utilizará una estructura MLD y se analizarán sus propiedades de estabilidad además de un esquema de control predictivo híbrido.

$$x(t+1) = 1.6z(t) - 0.8x(t) + u(t) \quad (3.39)$$

$$z(t) \leq 10\delta_1(t)$$

$$z(t) \geq -10\delta_1(t)$$

$$z(t) \leq x(t) + 10(1 - \delta_1(t))$$

$$z(t) \geq x(t) - 10(1 - \delta_1(t))$$

El sistema en estudio, en su estructura MLD, consta de un estado continuo $x(t)$, que se mide directamente, y 2 variables lógicas auxiliares $\delta_1(t)$ y $z(t)$. Para las simulaciones en lazo cerrado, se añadirá una perturbación estocástica aditiva a $x(t)$.

3.5.2. Análisis de Estabilidad en Lazo Abierto

De las definiciones de estabilidad, las ecuaciones del modelo y las restricciones de dominio, se aprecia que los pares de equilibrio del sistema se pueden determinar en función de la señal de control u como: $(x_e, u_e) = (u/0.2, u)$ para $u \in [0 \ 1]$ y de la forma $(x_e, u_e) = (u/1.8, u)$ para $u \in [-1 \ 0]$. Luego un punto candidato a punto de equilibrio debe cumplir con : $x \in [-0.5556 \ 5]$. De las definiciones de estabilidad es claro que todos los puntos de equilibrio cumplen con ser exponencialmente estables. Un punto cualquiera del dominio (x, u) será atraído por el punto de equilibrio $(x_e, u_e) = (u/0.2, u)$ para $u > 0$, mientras que para $u < 0$ el sistema convergerá al punto de equilibrio $(x_e, u_e) = (u/1.8, u)$.

Para un punto de equilibrio (x_e, u_e) , los valores de equilibrio de las variables auxiliares serán $(\delta_{e,1}, z_e) = (0, 0)$ si $x_e < 0$, o bien $(\delta_{e,1}, z_e) = (1, x_e)$ si $x_e \geq 0$.

3.5.3. Control Predictivo Híbrido

Ya estudiada la estabilidad del sistema en lazo abierto, se procede a diseñar un esquema de control predictivo híbrido que lleve al sistema a una referencia dada.

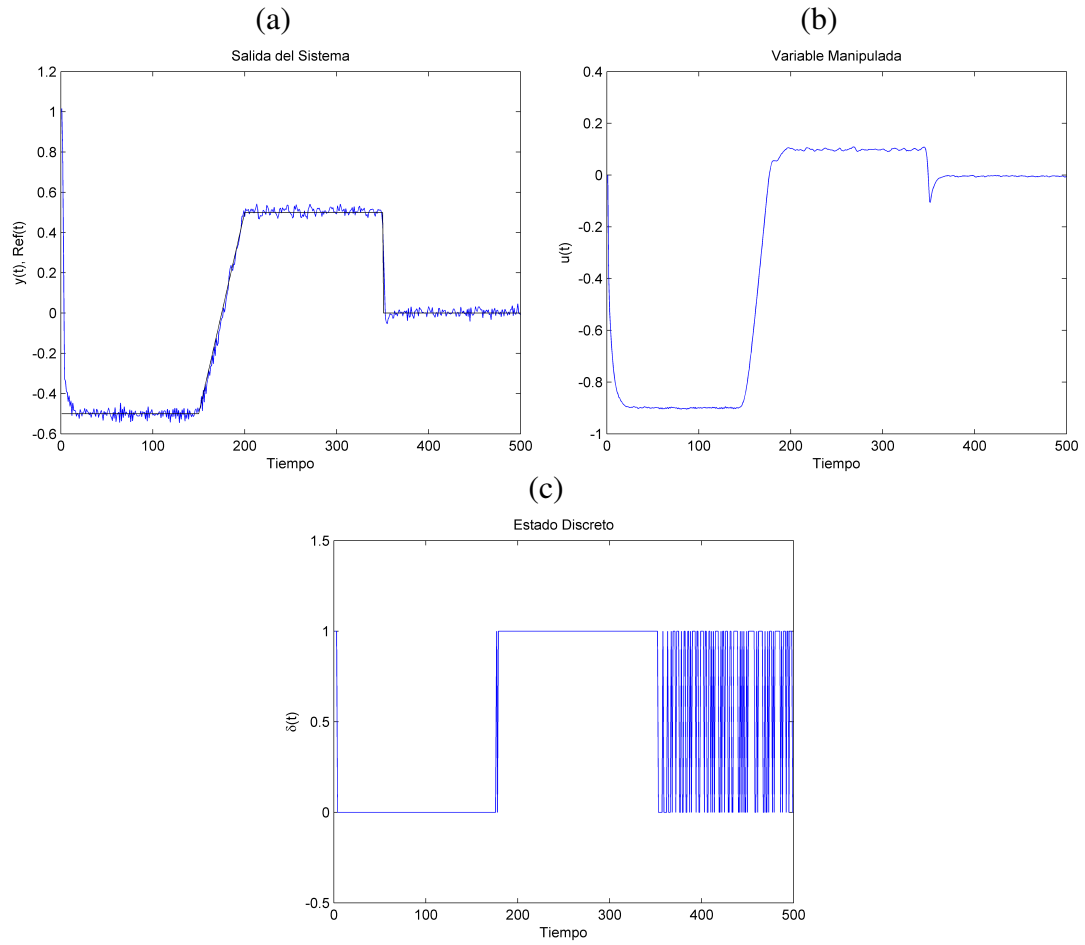


FIGURA 3.5. Resultados del control predictivo híbrido para el modelo MLD; (a): Salida del sistema (Azul) y referencia (Negro); (b): Evolución de la variable manipulada; (c): Evolución de la variable auxiliar δ_1 .

La propiedad de estabilidad en lazo cerrado, establece que el controlador predictivo logrará estabilizar al sistema en torno a la referencia para un par de equilibrio y un vector auxiliar admisible, si el estado inicial es tal que el problema de control predictivo tiene solución. Si como ejemplo se considera $x(t) = -0.5$, se tiene que el par de equilibrio es $(-0.5, -0.9)$ y el vector de variables auxiliares admisible es $(0, 0)$. Asimismo si se considera $x(t) = 0.5$, el par de equilibrio es $(0.5, 0.1)$ y el vector admisible es $(1, 0.5)$. Consideremos además el caso trivial $x(t) = 0$, donde el par de equilibrio es $(0, 0)$ y el vector admisible es $(0, 0)$. Según la propiedad de estabilidad en lazo cerrado, es directo que el controlador predictivo estabilizará al sistema en torno a los puntos analizados para

cualquier estado inicial.

Consideremos ahora $x(t) = -1$. Del análisis de estabilidad previo se tiene que no existe una señal de control factible que complete el par de equilibrio y que no viole las restricciones del modelo. Luego el controlador predictivo no será capaz de estabilizar al sistema en torno a $x(t) = -1$.

Con el fin de evaluar un sistema de control predictivo híbrido y comprobar el análisis de estabilidad en lazo cerrado, se realizaron simulaciones del sistema en lazo cerrado para un horizonte $t = 500$ con un periodo de muestreo unitario. Para el controlador predictivo se utilizó un horizonte de predicción $T = 10$. Los resultados obtenidos se presentan en las figuras 3.5 y 3.6 para señales de referencias variables en el tiempo.

Se aprecia que el controlador predictivo es capaz de compensar las perturbaciones estocásticas y logra llevar al sistema en torno a la referencia. Durante el cambio de referencia en forma de rampa el controlador es capaz de calcular el cambio en la señal de control para mantener al sistema en torno a la referencia; esto es posible debido a que la trayectoria descrita por la rampa está compuesta por puntos $x(t)$ para los cuales es posible obtener una señal de control $u(t)$ tal que el par $(x(t), u(t))$ sea un par de equilibrio. Sin embargo, como lo indicó el análisis de estabilidad en lazo cerrado, el controlador no es capaz de mantener al sistema en la referencia $x(t) = -1$ (figura 3.6) debido a que no es posible encontrar una señal de control $u(t)$ dentro del dominio del sistema tal que el par $(x(t), u(t))$ sea par de equilibrio, por ende el problema de control predictivo híbrido no tiene solución.

En las figuras se aprecia que la perturbación estocástica añadida, provoca la oscilación de la variable auxiliar δ_1 cuando se lleva al sistema en torno a $x(t) = 0$. Además se aprecia que la variable manipulada presenta saturación al tratar de llevar al sistema al punto $x(t) = -1$, lo que provoca que el sistema se estabilice en torno al punto factible más

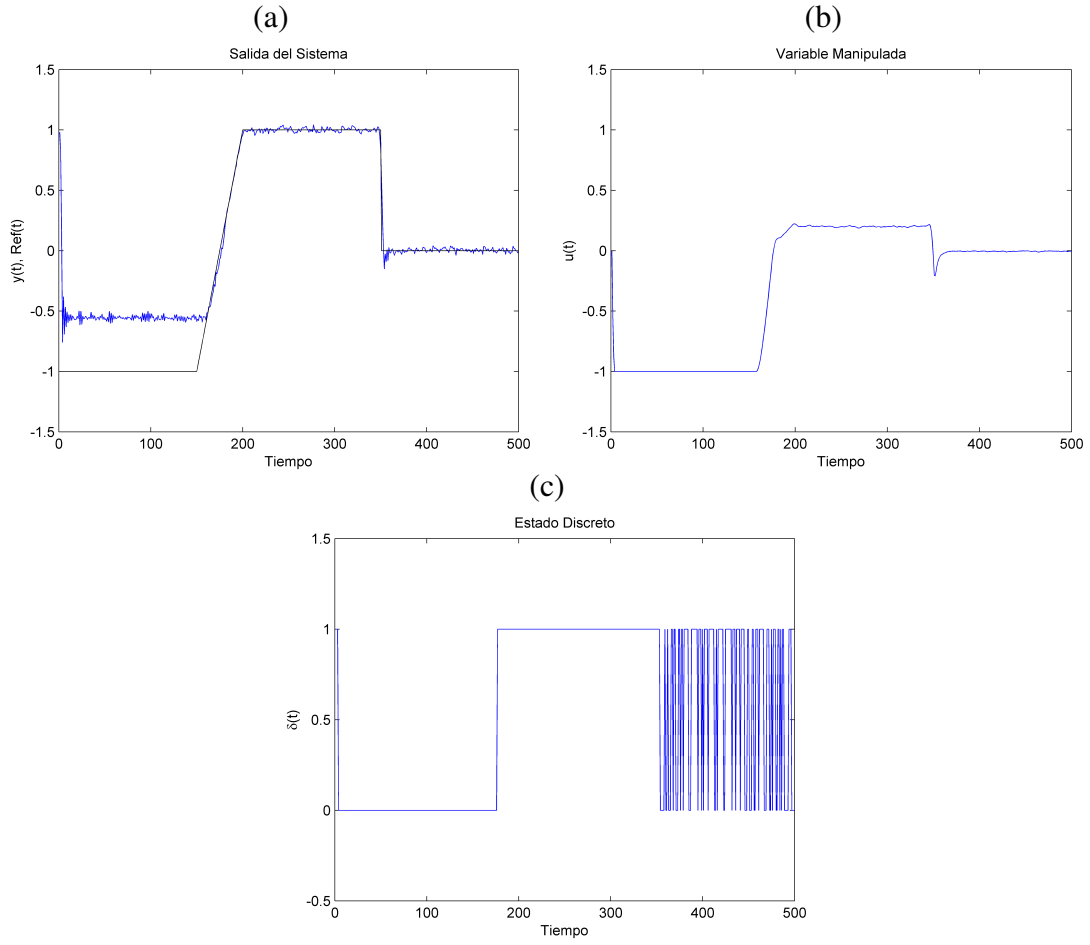


FIGURA 3.6. Resultados del control predictivo híbrido para el modelo MLD fuera de equilibrio; (a): Salida del sistema (Azul) y referencia (Negro); (b): Evolución de la variable manipulada; (c): Evolución de la variable auxiliar δ_1 .

cercano a la referencia: $x(t) = -0.5556$. La variable auxiliar $z(t)$ por definición tomará el valor $x(t)$ si $\delta_1(t) = 1$ o 0 si $\delta_1(t) = 0$, por lo que la perturbación estocástica añadida provocará discontinuidades cuando se lleva al sistema al estado $x(t) = 0$.

4. MODELOS DINÁMICOS DE PROCESOS VISUALES

4.1. Sensores Visuales

Un sensor visual entrega información del entorno en la forma de arreglos bidimensionales de píxeles en tiempo continuo o tiempo discreto. Se basa en la forma de adquisición de información de los humanos mediante los ojos; sin embargo, su espectro de visión puede superar al de los humanos.

4.1.1. Sensores de Espectro Visible

Los sensores de espectro visible, recogen información de la intensidad de la luz del entorno en el espectro visible para los humanos, el cual va entre 400 y 700 *nm* de longitud de onda.

Los sensores de espectro visible se basan en elementos electrónicos que convierten un patrón de luz en señales eléctricas, las cuales representan la intensidad de la luz presente en el entorno. Los más comunes son los llamados *Charge coupled device (CCD)* que utilizan elementos semiconductores fotosensibles en forma de arreglos matriciales. Los receptores activos de este arreglo se distribuyen en píxeles individuales. En un sensor CCD se transforma la energía de la luz incidente en energía eléctrica. La carga eléctrica almacenada en la celda es posteriormente transportada utilizando un registro de desplazamiento (shift register) para conformar una señal de vídeo, tal como se muestra en la figura 4.1.

Cabe señalar que en los sensores CCD se discretiza la imagen en píxeles; sin embargo, el valor de la carga eléctrica almacenado en cada celda no se digitaliza en el arreglo CCD, sino en una conversión posterior realizada por un conversor análogo-digital.

Las cámaras del tipo CCD son consideradas como dispositivos que poseen una muy baja deformación geométrica de la imagen, una muy buena sensibilidad a la luz y una muy buena resolución llegando hasta los 400.000 píxeles³, mientras que el tiempo de captura de la imagen puede llegar a 0.0001 s.

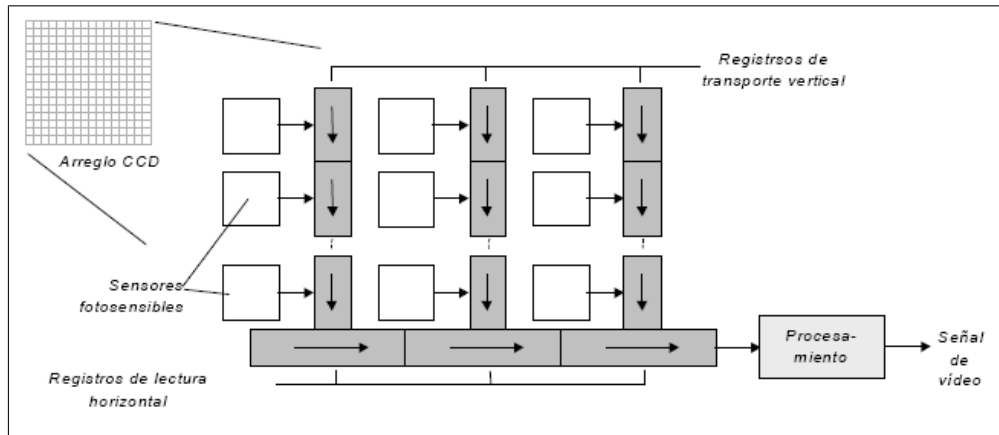


FIGURA 4.1. Esquema general de un sensor visual CCD.

Con el objeto de captar imágenes en colores, se han desarrollado métodos que discriminan la longitud de onda de la luz incidente mediante filtros. Este método consiste en formar una malla de elementos fotosensibles intercalando elementos sensibles a los 3 colores base rojo, verde y azul. De esta forma se obtienen 3 señales independientes que posteriormente se procesan para generar la imagen en colores.

4.1.2. Otros Sensores

En ciertos procesos interesa captar señales visuales que entregan información de variables de proceso, sin embargo, muchas veces éstas señales visuales están fuera del espectro visible. Un claro ejemplo es la radiación energética de los cuerpos que está directamente relacionada con la temperatura.

Para solucionar este problema se han desarrollado dispositivos capaces de captar señales luminosas de longitudes de onda fuera del rango visible, como por ejemplo cámaras infrarrojas. Este tipo de sensores no se utilizarán en los estudios presentados en esta tesis, pero al poseer el mismo principio de funcionamiento que los sensores de espectro visible, su uso no es una limitante para el desarrollo de un sistema de control predictivo híbrido basado en sensores visuales.

4.2. Texturas Dinámicas

Luego de la adquisición de la imagen, o secuencia de imágenes, por parte del sensor visual, se debe proceder a la extracción de características de éstas. Una imagen digital se interpreta como una matriz de píxeles dispuestos como en la figura 4.2. Cada píxel equivale a un número positivo, que es función de las propiedades de iluminación y movimiento de la escena, así como, de las propiedades físicas del sensor visual. Tomando en cuenta lo anterior, surge el concepto de textura que se puede entender como la realización de un proceso estocástico estacionario (Doretto, Chiuso, Wu, & Soatto, 2003) que resulta en los valores de cada píxel.

Para secuencias de imágenes es claro que cada realización, que da origen a cada cuadro, no es independiente de las otras; luego existe una coherencia temporal que debe ser incorporada. Es así como surge el concepto de textura dinámica o textura variante en el tiempo (Doretto et al., 2003).

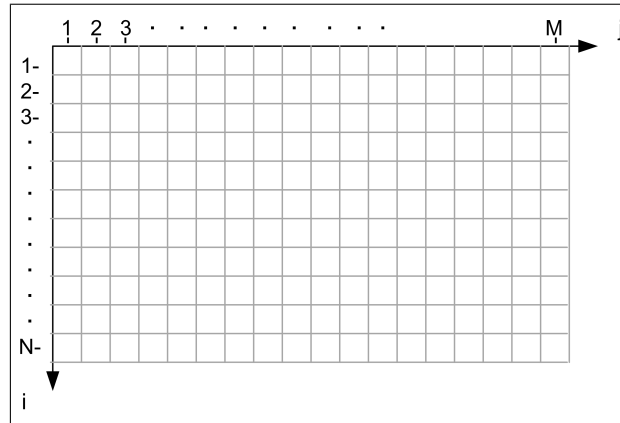


FIGURA 4.2. Esquema de una imagen digital de NxM píxeles

4.2.1. Modelación de Texturas Dinámicas

Sea $\{I(t)\}_{t=1}^{\tau}$, $I(t) \in \mathbb{R}^m$ una secuencia de τ imágenes. En cada instante t se adquiere una medición ruidosa de la imagen $y(t) = I(t) + w(t)$ donde $w(t)$ es una secuencia *iid* obtenida desde una distribución conocida $p_w(\cdot)$, luego se cuenta con una secuencia de mediciones $\{y(t)\}_{t=1}^{\tau}$, $y(t) \in \mathbb{R}^m$. Se dice que $\{y(t)\}_{t=1}^{\tau}$ es una textura lineal dinámica

si existe un conjunto de n funciones espaciales $\phi_\alpha : \mathbb{R} \mapsto \mathbb{R}^m$, $\alpha = 1, \dots, n$ y una distribución estacionaria $q(\cdot)$ tal que, definiendo $x(t) \in \mathbb{R}^n$ que cumple $I(t) = \phi(x(t))$, se tiene:

$$x(t) = \sum_{i=1}^k A_i x(t-i) + Bv(t) \quad (4.1)$$

donde $v(t)$ es una realización *iid* de la distribución $q(\cdot)$.

La expresión anterior implica que una textura lineal dinámica se puede asociar a la salida de un modelo ARMA, el que se puede formular en el espacio de estados como:

$$x(t+1) = Ax(t) + Bw(t) \quad (4.2)$$

$$y(t) = \phi(x(t)) + w(t) \quad (4.3)$$

con $x(0) = x_0$ y $w(t) \sim p_w(\cdot)$, $v(t) \sim q(\cdot)$.

La formulación anterior corresponde a texturas dinámicas lineales, pero es fácilmente extensible a modelos no lineales del tipo $x(t+1) = f(x(t), v(t))$, surgiendo el concepto de textura dinámica no lineal. Sin embargo, se ha demostrado que las texturas lineales abarcan un amplio espectro de procesos visuales (Doretto et al., 2003), (Vidal, 2007), (Vidal et al., 2007).

La definición anterior de texturas dinámicas involucra la elección de funciones espaciales ϕ_α , $\alpha = 1, \dots, n$. Existen diferentes criterios para la elección de las funciones, pero es claro verlo como un paso de reducción dimensional buscando la descomposición lineal de la imagen como:

$$I(t) = \sum_{i=1}^n x_i(t) \phi_i(t) \equiv Cx(t) \quad (4.4)$$

donde $C = [\phi_1, \dots, \phi_n] \in \mathbb{R}^{m \times n}$ es una base ortonormal, típicamente un conjunto de n componentes principales (Doretto et al., 2003), (Vidal et al., 2007).

Si se considera el proceso visual como un proceso estocástico estacionario de segundo orden y C como un conjunto de n componentes principales, se puede modelar la textura dinámica como la salida de un sistema excitado por ruido blanco Gaussiano de media 0 y covarianza arbitraria, lo que representa una aproximación que resulta en una caracterización

sub-óptima de la textura (Doretto et al., 2003). Luego el modelo ARMA asociado a la textura dinámica aproximada es:

$$x(t+1) = Ax(t) + v(t) \quad (4.5)$$

$$y(t) = Cx(t) + w(t) \quad (4.6)$$

donde $v(t) \sim \mathcal{N}(0, Q)$, $w(t) \sim \mathcal{N}(0, R)$.

4.2.2. Identificación de Texturas Dinámicas

Dada una secuencia de mediciones ruidosas $\{y(t)\}_{t=1}^{\tau}$, el aprendizaje de una textura dinámica consiste en identificar las matrices A , C , Q y R del modelo ARMA asociado definido por (4.5), (4.6). Para el problema de identificación anterior, es posible encontrar una solución cerrada siguiendo el siguiente procedimiento (Doretto et al., 2003):

Sea $Y_1^{\tau} \equiv [y(1), \dots, y(\tau)] \in \mathbb{R}^{m \times \tau}$ y $X_1^{\tau} \equiv [x(1), \dots, x(\tau)] \in \mathbb{R}^{n \times \tau}$ con $n > \tau$, y $W_1^{\tau} \equiv [w(1), \dots, w(\tau)] \in \mathbb{R}^{m \times \tau}$, luego:

$$Y_1^{\tau} = CX_1^{\tau} + W_1^{\tau} ; C \in \mathbb{R}^{m \times n} ; C^T C = I \quad (4.7)$$

Ahora sea $Y_1^{\tau} = U\Sigma U^T$, $U \in \mathbb{R}^{m \times n}$, $UU^T = I$; $V \in \mathbb{R}^{\tau \times n}$, $VV^T = I$ la descomposición en valores singulares (SVD) de Y_1^{τ} . Si se considera el problema de encontrar la estimación óptima, en el sentido de Frobenius:

$$\hat{C}(\tau), \hat{X}(\tau) = \arg \min_{C, X_1^{\tau}} \|W_1^{\tau}\|_F \quad (4.8)$$

sujeto a (4.7), la propiedad de aproximación de rango fijo de la SVD entrega la única solución óptima:

$$\hat{C}(\tau) = U \quad \hat{X}(\tau) = \Sigma V^T \quad (4.9)$$

$\hat{A}(\tau)$ puede ser determinada en forma única, nuevamente en el sentido de Frobenius, solucionando el problema de optimización lineal dado por:

$$\hat{A}(\tau) = \arg \min_A \|X_2^{\tau} - AX_1^{\tau-1}\|_F \quad (4.10)$$

Tomando en cuenta (4.9), (4.10) tiene por solución:

$$\hat{A}(\tau) = \Sigma V^T D_1 V (V^T D_2 V)^{-1} \Sigma^{-1} \quad (4.11)$$

Donde:

$$D_1 = \begin{pmatrix} 0 & 0 \\ I_{\tau-1} & 0 \end{pmatrix}$$

$$D_2 = \begin{pmatrix} I_{\tau-1} & 0 \\ 0 & 0 \end{pmatrix}$$

Finalmente, las matriz de covarianza del ruido del modelo Q se estima como:

$$\hat{Q}(\tau) = \frac{1}{\tau-1} \sum_{i=1}^{\tau-1} \hat{v}(i) \hat{v}^T(i) \quad (4.12)$$

donde $\hat{v}(i) = \hat{x}(i+1) - \hat{A}(\tau) \hat{x}(i)$. Asimismo, la matriz de covarianza R se estima como:

$$\hat{R}(\tau) = \frac{1}{\tau-1} \sum_{i=1}^{\tau-1} \hat{w}(i) \hat{w}^T(i) \quad (4.13)$$

donde $\hat{w}(i) = y(i) - \hat{C} \hat{x}(i)$.

Luego (4.9), (4.11), (4.12) y (4.13) constituyen la estimación del modelo ARMA que describe la textura dinámica, a partir de una secuencia de τ imágenes.

En el procedimiento anterior se asume que está dado el número de componentes principales n ; sin embargo, en la práctica se debe inferir de los datos (Vidal et al., 2007). Resultados del procedimiento anterior para la identificación de procesos visuales pueden revisarse en (Doretto et al., 2003), (Vidal et al., 2007).

4.2.3. Clasificación de Texturas Dinámicas

Dado un conjunto de texturas dinámicas, y sus modelos ARMA asociados, la clasificación de texturas consiste en comparar la similitud o diferencia de las texturas, basándose en sus modelos asociados. La medida clásica de similitud entre modelos es la llamada

distancia entre modelos ARMA que puede calcularse de diversas formas (Doretto et al., 2003), (Vidal et al., 2007), (Vishwanathan, Smola, & Vidal, 2006).

Estudios realizados muestran que los mejores resultados en la clasificación de texturas dinámicas se obtienen calculando la distancia entre los respectivos modelos ARMA utilizando los llamados Kernels de Binet-Cauchy y posteriormente aplicando el algoritmo *K-means* para agrupar las texturas similares (Vishwanathan et al., 2006).

Los kernels de Binet-Cauchy se basan en la llamada matriz compuesta, definida matemáticamente como sigue. Sea $A \in \mathbb{R}^{m \times n}$. Para $q < \min(m, n)$ se define $I_q^n = \{i = (i_1, i_2, \dots, i_q) : 1 \leq i_1 < \dots < i_q \leq n, i_l \in \mathbb{N}\}$, y asimismo se define I_q^m . La matriz compuesta $\mathcal{C}_q(A)$ de orden q se define entonces como:

$$[\mathcal{C}_q(A)]_{i,j} = \det(A(i_k, j_l))_{k,l=1}^q \quad (4.14)$$

donde $i \in I_q^n$ y $j \in I_q^m$.

Una vez definida la matriz compuesta, el Kernel de Binet-Cauchy de orden q entre los modelos M_1, M_2 se define como (Vishwanathan et al., 2006):

$$k_q(M_1, M_2) = \text{tra } \mathcal{C}_q(Y_1 Y_2^T) \quad (4.15)$$

Cuando $q = 1$, el Kernel se puede calcular explícitamente en función de los parámetros de los modelos ARMA que generan los datos $M_i = (x_i(0), A_i, C_i)$ como (Vishwanathan et al., 2006):

$$k_1(M_1, M_2) = x_1^T(0) Q x_2(0) \quad (4.16)$$

donde Q es solución de la ecuación de Sylvester:

$$A_1^T Q A_2 - Q = -C_1^T C_2 \quad (4.17)$$

A partir del Kernel de Binet-Cauchy, la distancia entre dos modelos ARMA se puede obtener como (Vidal et al., 2007):

$$d_1(M_1, M_2)^2 = k_1(M_1, M_1) - 2k_1(M_1, M_2) + k_1(M_2, M_2) \quad (4.18)$$

Cabe notar, que al contrario de otras medidas de distancia entre modelos, la distancia obtenida a partir de los Kernels de Binet-Cauchy depende explícitamente de la condición inicial de los procesos visuales. Esta característica es muy favorable cuando se trata de distinguir evoluciones temporales de procesos visuales independientes (Vidal et al., 2007), ya que no sólo considera la evolución del proceso considerando la dinámica (representada por la matriz Q), sino que también incluye el punto de partida del proceso facilitando la segmentación de un proceso visual en varios subprocessos consecutivos.

4.2.4. Modelación de Texturas Dinámicas como Sistemas Híbridos

Los métodos descritos anteriormente modelan una imagen o textura apoyándose en un modelo ARMA para la totalidad de la escena, lo que en muchos casos permite clasificar y predecir con exactitud procesos visuales. Sin embargo, es común que en una imagen, o en una secuencia de imágenes, interactúen diversos elementos. En éste caso la modelación de la imagen como una sola textura es inadecuada, debido a que se pueden distinguir porciones dentro de la imagen que no comparten la misma dinámica.

El problema ahora es modelar una secuencia de imágenes donde coexisten regiones que tienen características espacio-temporales diferentes. Asumiendo que la imagen en el tiempo t se puede dividir en $N(t)$ regiones desconocidas $\{\Omega_i(t) \subset \mathbb{R}^2\}_{i=1}^{N(t)}$, se modela la intensidad de cada pixel de la región $\Omega_i(t)$ como la salida de un modelo ARMA del tipo (4.2), (4.3) con estado $x_i(t)$ y parámetros $A_i(t)$, $B_i(t)$ y $C_i(t)$. Luego la secuencia completa de imágenes se puede modelar como la salida de $N \geq N(t)$ modelos ARMA.

Dada una secuencia de mediciones $\{y(t) \in \mathbb{R}^m\}_{t=1}^{\tau}$ con al menos 2 regiones con diferentes características dinámicas $\Omega_i(t)$, el problema de identificación asociado consiste en estimar las regiones $\Omega_i(t)$ y los parámetros de cada modelo ARMA asociado. Esto se conoce como la segmentación del proceso visual (Vidal et al., 2007).

4.2.5. Segmentación Espacial de Texturas Híbridas

La segmentación espacial de texturas, considera que existen N regiones Ω_i cuyos límites no cambian con el tiempo. Luego el problema se reduce a particionar una secuencia de imágenes en N regiones espaciales, cada una de ellas descritas por un modelo independiente.

El problema se resuelve mediante un proceso de minimización alternante, el cual comienza con una estimación de las regiones $\hat{\Omega}_i(0)$ y de los parámetros de los modelos asociados a cada región $\hat{A}_i(0)$, $\hat{B}_i(0)$, $\hat{C}_i(0)$ y los estados $\hat{x}_i(0)$. Luego para cada tiempo dado t , se buscan las actualizaciones de las regiones y los parámetros que minimizan una cierta funcional de costo dada. Típicamente se utiliza la norma de innovación integrada en el tiempo y el espacio (Vidal et al., 2007):

$$\sum_{i=1}^N \int_{\Omega_i} \sum_{t=n+1}^T (y(x, t) - \hat{y}_i(x, t | t-1))^2 dx \quad (4.19)$$

donde $\hat{y}_i(x, t | t-1)$ es el predictor de $y(x, t)$ a partir de las mediciones hasta el tiempo $t-1$ basada en el modelo i .

Para realizar la minimización se debe tener una representación de los contornos de las regiones $\delta\Omega_i$. Usando una representación del tipo *level-set* (Ghoreyshi & Vidal, 2006), se tiene: $\Omega_i = \{x : \phi_i(x) > 0\}$ y $\delta\Omega_i = \{x : \phi_i(x) = 0\}$ para alguna función $\phi_i : \Omega \rightarrow \mathbb{R}$.

Adicionalmente, se necesita una representación del modelo dinámico del predictor. Si se considera un modelo AR (Vidal et al., 2007), (Ghoreyshi & Vidal, 2006) dado por:

$$y(x, t) = \sum_{j=1}^n a_j^i y(x, t-j) \quad (4.20)$$

la segmentación espacial se puede formular como un problema de agrupar regiones definidas por modelos AR con parámetros similares. Para agrupar las regiones, se minimiza la generalización temporal de la funcional de Mumford-Shah (Vidal et al., 2007), (Ghoreyshi &

Vidal, 2006) dada por:

$$E(\{\phi_i\} \{a_j^i\}) = \sum_{i=1}^N \int_{\Omega} \lambda_i |\nabla H(\phi_i(x))| dx + \sum_{i=1}^N \int_{\Omega} \sum_{t=n+1}^T \left(y(x, t) - \sum_{j=1}^n a_j^i y(x, t-j) \right)^2 H(\phi_i(x)) dx \quad (4.21)$$

donde $H(\phi)$ es la función de Heaviside o escalón. El primer término de la funcional busca minimizar la longitud de los contornos, ponderados por un factor λ_i . El segundo término corresponde al error de innovación.

La minimización se realiza algorítmicamente alternando los siguientes pasos hasta la convergencia (Vidal et al., 2007), (Ghoreyshi & Vidal, 2006):

- Para ϕ_i fijo, se minimiza con respecto a a_j^i resolviendo un problema de mínimos cuadrados
- Para a_j^i dado, la minimización con respecto a ϕ_i se implementa mediante descenso del gradiente obteniéndose la siguiente ecuación diferencial parcial:

$$\frac{\partial \phi_i(x)}{\partial t} = \lambda_i \nabla \left(\frac{\nabla \phi_i(x)}{\|\nabla \phi_i(x)\|} \right) + \sum_{t=n+1}^T \left(y(x, t) - \sum_{j=1}^n a_j^i y(x, t-j) \right)^2 \quad (4.22)$$

Esta ecuación se simula numéricamente usando métodos del tipo *level set* hasta la convergencia a un mínimo local (Ghoreyshi & Vidal, 2006).

La condición de estacionariedad de los contornos, se puede levantar considerando ventanas móviles en el tiempo que consideran un número fijo de N imágenes. Esto permite aplicar el método a secuencias que contengan regiones con contornos dinámicos; sin embargo, el requerimiento computacional aumenta. De la misma manera se puede adaptar el método para operación en línea.

4.2.6. Segmentación Temporal de Texturas Híbridas

En lugar de particionar el dominio espacial en regiones, como en el apartado anterior, se puede particionar el dominio temporal en diversos eventos ocurridos en un instante t .

Dada la secuencia de imágenes $\{I(t) \in \mathbb{R}^m\}_{t=1}^T$, se desea agrupar imágenes de acuerdo a eventos identificados. El problema considera que existe una región única en cada imagen con un único modelo asociado, es decir, $N(t) = 1$ y $\Omega_i(t) = \Omega$.

Se considera que la intensidad de cada pixel cumple con un modelo SAR como (Vidal et al., 2007):

$$y(x, t) = \sum_{j=1}^n a_j(q_t) y(x, t - j) \quad (4.23)$$

donde el estado discreto $q_t = 1, \dots, N$. Además se define $b_i = (a_n(i), \dots, a_1(i), 1)$ y $z_t(x) = (y(x, t - n), \dots, y(x, t - 1), -y(x, t))$. Para cada x y t existe un i tal que:

$$b_i^T z_t(x) = 0 \quad (4.24)$$

Luego para todo x, t se cumple la condición de desacoplamiento híbrido y existe el polinomio de desacoplamiento híbrido asociado:

$$p_N(z_t(x)) = \prod_{i=1}^N (b_i^T z_t(x)) = 0 \quad (4.25)$$

Luego el problema de segmentación temporal se transforma en un problema de identificación híbrida tradicional, que se resuelve mediante la estrategia algebraica. Siguiendo el procedimiento algebraico, el polinomio de desacoplamiento híbrido se puede escribir linealmente en función del vector de coeficientes h y el vector de monomios $v_N(z_t(x))$ como:

$$h^T v_N(z_t(x)) = 0 \quad (4.26)$$

Los parámetros del modelo AR activo en el instante t se pueden calcular mediante:

$$b_i = \nabla p_N(z_t(x)) \quad (4.27)$$

Esta estrategia de segmentación, puede usarse para clasificación en línea de texturas a partir de modelos AR obtenidos con anterioridad.

4.2.7. Segmentación Espacio–Temporal de Texturas Híbridas

La segmentación espacio–temporal considera la segmentación tanto en el tiempo, como en el espacio, bajo el supuesto que cada región espacio–temporal se modela según un modelo AR. Esto da origen a un modelo del tipo SAR que describe el comportamiento espacio–temporal del proceso visual según:

$$y(x, t) = \sum_{j=1}^n a_j (q_t(x)) y(x, t - j) \quad (4.28)$$

Aquí el estado discreto $q_t(x)$ ahora depende no sólo del tiempo, sino también de la ubicación espacial del pixel x .

Luego, para realizar la segmentación, se pueden aplicar las mismas técnicas que en los apartados anteriores; sin embargo, en lugar de obtener un solo modelo para cada instante t , se obtendrán $N(t)$ modelos para cada imagen.

Mediante clustering usando *k-means* se obtiene la segmentación para cada instante t .

La coherencia temporal se incluye obteniendo una estimación $\hat{h}(t)$ del vector h a partir de las mediciones hasta el tiempo t . La estimación se realiza mediante un algoritmo recursivo de identificación basado en descenso del gradiente normalizado (Vidal, 2007).

4.3. Ejemplo de Modelación y Clasificación de Procesos Visuales

4.3.1. Descripción del Sistema en Estudio

El sistema en estudio corresponde a un tramo de una carretera la cual cambia su tráfico con el correr del día. Las secuencias de imágenes se obtuvieron a una tasa de 25 *fps* y resolución 320×240 pixeles, con una cámara estática por lo que el tramo de carretera en observación no varía. Las secuencias representan 2 horas de tráfico con condiciones de

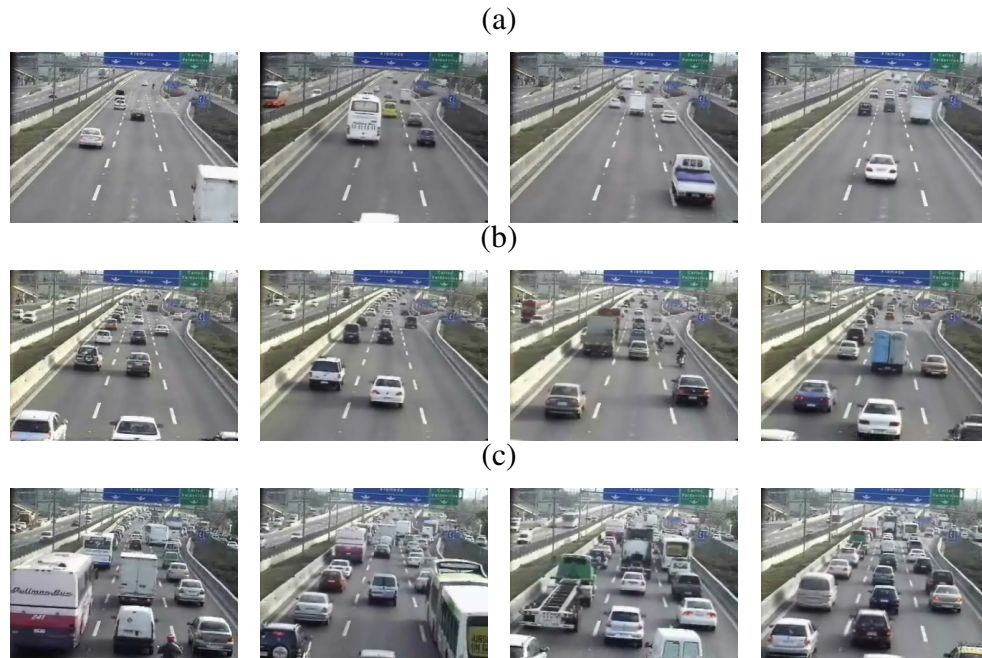


FIGURA 4.3. Estados operacionales definidos para el estudio; (a): Estado operacional 1; (b): Estado operacional 2; (c): Estado operacional 3.

iluminación variables debido al movimiento del Sol.

El objetivo del estudio es modelar la información visual con el fin de identificar el tráfico presente en la carretera, para lo cual se caracteriza el transcurso del tiempo como un sistema híbrido que atraviesa distintos estados de operación previamente definidos.

4.3.2. Caracterización por Estado de Operación

Según el tráfico que presenta la carretera, se realizó una discretización en 3 estados operacionales claramente distintos, los cuales son: tráfico bajo, tráfico medio y tráfico alto. La figura 4.3 presenta la apariencia visual de cada estado de operación para diferentes instantes. Se aprecia que la situación en cada estado es claramente diferente, usando como criterio el número de vehículos presentes en la imagen.

En la figura se aprecia que los estados definidos se diferencian en el número de vehículos presentes en la imagen. La modelación de la información visual captura las características espacio-temporales de cada estado, por lo que conceptualmente se pueden diferenciar

los estados de operación según la porción de la imagen que varía en el tiempo (movimiento de vehículos). Así una secuencia de imágenes que pertenece al estado 1 presentará una pequeña porción en movimiento, mientras que una perteneciente al estado 2 presentará una porción mayor. El estado 3 es el que presenta una mayor superficie ocupada por vehículos, sin embargo, el movimiento de éstos es a una menor velocidad en relación a los otros estados definidos. Lo anterior no representa un problema mayor, pues el algoritmo de modelación de texturas no es dependiente de la velocidad del movimiento si la secuencia utilizada para obtener el modelo es lo suficientemente larga.

A cada estado de operación de la figura 4.3 se le asignó un modelo patrón obtenido mediante identificación de texturas utilizando $n = 20$ componentes principales y secuencias de largo $\tau = 30$, parámetros que demostraron ser suficientes para capturar las características espacio-temporales de cada estado.

Los modelos ARMA obtenidos para cada estado quedan determinados por las matrices A y C de dimensiones 20×20 y 230400×20 respectivamente.

4.3.3. Resultados Experimentales

Con el fin de probar un sistema de reconocimiento de estados de operación, se programó en Matlab un algoritmo que se basa en la distancia entre modelos ARMA asociados a cada estado y kernels de *Binet-Cauchy*, que toma una secuencia de imágenes de largo $\tau = 30$, distintas en su totalidad a aquellas utilizadas en el proceso de identificación de los modelos patrones, y calcula la distancia entre su modelo ARMA y los modelos patrones que representan cada estado de operación. La secuencia se asigna al estado de operación al que presenta menor distancia. Los resultados obtenidos en las pruebas de detección se presentan en la tabla 4.1.

TABLA 4.1. Resultados del test de clasificación de estados operacionales.

Real/Detectado	Estado 1	Estado 2	Estado 3
Estado 1	30	0	0
Estado 2	0	30	0
Estado 3	0	0	30

Los resultados obtenidos en el reconocimiento de estados de operación son perfectos, en cuanto a que no existió confusión por parte del algoritmo en asignar estados a las secuencias de prueba. La capacidad de detectar estados de operación presenta la oportunidad de contar con un sistema capaz de determinar el instante de transición entre estados discretos, lo que en esta aplicación específica corresponde a identificar el instante en que el tráfico en la carretera cambia.

El algoritmo de modelación y reconocimiento puede también usarse potencialmente en análisis fuera de línea que busque determinar cuanto tiempo pasa el sistema en cada estado de operación para una secuencia de largo conocido.

5. CONTROL PREDICTIVO DE PROCESOS VISUALES

5.1. Descripción General

La capacidad de modelar procesos visuales, o información visual, deja abierta la posibilidad de realizar control directamente sobre los procesos visuales o sobre procesos donde la información visual es preponderante. A continuación se describirán estrategias de control utilizando información visual y esquemas predictivos, para el control de procesos basándose en interpretación de información visual.

5.2. Aplicaciones Desarrolladas

En la literatura se pueden encontrar aplicaciones de control de color utilizando información visual (Sbarbaro, 2002) y (Chen, Murphy, & Subbayan, 1996). Ambas aplicaciones utilizan un esquema predictivo para controlar la coloración en un sistema de mezclado de colorantes en una solución acuosa (Sbarbaro, 2002) y un sistema de manufactura de papel (Chen et al., 1996), utilizando como variable manipulada la adición de colorantes.

Ambas aplicaciones modelan el proceso con un modelo de Wiener multivariable, siguiendo la estructura de la figura 2.1, construido según la teoría de Kubelka-Munk, la cual considera que la solución pigmentada es uniforme y calcula el color en base a la luz reflejada por la superficie al ser iluminada por una fuente de luz. El sensor de color consiste en una fuente de luz y un detector que censa la luz reflejada por el objeto en estudio. El espectro de la luz reflejada comparada con el espectro de la fuente de luz es la reflectancia, la que se expresa como un vector con una componente por frecuencia. La calidad del control se mide comparando el espectro reflejado con un espectro patrón.

El objetivo del control es encontrar la cantidad de colorantes adecuada que se debe adicionar para lograr alcanzar el espectro patrón. El color es representado en el espacio CIE mediante las variables L , a y b definidas en función de los estímulos X , Y y Z (Sbarbaro,

2002) como:

$$\begin{aligned} L &= 116 \left(\frac{X}{X_0} \right)^{1/3} - 16 \\ a &= 500 \left(\left(\frac{X}{X_0} \right)^{1/3} - \left(\frac{Y}{Y_0} \right)^{1/3} \right) \\ b &= 200 \left(\left(\frac{Y}{Y_0} \right)^{1/3} - \left(\frac{Z}{Z_0} \right)^{1/3} \right) \end{aligned} \quad (5.1)$$

donde X_0 , Y_0 y Z_0 son los estímulos de la fuente de luz utilizada.

El sistema tiene como estados la concentración de n colorantes, los cuales son adicionados en una solución acuosa mediante bombas y un circuito de tuberías, por lo que se propone un sistema de primer orden más un retardo para modelar el proceso de adición de colorantes:

$$\frac{c^i(S)}{u^i(S)} = \frac{e^{-d^i S}}{S\tau^i + 1} \quad (5.2)$$

Finalmente el modelo del proceso de coloración viene dado por:

$$X_{t+1} = AX_t + BU_{t-d} \quad (5.3)$$

$$Y_t = F(X_t) \quad (5.4)$$

donde X_t es el vector de las concentraciones c_t^i , U_{t-d} es el vector de los flujos de colorantes con sus respectivos retardos u_{t-d}^i , e Y_t es el vector de salida que representa el color en el espacio CIE. F es una función no-lineal que relaciona la concentración de colorantes con el color del objeto en estudio.

El controlador predictivo se diseña con la siguiente función objetivo:

$$J = \sum_{i=1}^3 |y_{t+T}^i - yr_{t+T}^i| + \lambda \sum_{j=1}^{N_u} \sum_{i=1}^n |u_{t+j-1}^i - u_{t+j-2}^i| \quad (5.5)$$

la cual toma en cuenta el error absoluto en cada componente del color y los incrementos en el control para cada colorante. El problema de control predictivo consiste en minimizar la función objetivo sujeto a (5.3), (5.4), $\sum_{i=1}^n u_t^i = 1$ y $0 \leq u_t^i \leq 1$.

Resultados de simulaciones del sistema de control pueden encontrarse en (Sbarbaro, 2002) y (Chen et al., 1996).

5.3. Ejemplo de Control Predictivo de Procesos Visuales

5.3.1. Descripción del Sistema en Estudio

El sistema en estudio corresponde a una secuencia de imágenes de agua en el océano. La secuencia se generó con un modelo ARMAX, en el cual la componente autoregresiva se obtuvo mediante identificación de texturas dinámicas utilizando $n = 40$ componentes principales y una secuencia de largo $\tau = 100$, mientras que la entrada exógena fue generada artificialmente. La variable manipulada actúa en el espacio de las componentes principales del modelo y no tiene significado físico.

Las imágenes tienen una dimensión de 177×117 píxeles, y generan una secuencia artificial de tasa 25 *fps* y de duración arbitraria.

El objetivo del estudio es evaluar un sistema de control predictivo que busca modificar el color de una secuencia, manteniendo las características de la textura.

5.3.2. Modelación del Proceso Visual

El proceso visual en estudio se puede caracterizar por el siguiente modelo ARMAX formulado en variables de estado:

$$x_{k+1} = Ax_k + B_u u_k + B_w w_k \quad (5.6)$$

$$y_k = Cx_k + v_k \quad (5.7)$$

Donde w_k y v_k son el ruido de modelo y ruido de medición respectivamente, los cuales se asumen como variables aleatorias con distribución Gaussiana, u_k es el vector de variables manipuladas de dimensión 20, y_k es el vector de salida que corresponde a un vector columna cuyos elementos corresponden a la intensidad de los píxeles de la imagen en las componentes RGB, y A , B_u , B_w y C son matrices de dimensión 40×40 , 40×20 , 40×40 y 62127×40 respectivamente. La figura 5.1 muestra cuadros seleccionados de la secuencia en lazo abierto generadas con el modelo ARMAX.

La identificación de la parte autoregresiva del modelo mediante la descomposición en valores singulares de la matriz de mediciones garantiza que el modelo es estable y observable, y más aún la elección apropiada de la matriz B_u garantiza que el modelo sea controlable. Luego el modelo es apto para realizar control sobre la salida, que en este caso corresponde a la intensidad de los píxeles de la imagen en el espacio RGB.

5.3.3. Diseño del Controlador Predictivo

El controlador predictivo tiene por objetivo llevar el proceso visual a la referencia, con el menor esfuerzo de control asociado. El controlador predictivo se diseñó utilizando la siguiente función objetivo:

$$J(N_u, N_2) = \sum_{j=1}^{N_2} \delta [\hat{y}_{k+j} - yr_{k+j}]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u_{k+j-1}]^2 \quad (5.8)$$

Dada la linealidad del modelo, se supuso $\delta = 1$ dejando sólo un parámetro de penalización libre para la sintonía del controlador. Las predicciones se obtuvieron utilizando el modelo ARMAX en ausencia de perturbaciones estocásticas y no se consideraron restricciones de dominio para el problema de optimización asociado.

La referencia del proceso corresponde a una trayectoria de imágenes generadas artificialmente utilizando el modelo de la textura y variables manipuladas adecuadas en ausencia

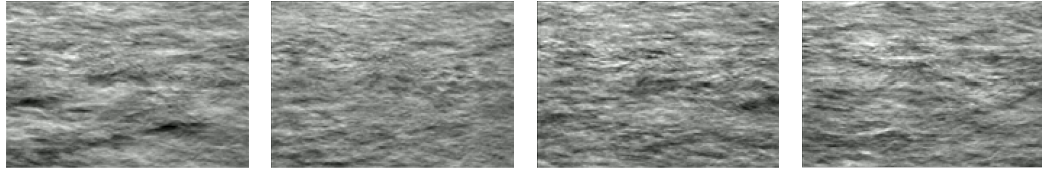


FIGURA 5.1. Cuadros obtenidos como salida del modelo para $k = 20, 40, 60, 80$.

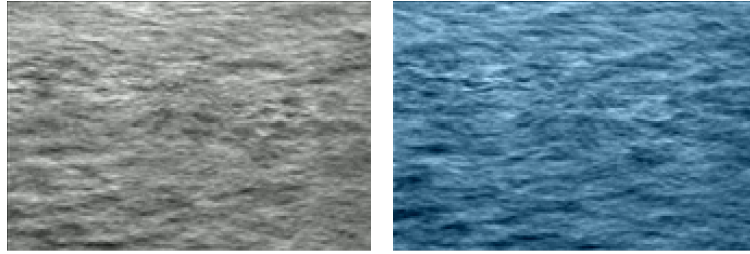


FIGURA 5.2. Cuadro de la secuencia real y su respectiva referencia.

de ruido. La figura 5.2 muestra un cuadro de la secuencia original y el cuadro respectivo de la referencia.

Para implementar el controlador, debido a la estructura del modelo, se hace necesaria la incorporación de un observador de estado al sistema en lazo cerrado con el fin de contar con la estimación del estado actual del proceso $\hat{x}_k$. Debido a la estructura lineal del modelo que caracteriza el proceso, el estimador de estado utilizado corresponde a un filtro de Kalman (Welch & Bishop, 2004).

La estructura del sistema en lazo cerrado se presenta en la figura 5.3.

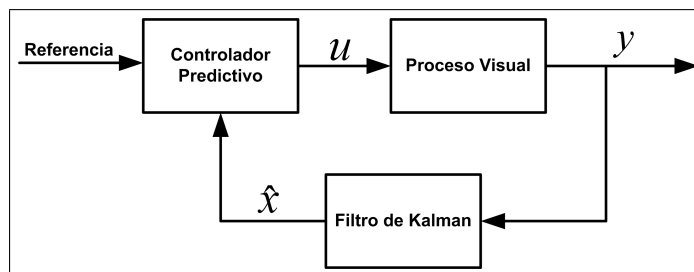


FIGURA 5.3. Esquema del sistema en lazo cerrado

Si se considera la función objetivo (5.8), el modelo (5.6),(5.7) y la ausencia de restricciones, se puede obtener una solución cerrada para la ley de control utilizando la ecuación de Maciejowski (Camacho & Bordons, 2004):

$$\Delta U_k = (\Theta^T \Theta + \lambda I)^{-1} \Theta^T (yr - \Psi \hat{x}_k - \Phi u_{k-1}) \quad (5.9)$$

Donde:

$$\Theta = \begin{pmatrix} CB & \cdots & 0 \\ C(AB + B) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{N_2-1} CA^i B & \cdots & \sum_{i=0}^{N_2-N_u} CA^i B \end{pmatrix} \quad (5.10)$$

$$\Psi = \begin{pmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_2} \end{pmatrix} \quad (5.11)$$

$$\Phi = \begin{pmatrix} CB \\ C(AB + B) \\ \vdots \\ \sum_{i=0}^{N_2-1} CA^i B \end{pmatrix} \quad (5.12)$$

Una vez obtenido el vector ΔU_k , se emplea el *Receding Horizon Principle* aplicando solo Δu_k y repitiendo el cálculo en el siguiente instante de tiempo.

5.3.4. Resultados Experimentales

El sistema se simuló en Matlab para una secuencia de largo 200. Los parámetros del controlador predictivo fueron $\lambda = 0.1$, $N_2 = 3$, $N_u = 3$. La figura 5.4 muestra cuadros seleccionados de la respuesta en lazo abierto, la referencia, y la respuesta en lazo cerrado.

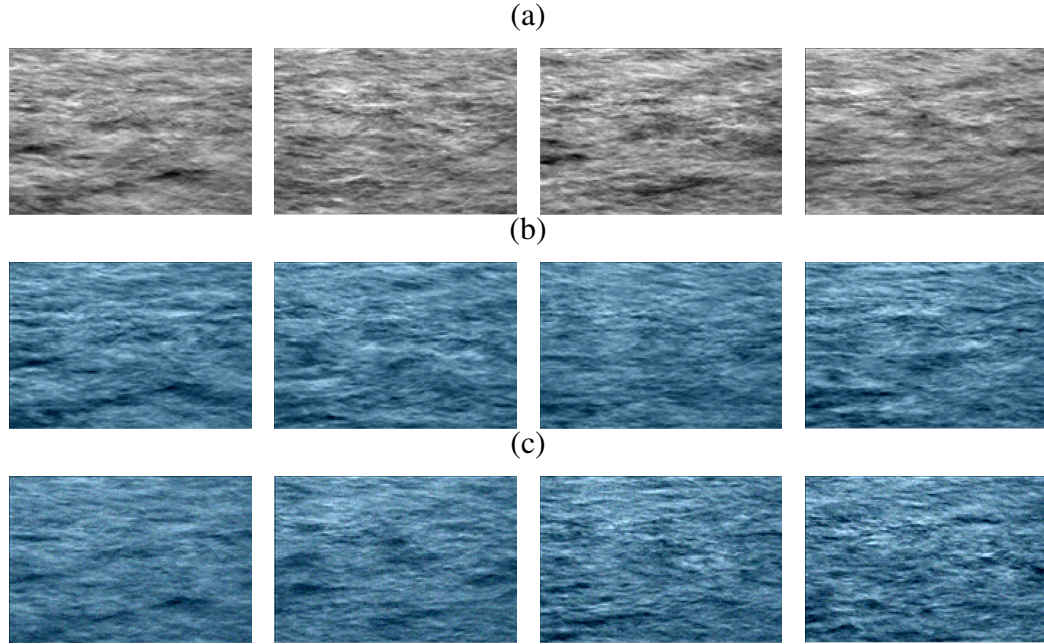


FIGURA 5.4. Resultados del control del proceso visual; (a): Respuesta en lazo abierto; (b): Referencia; (c): Respuesta en lazo cerrado.

Se aprecia que el controlador predictivo es capaz de llevar el sistema a la referencia, controlando efectivamente el color de la secuencia, sin perder la textura deseada.

Como índice de desempeño del controlador predictivo se propone la raíz del error de control medio cuadrático (RMSE) dado por:

$$RMSE(k) = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_k(i) - yr_k(i))^2} \quad (5.13)$$

La figura 5.5 muestra el error para cada cuadro para una simulación de largo $k = 200$, donde se aprecia que el error se mantiene estable a lo largo de la simulación, en un valor bajo 0.03 de una escala posible 0 a 1, debido a la codificación de la intensidad de los pixeles utilizando valores flotantes en lugar de escala 0 – 255 (8 bits).

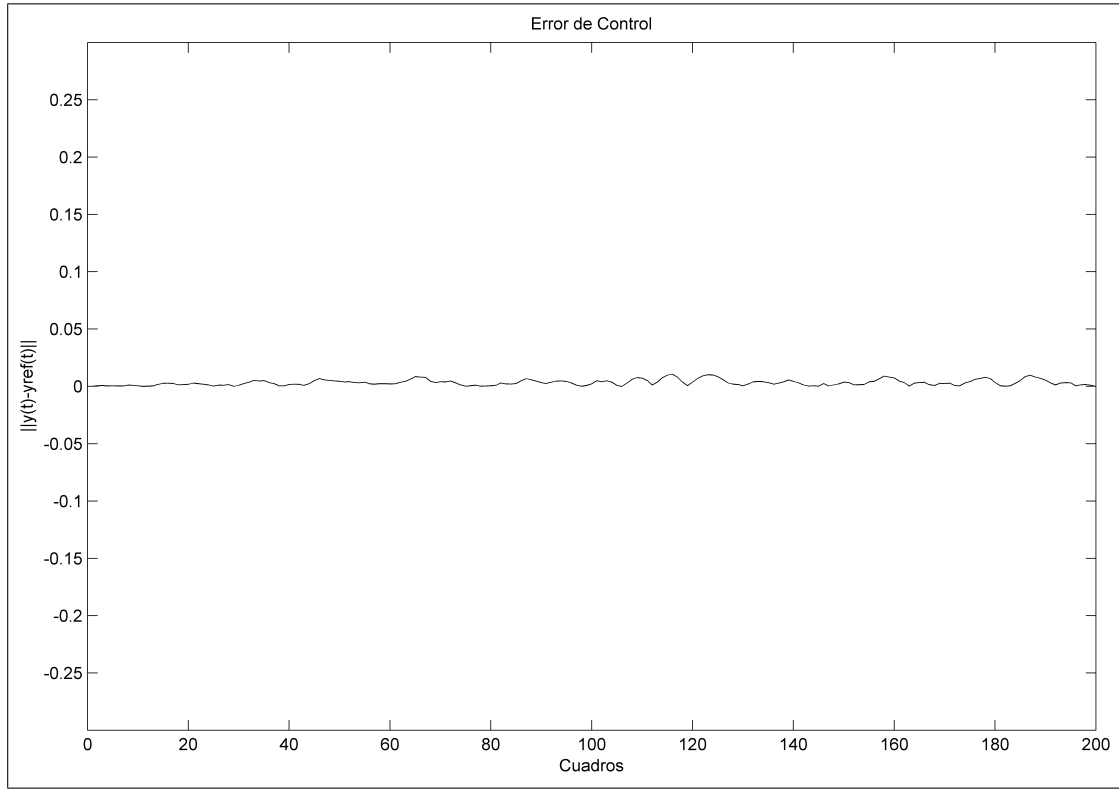


FIGURA 5.5. Error de control para una simulación de largo $k = 200$.

Los resultados obtenidos prueban que es posible controlar un proceso visual actuando directamente sobre la apariencia, sin extraer propiedades de la imagen y utilizarlas como variables controladas, como en (Sbarbaro, 2002) y (Chen et al., 1996). Si bien en el ejemplo presentado las variables manipuladas no tienen significado físico, en un proceso real se pueden identificar relaciones entre las variables físicas y su efecto en el espacio de las componentes principales del proceso visual, obteniéndose, por ejemplo, una función no lineal estática que relacione las variables físicas con el espacio de las componentes principales.

5.4. Control Predictivo Híbrido Basado en Sensores Visuales

El control predictivo híbrido basado en sensores visuales, considera un modelo híbrido del proceso para el cálculo de las predicciones, al igual que el control predictivo híbrido tradicional, pero incorpora información adicional obtenida de un sensor visual en forma de

imagen, o secuencia de imágenes. Esta información adicional se utiliza para determinar el estado de operación del sistema en conjunto con las variables controladas del proceso.

Considerando la posibilidad de determinar estados de operación a partir de información visual, mediante modelación, identificación y clasificación de texturas, se propone un esquema de control predictivo híbrido basado en visión donde el estado del sistema *switched affine* se obtiene a partir de información visual en forma de secuencias de imágenes modeladas como texturas.

Si se considera el esquema de un sistema *switched affine* dado en la figura 3.3, el generador de eventos toma como entrada la secuencia de imágenes y genera el evento correspondiente según las características de la textura. A partir de ello obtiene el nuevo estado en la máquina de estados que finalmente da origen al estado discreto del sistema, el que indexa a los modelos continuos que describen el proceso para propósitos de control. El esquema de control propuesto se muestra en la figura 5.6.

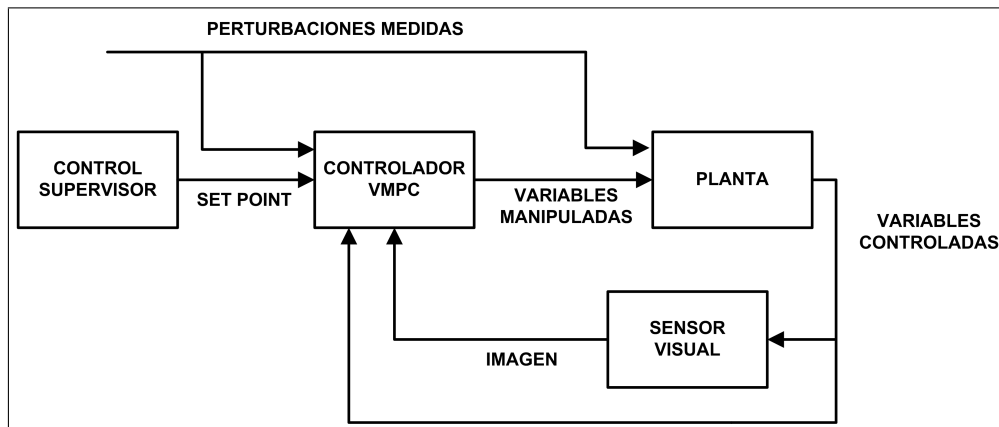


FIGURA 5.6. Lazo de control predictivo híbrido basado en sensores visuales

En el esquema propuesto el sensor visual entrega una imagen, o una secuencia de imágenes, a un bloque de preproceso que es el encargado de adecuar la información mediante filtros y ajustes de escala para luego entregarla al controlador predictivo híbrido basado en visión.

El controlador predictivo híbrido consta de un modelo de la apariencia visual de cada estado de operación que permite predecir la evolución de la apariencia basándose en la apariencia actual y las variables manipuladas. El controlador utiliza los modelos para reconocer el estado actual del sistema ya sea mediante técnicas de estimación de estado, o bien, identificando un nuevo modelo y calculando la distancia a los modelos patrones.

Una vez determinado el estado de operación, se realiza el cálculo de las predicciones en función de las variables manipuladas utilizando el modelo híbrido del sistema, que considera el cambio de estado discreto en función de la información entregada por el módulo de análisis visual, para finalmente calcular la ley de control a partir de las mediciones de las variables controladas, la información visual, las perturbaciones medidas y las referencias entregadas por el sistema de control supervisor.

6. CONTROL PREDICTIVO HÍBRIDO BASADO EN SENSORES VISUALES APLICADO A FLOTACIÓN DE MINERALES

6.1. El Proceso de Flotación de Minerales

6.1.1. Descripción del Proceso

El principal objetivo del proceso de flotación en una planta de tratamiento de mineral es aumentar la concentración del producto de la molienda. Este proceso puede ser segmentado en un conjunto de operaciones unitarias que se lleva a cabo en unidades básicas llamadas celdas de flotación, que son estanques por los cuales fluye una mezcla de agua, impurezas y el mineral a ser concentrado.

El producto obtenido del proceso de flotación se llama concentrado y posee una alta ley de mineral, definiéndose la ley de mineral como la razón porcentual entre el flujo de mineral y el flujo total de sólido. Una medida de la eficiencia de la flotación es la recuperación de mineral, que se define como la razón porcentual entre el flujo de mineral contenido en el concentrado y el flujo de mineral contenido en la alimentación. El material que se desecha, o que se reprocesa en otra celda, se conoce como cola y se evacúa por el fondo de la celda de flotación. La figura 6.1 muestra el esquema general de una celda de flotación (Wills, 1992).

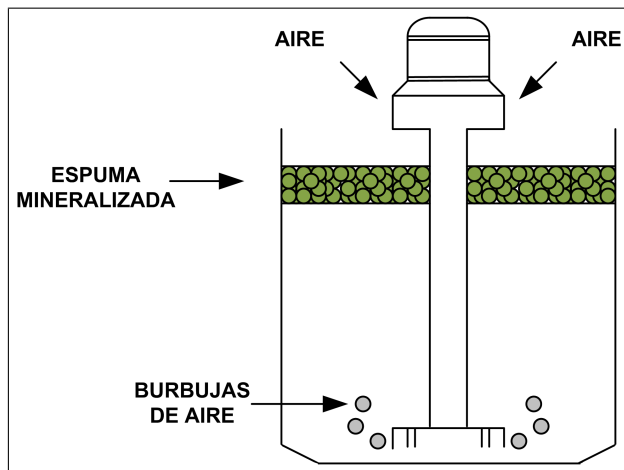


FIGURA 6.1. Esquema general de una celda de flotación

La flotación se basa en la diferencia de propiedades superficiales entre las especies mineralógicas que componen la pulpa y en la interacción de éstas con burbujas de aire que finalmente transportan las especies valiosas. Las burbujas se producen por efecto de inyección de aire al interior de la celda, las cuales suben continuamente originando en la superficie una capa de espuma que se descarga constantemente por rebalse. Mediante la adición de reactivos químicos se logra que las partículas de mineral se adhieran a las burbujas que conforman la espuma. Específicamente la adición de químicos provoca que las partículas de mineral se vuelvan hidrófobas adhiriéndose a las burbujas, la adición de colectores aumentan la fuerza de adhesión partícula-burbuja y espumantes estabilizan la dispersión de aire y tamaño de las burbujas (Melo & Laskowski, 2006).

A nivel industrial, la flotación se realiza en bancos de celdas conectadas en serie que cuentan con vías de entrada y salida para la pulpa, así como con mecanismos para mantener la dispersión del sólido y producir la diseminación del flujo de aire en burbujas de tamaño adecuado. La figura 6.2 muestra un diagrama de una planta de flotación rougher. (Casali, Gonzalez, Agosto, & Vallebuona, 2002), (Muñoz, 2000), (Wills, 1992).

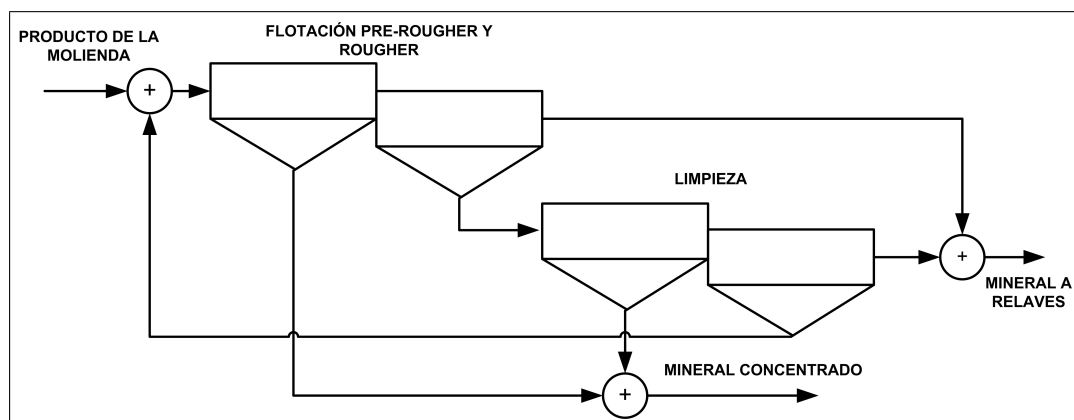


FIGURA 6.2. Esquema de proceso de flotación rougher y limpieza

6.1.2. Objetivos del Control en el Proceso de Flotación

La flotación constituye uno de los procesos de mayor complejidad en la extracción de mineral. Algunos factores como los grandes retardos, escasa instrumentación, la alta no-linealidad y la alta variabilidad en la composición del mineral proveniente de la molienda, presentan un gran desafío a la hora de automatizar el proceso. Al controlar una planta de flotación se persigue lograr los siguientes objetivos:

- Mantener la planta en operación estable a pesar de las perturbaciones que la afectan.
- Obtener una ley de concentrado sobre un valor predeterminado, manteniendo una baja ley de cola y un determinado nivel de recuperación.
- Maximizar la recuperación de mineral para una determinada ley de concentrado.
- Minimizar el uso de reactivos en el proceso.

6.1.3. Estrategias de Control Usadas en Flotación

Debido a la gran complejidad del proceso, no se cuenta con un modelo lo suficientemente confiable que permita aplicar técnicas de control avanzado al proceso. Sin embargo, operadores experimentados son capaces de caracterizar el estado del proceso en base a la calidad de la espuma en la superficie de las celdas de flotación y proponer acciones de control adecuadas. Estas acciones se basan en observaciones subjetivas de los operadores que toman en cuenta propiedades de las burbujas y la espuma tales como el tamaño, la forma y densidad de las burbujas además de la velocidad, estabilidad y color de la espuma.

Los investigadores especializados en el proceso de flotación han reconocido en las características de la espuma una fuente de información acerca del estado del proceso (Cipriano, Sepulveda, & Guarini, 1997), (Holtham & Nguyen, 2002), (Hyötyniemi & Ylinen, 2000). Esto ha dado origen a sensores que extraen, mediante procesamiento de imágenes, las características de la espuma que los operadores consideran preponderantes para tomar

una decisión de control. Los ingenieros de control han ido mas allá y han intentado plasmar el conocimiento que permite a los operadores tomar decisiones en sistemas computacionales de inteligencia artificial, conocidos como sistemas expertos, los cuales reciben como entrada las propiedades de la espuma obtenidas de los sensores y entregan sugerencias acerca de las acciones de control a seguir para optimizar el proceso.

6.1.4. Variables Relevantes en el Proceso

La flotación es un proceso de múltiples variables. Pese a que se han realizado numerosos estudios al respecto, no hay todavía un dominio tanto teórico como práctico que permita contar con un modelo que represente fielmente al proceso. Esto se debe principalmente a que aún no se conocen a cabalidad las interacciones entre todas las variables involucradas en la flotación.

En primer lugar se debe considerar un grupo de variables que corresponde a las naturaleza del mineral mismo, como la presencia de impurezas, el pH natural, etc. En seguida están los problemas que introducen los procesos de chancado y molienda, por los que debe pasar el mineral antes de la flotación, que incluyen la oxidación. Finalmente en el proceso de flotación mismo surgen una serie de factores que influyen como los reactivos, el tamaño granulométrico, la densidad de la pulpa, la tensión superficial, el flujo de aire en la celda, el nivel de la pulpa, etc.

La posibilidad de controlar el proceso mediante la observación de la espuma superficial y técnicas de inteligencia artificial, como el uso de sistemas expertos y lógica difusa, representa una alternativa a la modelación del proceso considerando las numerosas variables descritas anteriormente. Es por esto que en la actualidad los esfuerzos se han centrado en estudiar las características de la espuma que aportan información valiosa para el control y las técnicas para cuantificarlas de manera no subjetiva, como lo son las observaciones del

operador, e incluirlas en los algoritmos de control.

Los estudios realizados en el último tiempo coinciden en que existe una relación entre el tamaño de las burbujas y el tamaño de las partículas que se adhieren a ellas en la zona de colección; por ejemplo pequeñas partículas de mineral flotan predominantemente debido a pequeñas burbujas, mientras que grandes partículas flotan debido a burbujas mayores, incluso partículas muy grandes pueden flotar soportadas por varias burbujas.

Si se considera la espuma como un todo, su color representa una fuente importante de información acerca de la composición mineralógica. Así por ejemplo, los operadores saben que una espuma amarillenta tiene gran cantidad de pirita y una espuma de color gris tiene gran cantidad de aceite ((Cipriano et al., 1997)). El color óptimo va a depender del mineral que se está procesando.

Tomando en cuenta lo anterior, los investigadores coinciden en identificar como relevantes las siguientes propiedades de las burbujas y de la espuma que éstas forman (Botha, 1999), (Cipriano et al., 1997):

- Tamaño de las burbujas.
- Número de burbujas.
- Forma de las burbujas.
- Densidad de burbujas.
- Estabilidad de la espuma.
- Velocidad de la espuma.
- Color de la espuma.

6.1.5. Sensores Visuales Comerciales Especializados en Flotación

La oportunidad de controlar el proceso de flotación mediante el análisis de la espuma superficial de la celda, ha llevado a los proveedores de equipos y servicios de control de procesos y a las empresas especializadas en procesos mineros a desarrollar productos comerciales que extraen parámetros relevantes de la espuma mediante técnicas de procesamiento de imágenes.

A continuación se describen algunos productos industriales desarrollados en el último tiempo.

OutoTech FrothMaster 2: El sistema FrothMaster 2 fue desarrollado por la empresa OutoTech como parte de los numerosos instrumentos y programas computacionales para la minería que fabrica la empresa y corresponde a una versión mejorada de su antecesor FrothMaster. FrothMaster 2 realiza mediciones de estabilidad, velocidad, color y dirección de la espuma además de análisis de forma y tamaño de las burbujas. FrothMaster 2 cuenta con una salida de video opcional que permite observar en tiempo real la superficie de la celda de flotación

Metso VisioFroth: El sistema VisioFroth es el producto industrial desarrollado por la empresa Metso como parte de su línea de sensores visuales para procesos industriales. VisioFroth extrae características de velocidad, textura y estabilidad de la espuma, mientras que analiza la forma, tamaño y número de burbujas.

KnowledgeScape PlantVision: PlantVision es la solución industrial para flotación que fabrica la empresa KnowledgeScape. Se comercializa junto a un sistema experto propio de la empresa. PlantVision analiza la estabilidad, textura, velocidad y color de la espuma, mientras que para las burbujas identifica la forma y el tamaño de éstas.

JKFrothCam: JKFrothCam fue desarrollado en un ambiente académico por la universidad de Queensland en Australia (TUQ) en la división de estudios relacionados con minería. JKFrothCam analiza la velocidad y el tipo de espuma,

el cual determina a partir de una clasificación basada en propiedades como la estabilidad, color y textura. En cuanto a las burbujas, JKFrothCam entrega información acerca de la distribución del tamaño.

6.2. Caracterización Híbrida–Experta de la Apariencia Superficial

6.2.1. Adquisición de Imágenes de Proceso

Para propósitos de diseño, se obtuvieron filmaciones de la apariencia superficial de una celda de flotación utilizando una cámara industrial a una razón de 30 cuadros por segundo con una resolución de 320×240 píxeles. Las secuencias obtenidas representan cerca de una hora de operación, en distintos estados operacionales.

Las imágenes obtenidas, se dividieron en dos conjuntos utilizados para la identificación de modelos y entrenamiento de los algoritmos y para la evaluación del desempeño respectivamente.

6.2.2. Definición de los Estados de Operación

Basándose en conocimiento experto del proceso se adoptó una división clásica en 3 estados operacionales. Los estados definidos se presentan en la figura 6.3. Cada estado posee características propias, las cuales se detallan a continuación:

- Estado Operacional 1:
 - Burbujas grandes
 - Color grisáceo
 - Alta densidad de burbujas
 - Espuma estable
- Estado Operacional 2:
 - Burbujas pequeñas
 - Color amarillento

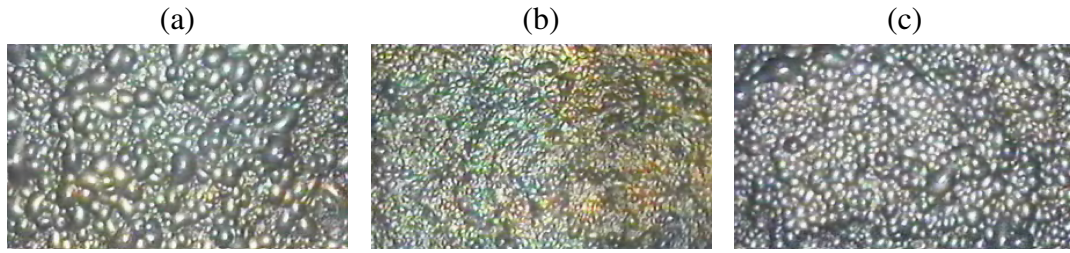


FIGURA 6.3. Estados operacionales definidos para el estudio; (a): Estado operacional 1; (b): Estado operacional 2; (c): Estado operacional 3.

- Áreas colapsadas
- Espuma inestable
- Estado Operacional 3:
 - Burbujas medianas
 - Color oscuro
 - Alta densidad de burbujas
 - Espuma estable

6.2.3. Modelación del Proceso Visual

La modelación del proceso visual consiste en la identificación de los parámetros de los modelos ARMA asociados a cada estado de operación, utilizando texturas dinámicas. El método de identificación basado en la descomposición en valores singulares de la matriz de mediciones, garantiza la estabilidad de los modelos. La identificación se realizó fuera de línea utilizando secuencias de largo τ entre 40 y 80 cuadros, y un número de componentes principales n entre 5 y 50.

La figura 6.4 muestra los resultados de la síntesis del proceso visual para cada estado de operación, utilizando $n = 40$ componentes principales y secuencias de largo $\tau = 60$ para la identificación de los modelos. Los modelos ARMA quedan determinados por sus matrices A y C de dimensiones 40×40 y 230400×40 respectivamente. Se observa que los modelos identificados, son capaces de capturar las características espacio-temporales

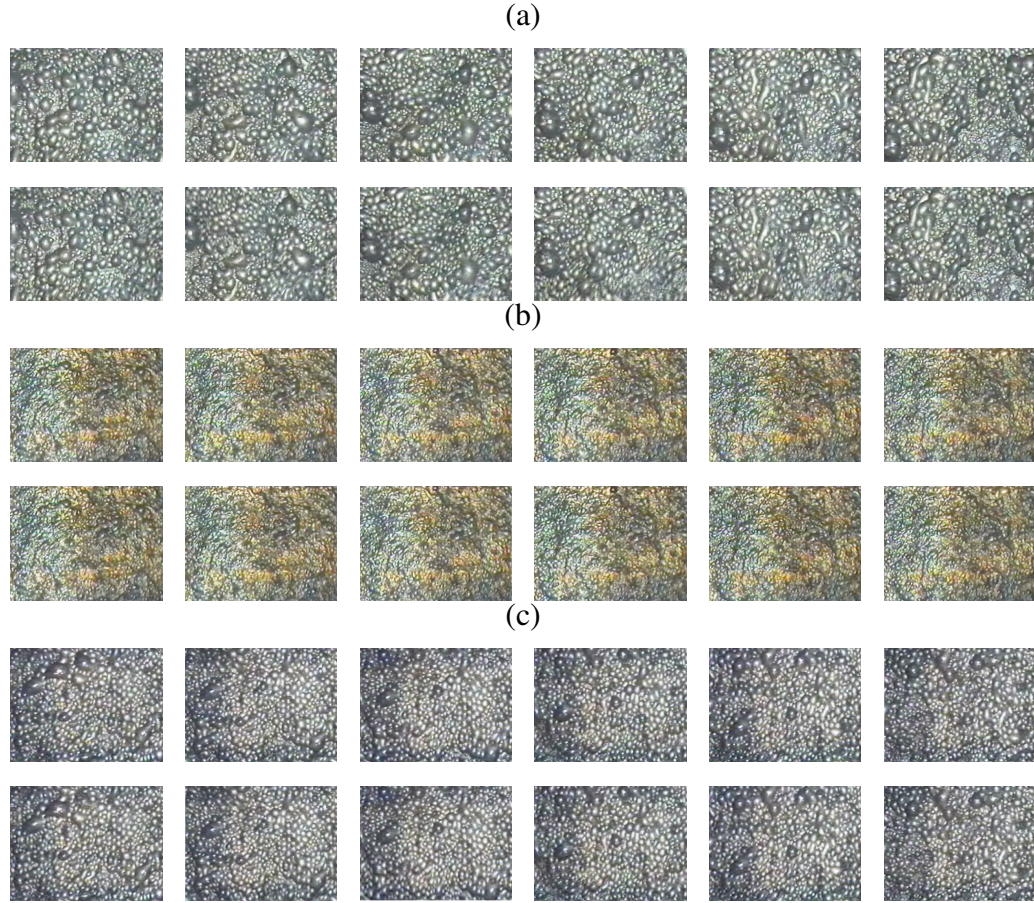


FIGURA 6.4. Cuadros seleccionados de las secuencias reales (línea superior) y los cuadros correspondientes obtenidos a partir de los modelos ARMA (línea inferior) para $t = 10, 20, 30, 40, 50, 60$; (a): Estado operacional 1; (b): Estado operacional 2; (c): Estado operacional 3.

de la espuma para los distintos estados.

Para evaluar el impacto del número de componentes principales en el desempeño del algoritmo de identificación, se realizaron pruebas variando el parámetro n para secuencias de largo fijo $\tau = 60$ en cada estado de operación. Las figuras 6.5, 6.6 y 6.7 presentan gráficamente los resultados obtenidos para el error de síntesis en función del número de componentes principales n utilizadas en el proceso de modelación.

En las figuras se aprecia que el error de síntesis decrece monotonamente a medida que se aumenta el número de componentes principales n . Sin embargo, el aumento indiscriminado del parámetro n tiene asociado un costo de procesamiento computacional. Por esto se estimó como valor óptimo para el estudio $n = 40$, valor que garantiza un error de síntesis inferior al 2% en los tres estados.

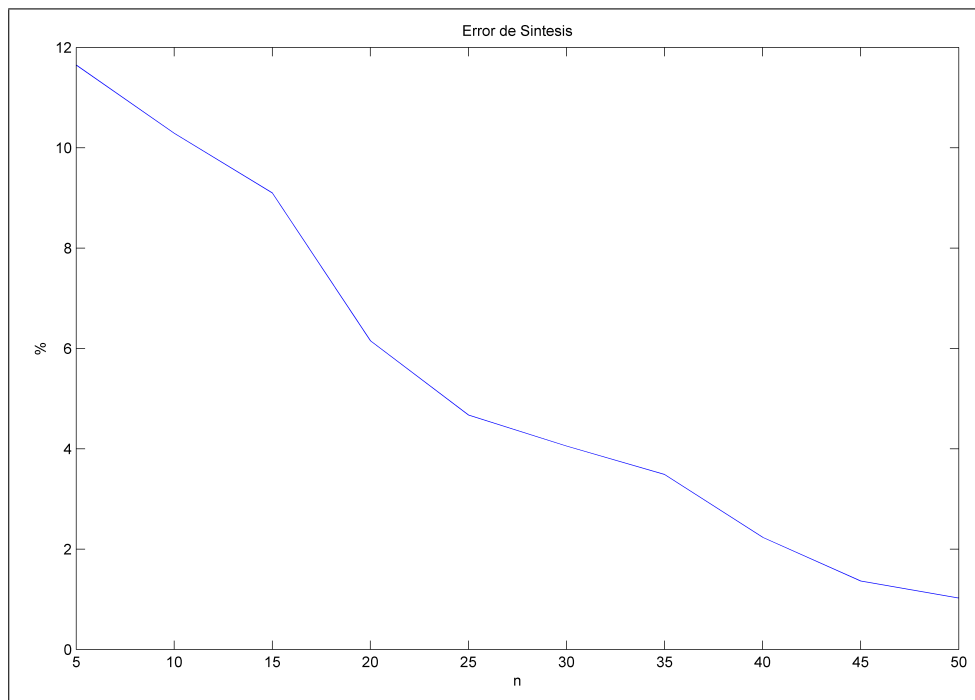


FIGURA 6.5. Error de síntesis para el estado 1

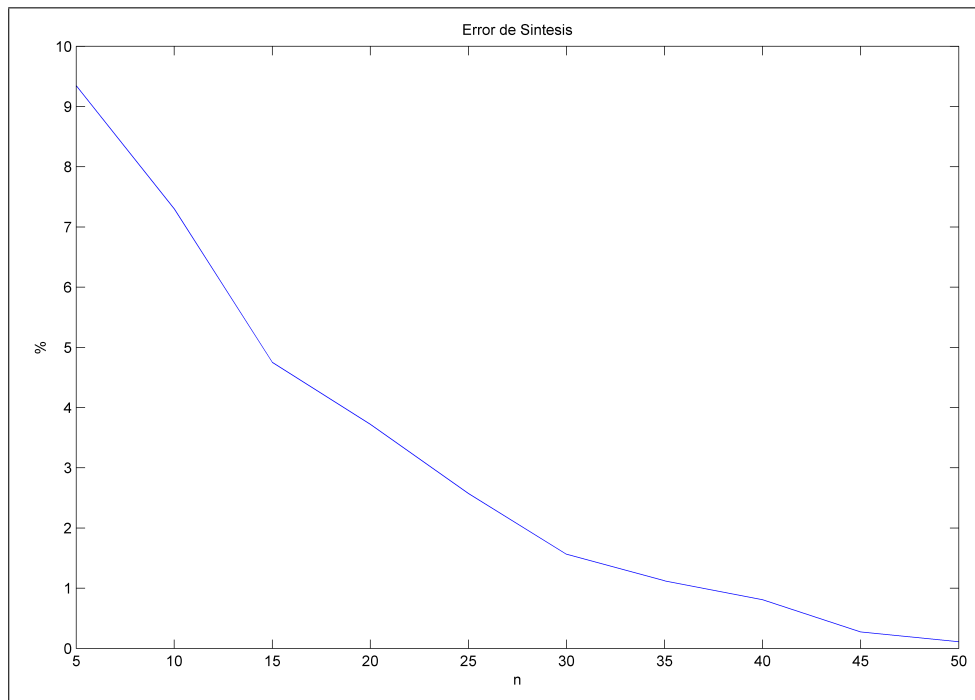


FIGURA 6.6. Error de síntesis para el estado 2

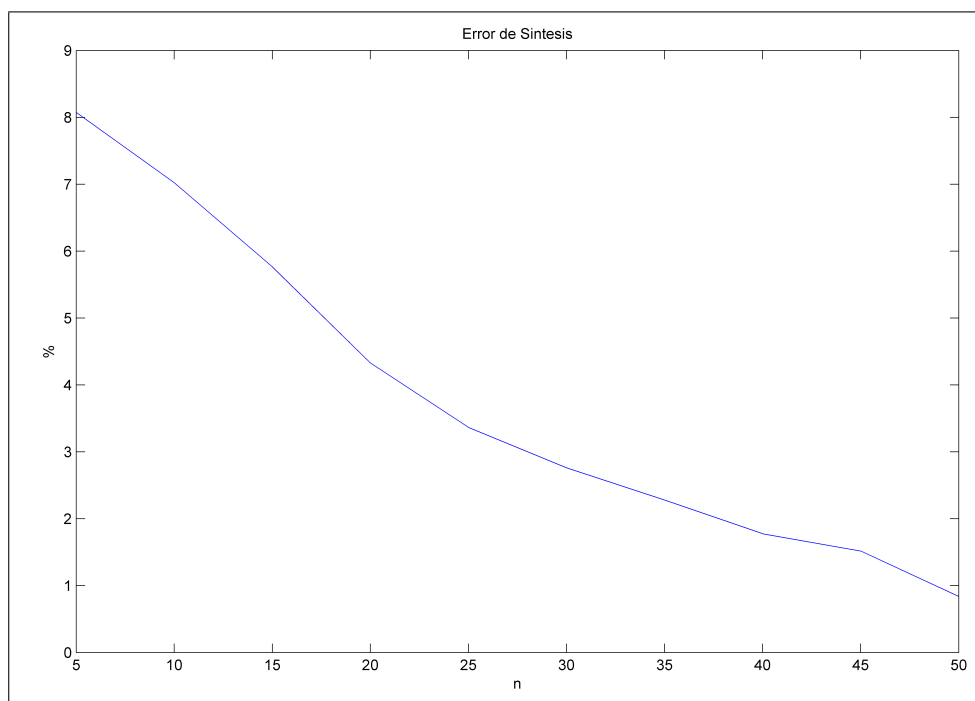


FIGURA 6.7. Error de síntesis para el estado 3

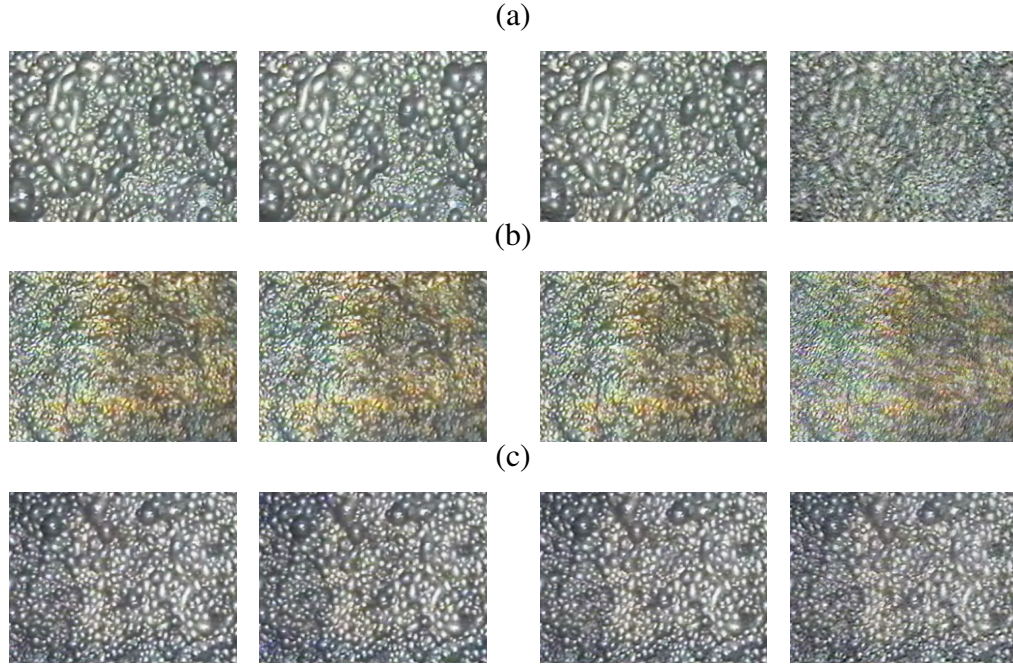


FIGURA 6.8. Resultados del test de predicción para los cuadros $\tau + 1$ y $\tau + 2$, Lado izquierdo: cuadros reales; Lado derecho: cuadros predichos; (a): Estado operacional 1; (b): Estado operacional 2; (c): Estado operacional 3.

El diseño de un controlador predictivo, presenta la necesidad de contar con un modelo que sea capaz de predecir con exactitud el comportamiento futuro de la espuma. Para probar las capacidades de predicción de los modelos identificados, se realizaron test de predicción para cada estado de operación. La figura 6.8 muestra los resultados obtenidos en los test de predicción de 2 pasos, que consiste en predecir los cuadros $\tau + 1$ y $\tau + 2$, para cada estado de operación, utilizando una secuencia de largo $\tau = 60$ y $n = 40$ componentes principales para el proceso de identificación del modelo.

Si bien las predicciones a 2 pasos son muy exactas el tiempo predicho es 0.07 s, que es despreciable en un proceso como la flotación. Por esto se realizaron predicciones con mayor horizonte para evaluar el deterioro en la predicción. La figura 6.9 muestra los resultados del test de predicción para los cuadros $\tau + 10$ y $\tau + 15$, que corresponde a una predicción de 0.3 s y 0.5 s respectivamente. Se observa que las predicciones se deterioran

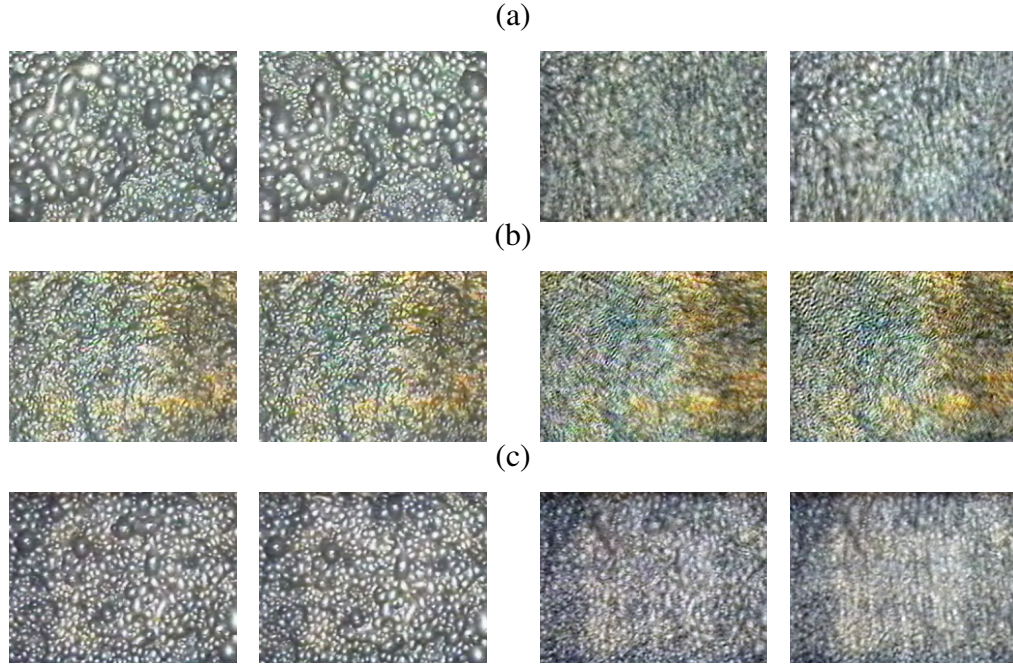


FIGURA 6.9. Resultados del test de predicción para los cuadros $\tau + 10$ y $\tau + 15$, Lado izquierdo: cuadros reales; Lado derecho: cuadros predichos; (a): Estado operacional 1; (b): Estado operacional 2; (c): Estado operacional 3.

a medida que el tiempo avanza, debido a que los modelos identificados rescatan las características espacio-temporales de la espuma a un plazo dependiente directamente del largo τ de la secuencia utilizada para la identificación.

El test de predicción es una herramienta poderosa de evaluación *a posteriori* de los modelos obtenidos. Por esto se realizaron pruebas de predicción a un paso variando el largo de la secuencia τ , con el fin de evaluar el efecto del parámetro τ en la exactitud de la predicción. Las figuras 6.10, 6.11 y 6.12 muestran gráficamente los resultados obtenidos para el error de predicción del cuadro $\tau + 1$ en función del largo de la secuencia τ para cada estado de operación, utilizando un número fijo de $n = 40$ componentes principales en el proceso de identificación de los modelos.

Nuevamente se aprecia que el error de predicción decrece monotonamente para los 3 estados, a medida que el parámetro τ aumenta. Sin embargo, el costo computacional

asociado al aumento de τ es muy grande como para aumentarlo en forma indiscriminada. Un valor de equilibrio entre bajo error de predicción y costo computacional aceptable es $\tau = 60$.

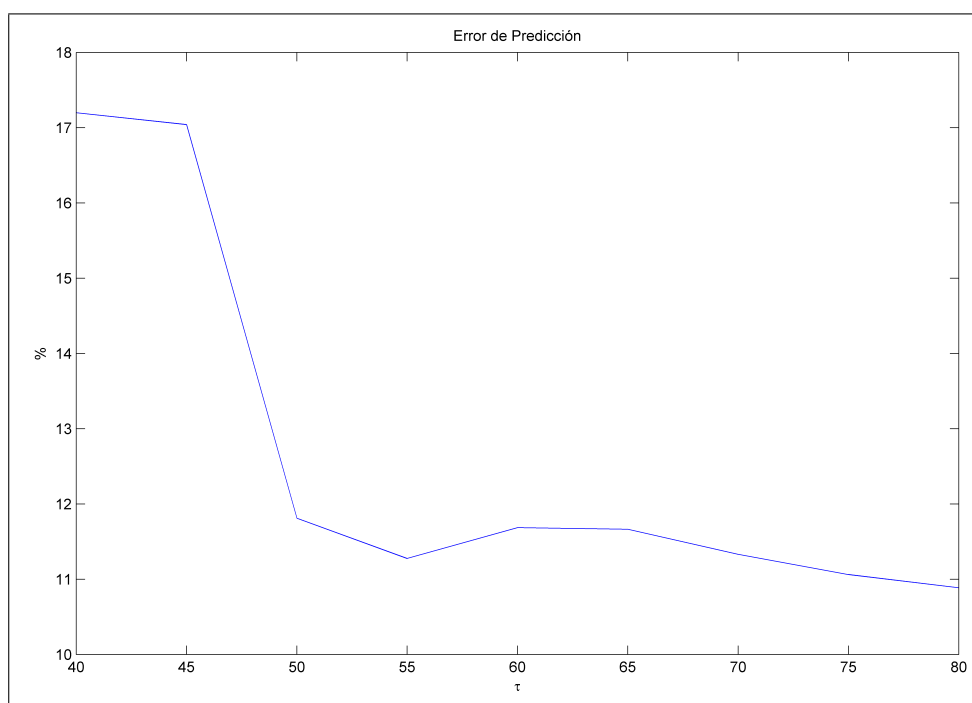


FIGURA 6.10. Error de predicción para el estado 1

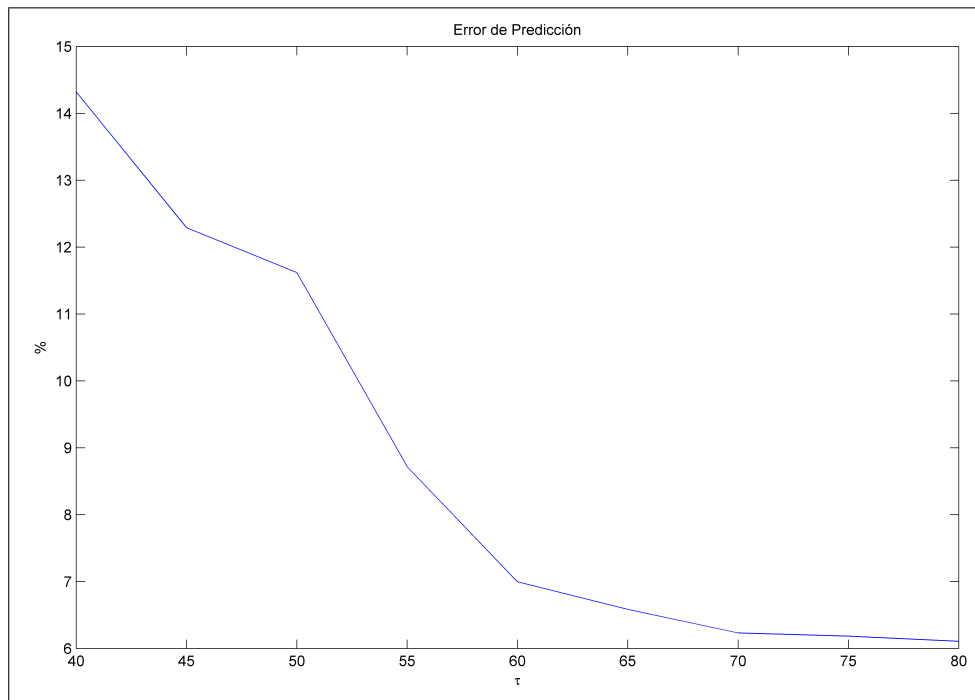


FIGURA 6.11. Error de predicción para el estado 2

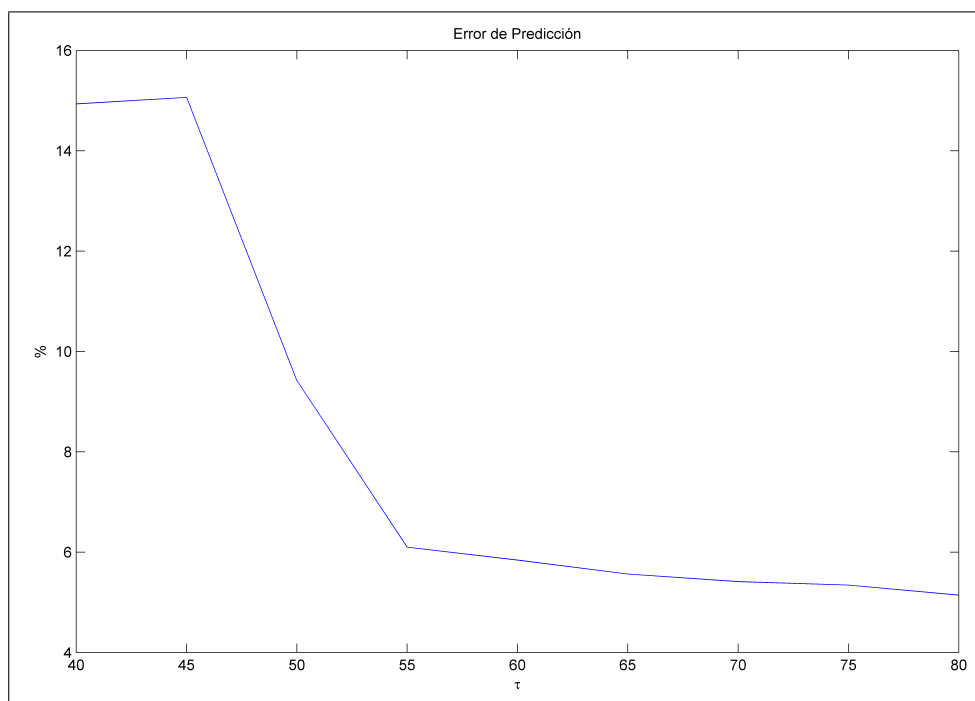


FIGURA 6.12. Error de predicción para el estado 3

6.2.4. Clasificación del Estado de Operación

Para propósitos de clasificación en línea del estado de operación, se programó un algoritmo basado en kernels de Binet-Cauchy y *nearest neighbor classification*. Para probar la robustez del algoritmo, se realizaron pruebas seleccionando secuencias de forma aleatoria de un largo $\tau = 50$ cuadros en los distintos estados de operación. La tabla 6.1 muestra los resultados obtenidos en el test de clasificación. Los resultados muestran que el algoritmo es capaz de reconocer en forma eficiente el estado de operación actual, con un corto tiempo de procesamiento asociado, lo que permite la implementación en línea del algoritmo de reconocimiento de texturas.

TABLA 6.1. Resultados del test de clasificación de estados operacionales.

Real/Detectado	Estado 1	Estado 2	Estado 3
Estado 1	37	0	3
Estado 2	0	40	0
Estado 3	2	0	38

La futura inclusión de variables manipuladas, tales como el nivel de pulpa, taza de adición de reactivos, flujo de aire, etc. en los modelos que describen la apariencia visual de la espuma permitirá implementar otros sistemas de reconocimiento del estado de operación, como por ejemplo, sistemas basados en estimación de estado similares a los utilizados en esquemas de detección y diagnóstico de fallas (Ferrari-Trecate et al., 2002).

7. CONCLUSIONES Y TRABAJO FUTURO

El estudio realizado muestra que es posible extender los conceptos de modelo predictor y control predictivo basado en modelos a procesos visuales. Como ejemplo se desarrolló un sistema de control predictivo para controlar el color de una secuencia generada artificialmente.

Asimismo se comprobó que las técnicas de modelación de procesos visuales se pueden aplicar a un proceso de alta complejidad. Específicamente se caracterizó el proceso visual descrito por las burbujas de una celda de flotación mediante modelos ARMA y se predijo la evolución de la apariencia superficial utilizando el modelo obtenido.

Utilizando modelación híbrida-experta se logró incorporar conocimiento experto previo acerca del comportamiento de la espuma, en forma de estado de operación discreto del modelo híbrido.

Los análisis de sensibilidad para los parámetros de identificación n y τ mostraron que tanto el error de síntesis, como el error de predicción decrecen monotonamente a medida que los parámetros aumentan. Sin embargo, el aumento indiscriminado tiene un alto costo computacional asociado que impide la implementación en línea y en tiempo real con el equipamiento disponible en estos momentos. En los análisis se demostró que un equilibrio entre bajo error y requerimientos computacionales aceptables se da con $n = 40$ y $\tau = 60$.

El análisis de reconocimiento del estado de operación, mostró que es posible identificar eficientemente el estado de operación mediante la distancia entre modelos ARMA asociados utilizando kernels de *Binet-Cauchy*.

Como trabajo futuro se propone incluir los efectos sobre la apariencia superficial de variables manipuladas del proceso de flotación, con el fin de obtener un modelo del tipo ARMAX que describa la evolución de la apariencia superficial. En una primera etapa se sugiere incluir el efecto del nivel de la pulpa, manteniendo la adición de reactivos proporcional al tonelaje de alimentación, para en una etapa posterior incluir como variable manipulada la adición de reactivos. También se propone realizar una modelación híbrida que incluya un mayor número de estados de operación, con el fin de obtener un modelo más fiel a la realidad del comportamiento de la apariencia de la espuma. El modelo esperado relaciona las variables físicas mediante funciones (lineales o no lineales) con el espacio de las componentes principales del proceso visual que describen las burbujas.

Finalmente se debe diseñar el controlador predictivo basado en sensores visuales según el esquema propuesto en la figura 5.6, que calcule la ley de control utilizando el modelo híbrido desarrollado para la predicción del comportamiento del proceso, los modelos ARMAX de la apariencia superficial y un sistema de reconocimiento de estados de operación basado en información visual.

BIBLIOGRAFIA

- Araya, J. F. (2004). *Estimación de parámetros para su aplicación en detección y diagnóstico de fallas*. Unpublished master's thesis, Departamento de Ingeniería Eléctrica, Pontificia Universidad Católica de Chile, Chile.
- Bemporad, A., & Morari, M. (1999). Control of systems integrating logic, dynamics and constraints. *Automatica*, v. 35(n. 3), 407–427.
- Botha, C. P. (1999). *An on-line machine vision flotation froth analysis platform*. Unpublished master's thesis, Department of Electronic Engineering, University of Stellenbosch, South Africa.
- Camacho, E., & Bordons, C. (2004). *Model predictive control* (2nd ed.). London: Springer.
- Casali, A., Gonzalez, G., Agosto, H., & Vallebuona, G. (2002). Dynamic simulator of a rougher flotation circuit for a copper sulphide ore. *Minerals Engineering*, v. 15(n. 4), 253–262.
- Chen, S., Murphy, T., & Subbayan, R. (1996). A color measurement and control system for paper-making processes. In *Proc. of the IEEE International Conference on Control Applications, 15–18 september 1996* (Vol. v. 1, pp. 136–142).
- Christofides, P. D., & El-Farra, N. H. (2005). *Control of nonlinear and hybrid process systems*. Berlin: Springer.
- Cipriano, A., Sepulveda, C., & Guarini, M. (1997). Expert system for supervision of mineral flotation cells using artificial vision. In *Proc. of the IEEE International Symposium on Industrial Electronics, 7–11 july 1997* (Vol. v. 1, pp. 149–153).
- Cruz, B. J. (2006). Modelación y análisis de un sistema híbrido: Un caso de estudio con un sistema de tanques. *Ingeniería, Revista académica de la FI-UADY*, v. 10(n. 2), 5–15.
- Doretto, G., Chiuso, A., Wu, Y., & Soatto, S. (2003). Dynamic textures. *International Journal of Computer Vision*, v. 51(n. 2), 91–109.

- Ferrari-Trecate, G., Mignone, D., & Morari, M. (2002). Moving horizon estimation for hybrid systems. *IEEE Trans. on Automatic Control*, v. 47(n. 2), 1663–1676.
- Ghoreyshi, A., & Vidal, R. (2006). Segmenting dynamic textures with ising descriptors, arx models and level sets. In *Proc. of the International Workshop on Dynamical Vision, may 2006* (pp. 127–141).
- Heemels, W., Schumacher, J., & Weiland, S. (2001). Equivalence of hybrid dynamical models. *Automatica*, v. 37(n. 7), 1085–1091.
- Holtham, P., & Nguyen, K. (2002). On-line analysis of froth surface in coal and mineral flotation using jkfrothcam. *International Journal on Mineral Procesing*, v. 64(n. 2), 163–180.
- Hyötyniemi, H., & Ylinen, R. (2000). Modeling of visual flotation froth data. *Control Engineering Practice*, v. 8(n. 3), 313–318.
- Ljung, L. (1999). *System identification, theory for the user*. Upper Saddle River, N.J: Prentice Hall.
- Melo, F., & Laskowski, J. (2006). Fundamental properties of flotation frothers and their effect on flotation. *Minerals Engineering*, v. 19(n. 6–8), 766–773.
- Muñoz, C. (2000). *Diseño de un método de control predictivo con restricciones y su aplicación y prueba por simulación para la optimización económica de plantas de molienda-flotación de minerales*. Unpublished doctoral dissertation, Departamento de Ingeniería Eléctrica, Pontificia Universidad Católica de Chile, Chile.
- Núñez, F., Navarro, S., Aguado, A., & Cipriano, A. (2008). State estimation based model predictive control for LHD vehicles. In *Proc. of the 17th IFAC World Congress, Seoul, Corea, july 2008*.
- Paoletti, S., Julosky, A., Ferrari-Trecate, G., & Vidal, R. (2007). Identification of hybrid systems: A tutorial. *European Journal of Control*, v. 13(n. 2–3), 242–260.
- Sbarbaro, D. (2002). An on-line color management system based on a non-linear predictive approach. In *Proc. of the 15th IFAC World Congress, Barcelona, Spain, june 2002*.

- Soliman, S. S., & Srinath, M. D. (1997). *Continuous and discrete signals and systems* (2nd ed.). USA: Prentice Hall.
- Vidal, R. (2007). Recursive identification of switched arx systems. *Automatica, Aceptado*.
- Vidal, R., Soatto, S., & Chiuso, A. (2007). Applications of hybrid system identification in computer vision. In *Proc. of the European Control Conference, Kos, Greece, july 2007*.
- Vishwanathan, S., Smola, A., & Vidal, R. (2006). Binet-cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes. *International Journal of Computer Vision*, v. 73(n. 1), 95–119.
- Welch, G., & Bishop, G. (2004). *An introduction to the Kalman Filter* (Tech. Rep.). University of North Carolina, Chapel Hill.
- Wills, B. A. (1992). *Mineral processing technology*. New York: Pergamon Press.

ANEXO A. STATE ESTIMATION BASED MODEL PREDICTIVE CONTROL FOR LHD VEHICLES

State Estimation Based Model Predictive Control for LHD Vehicles^{*}

Felipe Núñez^{*} Sergio Navarro^{*} Alberto Aguado^{**}
Aldo Cipriano^{*}

^{*} Department of Electrical Engineering,
Pontificia Universidad Católica de Chile, Santiago, Chile,
(e-mail: fenunez@ing.puc.cl, sanavarm@puc.cl, aciprian@ing.puc.cl)
^{**} Cybernetics, Mathematics and Physics Institute, La Habana, Cuba,
(e-mail: aguado@icmf.inf.cu)

Abstract: LHD (load-haul-dump) vehicles are used extensively in underground mining operations for ore transporting, primarily in tunnels where access is difficult or dangerous. To ensure underground efficient and safe LHD's performance, a robust feedback control strategy is needed. A state estimation based MPC scheme was designed for control purposes, and evaluated by simulation. The state estimator was developed by testing four approaches, in order to select the optimal one: the extended Kalman Filter, Particle Filter, Moving Horizon Estimator and a genetic algorithm based Moving Horizon Estimator. The simulation shows that non-linear MPC performs better than linear MPC for path tracking.

Keywords: Model Predictive Control; Nonlinear system control; Nonlinear observer and filter design; Monte Carlo methods; Genetic algorithms.

1. INTRODUCTION

LHD (load-haul-dump) vehicles are used extensively in underground mining operation for ore transporting, primarily in tunnels where access is difficult or dangerous. To ensure underground efficient and safe LHD's performance, a robust feedback control strategy is needed.

Model predictive control (MPC) is a widely used control concept, specially in the process industry. Most popular MPC applications use black-box linear models obtained from process data with the disadvantage that these models do not consider non-linear dynamics. State space based MPC seems to be an interesting LHD control alternative, since it can deal with multivariable non-linear dynamics and explicit constraints. However, the use of a state space model brings a new problem, the nonlinear state estimation, that must be successfully solved in a previous stage.

The most universally accepted state estimator is the Kalman Filter, due to its proved qualities as an optimal estimator and predictor for linear systems (Welch and Bishop, 2004). However, the corresponding nonlinear state estimator, the Extended Kalman Filter (EKF), can not claim a comparable success, due mainly to the fact that it is based on a linear approximation of the system model.

Recently, interest for state estimators based on Monte Carlo simulation as the so-called Particle Filters (PF) has reemerged. Also, the Moving Horizon Estimation (MHE)

method has been successfully compared with Extended Kalman Filtering (Haseltine and Rawlings, 2005) in the case of chemical reactors state estimation. Since nonlinear optimization is required the use of Genetic Algorithms (GA) represents an attractive option for on-line MHE implementation.

The purpose of this paper is to design a state space based model predictive control strategy for particular mining vehicles known as Load-Haul-Dump (LHD). Two MPC schemes, a linear and a non-linear MPC, have been compared in order to test their potentials on path tracking. Since a state estimator is needed, a comparison of four estimation methods was realized, in order to choose the optimal one for MPC design.

2. LHD VEHICLES

A LHD (Load-Haul-Dump) is a centrally articulated, four wheeled vehicle, steered by controlling the angle between its two constituent sections (Altafani, 1999), (Ridley and Corke, 2001). Each of the two sections has its own axle, and the wheels are not steerable. Thus, an LHD's control and path tracking characteristics differ significantly from those of conventional vehicles such as automobiles.

Figure 1 shows a scheme where the LHD main variables are specified. A simplified nonlinear model of a LHD, which only takes into account the cinematic relations among variables, can be formulated in the state space (Altafani, 1999) as follows:

$$\dot{x}_1 = u_1 \cos(\theta_1)$$

^{*} This work was supported by Fondecyt Project No. 1050684 and the Pontificia Universidad Católica de Chile

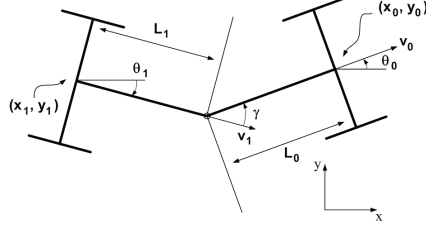


Fig. 1. Simplified scheme of a LHD

$$\begin{aligned} \dot{y}_1 &= u_1 \sin(\theta_1) \\ \dot{\theta}_1 &= \frac{u_1 \sin(\gamma) - L_0 \Gamma}{L_0 + L_1 \cos(\gamma)} \\ \dot{\gamma} &= \Gamma \end{aligned} \quad (1)$$

In this model, the variables usually measured are the angles θ_1 and γ . The manipulable variables are velocity u_1 and Γ , that is proportional to the applied torque.

3. STATE ESTIMATION

3.1 Extended Kalman Filter

The Extended Kalman Filter (EKF) is a suboptimal filter that minimizes the mean error of the state estimation of the LHD non-linear system given by the followings equations:

$$x_{k+1} = f(x_k, u_k, w_k) \quad (2)$$

$$y_k = Cx_k + v_k \quad (3)$$

In these equations, x_k is the state vector of the system, u_k is the vector of the manipulated variables, y_k is the measured output vector, $f(\cdot)$ is the non-linear function that define the LHD dynamics, C is the output matrix and w_k and v_k are the process and measurement disturbances which we assume characterized by white noise covariance matrices Q and R , respectively.

The EKF algorithm first estimates the state and measurement vectors at time k using the previous information and the system equations. Along with this, the covariance matrix P_k is also estimated:

$$x_{k+1}^- = f(\hat{x}_k, u_k, 0) \quad (4)$$

$$y_{k+1}^- = C\hat{x}_k \quad (5)$$

$$P_{k+1}^- = A_{k+1}\hat{P}_k A_{k+1}^T + Q \quad (6)$$

Using the covariance matrix R and the residual covariance matrix S_k , the Kalman gain K_k is obtained:

$$S_{k+1} = CP_{k+1}^- C^T + R \quad (7)$$

$$K_{k+1} = P_{k+1}^- C^T S_{k+1}^{-1} \quad (8)$$

Finally, the estimations are corrected using the Kalman gain and the difference between the estimation and the measurement:

$$\hat{x}_{k+1} = x_{k+1}^- + K_{k+1}(y_{k+1} - y_{k+1}^-) \quad (9)$$

$$\hat{y}_{k+1} = C\hat{x}_{k+1} \quad (10)$$

$$\hat{P}_{k+1} = (I - K_{k+1}C)P_{k+1}^- \quad (11)$$

In the preceding equations, A_k is the linearization of the non-linear function $f(\cdot)$ around an operating point at time k :

$$A_{k+1}(i, j) = \frac{\partial f_{[i]}}{\partial x_{[j]}}(\hat{x}_k, u_k, 0) \quad (12)$$

3.2 Particle Filter

Particle Filters (PF) are also suboptimal filters. They perform sequential Monte Carlo estimation based on point mass or particle representation of probability densities (Arulampalam et al., 2002). The most common method is Sequential Importance Sampling (SIS), which attempts to approximate the posterior probability density function (pdf) of the state $p(x_k/y_{1:k})$ with a set of realizations x_k^i , $i = 1 \dots N_p$, known as particles. $y_{1:k}$ represents the histogram of y from 1 to k . An approximation of the distribution is given by:

$$\hat{p}(x_k/y_{1:k}) = \frac{1}{N_p} \sum_{i=1}^{N_p} \delta(x_k^i) \quad (13)$$

Here $\delta(x_k^i)$ is a Dirac delta in (x_k^i) . This approximation can be used to obtain the mean value of the distribution:

$$E(x_k/y_{1:k}) = \int x_k p(x_k/y_{1:k}) dx_k \approx \frac{1}{N_p} \sum_{i=1}^{N_p} x_k^i \quad (14)$$

However, it is not possible to sample directly from the posterior distribution $p(x_k/y_{1:k})$. To generate samples, a known auxiliary distribution $q(x_k/y_{1:k})$ referred to as an Importance Function must be use, whose only constraint is that its support include the support of the real distribution. To represent the desired distribution, the samples must be weighted. The weights $\bar{w}(x_k)$ used to correct the samples are given by:

$$\bar{w}(x_k) = \frac{p(x_k/y_{1:k})}{q(x_k/y_{1:k})} \quad (15)$$

Solving for the distribution $p(\cdot)$ in (15) and then substitute it into (14) yields:

$$E(x_k/y_{1:k}) = \int x_k \bar{w}(x_k) q(x_k/y_{1:k}) dx_k \approx \frac{1}{N_p} \sum_{i=1}^{N_p} x_k^i \bar{w}_k^i \quad (16)$$

The samples x_k^i are now obtained from the importance function. If the prior state distribution is used as the importance function, then the weights satisfy the relation (Arulampalam et al., 2002):

$$\bar{w}_k^i = \bar{w}_{k-1}^i p(y_k/x_k^i) \quad (17)$$

3.3 Moving Horizon Estimation

Consider the system evolution as a Markov process, the quantity of interest in state estimation becomes the conditional probability density of the state evolution x_k , given the measurements y_k :

$$p(x_0, x_1, \dots, x_T | y_0, y_1, \dots, y_{T-1}) \quad (18)$$

The optimal state estimation is then a functional $L(\cdot)$ of the conditional probability density function (18). Typically $L(\cdot)$ is chosen as the maximum a posteriori Bayesian (MAP) estimation.

Using the Markov property and assuming normal distributions for the measurement noise, the disturbances and the initial state, the probabilistic expression could be transform into a mathematical programming problem.

Under the precedent assumptions, state estimation becomes the following optimization problem (Rao, 2000):

$$\min_{\hat{x}_0, \{\hat{w}_k\}_{k=0}^{T-1}} \Phi_T(\hat{x}_0, \{\hat{w}_k\}) \quad (19)$$

Where:

$$\begin{aligned} \Phi_T(\hat{x}_0, \{\hat{w}_k\}) = & \sum_{k=0}^{T-1} \{(y_k - C\hat{x}_k)^T R^{-1} (y_k - C\hat{x}_k) \\ & + \hat{w}_k^T Q^{-1} \hat{w}_k\} + (x_0 - \hat{x}_0)^T \Pi^{-1} (x_0 - \hat{x}_0) \end{aligned} \quad (20)$$

Subject to:

$$\hat{x}_{k+1} = f(\hat{x}_k, u_k, \hat{w}_k) \quad (21)$$

This is known as the full information problem where x_0 is an estimation of the initial state. The solution of the optimization problem gives $\hat{x}_0$ and the model disturbance sequence $\{\hat{w}_k\}$. The actual state is found by iterate in the state equation (21). In the full information problem the horizon length increase with time k , this makes impracticable an on-line implementation.

Considering the objective function $\Phi_T(\cdot)$, we may rearrange the terms by breaking the time interval into two pieces as follows:

$$\begin{aligned} \Phi_T(\hat{x}_0, \{\hat{w}_k\}) = & \sum_{k=T-N}^{T-1} \{(y_k - C\hat{x}_k)^T R^{-1} (y_k - C\hat{x}_k) + \\ & \hat{w}_k^T Q^{-1} \hat{w}_k\} + \sum_{k=0}^{T-N-1} \{(y_k - C\hat{x}_k)^T R^{-1} (y_k - C\hat{x}_k) \\ & + \hat{w}_k^T Q^{-1} \hat{w}_k\} + (x_0 - \hat{x}_0)^T \Pi^{-1} (x_0 - \hat{x}_0) \end{aligned} \quad (22)$$

The first term on the right depends only on the variables values at time $T - N$ so we can reformulate the problem as follows:

$$\begin{aligned} \min_{\hat{x}_{T-N}, \{\hat{w}_k\}_{k=T-N}^{T-1}} & \sum_{k=T-N}^{T-1} \{(y_k - C\hat{x}_k)^T R^{-1} (y_k - C\hat{x}_k) + \\ & \hat{w}_k^T Q^{-1} \hat{w}_k\} + Z_{T-N}(x_{T-N}) \end{aligned} \quad (23)$$

Subject to equation (21).

Where N is known as the fixed horizon length. The function $Z(\cdot)$ is known as the arrival cost function and represent the effect of the previous data on the state x_{T-N} . The formulation (23) is the basis of MHE.

Selecting the arrival cost penalty function properly MHE can approximate the full information problem, with processing time advantages.

Unfortunately an algebraic expression for arrival cost does not exist when the system is non linear or when constraints

are present. One strategy for approximating the arrival cost function is to take a first order Taylor approximation, yielding in an EKF covariance update formula (Rao, 2000).

Other strategies for approximating the arrival cost function have been studied recently (Tenny, 2002), (Haseltine, 2005).

3.4 Genetic Algorithm based Moving Horizon Estimation

Genetic algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures (Michalewicz, 1996).

The GA chosen for the present application has the following characteristics:

- (a) Individuals or chromosomes are encoded with real numbers (Michalewicz, 1996).
- (b) A ranking operation (Baker, 1987) assigns each individual z a natural number in accordance with its minimization criteria value $J(z)$. Thus each individual has a new criteria value $J_1(z)$ corresponding to its order number.
- (c) A selection operation known as Stochastic Universal Sampling (SUS) (Baker, 1987) is carried out that ensures the survival probability $p(z_i)$ of an individual z_i will be given by the equation:

$$p(z_i) = \frac{J_1(z_i)}{\sum_{j=1}^{N_{ind}} J_1(z_j)} \quad (24)$$

where N_{ind} is the number of individuals.

- (d) Chromosome crossover is performed, with probability pc , using an intermediate recombination operator defined by the following transformation:

$$\begin{aligned} z'_1 &= \alpha_1 z_1 + (1 - \alpha_1) z_2 \\ z'_2 &= \alpha_2 z_2 + (1 - \alpha_2) z_1 \\ \alpha_1, \alpha_2 &\in [0, 1] \end{aligned} \quad (25)$$

- (e) The mutation of a chromosome randomly selected with probability pm is obtained by changing a maximum of 20% of its genes.

3.5 Comparative Evaluation of State Estimators

The comparison of the estimators involves a trajectory composed of a straight-line segment and a curved segment in different scenarios. For a simulation horizon of $t = 50$ s in a MATLAB environment.

Each scenario is characterized by variations in the following parameters with respect to the nominal situation:

- (a) Covariance of process noise (Q_P) and measurement noise (R_P) for the model generating the data. The nominal values are given in the following matrices:

$$\begin{aligned} Q_P &= \begin{pmatrix} 10^{-3} & 0 & 0 & 0 \\ 0 & 10^{-3} & 0 & 0 \\ 0 & 0 & 10^{-5} & 0 \\ 0 & 0 & 0 & 10^{-5} \end{pmatrix} \\ R_P &= \begin{pmatrix} 10^{-1} & 0 \\ 0 & 10^{-3} \end{pmatrix} \end{aligned}$$

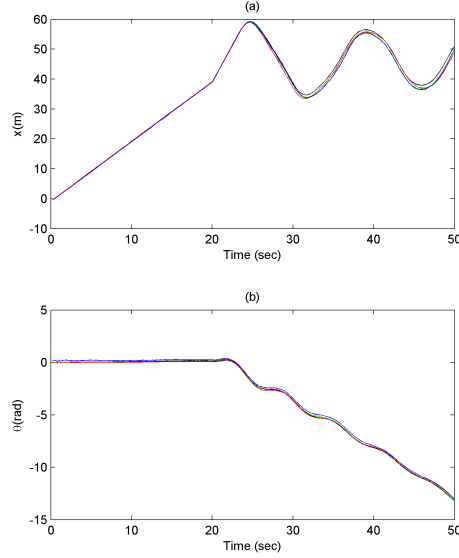


Fig. 2. State estimation, black: real state, green: EKF estimation, red: MHE estimation, blue: PF estimation; (a): non-direct observable state x ; (b): direct observable state θ .

- (b) To represent the effects of changes in the dispersion of process and measurement noise, the nominal covariance matrices are scaled by the parameter α , which may take the following values: 0.1, 1 or 10.
- (c) For EKF and MHE design in dispersion test, we assume that the values of matrices Q and R are known and equal to those for the model generating the data.
- (d) For sensitive analysis the values of matrices Q and R for the design of EKF and MHE are modified as follows $Q = \alpha_Q Q_P$ and $R = \alpha_R R_P$. As in the previous point, the parameters α_Q and α_R may take the following values: 0.1, 1 or 10.
- (e) Number of particles in the Particle Filter. For this parameter the following value were used: $N_P = 1200$.
- (f) Initial distribution of the filter particles. Normal distribution: mean 0, standard deviation 0.2.
- (g) Horizon length. A fixed horizon of $N = 10$ were used for moving horizon estimation.
- (h) Genetic algorithm parameters. The following parameters were chosen using trial and error for the genetic simulation: $N_{ind} = 20$, $pc = 0.8$, $pm = 0.4$, $\delta = 0.1$.

The quantitative indicator used to compare the estimators is the root mean square error (RMSE).

Figure 2 shows the estimation of non-direct observable state x and the direct observable state θ when $\alpha = 1$.

Table 1 shows that EKF performs a better estimation for the non direct observable state x in all covariance scenarios. The four estimators perform very similar results for the estimation of θ when the parameter α goes from 1 to 10. The estimation of PF, MHE and GA based MHE for

Table 1: State estimation errors depending on covariances.

	α	$x(m)$ RMSE	$\theta(rad)$ RMSE
EKF	0.1	0.07	0.01
	1	0.2	0.042
	10	1.32	0.09
PF	0.1	0.44	0.024
	1	0.52	0.042
	10	2.63	0.11
MHE	0.1	0.161	0.022
	1	0.29	0.037
	10	2.12	0.12
MHE-GA	0.1	0.168	0.022
	1	0.36	0.042
	10	2.55	0.14

the position x was poor at the same situation. However, for the direct observable state θ and $\alpha=1$, MHE performs a better estimation.

Tables 2, 3, 4 present the results of the sensibility test. Both estimators perform worse when parameters α_Q and α_R takes non nominal values. EKF performs, generally, better than MHE especially for the non direct observable state x . However, MHE perform better than EKF when $\alpha_Q = \alpha_R$.

Table 2: State estimation errors for the sensibility analysis with $\alpha_Q = 0.1$.

	α_R	$x(m)$ RMSE	$\theta(rad)$ RMSE
EKF	0.1	0.3405	0.0467
	1	0.4146	0.0447
	10	0.4483	0.0663
MHE	0.1	0.3072	0.0439
	1	0.9652	0.1264
	10	0.7411	0.0585

Table 3: State estimation errors for the sensibility analysis with $\alpha_Q = 1$.

	α_R	$x(m)$ RMSE	$\theta(rad)$ RMSE
EKF	0.1	0.5449	0.0534
	1	0.2	0.042
	10	0.3633	0.0371
MHE	0.1	0.5054	0.0591
	1	0.29	0.037
	10	1.457	0.1233

Table 4: State estimation errors for the sensibility analysis with $\alpha_Q = 10$.

	α_R	$x(m)$ RMSE	$\theta(rad)$ RMSE
EKF	0.1	0.236	0.0624
	1	0.2035	0.0469
	10	0.4047	0.0424
MHE	0.1	0.2975	0.0714
	1	0.225	0.0503
	10	0.3534	0.0373

4. MODEL PREDICTIVE CONTROL

MPC is designed as solving on-line a finite horizon open-loop optimization problem subject to system dynamics and constraints involving states and controls.

The LHD dynamic model in the state space is given by:

$$x_{k+1} = f(x_k, u_k, 0) \quad (26)$$

$$y_k = Cx_k \quad (27)$$

MPC search for the optimal control sequence that is solution of the following optimization problem (Camacho and Bordons, 2004):

$$\min_{\{u_k\}_{k=1}^{N_u}} \sum_{j=N_1}^{N_2} \delta(j) [\hat{y}_{t+j} - y_{ref,t+j}]^2 + \sum_{j=1}^{N_u} \lambda_{\Delta}(j) [\Delta u_{t+j-1}]^2 + \sum_{j=1}^{N_u} \lambda_u(j) [u_{t+j-1} - u_{ref,t+j-1}]^2 \quad (28)$$

Subject to equations (26), (27).

Where N_u is the control horizon, N_1 and N_2 are the prediction horizon limits, $\delta(j)$ is the reference deviation cost at time $t + j$, λ_u is the control deviation cost and $\lambda_{\Delta}(j)$ is the incremental control cost at time $t + j - 1$.

4.1 Linear MPC

Consider the non-linear model (26) (27) and the reference trajectory are given by:

$$x_{k+1}^* = f(x_k^*, u_k^*, 0) \quad (29)$$

$$y_{k+1}^* = Cx_k^* \quad (30)$$

x^* , u^* are the knowns optimal states and optimal control, that generates the trajectory (29), (30). Let we consider:

$$x = x^* + x^1 \text{ and } u = u^* + u^1 \quad (31)$$

Where x^1 and u^1 represents deviations from the optimal values x^* and u^* . Then taking a first order Taylor expansion of (29) we obtain a time variant linear model of the form:

$$x_{k+1}^1 = A_k x_k^1 + B_k u_k^1 \quad (32)$$

$$y_k = Cx_k^1 \quad (33)$$

The particular cost function is given by:

$$J(N_2, N_u) = \sum_{j=1}^{N_2} \delta [\hat{y}_{k+j} - y_{ref,k+j}]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u_{k+j-1}]^2 \quad (34)$$

MPC outputs are obtained by minimizing the deviations x^1 with minimum extra control effort u^1 .

The system outputs can be predicted as follows:

$$\begin{aligned} y_{k+1} &= C(A_k x_k + B_k u_{k-1} + B_k \Delta u_k) \\ y_{k+2} &= C(A_{k+1} A_k x_k + (A_{k+1} B_k + B_{k+1}) u_{k-1} \\ &\quad + (A_{k+1} B_k + B_{k+1}) \Delta u_k + B_{k+1} \Delta u_{k+1}) \\ &\vdots \end{aligned} \quad (35)$$

We could write (35) in matrix form:

$$Y_{k+1} = \Psi x_k + \Phi u_{k-1} + \Theta \Delta u_k \quad (36)$$

Where:

$$\Theta = \begin{pmatrix} CB_k & \cdots \\ C(A_k B_k + B_{k+1}) & \cdots \\ \vdots & \ddots \\ \sum_{i=k}^{k+N_2-1} C \prod_{j=k+1}^i A_j B_{2k+N_2-1-i} & \cdots \\ \vdots & 0 \\ \vdots & 0 \\ \vdots & \vdots \\ \sum_{i=k}^{k+N_2-N_u} C \prod_{j=k+1}^{k+i} A_j B_{2k+N_2-1-i} \end{pmatrix} \quad (37)$$

$$\Psi = \begin{pmatrix} CA_k \\ C \prod_{j=k+1}^{k+1} A_j \\ \vdots \\ C \prod_{j=k}^{k+N_2-1} A_j \end{pmatrix} \quad (38)$$

$$\Phi = \begin{pmatrix} CB_k \\ \sum_{i=k}^{k+1} C \prod_{j=k+1}^i A_j B_{2k+1-i} \\ \vdots \\ \sum_{i=k}^{k+N_2-1} C \prod_{j=k+1}^i A_j B_{2k+N_2-1-i} \end{pmatrix} \quad (39)$$

In the absence of constraints we can use the Maciejowsky equation (Camacho and Bordons, 2004), for time-invariant linear systems written in matrix form, to find a close solution for the optimal control sequence:

$$\Delta u_k = (\Theta^T \Theta + \lambda I)^{-1} \Theta^T (y_{ref,k} - \Psi \hat{x}_k - \Phi u_{k-1}) \quad (40)$$

4.2 Non-Linear MPC

Non-linear MPC is giving by solving on-line the minimization problem given by (26), (27), (28). Generally, this result in a non-convex non-linear optimization, which is difficult to solve because of the excessive computational requirements and multiple suboptimal local minima.

Non-linear optimization techniques or evolution based algorithms could be use to find the non-linear MPC outputs (Michalewicz, 1996).

4.3 Comparative Evaluation of State Estimation based Predictive Controllers

Both controller strategies were programmed in Simulink using an EKF as state estimator due his good results in the state estimation test. All simulations were executed in Simulink for a simulation horizon of $t = 35$ s.

Figure 3 shows the obtained LHD path tracking using the receding horizon control principle in three different scenarios: open loop, linear MPC and non-linear MPC, for following control and predictive horizons: $N_u = N_2 = 5$.

Table 5 shows the root mean square error in the coordinates x , y and the mean euclidean norm of the error vector.

Clearly, non-linear controller performs better than the linear MPC. However, the computational cost is higher, indeed, the non-linear MPC takes 12 times more time to run the simulation.

Table 5: Path tracking error for MPC schemes.

	$x(m)$ RMSE	$y(m)$ RMSE	Norm(m) Mean
Open Loop	2.73	2.33	3.25
Linear MPC	2.08	1.77	2.44
Non-Linear MPC	0.93	0.24	0.85

5. CONCLUSIONS AND FUTURE RESEARCH

In our study EKF performs, generally, better than MHE, PF and GA based MHE, especially for estimating the non direct observable state x . PF produces the worst results in estimation. However, it should be noted that we assumed that the disturbances are normally distributed, which represent an advantage for EKF and MHE.

Sensitivity test shows that EKF and MHE perform depends on the knowledge of the covariance matrices, which should not be the case in a real application.

GA seems to be an attractive alternative for on-line state estimation, since the results were very similar to the obtained with classical optimization methods.

Referring to the control schemes, linear MPC responds in between the non-linear control and the open loop scenario, with shorter processing time. However, the non-linear MPC presents smaller errors.

In the next future, we will test the four estimators using a more realistic LHD model, including explicit constraints, and non-Gaussian disturbances.

REFERENCES

- C. Altafini. A path tracking criterion for an LHD articulated vehicle. *Optimization and Systems Theory*. Royal Institute of Technology, Stockholm, Sweden, 1999.
- S. Arulampalam, S. Maskell, N. Gordon and T. Clapp. A tutorial on particle filters for online non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing.*, volume 50, pages 174–188, 2002.
- E. Baker. Reducing bias and inefficiency in the selection algorithms. *Proceedings of the Second International Conference on Genetic Algorithms*, Grefenstette, Volume 1, pages 14–21, 1987.
- E.F. Camacho and C. Bordons. *Model Predictive Control*. Springer, 2004.
- E.L. Haseltine. System analysis of stochastic and population balance models for chemically reacting systems. PhD Thesis. University of Wisconsin-Madison, 2005.
- E.L. Haseltine and J.B. Rawlings. Critical Evaluation of Extended Kalman Filtering and Moving Horizon Estimation. *Ind. Chem. Res.*, volume 44, pages 2451–2460, 2005.
- Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Program*. Springer series Artificial Intelligence, 3rd edition, 1996.
- C.V. Rao. Moving horizon strategies for the constrained monitoring and control of nonlinear discrete-time systems. PhD Thesis. University of Wisconsin-Madison, 2000.
- P. Ridley and P. Corke. Autonomous control of an underground mining vehicle. *Proceedings of the Australian Conference on Robotics and Automation*, Sydney, pages 26–31, 2001.
- M.J. Tenny. Computational strategies for nonlinear model predictive control. PhD Thesis. University of Wisconsin-Madison, 2002.
- G. Welch and G. Bishop. An introduction to the Kalman filter. University of North Carolina, Chapel Hill, NC, Tech. Rep. TR-95-041, 2004.

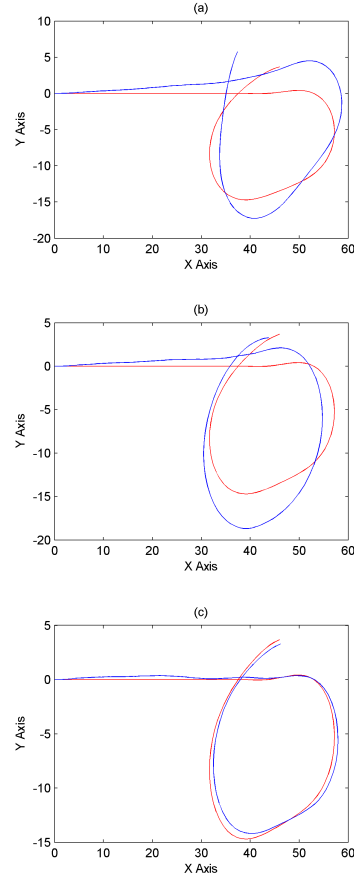


Fig. 3. Result of the simulations, red: reference trajectory, blue: LHD's real response; (a): open loop response; (b): linear MPC response; (c): non-linear MPC response.

ANEXO B. HYBRID MODELING OF FROTH FLOTATION SUPERFICIAL APPEARANCE APPLYING DYNAMIC TEXTURES ANALYSIS

HYBRID MODELING OF FROTH FLOTATION SUPERFICIAL APPEARANCE APPLYING DYNAMIC TEXTURES ANALYSIS

Felipe Núñez, Aldo Cipriano

Department of Electrical Engineering, Pontificia Universidad Católica de Chile, Chile
E-mail: femunez@ing.puc.cl, aciprian@ing.puc.cl

Abstract: Flotation is one of the most complex processes in mining industry. In fact, non linear characteristics, large delays, significant disturbances and important unmeasurable variables make the process very difficult for automatic control. However, experimented operators are able to take adequate control actions based on superficial froth appearance by defining several operational states. To ensure the obtaining of technical and economical benefits, it is necessary to better identify different operational states from the froth superficial appearance. In this work we propose a hybrid characterization of the froth surface applying dynamic texture techniques and texture classification. Results show that the models obtained represent accurately the froth dynamics and our detection algorithm is able to detect efficiently different operational states.

Key Words: Hybrid Systems, Dynamic Textures, Froth Flotation, ARMA models.

1 INTRODUCTION

Froth flotation is one of the most complex processes in mining industry. In fact, non linear characteristics, large delays, significant disturbances and important unmeasurable variables make the process very difficult for automatic control.

However, experimented operators are able to take adequate control actions based on superficial froth appearance. These subjective actions take into account froth characteristics like bubble size, bubble shape, bubble density, froth stability, froth speed and froth color.

Studies have been conducted to correlate these froth characteristics with concentrate grade, with variable success. Considering this fact several specialized visual sensors has been developed, that give quantitative measures of some representative froth characteristics like the mentioned above.

Next step in flotation control using image processing has been the design of expert systems that identify the current operational state and take control decisions based on operator knowledge [2]. However, these control strategies show variable success. To ensure the obtaining of technical and economical benefits, it is necessary to better identify different operational states from the froth superficial appearance.

Since there exist different operational states, hybrid modeling arises as an interesting option for froth flotation process. Moreover, switched affine models seem to be especially suitable for it since they consider explicitly different dynamics in each operational state [1].

However, for operational state identification purposes, we need to include visual information. Dynamic texture modeling seems to be an attractive option due to his good results in identification and classification of visual processes [3].

In this work we propose a switched affine hybrid modeling of the flotation froth, based on superficial froth appearance, applying dynamic texture techniques and texture classification. The aim of the modeling study is to develop a Hybrid Multivariable Predictive Controller based on dynamic texture classification.

2 FROTH FLOTATION

Froth flotation is a very widely used process for extracting valuable minerals from ore. It is based on chemical superficial differences between the particles that conforms the pulp.

In froth flotation the valuable mineral particles are made to be hydrophobic so that they attach to air bubbles generated by air injection at the bottom of the flotation cell, other particles should ideally remain (or be made) hydrophilic so that they remain in the pulp. Some of the reagents which are utilized in order to realize the chemical modification are: frother, depressant and collector [9].

In most flotation plants, it is customary for plant operators to monitor the froth surface visually and make adjustments to process parameters based on their interpretation of the surface appearance, by categorize froths mentally into a number of distinct types and develop appropriate operating strategies for each type, looking for the ideal froth appearance. Operator decisions are based on superficial

properties usually classified as dynamic: froth stability and froth speed; and static: bubbles size, bubbles shape, bubbles density and froth color [4],[5]. Manipulated variables include frother, depressant and collector addition rate, aeration rate, pulp level and froth height.

However, visual information from the surface froth in operating flotation plants has not been systematically cataloged. The interpretation of this information is frequently different from one operator to another mainly because it is based on intuitive experience. As a result, achieving optimal control is often not possible for the human operators.

3 HYBRID SYSTEMS

Hybrid systems are heterogeneous dynamical systems whose behavior is determined by the interaction between continuous and discrete dynamics [1]. The continuous dynamics is described by variables taking values from a continuous set, while the discrete dynamics is described by variables taking values from a discrete set, typically finite. The continuous or discrete-valued variables may depend on independent variables such as time, which in turn may be continuous or discrete [1].

Hybrid systems can be modeled using hybrid modeling techniques, resulting in a hybrid model that, depending on its structure, could be a *Mixed Logical Dynamics* (MLD) model, *PieceWise Affine* (PWA) model, *Linear Complementarity* (LC) model, etc. In this work we focus in the class of models named Switched Affine Models [6].

Switched affine models are defined as collections of linear/affine models, connected by switches that are indexed by a discrete-valued additional variable, called the discrete state. These models consider explicitly different dynamics for each discrete state. Models for which the discrete state is determined by a polyhedral partition of the state-input domain, are called piecewise affine models. Piecewise affine models are equivalent to several classes of hybrid models, and can therefore be used to describe systems that exhibit hybrid structure.

A general discrete-time switched affine model in input-output form is defined by introducing the regression vector [6]:

$$r_k = [y_{k-1}^T \cdots y_{k-n_a}^T u_k^T u_{k-1}^T \cdots u_{k-n_b}^T]^T \quad (1)$$

Where n_a and n_b are the model orders. And then by expressing the output y_k as a piecewise affine function of r_k namely:

$$y_k = \theta_{\sigma(k)}^T \begin{bmatrix} r_k \\ 1 \end{bmatrix} + e_k \quad (2)$$

Where $\sigma(k) \in \{1, \dots, s\}$ is the discrete state, s is the number of sub-models, $\theta_i, i = 1, \dots, s$ are the matrices of parameters defining each sub-model, and e_k is a noise/error term.

4 DYNAMIC TEXTURES

4.1 Modeling

Dynamic textures are sequences of images of moving scenes that exhibit certain stationary properties in time [3]. A linear dynamic texture is associated to a discrete time auto-regressive moving average process (ARMA) with unknown input distribution, which could be formulated in state space form as:

$$x(t+1) = Ax(t) + Bv(t) \quad (3)$$

$$y(t) = \phi(x(t)) + w(t) \quad (4)$$

Where $v(t)$ and $w(t)$ are random variables, and $\phi(\cdot)$ are a set of spatial filters [3].

There are several criteria for choosing a suitable class of filters, ranging from physical motivations to computational efficiency. In the simplest case, we can take $\phi(\cdot)$ to be the identity, and therefore look at the dynamics of individual pixels. We see the choice of filters as a dimensionality reduction step, and seek a decomposition of the image in a simple linear form, then the associated ARMA model yields [3]:

$$x(t+1) = Ax(t) + Bv(t) \quad (5)$$

$$y(t) = Cx(t) + w(t) \quad (6)$$

Let us consider the dynamic texture as a second-order stationary process with arbitrary covariance. It is well known that these processes can be modeled as the output of a linear dynamic system driven by white, zero mean Gaussian noise. Then we can approximate the visual process with the following ARMA model:

$$x(t+1) = Ax(t) + v(t) \quad (7)$$

$$y(t) = Cx(t) + w(t) \quad (8)$$

Where $v(t) \sim N(0, Q)$ and $w(t) \sim N(0, R)$.

4.2 Identification

The associated identification problem consists in estimating the model parameters A , C , Q , and R from the measurements obtained at time t [3].

Let $Y^\tau = [y(1), \dots, y(\tau)]$ and similarly for X^τ and W^τ , and notice that:

$$Y^\tau = CX^\tau + W^\tau \quad (9)$$

Now let $Y^\tau = \Lambda \Sigma Z^T$ be the singular value decomposition (SVD), and consider the problem of finding the best estimate of C in the sense of Frobenius. It follows immediately from the fixed-rank approximation property of the SVD that the unique solution is given by [3],[7]:

$$\hat{C} = \Lambda, \quad \hat{X} = \Sigma Z^T \quad (10)$$

Given $\hat{X}$, $\hat{A}$ can be determined uniquely again in the sense of Frobenius. This is trivially done in closed form using the estimate of $\hat{X}$ as:

$$\hat{A} = \Sigma Z^T D_1 Z (Z^T D_2 Z)^{-1} \Sigma^{-1} \quad (11)$$

Where:

$$D_1 = \begin{pmatrix} 0 & 0 \\ I_{t-1} & 0 \end{pmatrix} \quad (12)$$

$$D_2 = \begin{pmatrix} I_{t-1} & 0 \\ 0 & 0 \end{pmatrix} \quad (13)$$

Finally, the input and process noise covariance matrices can be estimated from:

$$\hat{Q} = \frac{1}{\tau-1} \sum_{t=1}^{\tau-1} \hat{v}(t) \hat{v}(t)^T \quad (14)$$

$$\hat{R} = \frac{1}{\tau-1} \sum_{t=1}^{\tau-1} \hat{w}(t) \hat{w}(t)^T \quad (15)$$

Where $\hat{v}(t)$ and $\hat{w}(t)$ are obtained as $\hat{x}(t+1) - \hat{A}\hat{x}(t)$ and $y(t) - \hat{C}\hat{x}(t)$ respectively.

It should be noted that $\hat{C}$, $\hat{X}$, $\hat{A}$, $\hat{Q}$ and $\hat{R}$ depends explicitly on the sequence length τ .

5 DISTANCES BETWEEN ARMA MODELS

In the following, we describe an approach to classification and recognition of dynamic textures based on distances defined on the space of models, which has been used for nearest-neighbor classification or *k-means* clustering [7].

To compute distances between dynamic models the so-called Binet-Cauchy kernels is used [8]. These kernels are obtained by computing the trace of a compound matrix of order q built from the output trajectories.

The Binet-Cauchy kernel of order q is defined as:

$$k_q(M_1, M_2) = \text{tr} E_q(Y_1 Y_2^T) \quad (16)$$

Where $E_q(x)$ is the compound matrix of order q of x , whose entries are given by the determinants of all the minors of x of size q . Y_1 and Y_2 are the output vectors obtained from the models M_1 and M_2 . As shown in [8], when $q = 1$, one can compute this kernel explicitly in terms of the ARMA models as:

$$k_1(M_1, M_2) = x_1^T(0) S x_2^T(0) \quad (17)$$

Where S is the solution to the Sylvester's equation:

$$A_1^T S A_2 - S + A_1^T C_1^T C_2 A_2 = 0 \quad (18)$$

From the Binet-Cauchy kernel, one may define a distance as [7]:

$$\begin{aligned} d_1(M_1, M_2)^2 &= k_1(M_1, M_1) - 2k_1(M_1, M_2) \\ &+ k_1(M_2, M_2) \end{aligned} \quad (19)$$

6 RESULTS

For modeling purposes we acquired image sequences of the cell surface, using a Firewire Industrial camera at a rate of 30 frames per second and a resolution of 320×240 pixels. We captured image sequences that represent about one hour of operation, in several operational states. For the training process we used sequences of 70 frames and 40 principal components for the texture synthesis, and to test the detection algorithm we used random sequences of 50 frames, whose do not overlap those used for the training process.

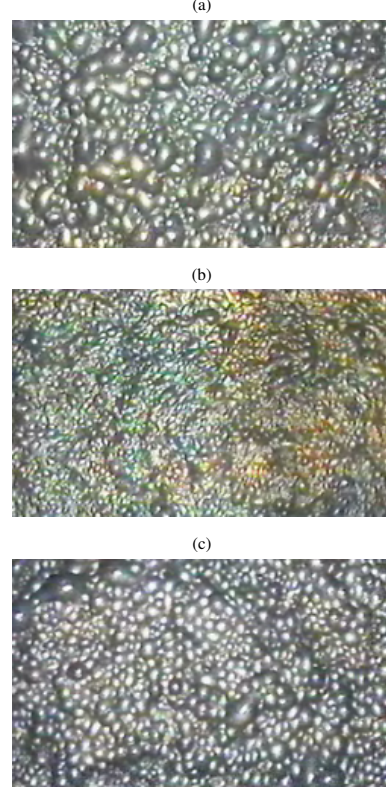


Figure 1: Operational states defined for the analysis; (a): Operational state 1; (b): Operational state 2; (c): Operational state 3.

6.1 Operational States

For the off line detection of the different operational states, based in superficial appearance, we define 3 operational states as show figure 1. These operational states have the following characteristics:

- Operational State 1:
 - Big size bubbles
 - Light gray froth
 - High density
 - Stable froth
- Operational State 2:
 - Small size bubbles
 - Yellow froth
 - Collapsed areas

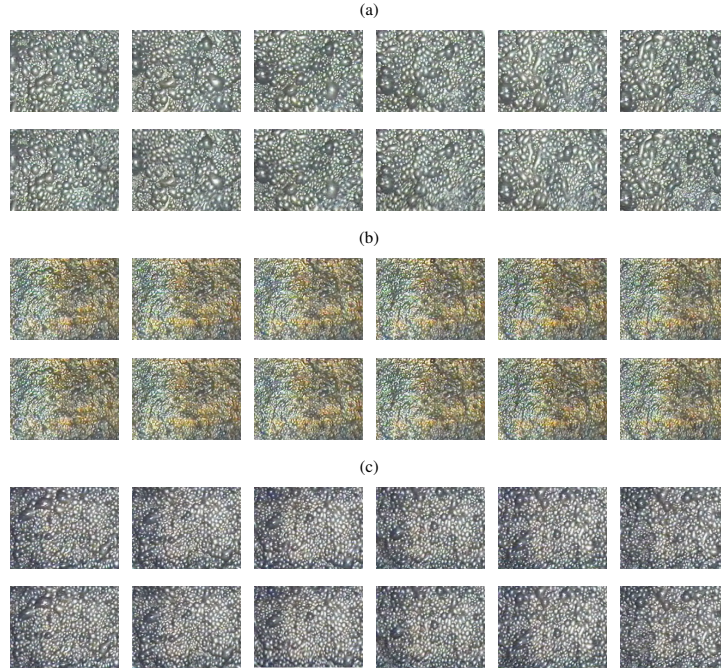


Figure 2: Selected frames from the real sequence (on top) and the corresponding frames obtained from the ARMA model (on bottom) for $t = 10, 20, 30, 40, 50, 60$; (a): Operational state 1; (b): Operational state 2; (c): Operational state 3.

- Unstable froth
- Operational State 3:
 - Middle size bubbles
 - Dark froth
 - High density
 - Stable froth

6.2 Texture Modeling and Synthesis

Once the different operational states were defined, we obtained the corresponding ARMA model associated to each operational state texture, applying dynamic textures techniques presented above.

Figure 2 shows selected frames from the real sequence and the corresponding frames obtained from the model synthesis, for each operational state. Clearly, the models obtained are able to capture froth spatio-temporal characteristics.

Figure 3 shows the results for two step prediction test, it consists in using the model obtained to predict the next two frames in the sequence. Result shows that the models obtained have good prediction capabilities, especially useful for the future design of our Hybrid Multivariable Predictive Controller (see introduction).

6.3 Operational States Detection

For detection purposes, we designed an identification algorithm based in Binet-Cauchy distance for the recognition of the current operational state using nearest neighbor classification. Table 1 shows the results obtained in detection tests.

Real/Detected	State 1	State 2	State 3
State 1	37	0	3
State 2	0	40	0
State 3	2	0	38

Result shows that our detection algorithm performs high effective detection for each operational state in short processing time, which made it very useful for on-line texture classification applications.

7 CONCLUDING REMARKS AND FUTURE RESEARCH

Hybrid modeling techniques gives the option to include expert knowledge as the form of an operational state, permitting to apply model based control techniques to froth flotation process.

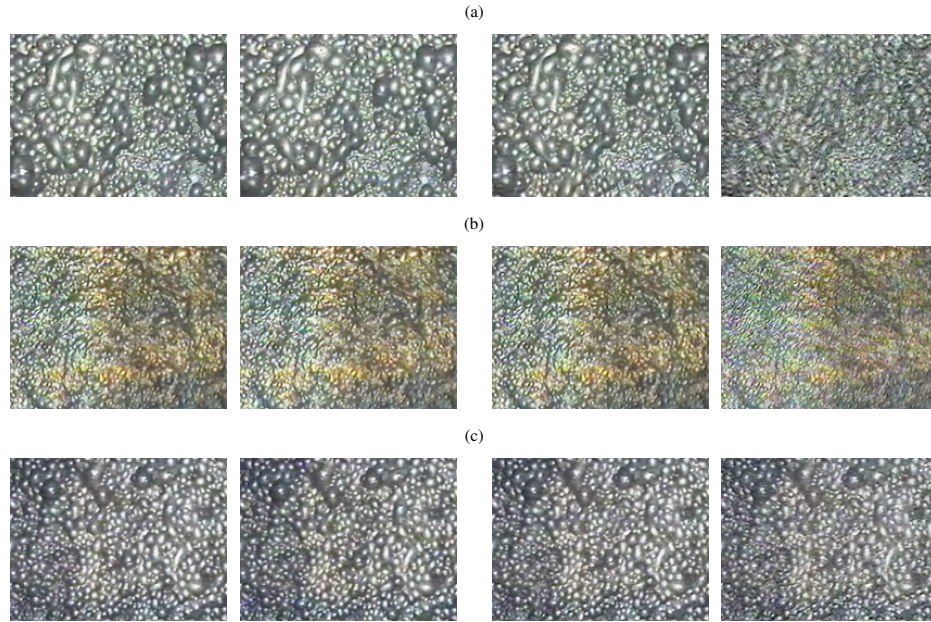


Figure 3: Results of the prediction test for frames $t + 1$ and $t + 2$, Left side: real frames; Right side: predicted frames; (a): Operational state 1; (b): Operational state 2; (c): Operational state 3.

Result shows that the models obtained are able to capture froth dynamics, giving the option to characterize the different operational states with an associate ARMA model.

The prediction test realized shows that the models obtained have good prediction capabilities for temporal froth evolution, which seem to be an attractive property for predictive control purposes. However, deterioration in predicted frames suggests the inclusion of manipulated variables in the model, yielding an ARMAX model to describe froth dynamics.

Detection test shows that our classification algorithm is capable of identifying efficiently the different operational states, based on distances between the associated models.

Future research should include analysis adding more operational states to study the detection algorithm robustness, and also the effects of process variables in froth behavior so we can design an hybrid predictive control scheme based on detection of operational states using textures classification.

REFERENCES

- [1] A. Bemporad and M. Morari, Control of Systems Integrating Logic, Dynamics and Constraints, *Automatica*, Vol.35, No.3, 407-427, 1999.
- [2] A. Cipriano, C. Sepulveda and M. Guarini, Expert System for Supervision of Mineral Flotation Cells Using Artificial Vision, *Proceedings of the IEEE international Symposium on Industrial Electronics*, 7-11, 1997.
- [3] G. Doretto, A. Chiuso, W. Wu and S. Soatto, Dynamic Textures, *International Journal of Computer Vision*, Vol.51, No.2, 91-109, 2003.
- [4] P.N. Holtham and K.K. Nguyen, On-line Analysis of Froth Surface in Coal and Mineral Flotation using JKFröthCam, *International Journal on Mineral Processing*, Vol 64, 163-180, 2002.
- [5] H. Hytyniemi and R. Ylinen, Modeling of Visual Flotation Froth Data, *Control Engineering Practice*, Vol 8, 313-318, 2000.
- [6] S. Paoletti, A. Julosky, G. Ferrari-Trecate and R. Vidal, Identification of Hybrid Systems: A Tutorial, *European Journal of Control*, Vol.13, 242-260, 2007.
- [7] R. Vidal, S. Soatto and A. Chiuso, Applications of Hybrid System Identification in Computer Vision, *Proceedings of the European Control Conference*, 2007.
- [8] S. Vishwanathan, A. Smola and R. Vidal, Binet-Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes, *International Journal of Computer Vision*, Vol.73, No.1, 95-119, 2006.
- [9] B. A. Wills and T. J. Napier-Munn, *Mineral Processing Technology*, Elsevier Science & Technology Books, 2006.

ANEXO C. SUPERFICIAL FROTH CHARACTERIZATION IN A FLOTATION CELL APPLYING DYNAMIC TEXTURES ANALYSIS AND ON-LINE TEXTURE CLASSIFICATION

SUPERFICIAL FROTH CHARACTERIZATION IN A FLOTATION CELL APPLYING DYNAMIC TEXTURES ANALYSIS AND ON-LINE TEXTURE CLASSIFICATION

Felipe Núñez, Aldo Cipriano
Electrical Engineering Department
Pontificia Universidad Católica de Chile

ABSTRACT

Flotation is one of the most complex processes in mining industry. In fact, non linear characteristics, large delays, significant disturbances and important unmeasurable variables make the process very difficult for automatic control. However, experimented operators are able to take adequate control actions based on superficial froth appearance. Considering this fact, several companies have developed specialized visual sensors that, applying image processing techniques, give quantitative measures of some representative froth characteristics.

Next step in flotation control using image processing has been the design of expert systems that identify the current operational state and take control decisions based on operator knowledge. However, these control strategies show variable results. To ensure the obtaining of technical and economical benefits, it is necessary to better identify different operational states from the froth superficial appearance.

In this work we propose a switched affine hybrid modeling of the flotation froth, based on superficial froth appearance, applying dynamic texture techniques and texture classification. The aim of the modeling study is to develop a Hybrid Multivariable Predictive Controller based on dynamic texture classification.

INTRODUCTION

Froth flotation is one of the most complex processes in mining industry. In fact, non linear characteristics, large delays, significant disturbances and important unmeasurable variables make the process very difficult for automatic control [10].

However, experimented operators are able to take adequate control actions based on superficial froth appearance. These subjective actions take into account froth characteristics like bubble size, bubble shape, bubble density, froth stability, froth speed and froth color [4],[6].

Several studies were conducted to correlate these froth characteristics with recuperation, with variable success. Considering this fact several specialized visual sensors has been developed, that gives quantitative measures of some representative froth characteristics like the mentioned above.

Next step in flotation control using image processing has been the design of expert systems that identify the current operational state and take control decisions based on operator knowledge [2]. However, these control strategies show variable success. To ensure the obtaining of technical and economical benefits, it is necessary to better identify different operational states from the froth superficial appearance.

Since there exist different operational states, hybrid modeling arises as an interesting option for froth flotation process. Moreover, switched affine models seem to be especially suitable for it since they consider explicitly different dynamics in each operational state [1].

However, for operational state identification purposes, we need to include visual information. Dynamic texture modeling seems to be an attractive option due to its good results in identification and classification of visual processes [3].

In this work we propose a switched affine hybrid modeling of the flotation froth, based on superficial froth appearance, applying dynamic texture techniques and texture classification. The aim of the modeling study is to develop a stabilizing froth Hybrid Multivariable Predictive Controller based on dynamic texture classification (controlled variables are froth properties).

DESIGN METHODOLOGY

Vision Based Froth Flotation Control

In most flotation plants, it is customary for plant operators to monitor the froth surface visually and make adjustments to process parameters based on their interpretation of the surface appearance, by categorizing froths mentally into a number of distinct types and developing appropriate operating strategies for each type, looking for the ideal froth appearance. Operator decisions are based on superficial properties usually classified as dynamic: froth stability and froth speed; and static: bubbles size, bubbles shape, bubbles density and froth color [4],[6]. Manipulated variables include frother, depressant and collector addition rate, aeration rate, pulp level and froth height [10].

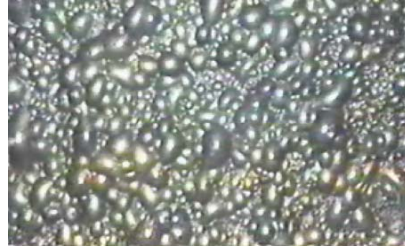
However, visual information from the surface froth in operating flotation plants has not been systematically catalogued. Depending on the design of the flotation cells and mineralogy of the ore, the visual information recognized by the plant operators exhibits its own idiosyncrasies. The interpretation of this information is frequently different from one operator to another mainly because it is based on intuitive experience. As a result, achieving optimal control is often not possible for the human operators.

Computer-based vision technology now offers a viable means of monitoring and controlling flotation performance at the cell level. The technology offers the following advantages:

- The measurement is non-intrusive and non-contact.
- No mechanical parts are involved, such as sampling and deaeration systems.
- Low capital cost.
- High sampling frequency.

Since the early 1990's, work related to flotation froth characterization using digital vision technology has received significant attention. In general, two approaches have been adopted. The first uses digital image analysis as a means to measure the parameters relating to the froth appearance that can be used to develop froth-based flotation models. The second attempts to relate froth feature measurements derived from image analysis to froth performance. At the same time several industrial sensors were developed, e.g.: *Outotech Frothmaster 2*, *Metso VisioFroth*, *JKTech JKFrothCam*, etc. These sensors give quantitative measures of some representative froth characteristics as: bubbles size, bubbles shape, froth speed, froth stability, froth color, etc.

The existence of different froth types suggest that froth flotation behavior could be characterized by separating the process in several operational states depending on froth characteristics. A classical sub-division in 3 operational states [4],[5],[6] has been used to test several image processing techniques as bubbles segmentation, froth speed estimation, etc.



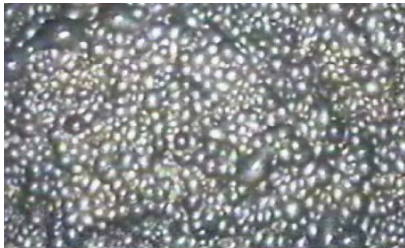
Operational State 1:

- Big size bubbles
- Light Gray Froth
- High density
- Stable Froth



Operational State 2:

- Small size bubbles
- Yellow Froth
- Collapsed areas
- Unstable Froth



Operational State 3:

- Middle size bubbles
- Dark Froth
- High density
- Stable Froth

Figure1: Operational states defined for the analysis

The 3 operational states shown in figure 1 represent the discrete states of our hybrid scheme, and our aim is to identify efficiently the current operational state based on froth texture characteristics. In the followings we describe the hybrid model structure that we associate to froth flotation process and the technique to obtain a froth texture model named dynamic texture techniques.

Froth Flotation Hybrid Modeling

Hybrid systems are heterogeneous dynamical systems whose behavior is determined by the interaction between continuous and discrete dynamics. The continuous dynamics is described by variables taking values from a continuous set, while the discrete dynamics is described by variables taking values from a discrete set, typically finite. The continuous or discrete-valued variables may depend on independent variables such as time, which in turn may be continuous or discrete.

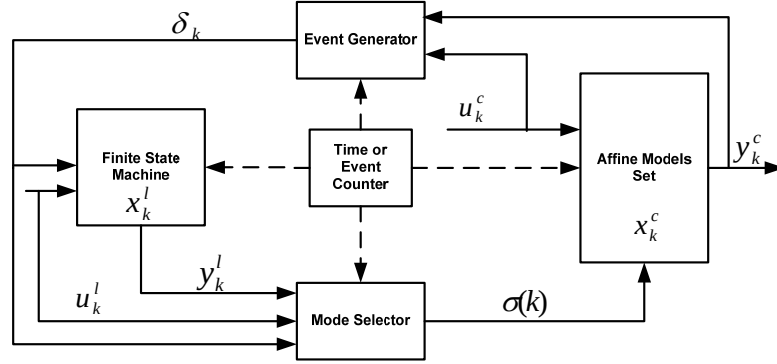


Figure 2: General scheme of a Switched Affine Hybrid System

Hybrid systems can be described using hybrid modeling techniques, resulting in a hybrid model that, depending on its structure, could be a Mixed Logical Dynamics (MLD) model [1], PieceWise Affine (PWA) model [1],[7], Linear Complementarity (LC) model [7], etc. In this work we focus in the class of models named Switched Affine Models, whose general scheme is shown in figure 2.

Switched affine models are defined as collections of linear/affine models, connected by switches that are indexed by a discrete-valued additional variable $\sigma(k) \in \{1, \dots, s\}$, called the discrete state, where s is the number of sub-models. Specifically in our model $\sigma(k) = 1$ for operational state 1, $\sigma(k) = 2$ for operational state 2 and $\sigma(k) = 3$ for operational state 3.

In the scheme of figure 2, the continuous and discrete variables can be distinguished by the labels c and l respectively. The interaction between these variables is determined by the event generator and the associated finite state machine. When state transition occurs the mode selector, based on state machine outputs and event generator signal, obtains the new value for $\sigma(k)$.

In this characterization the output vector y_k , that include continuous and discrete outputs, corresponds to froth properties that we pretend to control. The input vector u_k corresponds to manipulated variables as: frother, depressant and collector addition rate, aeration rate, pulp level and froth height.

Dynamic Textures

Dynamic textures are sequences of images of moving scenes that exhibit certain stationary properties in time [3]. A linear dynamic texture could be approximated by an associated autoregressive moving average model (ARMA) with Gaussian input distribution [3],[8]:

$$\begin{aligned} x(t+1) &= Ax(t) + v(t) \\ y(t) &= Cx(t) + w(t) \end{aligned} \quad (3)$$

Where $v(t)$ and $w(t)$ are the Gaussian process and measurement noise with covariance matrices Q and R respectively.

The associated identification problem consists of estimating the model parameters A , C , Q , and R from the measurements obtained at time t .

Let $Y^\tau \triangleq [y(1), \dots, y(\tau)]$ and similarly for X^τ and W^τ , and notice that:

$$Y^\tau = CX^\tau + W^\tau \quad (4)$$

Now let $Y^\tau = \Lambda \Sigma Z^T$ be the singular value decomposition, based on it we found the estimations of the model parameters from the fixed-rank approximation property of the SVD [3] [8].

In the following, we describe a novel approach to classification and recognition of dynamic textures based on distances defined on the space of models, which has been used for nearest-neighbor classification or k-means clustering, named Binet-Cauchy kernels distance [9].

The Binet-Cauchy kernel of order q is defined as:

$$k_q(M_1, M_2) = \text{tr} E_q(Y_1 Y_2^T) \quad (5)$$

Where $E_q(x)$ is the compound matrix of order q of x . As shown in [9], when $q=1$, one can compute this kernel explicitly in terms of the ARMA models $M_i = (x_i(0), A_i, B_i, C_i)$ as:

$$k_1(M_1, M_2) = x_1^T(0) S x_2(0) \quad (6)$$

Where Q is the solution to the Sylvester's equation:

$$A_1^T S A_2 - S + C_1^T C_2 = 0 \quad (7)$$

From the Binet-Cauchy kernel, one may define a distance as [8]:

$$d_1(M_1, M_2)^2 = k_1(M_1, M_1) - 2k_1(M_1, M_2) + k_1(M_2, M_2) \quad (8)$$

RESULTS

Texture Modeling and Synthesis

Once the different operational states were defined, we obtained the corresponding ARMA model associated to each operational state texture, applying dynamic textures techniques presented above. For the training process we used sequences of 70 frames, at 30 frames per second rate, and 40 principal components for the texture synthesis.

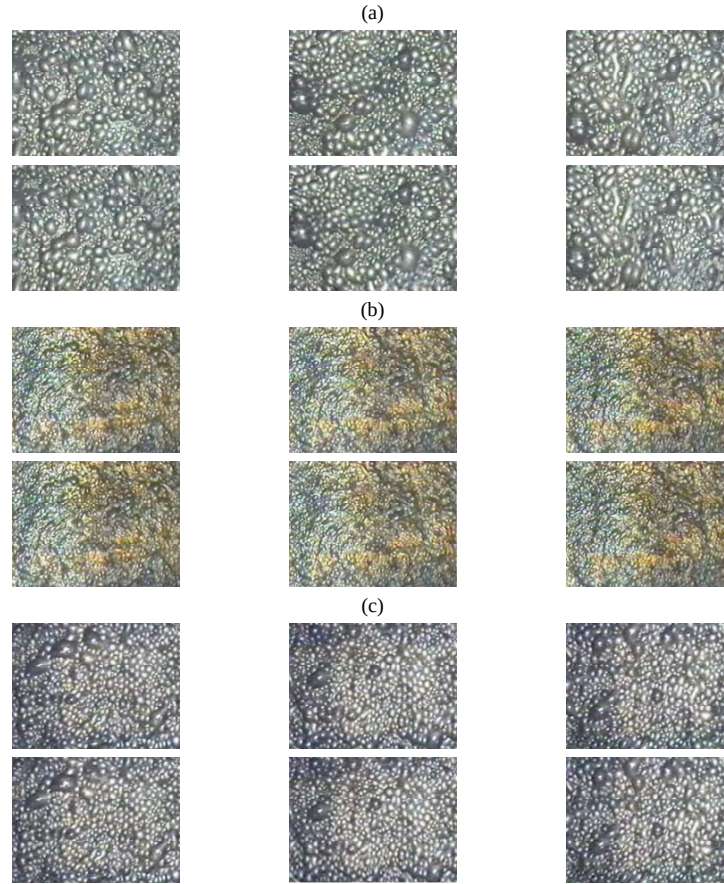


Figure3: Selected frames from the original sequence (on top) and the corresponding obtained from the ARMA model (on bottom) for $t=10,30,50$; (a): state 1; (b): state 2; (c): state 3

Figure 3 shows selected frames from the real sequence and the corresponding frames obtained from the model synthesis, for each operational state. Clearly, the models obtained are able to capture froth spatio-temporal characteristics.

Figure 4 shows the result for two step prediction test, it consists in using the model obtained to predict the next two frames in the sequence. Result shows that the models obtained have good prediction capabilities, especially useful for our hybrid approach.

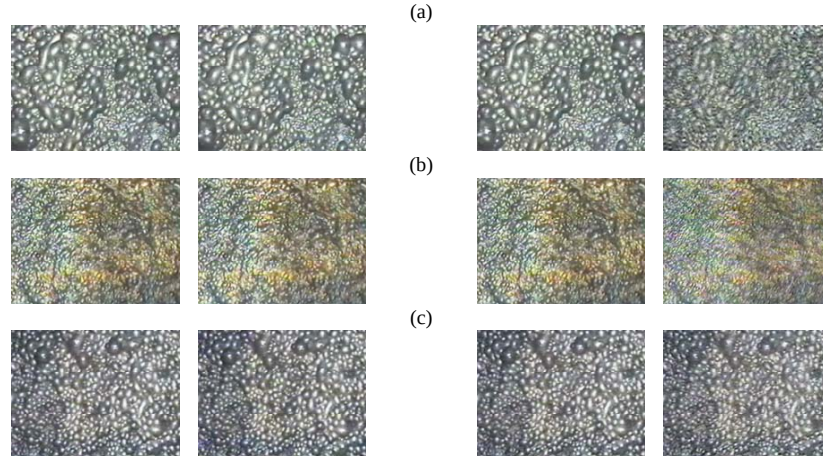


Figure 4: Results of the prediction test for frames $t+1$ and $t+2$, left side: real frames; right side: predicted frames; (a): state 1; (b): state 2; (c): state 3.

Operational States Detection

For detection purposes, we designed an identification algorithm based in Binet-Cauchy distance for the recognition of the current operational state using nearest neighbor classification. To test our detection algorithm, we used random sequences of 50 frames, obtained from the master clip. Table 1 shows the results obtained in detection tests.

Table1: Results of the detection tests.

Real / Detected	Operational State 1	Operational State 2	Operational State 3
Operational State 1	37	0	3
Operational State 2	0	40	0
Operational State 3	2	0	38

Result shows that our detection algorithm performs high effective detection for each operational state in short processing time, which made it very useful for on-line texture classification applications.

CONCLUSIONS AND FUTURE RESEARCH

Vision based froth flotation control seems to be an attractive option to control the process, since computational advances makes possible the acquisition and processing of great amount of process data. The use of specialized industrial visual sensors gives the option to make the control decisions independent of the operator knowledge, resulting in a control law that depends on the quantitative measures of these sensors.

Hybrid modeling techniques gives the option to include expert knowledge as the form of an operational state, permitting to apply classical model based control techniques to froth flotation process.

Result shows that dynamic textures techniques are able to capture froth dynamics, giving the option to characterize the different operational states with an associate ARMA model.

The prediction test realized shows that the models obtained have good prediction capabilities for temporal froth evolution, which seem to be an attractive property for predictive control purposes. However, deterioration in predicted frames suggests the inclusion of manipulated variables in the model, yielding an ARMAX model to describe froth dynamics.

Detection test shows that our classification algorithm is capable of identifying efficiently the different operational states, based on distances between the associated models.

Future research should include analysis adding more operational states to study the detection algorithm robustness, and also the effects of process variables in froth behavior so we can design a hybrid predictive control scheme based on detection of operational states using textures classification.

REFERENCES

- Bemporad A. and Morari M. (1999)** "Control of Systems Integrating Logic, Dynamics and Constraints". *Automatica*, 35(3), pp. 407-427. [1]
- Cipriano A., Sepulveda C. and Guarini M. (1997)** "Expert System for Supervision of Mineral Flotation Cells Using Artificial Vision". *Proceedings of the IEEE international Symposium on Industrial Electronics*, July, 1997, pp. 7-11. [2]
- Doretto G., Chiuso A., Wu W. and Soatto S. (2003)** "Dynamic Textures". *International Journal of Computer Vision*, vol 51(2), 2003, pp. 91-109. [3]
- Holtham P. N. and Nguyen K. K. (2002)** "On-line Analysis of Froth Surface in Coal and Mineral Flotation using JKFrthCam". *International Journal on Mineral Processing*, vol 64(2), 2002, pp.163-180. [4]
- Hytyniemi H. and Ylinen R. (2000)** "Modeling of Visual Flotation Froth Data". *Control Engineering Practice*, vol 8(3), 2000, pp. 313-318. [5]
- Moolman D.W., Aldrich C., Van Deventer J.S.J. and Stange W.W. (1994)** "Digital Image Processing as a Tool for On-Line Monitoring of Froth in Flotation Plants". *Minerals Engineering*, vol 7(9), 1994, pp. 1149-1164. [6]
- Paoletti S., Julosky A., Ferrari-Trecate G. and Vidal R. (2007)** "Identification of Hybrid Systems: A Tutorial". *European Journal of Control*, vol 13(2-3), 2007, pp. 242-260. [7]
- Vidal R., Soatto S. and Chiuso A. (2007)** "Applications of Hybrid System Identification in Computer Vision". *Proceedings of the European Control Conference*, Kos, Greece, July 2007. [8]
- Vishwanathan S., Smola A. and Vidal R. (2006)** "Binet-Cauchy Kernels on Dynamical Systems and its Application to the Analysis of Dynamic Scenes" *International Journal of Computer Vision*, vol 73(1), 2006, pp. 95-119. [9]
- Wills B. A. and Napier-Munn T. J. (2006)** "Mineral Processing Technology". Elsevier Science & Technology Books. [10]

ANEXO D. VISUAL INFORMATION PREDICTIVE MODELING FOR ADVANCED FLOTATION CONTROL

Visual information predictive modeling for advanced flotation control

Felipe Núñez ^a, Aldo Cipriano ^a

^aCollege of Engineering, Pontificia Universidad Católica de Chile

Abstract

Image processing sensors are emerging as an important measurement option in mineral processing, mainly due to their high sampling frequency and non-intrusive characteristics. Their principal application areas have been the determination of ore size distributions in grinding and froth features in flotation. On the other hand, in froth flotation processes model predictive control schemes have been systematically developed for tailings and concentrate grades using aeration and reagent addition rates as manipulated variables, but visual information has not been included. The design of predictive control strategies for froth flotation plants, using visual information, requires the generation of models capable of predicting the temporal evolution of froth features. In this paper we begin by reviewing classical image processing algorithms used in flotation froth feature extraction and then derive image processing-based predictive models. A new method is introduced for the characterization and recognition of visual information using dynamic texture techniques. A comparison of the models obtained from the two methods is also developed. Our results show that better predictions are obtained using dynamic texture-based models for froth color and froth speed. Finally we propose a hybrid scheme for surface froth appearance modeling.

Key words: Dynamic textures, Froth flotation, Expert systems, Modelling.

1. Introduction

The aim of mineral processing is to concentrate raw ore in preparation for the subsequent metal-extraction stage. Usually, the valuable minerals are first liberated from the ore matrix by comminution and size-separation processes and then separated from the gangue using processes capable of selecting the particles according to their physical or chemical properties. These include characteristics such as surface hydrophobicity, specific gravity, magnetic susceptibility and color (Wills, 1992).

Over the last two decades, image-processing sensors have emerged as an interesting measurement option in mineral processing, mainly due to their high sampling frequency and non-intrusive characteristics. The principal application fields of these sensors are the grinding and flotation stages.

In the grinding process, the most successful application of visual sensors is fragmentation measurement in traditional mining operations with large crushing and milling circuits that require efficiency in comminution to maximize profit on the final product. With the help of on-line image analysis systems, operators can actively monitor the various stages in the com-

minution circuit to ensure their operation is running within specification.

Regarding the flotation process, it is widely known that the color and morphology of the froth are closely related to mineral concentrations and process status, respectively. As a consequence, operators often make operating changes based on the visual appearance of the froth as well as past experience. Image processing sensors provide quantitative measures of some representative froth characteristics and also offer the option of making control decisions independently of the operator, resulting in a control law that depends exclusively on these measures. Most popular automatic control strategies that incorporate image processing sensors measures has been expert systems.

To incorporate visual information in model based control, a model is required that can relate visual features to manipulated and controlled variables. There are two possible approaches to this. The first is to obtain a model of features evolution while the second is to determine a model of visual process evolution and then extract the features of interest. The key question is then whether one of the approaches is better than the other. In this work we answer that question for the froth flotation process case. After reviewing the classical algorithms used in froth flotation image analysis, classical image processing-based

Email addresses: fenunez@ing.puc.cl (Felipe Núñez),
aciprian@ing.puc.cl (Aldo Cipriano).

predictive models are derived. We then introduce a new method for modeling visual information based on dynamic texture techniques and obtain the corresponding predictive models. This is followed by a comparison of the classical and dynamic texture models for three different froth types and two froth features. Finally, we propose a hybrid modeling scheme for flotation cell surface appearance based on froth-type recognition that would be highly useful in the future design of a hybrid multivariable predictive controller.

2. Visual information in froth flotation control

In most flotation plants it is customary for plant operators to monitor the surface froth visually and make adjustments to process parameters based on their interpretation of the surface appearance. To this end, operators categorize froths mentally into a number of distinct types and develop appropriate operating strategies for each one, looking for the ideal froth appearance. Figure 1 shows three different froth types that commonly appear in a froth flotation cell.

Visual information on surface froth from operating flotation plants has not been systematically catalogued. The interpretation of this information is based on intuitive experience and tends to vary from one operator to another. As a result, achieving optimal control with human operators is often not possible.

2.1. Froth features and industrial sensors

Operator control decisions are based on surface properties such as froth stability, froth speed, bubble size, bubble shape, bubble density and froth color (Hyötyniemi & Ylinen, 2000), (Holtham & Nguyen, 2002).

Since automatic control algorithms must include these surface features in control law determination, a quantitative measurement system is needed. Several instrumentation companies have developed specialized industrial sensors that apply image processing techniques to produce quantitative measures of froth features (e.g., FrothmasterTM, VisiofrothTM, JKProthcamTM). These measures can be classified as:

- Static Features
 - Bubble size
 - Bubble count
 - Bubble density
- Dynamic features
 - Froth speed
 - Froth stability
- Statistical features
 - Bubble size distribution
 - Bubble shape analysis
 - Froth color histogram

The output of the specialized sensors consists of signals in industrial format (e.g., 4 – 20 mA) for direct incorporation in

distributed control systems.

2.2. Vision based expert systems

A first approach for incorporating visual characteristics in control is an expert system. An expert system is a computational algorithm that simulates the control decisions of a “best operator”, using a knowledge base.

Using analyses of captured images and information provided by expert flotation operators, control engineers design a rule-structured knowledge base that determines the operating point of each flotation cell based on the static, dynamic and statistical froth feature signals outputted by the specialized sensor.

A color analysis is undertaken to obtain the processed ore’s coloration characteristics. A geometric analysis allows several types of froth to be defined such as fine, coarse, dense or others. A dynamic analysis characterizes the froth speed and stability, distinguishing between cases such as slow or fast froth, quiet or turbulent froth, and stable or unstable froth.

If the identified operating point indicates abnormal operation, the expert system searches in a table for the possible causes. These may include inappropriate reagent dosages, froth contamination by external agents (petroleum, oil, detergents, etc.) or inaccurate classification during the previous comminution process. The expert system then suggests what corrective actions should be taken by the plant operator, such as adjusting reagent dosage, opening the cell cap or reducing the water supply (Cipriano et al., 1997).

2.3. Vision based predictive control

Model predictive control schemes have been systematically developed for concentrate and tailings grades in froth flotation processes (Remes et al., 2007) using the aeration and reagent addition rates as manipulated variables. However, visual information has not been included in these schemes, mainly due to the absence of accurate predictive models.

The design of vision based predictive control loops involves the additional complication of generating models that can predict the temporal evolution of froth characteristics. In the present context there exist two options: obtain a model of froth features and use it for prediction, or obtain a model of the visual process described by the froth and then extract the features of interest from the predicted frames. In the following sections we develop a theoretical introduction and practical implementation of both methods.

3. Feature modeling for control purposes

Deriving models that describe the temporal evolution of froth features is crucial for advanced control. The design of a predictive control scheme requires models that are able to predict

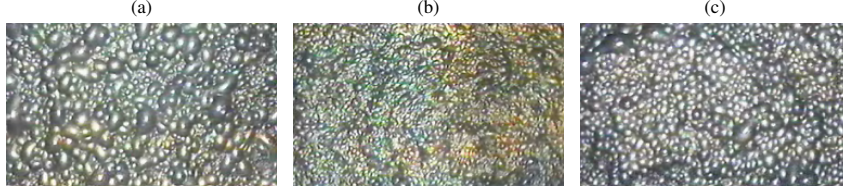


Fig. 1. Examples of froth types; (a): Froth type 1; (b): Froth type 2; (c): Froth type 3.

evolution over a finite horizon. In what follows, we describe several algorithms used to extract froth features from image sequences and the generation of auto-regressive models capable of predict features evolution.

3.1. Feature extraction algorithms

The most important step in froth image processing and analysis is to delineate every bubble. In most actual industry cases the froth images are difficult to segment, however. The number of bubbles can be as high as 2,000 and the bubbles often touch each other, with no void space between them. Moreover, the illumination on the bubble surface is uneven. The edges between bubbles are normally weak, whereas strong edges are found on the border of specular regions (referred to here as white spots) on top of almost every bubble. All of these characteristics complicate the development of a segmentation algorithm. Intensity similarity cannot be used to segment the bubbles and ordinary edge detection functions fail to segment froth images properly.

Morphological segmentation algorithms, particularly watersheds (Vincent & Soille, 1991), (Vincent, 1993), are the most widely used technique for froth image segmentation (Sadr-Kazemi & Cilliers, 1997), (Cipriano et al., 1998), (Bonifazi et al., 1999), (Grau & Heiskanen, 2002). The notion of a watershed is best understood by visualizing a grayscale image as a three dimensional landscape or topographic relief map where pixel intensity represents the landscape altitude at a given Cartesian position. The valleys and hills making up the landscape correspond to the darker and lighter portions of the image. If a drop of water were to fall at any position on the landscape it would flow down and come to rest in the deepest part of the closest valley. This deepest part represents a local minimum in the image, and the entire surrounding area is its catchment basin. The watersheds are thus the borders that define and separate catchment basins belonging to different local minima.

The purpose of a grayscale watershed is to extract watershed lines, that is, contiguous lines of high-intensity pixels separating local intensity minima in the image.

Once the segmentation process is complete we can extract the image features. The per-bubble segmentation yields a tessellation of the surface froth image in which all the pixels

belonging to a specific bubble are tagged as such in an array of two dimensions corresponding to those of the image. We use this array to obtain the froth features as follows.

Bubble area is calculated by counting the exact number of pixels contained in the array's tessellated two-dimensional projection. This is done for each and every bubble in the image.

Bubble count refers to the total number of bubbles present in a specific froth image and is obtained by counting the white spots or markers. The resulting figure serves as an indicator of froth stability when continuous video is analyzed. A bubble count that exhibits dramatic variations in sequences of consecutive images indicates that the froth is unstable.

The horizontal and vertical differential motion pair is a two-element vector describing how many pixels the markers have moved in the time interval between the two input images. This vector is a measure of froth speed.

Color information corresponds to the average of the red, green and blue components in each frame.

3.2. Feature predictive models

For predictive modeling purposes we acquired image sequences of the flotation cell surface using a FireWire industrial camera at a rate of 30 frames per second and a resolution of 320×240 pixels. These sequences represent about one hour of operation for the different froth types. Sequences were acquired during periods of constant values for manipulated variables.

To obtain predictive models of froth features, we proceeded as follows. First, we obtained the corresponding feature value at each frame for a sequence of length $\tau = 60$. Then, by applying least squares we derived autoregressive models of the following form:

$$y_k = \theta^T \begin{bmatrix} r_k \\ 1 \end{bmatrix} \quad (1)$$

where r_k is the regression vector of order n_1 and θ is the constant parameter vector.

The prediction capabilities of the models so obtained were tested by comparing ten predicted values with real values. Figure 2 shows the comparisons of froth color and froth speed for

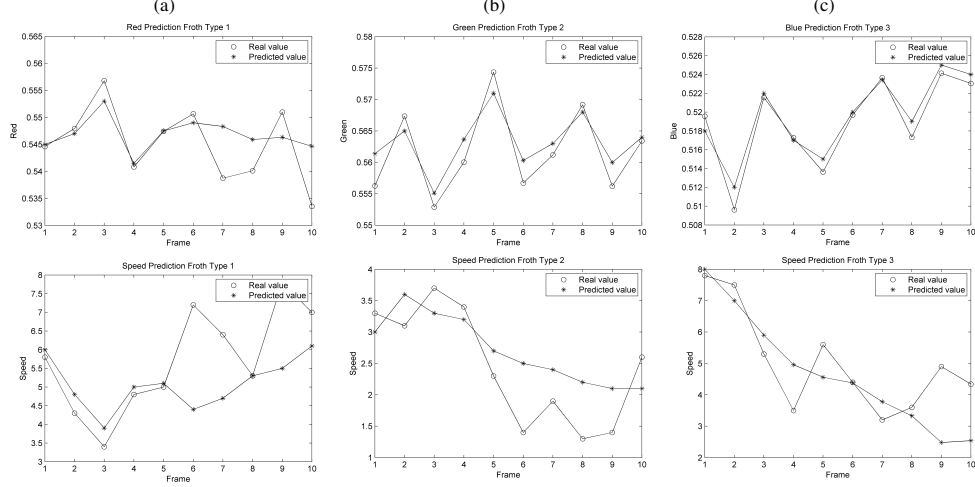


Fig. 2. Results of the prediction test for froth color and froth speed using feature based models; (a): Red prediction (on top), speed prediction (bottom) for froth type 1; (b): Green prediction (on top), speed prediction (bottom) for froth type 2; (c): Blue prediction (on top), speed prediction (bottom) for froth type 3

each froth type and a model order of $n_1 = 3$ (for both features).

These results show that the models obtained are capable of predicting future values of the studied features. Some deterioration of the prediction does occur, however. This is especially true for froth speed, due mainly to the absence of other froth feature effects in the autoregressive models describing feature evolution.

4. Visual processes modeling for control purposes

In this section we introduce a new method for modeling visual processes known as dynamic texture analysis. The aim is to obtain a model that captures the spatio-temporal characteristics of the visual process under analysis.

4.1. Dynamic textures

4.1.1. Modeling

Dynamic textures are sequences of images of moving scenes that exhibit certain stationary properties in time (Doretto et al., 2003). A linear dynamic texture is associated with a discrete-time auto-regressive moving average process (ARMA) with unknown input distribution, which can be formulated in state space form as follows:

$$x(t+1) = Ax(t) + Bv(t) \quad (2)$$

$$y(t) = \phi(x(t)) + w(t) \quad (3)$$

where $v(t)$ and $w(t)$ are random variables and $\phi(\cdot)$ is a set of spatial filters (Doretto et al., 2003).

There are several criteria for choosing a suitable class of filters, ranging from physical motivations to computational efficiency. In the simplest case, $\phi(\cdot)$ is taken to be the identity, meaning that we are observing the dynamics of individual pixels. The choice of filters is seen as a dimensionality reduction step, and we attempt to achieve a decomposition of the image in a simple linear form. The associated ARMA model then yields (Doretto et al., 2003)

$$x(t+1) = Ax(t) + Bv(t) \quad (4)$$

$$y(t) = Cx(t) + w(t) \quad (5)$$

Let us now consider the dynamic texture as a second-order stationary process with arbitrary covariance. It is well known that such a process can be modeled as the output of a linear dynamic system driven by white, zero-mean Gaussian noise. We can then approximate the visual process with the following ARMA model:

$$x(t+1) = Ax(t) + v(t) \quad (6)$$

$$y(t) = Cx(t) + w(t) \quad (7)$$

where $v(t) \sim N(0, Q)$ and $w(t) \sim N(0, R)$.

4.1.2. Identification

The associated identification problem consists in estimating the model parameters A , C , Q , and R from the measurements obtained at time t (Doretto et al., 2003).

Let $Y^\tau = [y(1), \dots, y(\tau)]$ and similarly for X^τ and W^τ . Thus,

$$Y^\tau = CX^\tau + W^\tau \quad (8)$$

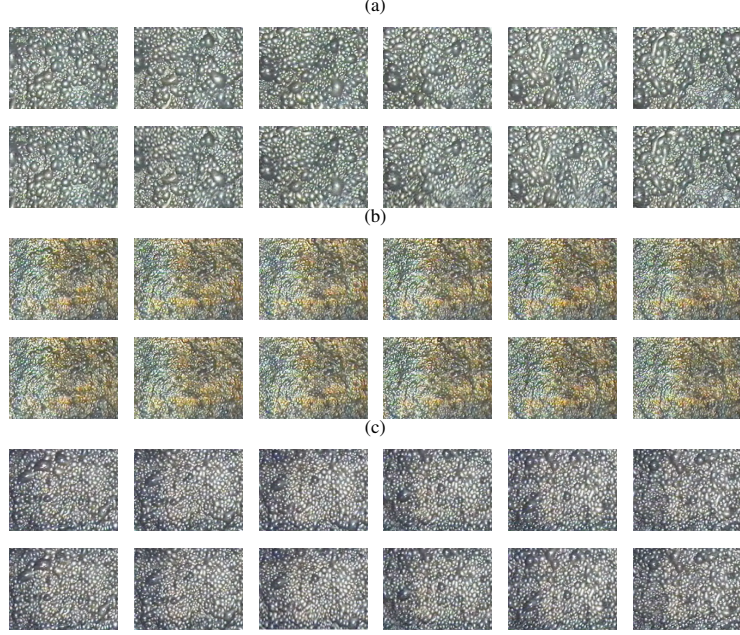


Fig. 3. Selected frames from the real sequence (on top) and the corresponding frames obtained from the ARMA model (on bottom) for $t = 10, 20, 30, 40, 50, 60$; (a): Froth type 1; (b): Froth type 2; (c): Froth type 3.

Now let $Y^\tau = \Lambda \Sigma Z^T$ be the singular value decomposition (SVD), and consider the problem of finding the best estimate of C in the Frobenius sense. It follows immediately from the fixed-rank approximation property of the SVD that a unique solution is given by (Doretto et al., 2003), (Vidal et al., 2007):

$$\hat{C} = \Lambda, \quad \hat{X} = \Sigma Z^T \quad (9)$$

Given $\hat{X}$, $\hat{A}$ can be determined uniquely, again in the Frobenius sense. This is trivially done in closed form using the estimate of $\hat{X}$ as

$$\hat{A} = \Sigma Z^T D_1 Z (Z^T D_2 Z)^{-1} \Sigma^{-1} \quad (10)$$

where

$$D_1 = \begin{pmatrix} 0 & 0 \\ I_{t-1} & 0 \end{pmatrix} \quad (11)$$

$$D_2 = \begin{pmatrix} I_{t-1} & 0 \\ 0 & 0 \end{pmatrix} \quad (12)$$

Finally, the input and process noise covariance matrices can be estimated from

$$\hat{Q} = \frac{1}{\tau-1} \sum_{t=1}^{\tau-1} \hat{v}(t) \hat{v}(t)^T \quad (13)$$

$$\hat{R} = \frac{1}{\tau-1} \sum_{t=1}^{\tau-1} \hat{w}(t) \hat{w}(t)^T \quad (14)$$

where $\hat{v}(t)$ and $\hat{w}(t)$ are obtained as $\hat{x}(t+1) - \hat{A}\hat{x}(t)$ and $y(t) - \hat{C}\hat{x}(t)$, respectively.

It should be noted that $\hat{C}$, $\hat{X}$, $\hat{A}$, $\hat{Q}$ and $\hat{R}$ depend explicitly on the sequence length τ .

4.2. Froth predictive models

We obtained the ARMA model associated with each froth type using the dynamic texture techniques presented above. For the training process we used the same sequences of $\tau = 60$ frames employed in deriving the feature predictive models in 3.2, with $n = 40$ principal components for the texture synthesis.

Figure 3 shows selected frames of the real sequence and the corresponding frames obtained from the model synthesis for each froth type. Clearly, the models obtained are able to capture spatio-temporal froth characteristics.

Figure 4 shows the results for our two-step prediction test in which the obtained models are used to predict the next two frames in the sequence. The results demonstrate that they display good prediction capabilities, maintaining the prediction intensity error below 3%.

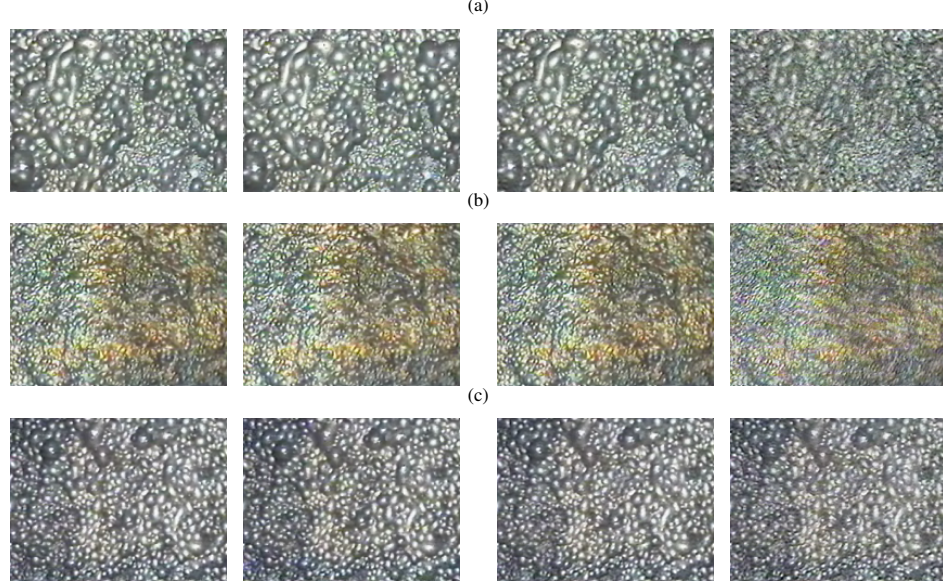


Fig. 4. Results of the prediction test for frames $t + 1$ and $t + 2$. Left side: real frames; Right side: predicted frames; (a): Froth type 1; (b): Froth type 2; (c): Froth type 3.

In order to compare the froth feature values obtained from predicted frames with those generated by the autoregressive feature model (1), we first generate a 10-frame prediction and then the corresponding feature values from the predicted frames for froth color and froth speed. The comparisons, shown in Figure 5, clearly indicate that the froth models are able to represent the future evolution of the froth features covered in the study. The results from the predicted frames are better than those produced by model (1), especially for color prediction. This may be explained by the fact that the dynamic texture-based models consider the interactions between all froth features in the next frame generation.

4.3. Classification of visual information

Once the different froth types have an associated ARMA model, we can characterize them using the ARMA model matrices A , C .

Below we describe an approach to classification and recognition of dynamic textures based on distances defined in ARMA model space, which has been used for nearest-neighbor classification or *k-means* clustering (Vidal et al., 2007).

To compute distances between dynamic models the so-called Binet-Cauchy kernels are used (Vishwanathan et al.,

2006). These kernels are obtained by computing the trace of a compound matrix of order q built from the output trajectories.

The Binet-Cauchy kernel of order q is defined as

$$k_q(M_1, M_2) = \text{tr} E_q(Y_1 Y_2^T) \quad (15)$$

where $E_q(x)$ is the compound matrix of order q of x whose entries are given by the determinants of all the minors of x of size q . Y_1 and Y_2 are the output vectors obtained from the models M_1 and M_2 . As shown in (Vishwanathan et al., 2006), when $q = 1$ this kernel can be computed explicitly in terms of the ARMA models as

$$k_1(M_1, M_2) = x_1^T(0) S x_2^T(0) \quad (16)$$

where S is the solution to the Sylvester equation

$$A_1^T S A_2 - S + A_1^T C_1^T C_2 A_2 = 0 \quad (17)$$

From the Binet-Cauchy kernel we may define a distance as (Vidal et al., 2007)

$$d_1(M_1, M_2)^2 = k_1(M_1, M_1) - 2k_1(M_1, M_2) + k_1(M_2, M_2) \quad (18)$$

5. Froth flotation hybrid modeling

The existence of several froth types exhibiting different dynamics provides an opportunity to apply hybrid modeling techniques to the froth flotation process. Moreover, switched affine

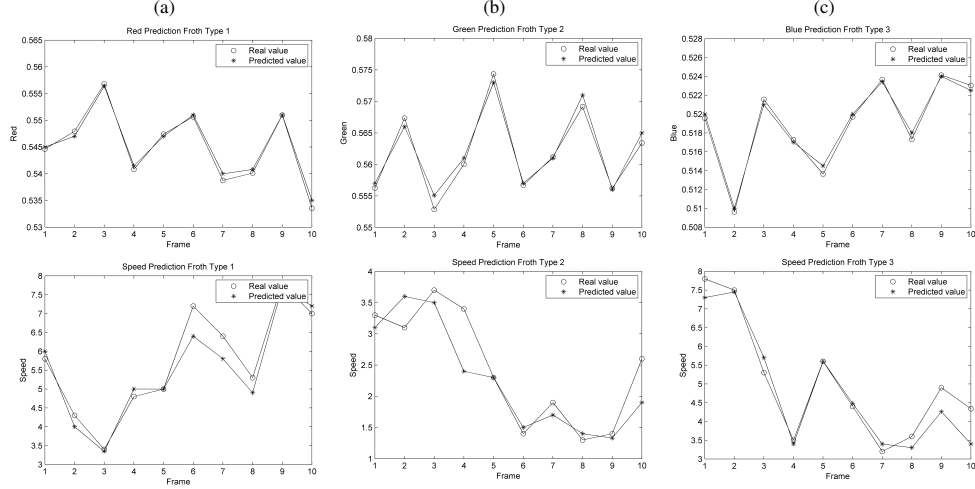


Fig. 5. Results of the prediction test for froth color and froth speed using dynamic textures based models; (a): Red prediction (on top), speed prediction (bottom) for froth type 1; (b): Green prediction (on top), speed prediction (bottom) for froth type 2; (c): Blue prediction (on top), speed prediction (bottom) for froth type 3

models would seem to be particularly suitable for the task since they explicitly consider different dynamics in each operating state (Bemporad & Morari, 1999), (Paoletti et al., 2007).

5.1. Hybrid systems

Hybrid systems are heterogeneous dynamical systems whose behavior is determined by the interaction between continuous and discrete dynamics (Bemporad & Morari, 1999). The continuous dynamics are described by variables taking values from a continuous set while the discrete dynamics are described by variables taking values from a discrete set, typically finite. The continuous or discrete-valued variables may depend on independent variables such as time, which in turn may be continuous or discrete (Bemporad & Morari, 1999).

Hybrid systems can be modeled using specific techniques that produce a hybrid model which, depending on its structure, can be a *mixed logical dynamics* (MLD) model, a *piecewise affine* (PWA) model, a *linear complementarity* (LC) model, etc. We will focus on the class of models known as switched affine (Paoletti et al., 2007).

Members of this class are defined as collections of linear/affine models connected by switches that are indexed by a discrete-valued additional variable called the discrete state. These models explicitly consider different dynamics for each discrete state. Those for which the discrete state is determined by a polyhedral partition of the state-input domain are called piecewise affine models. The latter are equivalent to several classes of hybrid models and can therefore be used to describe systems that exhibit hybrid structure.

A general discrete-time switched affine model in input-output form is defined by introducing the regression vector (Paoletti et al., 2007)

$$r_k = [y_{k-1}^T \cdots y_{k-n_a}^T u_k^T u_{k-1}^T \cdots u_{k-n_b}^T]^T \quad (19)$$

where n_a and n_b are the model orders. The output y_k is then expressed as a piecewise affine function of r_k :

$$y_k = \theta_{\sigma(k)}^T \begin{bmatrix} r_k \\ 1 \end{bmatrix} + e_k \quad (20)$$

where $\sigma(k) \in \{1, \dots, s\}$ is the discrete state, s is the number of sub-models, $\theta_i, i = 1, \dots, s$ are the matrices of parameters defining each sub-model and e_k is a noise/error term.

5.2. Surface appearance hybrid modeling

5.2.1. Operational States

For the hybrid model of surface appearance we define three operating states corresponding to the three froth types shown in Figure 1. These states have the following characteristics:

- Operating State 1:
 - Large bubbles
 - Light gray froth
 - High density
 - Stable froth
- Operating State 2:
 - Small bubbles
 - Yellow froth
 - Collapsed areas

- Unstable froth
- Operating State 3:
 - Medium bubbles
 - Dark froth
 - High density
 - Stable froth

The three operating states shown in Figure 1 represent the discrete states of our hybrid scheme, and our aim is to identify efficiently the current operating state based on froth texture characteristics.

5.2.2. Automatic Operational States Detection

For automatic detection purposes we designed an identification algorithm based on Binet-Cauchy distance for recognizing the current operating state using nearest neighbor classification. Table 1 shows the data obtained from detection tests.

These results demonstrate that our detection algorithm provides highly effective automatic detection for each operating state in a short processing time, making it very useful for online froth-type classification applications.

Real/Detected	State 1	State 2	State 3
State 1	37	0	3
State 2	0	40	0
State 3	2	0	38

6. Concluding remarks and future research

Our results show that the dynamic texture-based models we obtained were able to capture froth dynamics, with the option of characterizing the different operating states via an associated ARMA model. Feature prediction tests demonstrated that these models offer better prediction than image processing-based models, especially for color.

Hybrid modeling techniques afford the option of including expert knowledge in the form of operating states, thus allowing model-based control techniques to be applied to the froth flotation process.

Automatic detection tests confirmed that our classification algorithm is capable of efficiently identifying the different operating states (froth types) based on distances between the associated ARMA models.

Future research should incorporate additional operational states (froth types) to study the robustness of the detection algorithm, and should also address the effects of process variables on froth behavior so that a hybrid predictive control scheme can be designed based on texture classification.

References

- Bemporad, A. & Morari, M., 1999, Control of systems integrating logic, dynamics and constraints, *Automatica*, Vol.35, No.3, 407-427.
- Bonifazi, G., Serranti, S., Volpe, F., & Zuco, R., 1999, A combined morphological and color based approach to characterize flotation froth bubbles, *Proceedings of the Second International Conference on intelligent Processing and Manufacturing of Materials*.
- Cipriano, A., Sepulveda, C., & Guarini, M., 1997, Expert system for supervision of mineral flotation cells using artificial vision, *Proceedings of the IEEE International Symposium on Industrial Electronics*, 7-11.
- Cipriano, A., Guarini, M., Vidal, R., Soto, A., & Sepulveda, C., 1998, A real time visual sensor for supervision of flotation cells, *Minerals Engineering*, Vol.11, No.6, 489-499.
- Doretto, G., Chiuso, A., Wu, W., & Soatto, S., 2003, Dynamic textures, *International Journal of Computer Vision*, Vol.51, No.2, 91-109.
- Grau, R.A., & Heiskanen, K., 2002, Visual technique for measuring bubble size in flotation machines, *Minerals engineering*, Vol.15, No.6, 507-513.
- Holtham, P.N., & Nguyen, K.K., 2002, On-line analysis of froth surface in coal and mineral flotation using JKFroth-Cam, *International Journal on Mineral Processing*, Vol 64, No.2, 163-180.
- Hyötyniemi, H., & Ylinen, R., 2000, Modeling of visual flotation froth data, *Control Engineering Practice*, Vol 8, No.3, 313-318.
- Paoletti, S., Julosky, A., Ferrari-Trecate, G., & Vidal, R., 2007, Identification of Hybrid Systems: A tutorial, *European Journal of Control*, Vol.13, No.2-3, 242-260.
- Remes, A., Saloheimo, K. & Jämsä-Jounela, S.L., 2007, Effect of speed and accuracy of on-line elemental analysis on flotation control performance, *Minerals Engineering*, Vol.20, No.11, 1055-1066.
- Sadr-Kazemi, N., & Cilliers, J.J., 1997, An image processing algorithm for measurement of flotation froth bubble size and shape distributions, *Minerals Engineering*, Vol.10, No.10, 1075-1083.
- Vidal, R., Soatto, S., & Chiuso, A., 2007, Applications of hybrid system identification in computer vision, *Proceedings of the European Control Conference*.
- Vincent, L., & Soille, P., 1991, Watershed in digital spaces: An efficient algorithm based on immersion simulations, *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol.13, No.6, 583-598.
- Vincent, L., 1993, Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms, *IEEE Trans on Image Processing*, Vol.2, No.2, 176-201.
- Vishwanathan, S., Smola, S. & Vidal, R., 2006, Binet-Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes, *International Journal of Computer Vision*, Vol.73, No.1, 95-119.

Wills, B.A., 1992, Mineral Processing Technology, Pergamon Press, New York.