

# SOLUCIONES ALTERNATIVAS AL CONTROL PREDICTIVO BASADO EN MODELO DE VOLTERRA

J.K. Gruber

Unidad de Sistemas Eléctricos, Instituto IMDEA Energía, Móstoles (Madrid)  
jorn.gruber@imdea.org

I. Peñarrocha

Departament d'Enginyeria de Sistemes Industrials i Disseny, Universitat Jaume I de Castelló  
ipenarro@uji.es

## Resumen

*El concepto básico de control predictivo basado en modelos (MPC) ofrece la posibilidad de usar modelos no lineales para predecir la evolución futura del sistema. La consideración de modelos no lineales en una función de coste cuadrática resulta en un problema de optimización posiblemente no convexo con varios mínimos. En este trabajo se presentan diferentes alternativas para la solución del problema del control predictivo no lineal basado en modelos de Volterra. Las opciones planteadas consisten en la utilización de un solver global no lineal incluido en Matlab, la utilización de la técnica de los momentos y, para el caso de modelos de Volterra de grado dos, la transformación del problema en uno de programación semidefinida. Se muestra mediante un ejemplo la bondad de las diferentes propuestas y la relación del coste computacional de cada opción en función de los horizontes de predicción y control.*

**Palabras clave:** Control predictivo, optimización sobre polinomios, momentos, programación semidefinida.

## 1. Introducción

El control predictivo basado en modelo (MPC, del inglés *Model Predictive Control*) ha experimentado un gran desarrollo en las últimas décadas y se ha convertido en una de las técnicas de control avanzado más populares en la comunidad científica y la industria. Varios factores contribuyen al éxito de MPC, entre los más importantes, la formulación intuitiva del problema de control, la posibilidad de controlar una gran variedad de procesos, la consideración de restricciones y, al menos en el caso de MPC lineal, la fácil implementación de la ley de control. La posibilidad de utilizar modelos que se pueden obtener fácilmente del proceso

considerado, p.ej. modelos de respuesta ante impulso o escalón, representa otra ventaja de MPC, especialmente en aplicaciones industriales.

Prácticamente todos los procesos dinámicos de importancia industrial exhiben cierto comportamiento no lineal. Sin embargo, la gran mayoría de técnicas de MPC han sido desarrolladas para sistemas lineales. La aplicación de MPC lineal a procesos con dinámica fuertemente no lineal puede resultar en un rendimiento de control deficiente, debido a la discrepancia entre el sistema y el modelo. Con el fin de obtener un mejor rendimiento de control, se puede considerar el uso de control predictivo no lineal (NMPC, del inglés *Nonlinear Model Predictive Control*). Desafortunadamente, el NMPC tiene algunos inconvenientes para su uso en la industria como la dificultad de obtener modelos apropiados del proceso considerado y la solución necesaria de un problema de optimización posiblemente no convexo. A pesar de que hay evidencia de que NMPC habitualmente mejora el rendimiento de control, el uso de este tipo de estrategia en la industria es todavía relativamente escaso.

Para la aproximación de procesos no lineales se usan modelos de Volterra en muchos campos, entre otros en aplicaciones biomédicas, acústica, electrónica y control de procesos, pero especialmente en el procesamiento de señales para el diseño de filtros no lineales. Los modelos de Volterra, a pesar de representar modelos no lineales, son lineales en sus parámetros. Debido a ello se pueden estimar los parámetros de los modelos de Volterra mediante técnicas de identificación habitualmente usadas para modelos lineales, p.ej. métodos de mínimos cuadrados. Además existe la posibilidad de estimar los parámetros de los modelos de Volterra a partir de datos experimentales de tipo entrada-salida. Por tanto, no hace falta un conocimiento profundo del proceso que se aproxima. Finalmente, los modelos de Volterra pueden ser

usados para modelar una gran variedad de dinámicas no lineales diferentes, como sistemas de fase no mínima.

Las razones mencionadas convierten los modelos de Volterra en buenos candidatos para el desarrollo de nuevas técnicas de NMPC. Probablemente el mayor inconveniente es la minimización de la función de coste frecuentemente no convexa. No obstante, en los últimos años se han propuesto varios métodos para determinar la secuencia de control mediante NMPC basado en modelos de Volterra. En [1, 2] la optimización se realiza sin considerar restricciones mediante un algoritmo iterativo basado en la separación de los términos lineales y no lineales de un modelo de Volterra de segundo orden. Las dos estrategias de NMPC subóptimas basadas en modelos de Volterra de segundo orden [3] consideran una única variable de decisión y convierten el problema de optimización posiblemente no convexo en un polinomio de grado cuatro fácilmente de resolver. En [4, 5] se ha desarrollado un enfoque de convexificación para estrategias de NMPC basado en modelos de Volterra de segundo orden que aproxima la función de coste original mediante una serie de funciones cuadráticas convexas.

La utilización de modelos de Volterra hace del problema de optimización no lineal un problema de optimización sobre polinomios cuya estructura puede ser explotada [6]. La técnica de los momentos [7, 8] reduce el problema de optimización sobre polinomios en una secuencia de problemas de desigualdades lineales matriciales (LMI, del inglés *Linear Matrix Inequalities*) tratable computacionalmente. La aplicación de esta técnica al problema en cuestión es directa gracias a herramientas como Gloptipoly o Yalmip [9, 10]. Sin embargo, tiene un coste computacional elevado debido a la naturaleza multivariable del polinomio a optimizar, puesto que el número de variables de decisión crece con el horizonte de control a considerar. Dada la estructura de la función objetivo, también es posible transformar el problema en uno de desigualdades bilineales matriciales (BMI) que puede resolverse a su vez por dos vías: mediante solvers comerciales dedicados para este problema como PENBMI [11, 12], y de forma iterativa mediante aproximaciones convexas [13, 14] con un coste computacional menor que el método de los momentos. La ventaja de la vía iterativa es que sólo se requiere de un solver de programación semidefinida, pero, sin embargo, requiere la implementación de un algoritmo iterativo por parte del usuario, a diferencia de la escritura directa que permiten las otras dos técnicas (momentos y BMI). En este trabajo se muestra como resolver el problema NMPC para modelos Volterra mediante la técnica de los

momentos, así como la transformación del mismo en un problema BMI y la propuesta de solución iterativa del mismo.

El resto del trabajo se organiza de la siguiente manera: la Sección 2 describe la idea de control predictivo basado en modelos de Volterra y los posibles problemas. En la Sección 3 se proponen dos soluciones para el problema de optimización: una basada en la técnica de los momentos, y otra basada en la resolución de un problema BMI. La Sección 4 presenta la aplicación de las soluciones propuestas a un ejemplo simple. Finalmente se exponen las conclusiones principales en la Sección 5.

## 2. Planteamiento del problema

Los modelos de Volterra [15] representan la extensión simple y lógica de modelos de respuesta ante impulso y vienen definidos por:

$$y(k) = h_0 + \dots \sum_{n=1}^{\infty} \sum_{i_1=0}^{\infty} \dots \sum_{i_n=0}^{\infty} h_n(i_1, \dots, i_n) \dots u(k-i_1) \dots u(k-i_n) \quad (1)$$

siendo  $y(k)$  y  $u(k)$  la salida y entrada del modelo, respectivamente. El parámetro  $h_0$  es la parte fija del modelo,  $h_1$  denota los parámetros del término lineal y  $h_n \forall n \geq 2$  representa los parámetros de los términos no lineales. En [16] se ha demostrado que se puede aproximar cualquier sistema con memoria acotada mediante un modelo de Volterra de orden suficientemente alto. Este tipo de modelo exhibe normalmente un buen comportamiento y se puede explotar su estructura en el diseño de controladores, especialmente los modelos de segundo orden. Para un sistema con memoria acotada, el modelo de Volterra de segundo orden viene definido por [17]:

$$y(k) = h_0 + \sum_{i=1}^{N_1} h_1(i)u(k-i) + \dots \sum_{i=1}^{N_2} \sum_{j=i}^{N_2} h_2(i, j)u(k-i)u(k-j) \quad (2)$$

donde  $N_1$  and  $N_2$  denotan el orden de truncamiento de los términos lineales y no lineales del modelo. Cabe destacar que la salida  $y(k)$  (2) en el muestreo  $k$  depende exclusivamente de las entradas pasadas  $u(k-i)$  del modelo. Con la idea mantener la notación simple se usará  $N_1 = N_2 = N_t$  en las siguientes secciones.

Con un horizonte de predicción  $N$ , un horizonte de control  $N_u$  y un orden de truncamiento  $N_t$ , un modelo de predicción basado en un modelo de

Volterra de segundo orden tiene la forma [17]:

$$\mathbf{y} = G\mathbf{u} + \mathbf{f}(\mathbf{u}) + \mathbf{c} \quad (3)$$

$$\mathbf{c} = H\mathbf{u}_p + \mathbf{g} + \mathbf{d} \quad (4)$$

siendo  $\mathbf{y} \in \mathbb{R}^N$  el vector de salida del sistema a lo largo del horizonte de predicción,  $\mathbf{u} \in \mathbb{R}^{N_u}$  es el vector de entradas futuras y  $\mathbf{u}_p \in \mathbb{R}^{N_t}$  es el vector de entradas pasadas. Los términos  $G\mathbf{u}$  con  $G \in \mathbb{R}^{N \times N_u}$  y  $H\mathbf{u}_p$  con  $H \in \mathbb{R}^{N \times N_t}$  representan las partes lineales basadas en entradas futuras y pasadas, respectivamente. El término  $\mathbf{f}(\mathbf{u}) \in \mathbb{R}^N$  denota la parte no lineal del modelo que depende de las entradas futuras. La parte no lineal que depende exclusivamente de las entradas pasadas se considera en el vector  $\mathbf{g} \in \mathbb{R}^N$ . El vector  $\mathbf{d} \in \mathbb{R}^N$  con  $\mathbf{d} = [d(k), \dots, d(k)]^T$  contiene el error de estimación actual (la diferencia entre la salida del sistema y la predicción). En MPC se determina las entradas futuras minimizando una función objetivo basada en el modelo de predicción del sistema considerado. Frecuentemente se usa la siguiente función cuadrática[18]:

$$J(\mathbf{u}) = (\mathbf{y} - \mathbf{r})^T(\mathbf{y} - \mathbf{r}) + \lambda \Delta \mathbf{u}^T \Delta \mathbf{u} \quad (5)$$

donde el escalar  $J(\mathbf{u})$  es el coste,  $\mathbf{r} \in \mathbb{R}^N$  denota la referencia a lo largo del horizonte de predicción y  $\Delta \mathbf{u} \in \mathbb{R}^{N_u}$  denota el vector de incrementos en la acción de control. El parámetro  $\lambda$  representa la penalización de la acción de control y permite ajustar el MPC hacia un control más agresivo ( $\lambda$  bajo) o más suave ( $\lambda$  alto). La secuencia de control óptima se calcula usualmente minimizando la función de coste (5):

$$\mathbf{u}^* = \arg \min_{\mathbf{u}} J(\mathbf{u}) \quad (6)$$

$$\text{s.t. } \mathbf{q}(\mathbf{u}) \leq \mathbf{b}_c$$

considerando las restricciones  $\mathbf{q}(\mathbf{u}) \leq \mathbf{b}_c$ .

Usando el modelo de predicción no lineal (3)-(4) convierte la función de coste (5) en un polinomio de orden cuatro. El problema de optimización (6) no necesariamente convexo da lugar a varios mínimos. El objetivo del presente trabajo es el desarrollo de métodos para resolver el problema de optimización y determinar la secuencia de control óptima de forma eficiente y rápida.

### 3. Soluciones propuestas

A continuación se exponen las diferentes propuestas para resolver el problema de optimización en cada iteración:

1. La utilización de un solver genérico incluido en Matlab mediante la utilización de la función `fmincon`. Este solver suele ser bastante

robusto y consigue resolver los problemas polinómicos planteados.

A continuación se propone la utilización de la técnica de los momentos. Para ello se dispone de dos herramientas:

2. El uso del software gratuito Gloptipoly [9], que tiene una sintaxis propia y corre sobre Matlab.
3. La utilización del método de Lasserre [7] y explotar la estructura dispersa de las matrices que definen el problema. Para ello se puede utilizar el interfaz Yalmip junto a la función `solvemoment`.
4. Para el caso de modelos de Volterra de grado 2 (sólo hay productos de dos acciones de control en el modelo como en (2)) se propone la transformación del problema en un problema de desigualdades de matriz bilineal (BMI). El problema (6) se puede transformar en primer lugar en el problema:

$$\mathbf{u}^* = \arg \min_{\mathbf{u}} t \quad (7)$$

$$\text{s.t. } \mathbf{q}(\mathbf{u}) \leq \mathbf{b}_c$$

$$J(\mathbf{u}) \leq t$$

La última desigualdad se puede transformar en la desigualdad matricial:

$$\begin{bmatrix} t & (\mathbf{y} - \mathbf{r})^T & \Delta \mathbf{u}^T \\ \star & I & 0 \\ \star & \star & \lambda^{-1}I \end{bmatrix} \succeq 0 \quad (8)$$

mediante la aplicación de los complementos de Schur. El término  $\mathbf{y} - \mathbf{r}$  es cuadrático en las variables de decisión contenidas en el vector de acciones de control, con lo que se trata de un problema de BMI resoluble mediante el solver PENBMI que puede ser llamado desde Yalmip [11].

5. En último lugar, se propone solucionar el problema BMI anterior mediante una secuencia de problemas convexos tras linealizar los elementos cuadráticos que aparecen en el término  $\mathbf{y} - \mathbf{r}$  con la técnica expuesta en [14] (se omiten los detalles por brevedad). Este problema se puede escribir de forma sencilla con ayuda de Yalmip y se resuelve con un solver estándar para problemas de programación semidefinida (como Sedumi).

### 4. Ejemplo

Finalmente, se han implementado los métodos para resolver los problemas polinómicos de control predictivo basado en modelos de Volterra de segundo orden. En simulaciones se han aplicado las

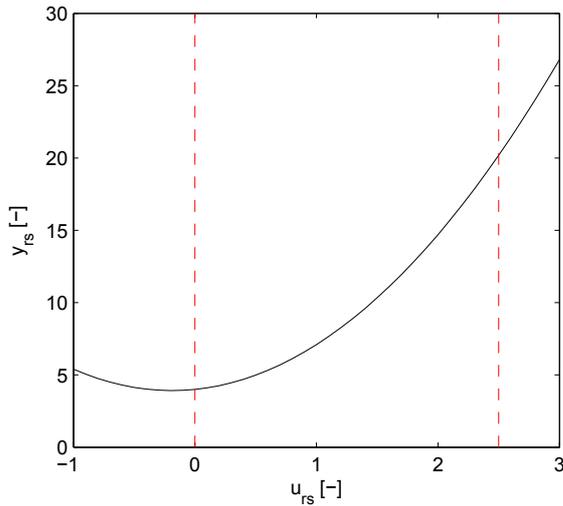


Figura 1: Curva estacionaria del sistema no lineal (salida del modelo para entrada constante) y restricciones en la entrada consideradas (líneas discontinuas).

diferentes propuestas a un sistema no lineal con el objetivo de comparar el rendimiento.

El sistema no lineal considerado puede ser expresado como un modelo de Volterra de segundo orden (2) con memoria acotada. Los parámetros del sistema son los siguientes:

$$\begin{aligned} h_0 &= 4 \\ h_1 &= [0.5 \ 0.25 \ 0.1] \\ h_2 &= \begin{bmatrix} 1 & 0.5 & 0.2 \\ 0 & 0.3 & 0.15 \\ 0 & 0 & 0.1 \end{bmatrix} \end{aligned} \quad (9)$$

En el NMPC se usa el modelo de Volterra del sistema no lineal (9) en combinación con los métodos de optimización propuestos para determinar la secuencia de control. Se supone que no hay error de modelado, es decir, el modelo de control usado en la optimización y el sistema controlado son idénticos. La relación entre entrada y salida del sistema en régimen estacionario y las restricciones consideradas en la optimización se pueden ver en la Fig. 1.

En las simulaciones con el NMPC usando los diferentes métodos para resolver el problema de optimización se ha limitado la acción de control al intervalo  $u \in [0, 2.5]$ . Con un orden de truncamiento de  $N_t = 3$  del modelo de Volterra se ha fijado los horizontes de predicción y control de los NMPC en  $N_p = 3$  y  $N_u = 3$ , respectivamente. Los incrementos en la acción de control se han penalizado en la función de coste con  $\lambda = 3$ . En las figuras 2 y 3 se muestra la evolución temporal de la salida y las acciones de control, así como el tiempo

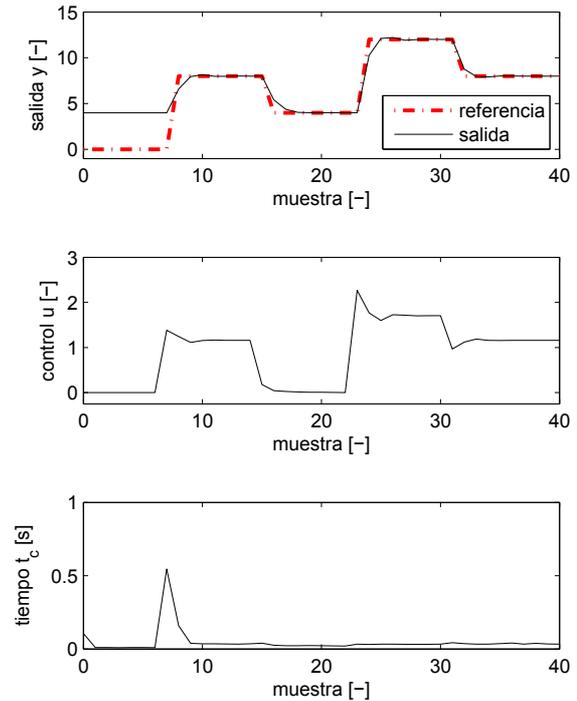


Figura 2: Resultados obtenidos con el NMPC usando la función `fmincon` de Matlab. Arriba: salida del sistema  $y$  y referencia  $r$ , centro: acción de control  $u$ , abajo: tiempo de optimización  $t_c$ .

de cómputo necesario en cada iteración para la primera solución propuesta (función `fmincon` integrada en Matlab) y la quinta (aproximación convexa para resolver el problema BMI), que son las de menor y mayor coste computacional, respectivamente, para el horizonte  $N_p = N_c = 3$  indicado.

En la figura 4 se muestran los tiempos de cómputo medios por periodo para diferentes horizontes  $N_p = N_c$ , dando todas las técnicas un índice de coste y una respuesta temporal prácticamente idénticos. Se puede observar que el método `gloptipoly` es el primero en perder prestaciones cuando el horizonte aumenta ligeramente, seguido del método `solvemoment`. El método del uso del solver BMI muestra un menor empeoramiento de prestaciones con el aumento del horizonte que las opciones anteriores. Finalmente, la aproximación convexa muestra un comportamiento más robusto frente al aumento del horizonte, mejorando las prestaciones respecto de `fmincon` para horizontes mayores de 30.

## 5. Conclusiones

En este trabajo se han planteado varias alternativas para la solución del problema del control predictivo no lineal para sistemas modelados mediante series de Volterra. Para ello se han planteado

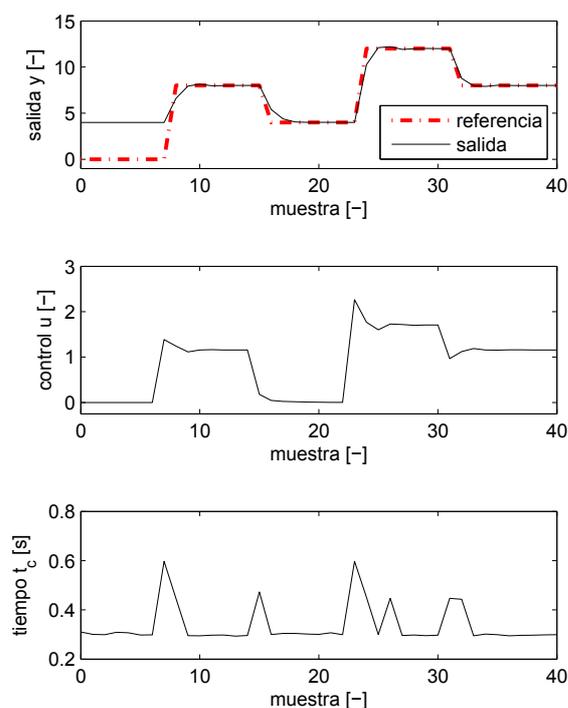


Figura 3: Resultados obtenidos con el NMPC usando la aproximación convexa para resolver el problema BMI. Arriba: salida del sistema  $y$  y referencia  $r$ , centro: acción de control  $u$ , abajo: tiempo de optimización  $t_c$ .

diferentes opciones para resolver la optimización sobre polinomios que aparece en cada iteración, tratando de explotar la naturaleza polinómica del problema.

La primera opción es el uso de un solver genérico como el integrado en Matlab con la función `fmincon`. Otra opción es el uso de la técnica de los momentos, que transforma el problema en una secuencia de problemas de programación semidefinida resoluble mediante solvers estándar. En este caso existen dos herramientas (`gloptipoly` y `solvemoment`), aunque la última muestra un mejor comportamiento al explotar la estructura dispersa de las matrices que formulan el problema de los momentos. En el caso de utilizar modelos de Volterra de grado dos, también es posible plantear el problema mediante un problema de desigualdades bilineales matriciales que puede resolverse mediante solvers existentes para este propósito (`penbmi`), o mediante una linealización del problema y la resolución de una secuencia de problemas convexos mediante un solver estándar (como `sedumi`).

Se han comparado mediante un ejemplos las prestaciones en tiempo de cómputo que ofrecen cada una de las posibles soluciones al problema, y se ha mostrado que, para el caso de modelos de Volterra de grado dos, el planteamiento mediante un

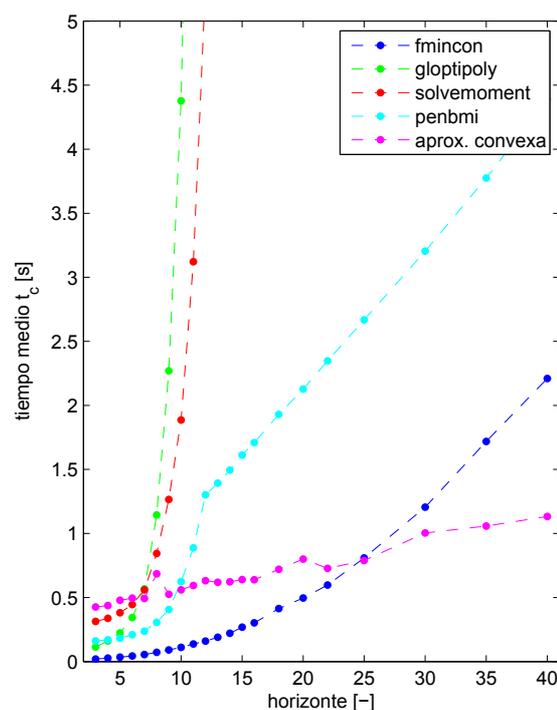


Figura 4: Tiempo de optimización  $t_c$  medio por iteración en función de los horizontes de control y predicción.

problema BMI y su aproximación convexa ofrece mejores prestaciones para grandes horizontes.

### Agradecimientos

Este trabajo ha sido financiado por la Universitat Jaume I de Castelló a través del proyecto con referencia P11B2015-42, el Ministerio de Economía y Competitividad a través del proyecto con referencia TEC2015-69155-R y la Comunidad de Madrid a través del proyecto PRICAM con referencia S2013/ICE-2933.

### Referencias

- [1] Doyle, F. J., Ogunnaike, B. A., and Pearson, R. K. (1995) Nonlinear model-based control using second-order Volterra models. *Automatica*, **31**, 697–714.
- [2] Maner, B. R., Doyle, F. J., Ogunnaike, B. A., and Pearson, R. K. (1996) Nonlinear model predictive control of a simulated multivariable polymerization reactor using second-order Volterra models. *Automatica*, **32**, 1285–1301.
- [3] Haber, R. (1995) Predictive control of nonlinear dynamic processes. *Applied Mathematics and Computation*, **70**, 169–184.

- [4] Gruber, J. K. (2010) *Efficient and Robust Techniques for Predictive Control of Nonlinear Processes*. Ph.D. thesis, Departamento de Ingeniería de Sistemas y Automática. Universidad de Sevilla.
- [5] Gruber, J. K., Ramirez, D. R., Limon, D., and Alamo, T. (2015) A convex approach for NMPC based on second order Volterra series models. *International Journal of Robust and Nonlinear Control*, **25**, 3546–3571.
- [6] Chesi, G. (2010) Lmi techniques for optimization over polynomials in control: a survey. *IEEE Transactions on Automatic Control*, **55**, 2500–2510.
- [7] Lasserre, J. B. (2001) Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, **11**, 796–817.
- [8] Laurent, M. (2009) Sums of squares, moment matrices and optimization over polynomials. *Emerging applications of algebraic geometry*, pp. 157–270, Springer.
- [9] Henrion, D., Lasserre, J.-B., and Löfberg, J. (2009) Gloptipoly 3: moments, optimization and semidefinite programming. *Optimization Methods & Software*, **24**, 761–779.
- [10] Löfberg, J. (2004) Yalmip: A toolbox for modeling and optimization in matlab. *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pp. 284–289, IEEE.
- [11] Kocvara, M. and Stingl, M. (2005) Penbmi user guide. Available from <http://www.penopt.com>.
- [12] Henrion, D., Löfberg, J., Kocvara, M., and Stingl, M. (2005) Solving polynomial static output feedback problems with penbmi. *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 7581–7586, IEEE.
- [13] Han, J. and Skelton, R. E. (2003) An lmi optimization approach for structured linear controllers. *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 5, pp. 5143–5148, IEEE.
- [14] De Oliveira, M., Camino, J., and Skelton, R. (2000) A convexifying algorithm for the design of structured linear controllers. *Decision and Control, 2000. Proceedings of the 39th IEEE Conference on*, vol. 3, pp. 2781–2786, IEEE.
- [15] Volterra, V. (1959) *Theory of Functionals and of Integral and Integro-Differential Equations*. Dover Publications.
- [16] Boyd, S. and Chua, L. (1985) Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems*, **32**, 1150–1161.
- [17] Doyle, F., Pearson, R., and Ogunnaike, B. (2001) *Identification and Control Using Volterra Models*. Springer.
- [18] Camacho, E. F. and Bordons, C. (2004) *Model Predictive Control*. Springer, second edn.