Robotics and Computer-Integrated Manufacturing 41 (2016) 23–30

Contents lists available at ScienceDirect

Robotics and Computer-Integrated Manufacturing

journal homepage: www.elsevier.com/locate/rcim

Non-linear model predictive control schemes with application on a 2 link vertical robot manipulator

Jacob Wilson, Meaghan Charest, Rickey Dubay*

University of New Brunswick, Department of Mechanical Engineering, 15 Dineen Drive, P.O. Box 4400, Fredericton, New Brunswick, Canada E3B 5A3

ARTICLE INFO

Article history:
Received 28 November 2014
Received in revised form
28 January 2016
Accepted 23 February 2016

Keywords:
Nonlinear model predictive control
2-Link vertical robotic manipulator
PID
Adaptive nonlinear modeling

ABSTRACT

The industrial requirements for controllers able to perform tasks in the presence of plant nonlinearities are growing. In addition, an increase in industrial computation power is allowing the implementation of more complex control algorithms in the fast processing industry. In this investigation three different nonlinear model predictive control algorithms are tested and evaluated in simulation and experimentally. The methodologies are adaptive nonlinear model predictive control (nMPC), PID based nMPC (PIDnMPC), and a novel simplified nMPC (SnMPC). These are tested in simulation with an inverted pendulum, a Van der Pol oscillator, and a planar 2-link vertical robotic arm. The controllers are tested experimentally using a fabricated planar 2-link vertical robotic arm apparatus. A comparison of the different algorithms is made with special attention to trajectory tracking, computational complexity and transient response dynamics.

1. Introduction

Since the 1970s MPC has gained popularity in the process industry and academia as a robust control form able to handle problem features such as constraints, disturbances, and complex modeling. The importance of control algorithms capable of negotiating nonlinear systems in current and future automation industries is undeniable. A number of controller methodologies have been developed to address this class of problems. These nonlinear schematics often present themselves as adaptations of classic control algorithms such as PID control and model predictive control (MPC) [20,11,18,7].

In many cases different adaptations of MPC for nonlinear systems are designed for a class of problems, or to emphasize a control objective [4,13,21]. Other examples of advanced nonlinear MPC techniques incorporate tools such as artificial neural networks, and global optimization methods [6,3]. The trade-off of these advanced nonlinear MPC methods is complexity and longer computer processing time [23]. This can be mitigated by efficient nonlinear MPC algorithm formulations, and becomes relevant when considering the more recent extension of MPC to applications, beyond the process industry, into electro-mechanical systems [16,15,10]. Examples of nonlinear MPC control formulations applied to fast systems include robotics, active vibration control, and unmanned aerial vehicles [12,24,17,14,8].

Other forms of state dependent MPC that address nonlinear dynamics have been formulated [1,19]. While these schemes work well, they are complex in formulation and cannot be readily applied to fast response systems, such as robotic manipulators, that involve kinematic solutions within the control sampling instant. In addition, these methodologies did not address the challenge of tracking irregular reference trajectories that are prevalent in robotic systems. This study focuses on developing nonlinear formulations that can be applied to the general field of MPC with respect to fast response systems to address some of these challenges.

The development and the application of three different formulations of nonlinear adaptive MPC are considered here. The first, nMPC, is based on standard nMPC methods. The second is a hybrid form of MPC that includes a PID component inside the nMPC control loop (PIDnMPC). The first two presented approaches can be applied to most adaptive variations of MPC schemes such as generalized predictive control (GPC), M-shifted MPC, and extended predictive control (EPC) [2,9]. The third methodology is a novel simplified version of nonlinear MPC (SnMPC). The performance of these nMPC formulations are evaluated with respect to fast processes in industry. Knowledge of underlying characteristics of these nonlinear methodologies can be helpful in selecting the proper control type for a given application. The algorithms designed here are tested first in simulation on standard nonlinear plant examples and then with a planar two link vertical robot manipulator. The simulation results are validated with experiments conducted using the modeled robot manipulator. Features

* Corresponding author.
E-mail addresses: Jacob.Wilson@unb.ca (J. Wilson), charest.meaghan@unb.ca (M. Charest), dubayr@unb.ca (R. Dubay).

http://dx.doi.org/10.1016/j.rcim.2016.02.003

of the nonlinear controllers are summarized and suggestions for future research directions are made.

2. Model predictive control

The standard Model Predictive control algorithm rests on the optimization of control moves, $\mathbf{u}$, in order to minimize future errors. The optimization is done over a prediction horizon, N using a model in order to construct predicted outcomes of plant actuations $\hat{\mathbf{y}}$. The MPC control algorithm considered here is dynamic matrix control (DMC) which is characterized by a dynamic matrix model $\mathbf{A}$:

$$\mathbf{A} = \begin{bmatrix} a_1 & 0 & \cdots & 0 \\ a_2 & a_1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ a_N & a_{N-1} & \cdots & a_{N-nu} \end{bmatrix} \quad (1)$$

The dynamic matrix is constructed with a normalized open loop response vector, $\mathbf{a}$, to an impulse or step actuation. When considering linear control problems the dynamic matrix is constant throughout controller implementation and the change in control move, $\boldsymbol{\Delta}\mathbf{u}$, is evaluated at every sampling instant.

The objective function used to optimize $\boldsymbol{\Delta}\mathbf{u}$ for DMC control can be written as:

$$J = \sum_{j=1}^{N} [\hat{\mathbf{y}}(t+j|t) - \hat{\mathbf{r}}(t+j)]^2 + \sum_{j=1}^{n_u} \lambda(j)[\boldsymbol{\Delta}\mathbf{u}(t+j-1)]^2 \quad (2)$$

where $\hat{\mathbf{r}}$ is a desired reference value or profile and the error to be minimized is the difference $\hat{\mathbf{y}} - \hat{\mathbf{r}}$. Least squares optimization is used to evaluate Eq. (2):

$$\boldsymbol{\Delta}\mathbf{u} = (\mathbf{A}^T\mathbf{A} - \lambda\mathbf{I})^{-1}\mathbf{A}^T\mathbf{E} \quad (3)$$

The prediction vector for the current time step $\hat{\mathbf{y}}|_t$ is then determined with the prediction vector of the previous time step $\hat{\mathbf{y}}|_{t-1}$ and the predicted response to the change in $\mathbf{u}$ made since the previous time step $\mathbf{A}\boldsymbol{\Delta}\mathbf{u}$:

$$\hat{\mathbf{y}}|_t = \hat{\mathbf{y}}|_{t-1} + \mathbf{A}\boldsymbol{\Delta}\mathbf{u} + \phi|_t \quad (4)$$

where ϕ is a model correction value, calculated at each time instant, that adjusts the $\hat{\mathbf{y}}$ to account for differences between model predictions and plant measurements y_m:

$$\phi|_t = y_m - \hat{\mathbf{y}}_1|_{t-1} \quad (5)$$

The three variations of nonlinear control algorithms discussed in this section are all derived from this standard linear MPC control algorithm. Good complex trajectory tracking performance is expected when considering robot manipulators [22]. For this reason, the MPC tracking enhancement described in [5] is added to all implemented and simulated control algorithms presented here. Examples of controller tracking capacity are shown experimentally with the robot manipulator.

2.1. Nonlinear adaptive MPC – nMPC

When deriving nMPC formulation, $\mathbf{A}$ in Eq. (1) is not constant. Every sampling instant a new $\mathbf{A}$ is generated to reflect the varying dynamics of a nonlinear plant. The $\mathbf{A}|_\mathbf{t}$ in this case is constructed using simulated normalized step or impulse responses. The simulations are generated using the current states to capture the associated system dynamics:

$$\boldsymbol{\Delta}\mathbf{u} = (\mathbf{A}^T|_t\mathbf{A}|_t - \lambda\mathbf{I})^{-1}\mathbf{A}^T|_t\mathbf{E} \quad (6)$$

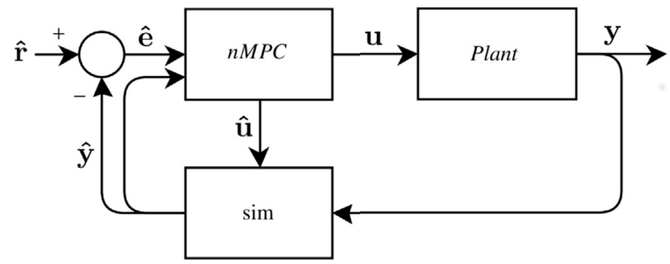

Fig. 1. Block diagram representation of nMPC.

Note that least squares optimization is used, this is appropriated because the adaption is a linearization of the nonlinear plant and Eq. (4) still holds true. In order to evaluate a more accurate prediction, $\hat{\mathbf{y}}$ is constructed using a simulation of the nonlinear plant being controlled. The adaptive nonlinear MPC algorithm is analogous to the standard linear MPC algorithm in all other respects. A block diagram representation of this control algorithm is shown in Fig. 1.

2.2. PID based nonlinear MPC – PIDnMPC

The PIDnMPC algorithm is an extension to nMPC, where a PID feed back loop is incorporated into the controller formulation as depicted in Fig. 2. The nMPC model is that of a closed loop PID controlled system as opposed to the usual plant model. This structure allows the nMPC component to strategically adjust the setpoint of the internal PID controller. The system nonlinearities can then be accounted for and their undesirable controlled features can be mitigated. In fast processing industrial applications this hybrid approach is being used [12]. The method is limited to linear MPC as the lead control algorithm due to the additional computational effort of adaptive MPC algorithms.

2.3. Simplified nonlinear MPC – SnMPC

The final form of nonlinear MPC considered in this investigation is a simplified nMPC abbreviated, SnMPC. The objective of this control scheme is to eliminate predicted errors. As with nMPC, this is achieved by minimizing the mean square error of the future response. The SnMPC objective cost function is shown in the following equation:

$$\mathbf{J} = \sum_{i=1}^{Nu} \left(\hat{\mathbf{r}}_i - \hat{\mathbf{y}}_i\right)^2 \cdot \boldsymbol{\alpha}_i \quad (7)$$

where $\boldsymbol{\alpha}_i$ is a vector of weights that can be used to adjust controller aggressiveness and i denotes the control time step. The setpoint at i is $\hat{\mathbf{r}}_i$, the prediction is $\hat{\mathbf{y}}_i$, and Nu is the number of control steps into the future that will be considered in the error optimization. In the simulations presented here $\boldsymbol{\alpha}_i = 1$.

The predicted response of the system can be broken up into two components. The first is the response of the system if the control action remains constant ($\hat{\mathbf{y}}'$). The second component is the change in the response of the system due to a change in the

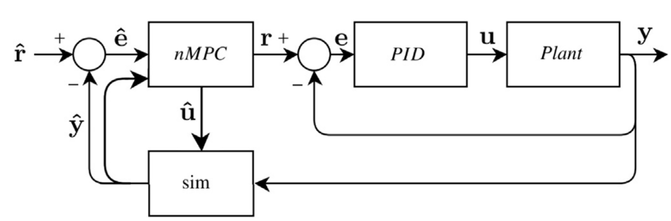

Fig. 2. Block diagram representation of PIDnMPC.