Nonlinear Receding Horizon Control based on Pontryagin Optimum Principle ★

João M. Lemos ★
★ INESC-ID/IST, R. Alves Redol 9, 1000-029 Lisboa, Portugal (e-mail: jlml@inesc.pt)

Abstract: An algorithm based on the Pontryagin’s Optimum Principle (POP) is proposed to solve nonlinear receding horizon control problems. Based on a nonlinear state-space plant model, at each time, the corresponding optimal control problem for a finite horizon is solved by imposing the POP conditions that are solved iteratively. These conditions impose a structure on the approximate solution that improves its convergence. The algorithm is illustrated by means of two examples, one concerning motion control and the other HIV-1 infection control.

1. INTRODUCTION

The optimization of a receding horizon (RH) cost functional for plants with nonlinear dynamics is difficult, among other aspects, due to the possible existence of multiple local minima. This effect may thus hamper the convergence of the manipulated variable signal when using a staircase approximation and requires the initial estimate to be close to the optimum. In this work an algorithm based on the Pontryagin’s Optimum Principle (POP) is proposed to solve nonlinear receding horizon control problems. Based on a nonlinear state-space plant model, at each time, the corresponding optimal control problem for a finite horizon is solved by imposing the POP conditions that are solved iteratively. These conditions impose a structure on the approximate solution that improves its convergence. According to a RH strategy, the resulting control is then applied to the plant up to time, with much smaller than the prediction horizon, and the procedure is repeated.

The algorithm is illustrated through its application to the control of infection by HIV-1. The nonlinear model used in the simulations comprises 3 states corresponding to the concentrations of un-infected T-CD4+ cells, infected T-CD4+ cells and free virus particles, and 2 control variables, corresponding to RTI and PI drug administration. These drugs have undesirable side effects and the dosage must achieve the therapeutic effect (reducing free virus particle counts below a specified threshold) with the least amount. Since the model predicts that an individual, once infected, will always remain so, no matter the control applied, the use of finite horizon control is prevented. Nonlinear RH provides a sub-optimal solution to the continuous administration of drugs that has the advantage of embedding feedback with all its advantages.

The contributions of the algorithm consist in the nonlinear receding horizon algorithms based on Pontryagin’s Optimum Principle and their demonstration in two case studies.

★ This work was supported by FCT (Portugal) under the project PTDC/EEA-CRO/100128/2008, ”HIVCONTROL - Control based on dynamic modeling of HIV-1 infection for therapy design”.

The paper is organized as follows: After this introduction (section 1), that motivates and presents the problem to be solved, sections 2 and 3 describe, respectively, the discrete and continuous time versions of the algorithm. Section 4 is devoted to the motion control example and section 5 to the HIV-1 infection control example. Sections 6 draws conclusions.

2. CONTROL ALGORITHM – DISCRETE TIME

For a formulation in discrete time, let the plant to be controlled be modeled by the state difference equation

\[ x(k + 1) = f(x(k), u(k)) \]  

(1)

with initial condition \( x(0) = x_0 \) and where \( x \in \mathbb{R}^n \) is the plant state (for simplicity hereafter assumed available for direct measure), \( u \in \mathbb{R}^m \) is the manipulated variable and \( f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n \).

In a receding horizon (RH) setting, let \( k \in \mathbb{N}_0 \) denote the current value of discrete time. In order to compute the value of the manipulated variable \( u \) to apply to the plant at time \( k \), \( u(k) \), consider predictors of the plant state, denoted \( \hat{x} \), from time \( k + 1 \) up to time \( k + N_H \) (with \( N_H \in \mathbb{N} \) the ”prediction horizon”), obtained under the assumption that, from time \( k + 1 \) up to time \( k + N_H - 1 \) the manipulated variable to the plant is \( \hat{u}(i) \), the initial condition being \( x(k) \). Along the prediction horizon, both the predicted states and the virtual controls \( \hat{u} \) are indexed by a ”virtual time” index \( i \) that starts at 0 for present time \( k \).

The sequence of virtual controls \( \hat{u} \) is computed such as to optimize the RH performance index

\[ J(k) = \psi(\hat{x}(N_H)) + \sum_{i=0}^{N_H-1} L(\hat{x}(i), \hat{u}(i)) \]  

(2)

where \( \psi : \mathbb{R}^n \rightarrow \mathbb{R} \), and \( L : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R} \). This is accomplished by the following algorithm:

Algorithm A1

At each time \( k \), recursively perform the following steps:

0. Initialize the virtual control sequence, \( \hat{u}(i) \) for \( i = 0, \ldots, N_H - 1 \). This may be done in several ways. For
instance, set \( \hat{u}(i) = 0 \) for all \( i \), or set \( \hat{u}(i) \) to its value at time \( k-1 \).

1. With the current sequence of virtual control, \( \hat{u}(i), i = 0, \ldots, N_H - 1 \), compute the predicted values of the state \( \hat{x} \) over the horizon, by iterating for \( i = 1, \ldots, N_H \)

\[
\hat{x}(i + 1) = f(\hat{x}(i), \hat{u}(i)) \quad (3)
\]

starting from the initial condition

\[
\hat{x}(0) = x(k) \quad (4)
\]

2. Compute the co-state terminal condition by

\[
\lambda(N_H) = \psi_x(x)_{|x=\hat{x}(N_H)} \quad (5)
\]

where \( \psi_x \) denotes the gradient of \( \psi \) with respect to \( x \).

3. Starting from (5), iterate backwards in time the adjoint equation

\[
\lambda(i) = f^T(\hat{x}(i), \hat{u}(i))\lambda(i + 1) + L^*_x(\hat{x}(i), \hat{u}(i)) \quad (6)
\]

for \( i = N_H - 1 \) down to \( i = 0 \), in order to compute the co-state. Here \( f^*_x \) denotes the Jacobian matrix of \( f \) with respect to \( x \), and \( L^*_x \) denotes the gradient vector of \( L \) with respect to \( x \).

4. Compute the virtual control updating equation by

\[
\Delta \hat{u}(i) = \gamma_d [(\lambda^T(i + 1) + f^*_x(\hat{x}(i), \hat{u}(i))] \quad (7)
\]

for \( i = 0, \ldots, N_H \), with \( \gamma_d \in \mathbb{R} \).

5. Update the virtual control sequence by

\[
\hat{u}(i) \leftarrow \hat{u}(i) + \Delta \hat{u}(i) \quad (8)
\]

for \( i = 0, \ldots, N_H \), in the case of performance index maximization. Otherwise, for minimization, replace the \( + \) by \( - \).

6. If

\[
\| \Delta \hat{u} \| \leq \varepsilon \quad (9)
\]

with \( \varepsilon \in \mathbb{R} \) a small parameter, set the control to actually apply to the plant at time \( k \) to

\[
u(k) = \hat{u}(0) \quad (10)
\]

Otherwise, go to step 1.

\[\Box\]

Actually, this is nothing more than a gradient algorithm to approximate the solution of POP conditions (Bryson (1999)), with the innovation of being embedded in a RH framework. The major advantage with respect to other nonlinear RH control computation algorithms consists in the fact that the adjoint equations impose a "shape" to the virtual control function over the prediction horizon that allows convergence to the optimum even when the initial optimal control sequence guess is rather poor.

The gradient minimization in steps 4 and 5 may be replaced with other methods, with or without constraints [Polak (1999); Pytlak (1999); Bryson (1999)].

3. CONTROL ALGORITHM – CONTINUOUS TIME

For a formulation in continuous time, model the plant to control by the nonlinear state equation

\[
\dot{x}(t) = f(x(t), u(t)) \quad (11)
\]

with initial condition

\[
x(0) = x_0 \quad (12)
\]

where \( x \in \mathbb{R}^n \) is the plant state (for simplicity assumed available for direct measure), \( u \in \mathbb{R}^m \) is the manipulated variable and \( f : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n \).

In a RH setting, let \( t \) denote current time. The problem consists in finding the control \( u \) to apply to the plant between \( t \) and \( t + \delta \), \( \delta \in \mathbb{R} \) by optimizing in order to \( \dot{u} \) the performance index

\[
J(t) = \psi(\hat{x}(T_H)) + \int_0^{T_H} L(\hat{x}(\tau), \dot{\hat{x}}(\tau))d\tau \quad (13)
\]

in a RH sense, where \( \hat{x} \) satisfies (11) with initial condition

\[
\hat{x}(0) = x(t) \quad (14)
\]

This is accomplished by the following algorithm:

\textbf{Algorithm A2}

At time \( t \), recursively perform the following steps:

0. Initialize the virtual control function, \( \hat{u}(\tau) \), for \( 0 \leq \tau \leq T_H \).

1. With the current virtual control function \( \hat{u}(\tau) \), \( 0 \leq \tau \leq T_H \), compute the predicted values of the state over the prediction horizon, \( \hat{x}(\tau) \), \( 0 \leq \tau \leq T_H \), by solving the ordinary differential equation

\[
\frac{d\hat{x}(\tau)}{d\tau} = f(\hat{x}(\tau), \hat{u}(\tau)) \quad (15)
\]

with initial condition

\[
\hat{x}(0) = x(t) \quad (16)
\]

2. Compute the terminal condition of the co-state by

\[
\lambda(T_H) = \psi_x(\hat{x})_{|\hat{x}=\hat{x}(T_H)} \quad (17)
\]

3. Solve backwards the adjoint equation

\[
\frac{d}{d\tau} \lambda(\tau) = -f^T(\hat{x}(\tau), \hat{u}(\tau))\lambda(\tau) - L^*_x(\hat{x}(\tau), \hat{u}(\tau)) \quad (18)
\]

in order to compute the co-state in the interval \( 0 \leq \tau \leq T_H \).

4. Compute the virtual control updating function by

\[
\Delta \hat{u}(\tau) = \gamma_C [\lambda^T(\tau) + f^*_x(\hat{x}(\tau), \hat{u}(\tau))] + L_\mu(\hat{x}(\tau), \mu(\tau)) \quad (19)
\]

with \( \gamma_C \in \mathbb{R} \) a gain.

5. Update the virtual control function by

\[
\hat{u}(\tau) \leftarrow \hat{u}(\tau) + \Delta \hat{u}(\tau) \quad (20)
\]

for \( 0 \leq \tau \leq T_H \), in the case of performance index maximization. Otherwise, for minimization, replace the \( + \) by \( - \).

6. If

\[
\sqrt{\int_0^{T_H} (\Delta \hat{u}(\tau))^2 d\tau} \leq \varepsilon \quad (21)
\]

with \( \varepsilon \in \mathbb{R} \) a small parameter, set the control to actually apply to the plant to

\[
u(t + \tau) = \hat{u}(\tau) \quad 0 \leq \tau \leq \delta \quad (22)
\]

and restart the whole algorithm with \( t \leftarrow t + \delta \). Otherwise, go to step 1.

\[\Box\]

The remarks made to the discrete time version of the algorithm apply in continuous time as well.
4. EXAMPLE 1: MOTION CONTROL PROBLEM

As a first example, a motion control problem is considered, namely the Velocity Direction Programming problem [Bryson (1999)]. Consider a bead that slides along a rail, subject only to the gravitational force. The problem consists in finding the shape of the rail that maximizes the horizontal range attained in a fixed interval of time \([0, T_H]\). The manipulated variable \(u\) of this control problem consists in the direction of the velocity vector, that defines the angle of the tangent of the rail below the horizontal, as a function of time.

According to figure 1, let \(v\) be the velocity, \(z\) the horizontal coordinate and \(y\) the vertical coordinate. From [Bryson (1999)], the equations of motion in discrete time are
\[
\begin{align*}
v((k+1)h) &= v(kh) + gh \sin(u(kh)) \\
z((k+1)h) &= z(kh) + \Delta S(kh) \cos(u(kh))
\end{align*}
\]
where
\[
\Delta S(kh) = hv(kh) + \frac{1}{2} gh^2 \sin(u(kh))
\]
g = 9.8 \(ms^{-1}\) is the acceleration of gravity and \(h\) is the sampling interval.

The variables \(v\) and \(z\) form a state:
\[x(k) = \begin{bmatrix} v(kh) \\ z(kh) \end{bmatrix}\]

The \(y\) coordinate does not need to be included in the state, but it is important for plotting the shape of the rail. It may be computed from the kinematic equation:
\[y((k+1)h) = y(kh) + \Delta S \sin(u(kh))\]
The initial conditions are \(v(0) = 0\), \(z(0) = 0\) and \(y(0) = 0\).

I the case at hand, the performance index to optimize (maximize) is
\[J(k) = z(N_H)\]

Figure 2 shows the optimized control for various number of iterations when the solution of POP conditions are approximated with a gradient algorithm. This corresponds to obtaining the virtual control with algorithm A1 starting from \(k = 0\). Figure 3 shows the corresponding shapes of the rail on which the bead slides. The results are shown after 10, 100, 250 and 1242 iterations, this last one corresponding to convergence being attained for the specified value of \(\varepsilon\).

In order to illustrate the fact that the algorithm is able to converge to the optimum even when the initial guess of the optimal control is a poor one, the control has been initialized with a sinusoid, which is far away from the shape obtained at convergence (a straight line). Despite converging to the correct point, the algorithm requires a large number of iterations. In order to circumvent this problem, one may run this algorithm for a number of iterations, and then, once close to the optimum, switch to a faster algorithm (but with a smaller convergence region).

Figure 4 shows an example of a rail shape when the system is controlled with a RH strategy using algorithm A1 and figure 5 shows a plot of the range attained after a fixed time interval as a function of the prediction horizon.

5. EXAMPLE 2: HIV1 INFECTION

Strategies for counteracting HIV infection designed using control methods currently receive more and more attention [Chang and Astolfi (2009), Barão and Lemos (2007)]. Detailed studies that combine modeling analysis with clinical results show that the initial infection phase may be represented using simple nonlinear state models [Perelson et al. (1999), Wodarz (2001)]. This fact boosted the production of an increasing number of papers where therapy strategies are derived from control principles. Examples related to the present work include Optimal Control [Souza et
The model used here to describe the HIV-1 infection [Xia et al. (2004), Perelson et al. (1999)] is a deterministic one-compartment model with the following three state variables:

- $x_1$: Concentration of healthy cells
- $x_2$: Concentration of infected cells
- $x_3$: Concentration of virions (free virus particles)

The equations connecting these variables read as follows:

$$
\begin{align*}
\dot{x}_1 &= s - dx_1 - (1 - u_1)\beta x_1 x_3 \\
\dot{x}_2 &= (1 - u_1)\beta x_1 x_3 - \mu x_2 \\
\dot{x}_3 &= (1 - u_2)kx_2 - cx_3. \\
\end{align*}
$$ (28)

In the first equation, $s$ represents the production rate of healthy cells, the coefficient $d$ the natural death of the cells and $\beta$ the infection rate coefficient. The infection rate of healthy cells is proportional to the product of healthy cells $x_1$ and free virus $x_3$. This process can be influenced by drugs (Reverse Transcriptase Inhibitors – RTI) that reduce the virus performance. This influence is represented by the manipulated variable $u_1$, in which $u_1 = 0$ corresponds to absence of drug and $u_1 = 1$ to a drug efficiency in preventing infection of 100%. Actually, with the available drugs, the efficiency is below 100%, and $u_1$ is constrained to the interval $[0, u_{\text{max}}]$ with $u_{\text{max}} < 1$. The second equation comprises two terms that represent, respectively, the transition of healthy cells to infected cells and the death of infected cells, with $\mu$ the death coefficient.

An infected cell liberates free virus. This process is represented in the third equation, where the first term represents the liberation of virus by infected cells and the second the "death" of free virus with $c$ the corresponding coefficient. The manipulated variable $u_2$ represents the action of drugs (Protease Inhibitors – PI) that prevent infected cells to produce free particle virus (virions). Once again $u_2$ is constrained to the interval $[0, u_{\text{max}}]$ with $u_{\text{max}} < 1$. Other possibilities to model drug effect is either to assume that parameter $\beta$ is proportional to $e^{-C u_i}$, with $C - e$ a constant or to resort to the Hill equation, commonly used in pharmacodynamic models [BH05, Lemos et al., (2005)].

Figure 6 shows the transient time response to an HIV-1 infection. The parameters used [Xia et al. (2004)] are the ones of table 1. It must be remarked that it is unrealistic to assume an unique model to represent a wide group of individuals. Indeed, there is a strong variability, both from inter-individual and due to virus resistance mechanisms that prevents this. Current efforts aim at characterizing the dynamics of an individual, based not only on sex, weight, age, etc., but also on patient genetic features. The resulting a priori model is then corrected by observations made along time. This case study concerns only the control design and assumes that the modeling issues are solved.

In actual clinical practice only the viral load is available. The methods in this paper must thus be complemented with a state observer that takes into consideration that measures are sparse in time.
The initial conditions correspond to an healthy person infected with a virus concentration of 1 copy per mm$^3$. As can be seen in figure 6, during an initial phase of the illness, lasting for about 30 days, the concentration of infected cells and free virus in the body is very small. After this period, a fast growth of the concentration of virus and infected cells is noticed, together with a major decrease of healthy CD4+T cells. After 6 months the infection stabilizes, and its kept in an approximated steady state for a period that may last between 2 and 10 years or even more. After this period, through a mechanism not modeled in (28), the number of healthy CD4+T cells is drastically reduced and the patient develops AIDS.

Figure 7 shows some state trajectories obtained with different initial infection conditions. This nonlinear model has two equilibrium points: An unstable point corresponding to the "healthy" individual (over the $x_1$ axis) and a stable point corresponding to infection, around which the behavior is oscillatory.

It is shown in [Barão and Lemos (2007)] that model (28) implies that there is no control function able to drive the state from an initial condition outside the $x_1$ axis back to it. In other words, the model predicts that, once an individual is infected, it will remain so forever, no matter the combination of drugs applied. Therefore, what can be expected is to reduce the viral load below a threshold under which it is undetected with current technology (e. g. 50 copies/mm$^3$).

Since anti-retroviral drugs have major damaging side effects, this objective has to be balanced with the minimization of the amount of drugs applied. A possibility is, therefore, to minimize the cost functional

$$J(t) = \psi_3 x_3^2(T_H) + \frac{1}{2} \int_0^{T_H} [u_1^2(\tau) + u_2^2(\tau) + \gamma_3 x_3^2(\tau)]d\tau$$

(29)

where $\psi_3$ and $\gamma_3$ are constant parameters to be selected, the term proportional to the integral of $x_3^2(\tau)$ is a regularization term that proved to be much useful.

Figure 8 shows the optimal profile of administration of the RTI drug computed using Pontryagin’s Optimum Principle in continuous time. It corresponds to the use of algorithm A2 starting from $t = 0$. Besides being an open-loop control law, this has the disadvantage of cutting the amount of drug down to zero close to the final optimization time. If this was followed, the viral load would rise again, with disastrous health consequences.

In order to avoid the serious drawback of POP, the use of a receding horizon strategy is strongly suggested. Figures 9 and 10 show the results obtained with the RH algorithm proposed (algorithm A2). The problem could be formulated in discrete time as well, in which case algorithm A1 would be used. The parameters configuring the controller were selected as $T = 500$, $\delta = 50$, $\gamma_3 = 1$ and $\psi_3 = 10$. As seen in figure 10, the manipulated variables do not go to zero, but keep a constant value that prevent the viral load to increase again.
6. CONCLUSION

Algorithms for nonlinear receding horizon control of nonlinear systems that are based on Pontryagin’s Optimum Principle have been proposed and illustrated with a motion control problem and a simple model of HIV-1 infection. Both discrete time and continuous time cases are considered. Since the adjoint equation imposes a constraint on the "shape" of the optimized control over the prediction horizon, these algorithms allow the initial guess to be far from the optimum. This type of algorithms may be used to drive the control estimate to a neighborhood of the optimum, thereby providing an initialization to other type of methods that converge faster but whose convergence region is smaller.

Acknowledgement

The simulations in figures 9-11 have been performed in cooperation with R. Malhado.

REFERENCES


