Proc. Estonian Acad. Sci. Eng., 2003, **9**, 1, 25–33
https://doi.org/10.3176/eng.2003.1.02

A parametric optimization technique for model-predictive control simulation

Ingmar Randvee

Institute of Cybernetics, Tallinn Technical University, Akadeemia tee 21, 12618 Tallinn, Estonia; Ingmar@ioc.ee

Received 19 March 2002, in revised form 22 November 2002

Abstract. A parametrization technique, which has been introduced in the 1980s in the context of multilevel systems, is revisited. It is shown that a version of this optimization technique (which skips the introduction of co-states and the solution of the two-point boundary value problem) may be applicable in the numerical simulation of one-step ahead unconstrained model-predictive control strategies and other suboptimal real-time dynamic systems that use predicted closed-loop system trajectories. The representation is given in discrete-time setting using state-space models.

Key words: predictive control, parametric optimization.

1. DESCRIPTION OF THE PROBLEM

The past decades have witnessed a great success in the use of a variety of model-predictive control (MPC) methods [1–6]. The basic idea of MPC is to predict the controlled variables over a future horizon using a prediction model of the process, to calculate the controller outputs by minimizing an objective function, and finally to apply only the first control action. Since all optimizations involved in MPC are to be solved on-line at every sampling instant, they become the bottleneck when applying MPC methods to large-scale or rapid systems. To be more specific, there are three types of trajectories related to the minimization of a performance index.

1. *Projected desired trajectory* (PDT) that starts from the actual value of the process output and reaches the set point in a smooth manner. It is recalculated for each sampling period k by solving a non-linear n-step (n is relatively large) quadratic programming (QP) problem. The given set point may be time-varying. As a result of minimization of the sum of weighted square errors between the modelled output and given set points using (adaptable) process model equations,

the optimal (relative to stability, robustness, tracking, and other main properties of the system) PDT is calculated. As an illustration, the expression for finite horizon predicted cost is often used as a control Lyapunov function [1].

2. *Driving desired trajectory* (DDT) is produced in each sampling period by solving a *n*-step (n is not large) QP problem using first values of the PDT. This trajectory, in fact, guides the process output to the PDT in the desired way since it corresponds to the realized control action, which is calculated using DDT and the model of the process.

3. *Measured process output trajectory* (POT) approches the DDT, PDT, and the set point if the control system is stable and the used process model is sufficiently adequate. This trajectory is used for the analysis of the control quality and for the process model adaptation. Adaptation mechanism again includes the multistep QP algorithm.

As an example, we consider the multivariable DDT problem for an unconstrained discrete-time linear system [5].

At each sampling instant k find the control sequence $u_{k,k}, u_{k+1,k}, ..., u_{k+N-1,k}$ such that the specific cost function

$$J(k) = \frac{1}{2}\sum_{j=1}^{N_1}(x_{k+j,k} - x^{\text{ref}}_{k+j,k})'Q(x_{k+j,k} - x^{\text{ref}}_{k+j,k}) + \frac{1}{2}\sum_{j=0}^{N-1}u'_{k+j,k}Ru_{k+j,k} \tag{1}$$

is minimum subject to the state predictive model

$$x_{k+j,k} = Ax_{k+j-1,k} + Bu_{k+j-1,k}, \quad j = 1, 2, ..., N, \tag{2}$$

where $x_{k+j,k}$ is the closed-loop state vector predicted at the instant k for the instant $k+j$, $u_{k+j-1,k}$ is the predicted control vector, and $x^{\text{ref}}_{k+j,k}$ is the reference (PDT_k) state; the matrices $A, B, R > 0, Q \geq 0$ and the initial state $x_{k,k}$ are given.

The predictive model is updated at every sampling instant k using the measured actual state which is used as initial value of the problem (1)–(2) for the next sampling period.

The dynamic DDT problem is usually transformed into the static form and then solved directly [3,5,7]. Recently a number of approaches has been proposed to reduce computational requirements of MPC [6–8].

In this paper, for the same reason, we suggest a version of a direct optimization scheme, which has been introduced in [9] and later applied to the Nash-optimal set of subsystems [10]. The suggested scheme is directly applicable in the numerical simulation of unconstrained multivariable MPC systems, defined in the state space. The technique is restricted to linear and non-linear control objects for which the so-called influence matrix is known. The meaning of the influence matrix can be explained by a simple example.

Consider a single-stage (static) optimization problem:

$$\min_u J(x,u), \text{ subject to } f(x,u) = 0, x \in R^n, u \in R^m.$$

Assume that $x = g(u)$, which means that

$$\partial g/\partial u = \partial x/\partial u\,. \tag{3}$$

From the stationarity condition

$$\frac{\partial J}{\partial u} + \left(\frac{\partial x}{\partial u}\right)' \frac{\partial J}{\partial x} = 0 \tag{4}$$

and the differential of the equality constraint

$$\frac{\partial f}{\partial u} + \left(\frac{\partial x}{\partial u}\right)' \frac{\partial f}{\partial x} = 0 \tag{5}$$

we can state that if J is convex, $\partial f/\partial u$ exists, and $\partial f/\partial x$ is nonsingular, then

$$\frac{\partial x}{\partial u} = -\left(\frac{\partial f}{\partial x}\right)^{-1} \frac{\partial f}{\partial u}\,. \tag{6}$$

The term $\partial g/\partial u$ in (3) is named the influence matrix [9]. If this matrix is control-independent, we can solve the problem directly.

2. MAIN PROBLEM. OPTIMIZATION OF DISCRETE-TIME SYSTEMS USING PERFORMANCE CRITERIA PARTITION

In the multistage (dynamic) case we can apply the single-stage optimization approach (3)–(6) using the performance criterion partition.

Consider again the multivariable DDT problem (1)–(2), in which the controlled object is now non-linear. For the simplification of the expressions assume also that the reference level is constantly zero $(x^{\mathrm{ref}} = 0)$. Then

$$J = \frac{1}{2}\sum_{k=0}^{N-1}(x'_{k+1}Qx_{k+1} + u'_k R u_k), \tag{7}$$

and

$$x_{k+1} = f(x_k, u_k, k),\ \ x_k \in R^n,\ \ u_k \in R^m,\ \ R > 0, Q \ge 0. \tag{8}$$

Here, in discrete-time setting, we use the same partial differentiation notation.

The optimal control law must satisfy the stationarity condition

$$\frac{\partial J}{\partial u_k} = 0,\ \ k = 0, 1, \ldots, N-1. \tag{9}$$

In [9,10] it is shown that if the influence matrix is control-independent then the condition (9) can be satisfied and the problem (7)–(8) iteratively solved without introducing co-states.

2.1. Solution procedure

The performance index is partitioned $J = J_k + \bar{J}_k$, and the second term of the stationarity condition

$$\frac{\partial J_k}{\partial u_k} + \frac{\partial \bar{J}_k}{\partial u_k} = 0 \tag{10}$$

is parametrized, i.e.,

$$\frac{\partial J_k}{\partial u_k} + \rho_k = 0, \;\; k = 0, 1, \ldots, N-1. \tag{11}$$

There are several possible ways of partition. For the stated control problem the natural choice would be:

$$J_k = \frac{1}{2}(x'_{k+1} Q x_{k+1} + u'_k R u_k), \tag{12}$$

$$\bar{J}_k = \frac{1}{2} \sum_{l=0, l\neq k}^{N-1} (x'_{l+1} Q x_{l+1} + u'_l R u_l), \;\; k = 0, 1, ..., N-1. \tag{13}$$

The condition for optimality at the stage k is

$$\frac{\partial x'_{k+1}}{\partial u_k} Q f(x_k, u_k, k) + R u_k + \rho_k = 0, \tag{14}$$

where

$$\rho_k = \left(\frac{\partial x_{k+1}}{\partial u_k} \right)' \frac{\partial \bar{J}_k}{\partial x_k}. \tag{15}$$

Now instead of the N-stage problem we must solve a sequence of modified static problems (3)–(6).

If the influence matrix is independent of the control, i.e.,

$$f(x_k, u_k, k) = g(x_k, k) + D(x_k) u_k, \tag{16}$$

where

$$\partial x_{k+1} / \partial u_k = D(x_k), \tag{17}$$

then the condition for optimality at stage k gives

$$u_k = M(x_k)^{-1} (D'(x_k) Q g(x_k, k) + \rho_k). \tag{18}$$

The function $g(x_k, k)$ represents that part of $f(x_k, u_k, k)$, which depends only on x_k; nonconstant matrices $M(x_k)$ and $D(x_k)$ are obtained when substituting $f(x_k, u_k, k)$, into the relation (14). The procedure of obtaining Eq. (18) is transparent in the case in which the non-linear object is linear with respect to the control:

$$f(x_k, u_k, k) = \varphi(x_k, k) + Bu_k. \tag{19}$$

For the problem (12), (19), the partially closed-loop control law is obtained directly:

$$u_k = -[B'QB + R]^{-1}(B'Q\varphi(x_k, k) + \rho_k). \tag{20}$$

In the case of a linear object, the term $\varphi(x_k, k)$ is replaced by Ax_k.

It is natural that the controller is also partitioned. The first part represents a one-stage ahead control law in feedback form. The second, feed-forward term, represents open-loop control, which takes into consideration the influence of the remained control horizon.

For the exposed partition (12), the vector ρ_k is calculated sequentially, backward in time, along the closed-loop system trajectory $x_{k+1} = \tilde{f}(x_k)$:

$$\rho_k = \left(\frac{\partial x_{k+1}}{\partial u_k}\right)' v_k, \tag{21}$$

where

$$v_{k-1} = \left(\frac{\partial \tilde{f}}{\partial x_k}\right)' Qx_{k+1} + \left(\frac{\partial u_k}{\partial x_k}\right)' Ru_k + \left(\frac{\partial \tilde{f}}{\partial x_k}\right)' v_k, \ k = 0, 1, ..., N-1, \ v_{N-1} = 0. \tag{22}$$

The inner structure of the recursion (22) can be easily explained on the example of a linear object. It is supposed that the trajectory $x_{k+1} = \tilde{f}(x_k)$ under the controls (18) is stable and unique.

2.2. Solution algorithm

1. Take $\rho_k = 0$ and find sequentially, using (18) and (8), the controls u_k together with the closed-loop trajectory $x_{k+1} = \tilde{f}(x_k)$, $k = 0, 1, 2, \ldots, N-1$.
2. Find the value of ρ_k along the closed-loop trajectory (21)–(22).
3. Substitute ρ_k into the equation for u_k (20), find $x_{k+1} = \tilde{f}(x_k)$, and repeat step 2.

It is assumed that this simple iteration (steps 2 and 3) converges at least for closed-loop systems, which are stable at $\rho_k = 0$. Numerical simulation supports this assumption. To improve the convergence, at each iteration c the calculated value of the coordination term ρ_k can be relaxed, i.e,

$$\rho_k^{c+1} = \Omega\rho_k + (I - \Omega)\rho_k^c, \tag{23}$$

where Ω is the so-called relaxation matrix.

According to the MPC scheme, the final value of u_0 is implemented and the measured real state is taken as the basis for the next solution of the problem (7)–(8).

2.3. Application domain

The algorithm described is directly applicable to several numerical simulation problems of MPC stated in state space and discrete time in the following cases:
– linear system;
– non-linear system, which is linear with respect to the controls (19);
– non-linear system, in which the influence matrix is independent of the control (16);
– Nash-optimal linear system [10], defined on the set of m interconnected subsystems:

$$x_{j,k+1} = A_j x_{j,k} + \sum_{i \neq j}^{m} A_i x_{i,k} + B_j u_{j,k}, \tag{23}$$

$$J_j = \frac{1}{2}\sum_{k=0}^{N-1}((x_{j,k+1} - x_{j,k+1}^{\text{ref}})' Q_j (x_{j,k+1} - x_{j,k+1}^{\text{ref}}) + u'_{j,k} R_j u_{j,k}),$$
$$k = 0, 1, \ldots, N-1;\ j = 1, 2, \ldots, m, \tag{24}$$

performance indices J_j may also contain the controls of other subsystems, which will lead to interesting algorithms of incentive control [11];
– variants generated by suitable partition of performance criteria; for example, we can separate the performance index (7) so that

$$J_k = \frac{1}{2}(x'_{k+1} Q x_k + u'_k R u_k), \tag{25}$$

$$\bar{J}_k = J - J_k. \tag{26}$$

Now, in the case in which the influence matrix is independent of the control (16), this partition simplifies calculations since inversion of the non-constant matrix $M(x_k)$ is avoided. The expression for coordination term in this case is also slightly modified [9].

3. NUMERICAL SIMULATION OF THE SOLUTION PROCEDURE

Consider the problem (7)–(8) with the partition (13) to which the steps 2 and 3 of the solution algorithm are c times applied. Simple comparison of the final value of the performance index after each iteration illustrates to some extent the convergence process.

Let us take a simple scalar linear system

$$x_{k+1} = ax_k + bu_k, \; k = 0, 1, ..., N-1, \tag{27}$$

and a non-linear system

$$x_{k+1} = ax_k^2 + bu_k, \; k = 0, 1, ..., N-1, \tag{28}$$

with positive initial state value x_0.

The performance index for both systems is the same:

$$J = \frac{1}{2}\sum_{k=0}^{N-1}(x_{k+1}qx_{k+1} + u_k r u_k). \tag{29}$$

Simulation procedure has been carried out for different values of the parameters and state transitions. One version of both, of the linear and the non-linear system together with the obtained values of the performance index J_c after c iterations are presented in Tables 1 and 2, where ω is the relaxation parameter.

As it was to be expected, for a stable object the process converges rapidly. For unstable objects and stabilized closed-loop systems, the convergence decreases with the increase of the state transition a. The reduction of the relaxation parameter ω (Table 1, row 6 and Table 2, row 5) supresses oscillations and restores rapid convergence.

Table 1. Simulation results of the linear system for $x_0 = 50$, $q = 1.0$, $r = 1.0$, $b = 1.0$, $N = 15$

ω	a	J_1	J_2	J_5	J_{10}	J_{20}
1.0	0.8	634	577	576	576	576
1.0	1.0	834	777	776	773	773
1.0	1.1	907	865	862	860	860
1.0	1.2	960	933	942	935	933
1.0	1.3	993	1051	992	998	995
0.5	1.3	993	998	996	996	995

Table 2. Simulation results of the non-linear system for $x_0 = 10$, $q = 1.0$, $r = 1.0$, $b = 1.0$, $N = 15$

ω	a	J_1	J_2	J_5	J_{10}	J_{20}
1.0	0.08	32.82	32.75	32.75	32.75	32.75
1.0	0.10	53.14	52.62	52.55	52.55	52.55
1.0	0.12	81.49	79.87	78.56	78.52	78.52
1.0	0.15	151.70	179.43	149.83	173.47	149.56
0.5	0.15	151.70	134.45	131.06	131.06	131.06

4. CONCLUDING REMARKS

Model-predictive control has become an accepted standard for many control problems in industry. Its serious drawback is the relatively large amount of on-line computations, which limit the applicability to relatively slow or small problems. In this paper, a version of the parametric optimization technique is suggested which can find an application in real-time simulation of several modifications of MPC strategies, based on predicted closed-loop system trajectories.

The technique is quite flexible. It may yield linear or non-linear partial feedback laws, depending on the type of non-linearities, on the nature of influence matrix, and on the chosen decomposition of the performance index. Also, the used state-space representation is well suited for complex multivariable systems. Among others, the Nash-optimal set of subsystems and some schemes of incentive control can be successfully treated.

The optimization technique is not very demanding since it does not explicitly introduce the co-state vector to derive the stationarity conditions to be satisfied by the controls. Instead, the resulting equations are solved iteratively by means of parametrization. The calculated controls are treated as suboptimal and are implementable at any iteration step.

ACKNOWLEDGEMENT

This work was supported by the Estonian Science Foundation under grant No. 5086.

REFERENCES

1. Primbs, J. A. The analysis of optimization based controllers. *Automatica*, 2001, **37**, 933–938.
2. Lee, Y. I. and Kouvaritakis, B. Robust receding horizon predictive control for systems with uncertain dynamics and input saturation. *Automatica*, 2000, **36**, 1497–1504.
3. Kambhampati, C., Mason, J. D., and Warwick, K. A stable one-step ahead predictive control of non-linear systems. *Automatica*, 2000, **36**, 485–495.
4. Van Antwerp, J. G. and Braatz, R. D. Model predictive control of large scale process. *J. Process Control*, 2000, **10**, 1–8.
5. Sanches, M. and Rodellar, J. *Adaptive Predictive Control. From Concepts to Plant Optimization.* Prentice Hall Europe, Cornwall, 1996.
6. Ansari, R. M. and Tade, M. O. *Non-linear Model-based Process Control. Applications in Petroleum Refining*. Springer-Verlag, London, 2000.
7. Zheng, A. Reducing on-line computational demands in model predictive control by approximating QP constraints. *J. Process Control*, 1999, **9**, 279–290.
8. Rossiter, J. A., Kouvaritakis, B., and Rice, M. J. A numerically robust state-space approach to stable predictive control strategies. *Automatica*, 1998, **34**, 65–73.
9. Abou-Kandil, H., Drouin, M., and Bertrand, P. Two-level control laws for discrete-time non-linear systems. In *Proc. American Control Conference.* San Diego, 1984, 1493–1497.
10. Randvee, I. Two-level control law for nash-optimal nonlinear discrete-time systems. *Proc. Estonian Acad. Sci. Phys. Math.*, 1989, **38**, 107–110 (in Russian).
11. Randvee, I. One-step memory closed-loop control of linear interconnected subsystems. *Proc. Estonian Acad. Sci. Phys. Math.*, 2000, **49**, 100–107.

Optimeerimistehnika kasutamine ennustava juhtimise modelleerimisel

Ingmar Randvee

Esitatakse ennustava juhtimise algoritmis sisalduvate optimeerimisülesannete lahendamise tehnika. See baseerub sihifunktsiooni dekomponeerimisel, mille tulemusel saadakse osaliselt tagasisidestatud regulaator. Regulaatori avatud ahelaga juhttoime komponenti täpsustatakse iteratiivselt, kasutades sihifunktsiooni parametriseerimist. Tehnika on rakendatav nii lineaarsete kui ka eralduvate muutujatega mittelineaarsete objektide puhul ning see on oluliselt lihtsam klassikalisest lahendusskeemist, kuna ei nõua kaasmuutujatega rajaülesannete lahendamist. See asjaolu võimaldab kasutada ennustavat juhtimist ka keerukamate objektide, sealhulgas mitme lokaalse regulaatoriga Nash-optimaalsete süsteemide korral.