



Generalized predictive control in the delta-domain

Lauritsen, Morten Bach; Jensen, Morten Rostgaard; Poulsen, Niels Kjølstad

Published in:
Proceedings of the American Control Conference

Publication date:
1995

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Lauritsen, M. B., Jensen, M. R., & Poulsen, N. K. (1995). Generalized predictive control in the delta-domain. In *Proceedings of the American Control Conference* (Vol. Volume 5, pp. 3709-3713). IEEE.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

GENERALIZED PREDICTIVE CONTROL IN THE DELTA-DOMAIN

Morten B. Lauritsen, Morten Rostgaard, Niels K. Poulsen
Institute of Mathematical Modelling

Technical University of Denmark
DK-2800 Lyngby, Denmark

mbl@imm.dtu.dk, mrj@imm.dtu.dk, nkp@imm.dtu.dk

Abstract

This paper describes new approaches to generalized predictive control formulated in the delta (δ) domain. A new δ -domain version of the continuous-time emulator-based predictor is presented. It produces the optimal estimate in the deterministic case whenever the predictor order is chosen greater than or equal to the number of future predicted samples, however a "good" estimate is usually obtained in a much longer range of samples. This is particularly advantageous at fast sampling rates where a "conventional" predictor is bound to become very computationally demanding. Two controllers are considered: one having a well-defined limit as the sampling period tends to zero, the other being a close approximation to the conventional discrete-time GPC. Both algorithms are discrete in nature and well-suited for adaptive control. The fact, that δ -domain model are used does not introduce an approximation since such a model could be obtained by an exact sampling of a continuous-time model.

1. Introduction

In recent years it has become widely accepted, that the shift operator is not well suited to describe sampled systems at fast sampling rates. It has been suggested to use the so-called delta-operator [1]

$$\delta = \frac{q-1}{T} \quad \text{or} \quad q = 1 + T\delta \quad (1)$$

where T is the sampling period and q is the forward shift-operator. Using system models parameterized by the δ -operator gives a closer relationship between the sampled system and the underlying continuous-time system at fast sampling rates. This is due to the fact, that for a signal $x(t)$

$$\lim_{T \rightarrow 0} \delta x(t) = \frac{d}{dt} x(t) \quad (2)$$

In accordance with this relationship we will call $\delta^n x(t)$ the n 'th order δ -derivative of $x(t)$. The fact, that δ -domain descriptions converge to their corresponding continuous-time descriptions as $T \rightarrow 0$ makes it possible to utilize physical a priori knowledge. Also non-minimum phase problems introduced by the sampling process can be handled more easily than in the q -domain. An important advantage in adaptive control systems is, that better numerical and convergence properties of least squares parameter estimators are obtained when formulated in the δ -domain, see [2].

In order to make use of the above mentioned advantages in adaptive control systems we need to formulate control algorithms based on δ -domain models. Examples of such control algorithms are given in e.g. [1, 3, 4]. In this paper δ -domain versions of generalized predictive control will be considered. Some of the work has been presented in [5].

The δ -operator offers the same flexibility (and restrictions) in modeling as the q -operator which makes it possible to transform any q -domain control algorithm to the δ -domain. A directly transformed version of the basic GPC algorithm, see [6], has been presented in [3] for systems on δ -domain ARMAX form. The calculations involve a reformulation of optimal output predictions from a q -domain formulation to a δ -domain equivalent. This is obtained by means of the two Diophantine equations

$$C(\delta)(1 + T\delta)^{k-1} = A(\delta)E_k(\delta) + F_k(\delta) \quad (3)$$

$$(1 + T\delta)B(\delta)E_k(\delta) = C(\delta)G_k(\delta) + H_k(\delta) \quad (4)$$

These Diophantine equations can be solved in a recursive manner, see e.g. [4]. However, they become singular when $T \rightarrow 0$. This is impossible to avoid when considering optimal (MV) prediction because this is inherently connected to the shift-operator, i.e. the output is predicted a number of samples ahead rather than at absolute time instants independent from the sampling period.

In this paper a new discrete-time predictor which overcomes this problem is presented. The predictor is based on ideas from the continuous-time emulator-based predictor presented in [7]. It is shown to have a certain optimality property in the deterministic case.

The algorithms presented here avoid the problems of constructing continuous-time estimators and implementing continuous-time control algorithms. The control algorithms are based on discrete-time δ -domain models possibly obtained using a discrete-time identification algorithm and are in a sense exact—they are not derived as simple approximations of continuous-time algorithms. The δ -domain models are also exact in the sense, that they should *not* be thought of as approximations of continuous-time models and are fully equivalent to q -domain models. The reason for spelling out these observations is that many people may think of the δ -operator as simply a tool for implementation of continuous-time algorithms. However, as will be shown, the nature of the new δ -domain emulator-based predictor makes it possible to construct algorithms that approximate both continuous-time algorithms with a well-defined limit as $T \rightarrow 0$ and discrete-time algorithms which unavoidably become unrealizable when $T \rightarrow 0$.

2. Emulator-based prediction in the δ -domain

As mentioned optimal prediction is inherently connected to the shift-operator and is therefore not feasible at fast sampling rates. To obtain a predictor which behaves well when $T \rightarrow 0$ we will start out by looking at the continuous-time emulator based predictor presented in [7]. In continuous time we can approximate a function $y(t)$ in the neighborhood of t by the Taylor-series expansion

$$y(t + \tau) \approx y(t) + \sum_{j=1}^{N_y} \frac{\tau^j}{j!} \frac{d^j}{dt^j} y(t) \quad (5)$$

or in a matrix notation

$$y(t + \tau) \approx \mathbf{T}_{N_y}(\tau) \left[y(t) \frac{d}{dt} y(t) \dots \frac{d^{N_y}}{dt^{N_y}} y(t) \right]^T \quad (6)$$

where

$$\mathbf{T}_{N_y}(\tau) = \begin{bmatrix} 1 & \tau & \frac{\tau^2}{2!} & \dots & \frac{\tau^{N_y}}{N_y!} \end{bmatrix} \quad (7)$$

This expression can be used to predict $y(t + \tau)$ for positive τ assuming knowledge of the derivatives of $y(t)$. Based on (2) a δ -domain approximation of the above may be obtained simply by replacing d/dt with δ . However, as knowledge of N_y δ -derivatives would be assumed we may do better by utilizing all these δ -derivatives in the approximation of each time-derivative. Denote the transform variables associated with δ and q by $\gamma \in \mathbb{C}$ and $z \in \mathbb{C}$, respectively. Knowing, that $z = \exp(sT)$ where s is the complex Laplace-operator it is not difficult to see, that

$$s = \frac{1}{T} \ln(1 + \gamma T) \quad (8)$$

Using Taylor-series expansion now gives

$$s = \gamma - T \frac{\gamma^2}{2} + T^2 \frac{\gamma^3}{3} - + \dots = \sum_{i=1}^{\infty} v_{1,i} \gamma^i \quad (9)$$

By raising the above expression in the j 'th power it is obvious, that s^j can be expressed in terms of $\gamma^j, \gamma^{j+1}, \dots$

$$s^j = \left(\sum_{i=1}^{\infty} v_{1,i} \gamma^i \right)^j = \sum_{i=j}^{\infty} v_{j,i} \gamma^i \quad (10)$$

where the coefficients $v_{j,i}$ can be determined from $v_{1,1}, \dots, v_{1,i}$ by repeated convolution of (9). By breaking of all these expansions at the N_y 'th power of γ we end up with the following matrix formula for approximation of s, s^2, \dots, s^{N_y}

$$\begin{bmatrix} s^0 \\ s \\ s^2 \\ \vdots \\ s^{N_y} \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & v_{1,1} & v_{1,2} & \dots & v_{1,N_y} \\ 0 & 0 & v_{2,2} & \dots & v_{2,N_y} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & v_{N_y,N_y} \end{bmatrix} \begin{bmatrix} \gamma^0 \\ \gamma \\ \gamma^2 \\ \vdots \\ \gamma^{N_y} \end{bmatrix} \quad (11)$$

or

$$\mathbf{S} \approx \mathbf{M}_{s\gamma}(N_y) \mathbf{\Gamma} \quad (12)$$

with obvious definitions of \mathbf{S} and $\mathbf{\Gamma}$. The matrix $\mathbf{M}_{s\gamma}(N_y)$ is upper-triangular with unit diagonal elements and hence of full rank. Combining the time-domain version of the above with (5) yields

$$y(t + \tau) \approx y(t) + \sum_{j=1}^{N_y} \sum_{i=j}^{N_y} \frac{\tau^j}{j!} v_{j,i} \delta^i y(t) \quad (13)$$

or with $y(t, \tau)$ being the approximation of $y(t + \tau)$

$$y(t, \tau) = \mathbf{T}_{N_y}(\tau) \mathbf{M}_{s\gamma}(N_y) \mathbf{y} \quad (14)$$

where \mathbf{y} contains the δ -derivatives of $y(t)$

$$\mathbf{y} = [y(t) \delta y(t) \delta^2 y(t) \dots \delta^{N_y} y(t)]^T \quad (15)$$

An interesting feature of this improved predictor is shown in the following theorem.

Theorem 1 Let k denote a positive integer. Then

$$\mathbf{T}_{N_y}(kT) \mathbf{M}_{s\gamma}(N_y) \mathbf{\Gamma} = (1 + \gamma T)^k \quad (16)$$

if the predictor order is $N_y \geq k$.

Proof: Using a Taylor-series expansion for $\exp(\cdot)$ we can write

$$(1 + \gamma T)^k = \exp(k \ln(1 + \gamma T)) \quad (17)$$

$$= 1 + \sum_{i=1}^{\infty} \frac{(kT)^i}{i!} \left(\frac{1}{T^i} \ln^i(1 + \gamma T) \right) \quad (18)$$

or in a matrix notation

$$(1 + \gamma T)^k = \quad (19)$$

$$\begin{bmatrix} 1 & kT & \frac{1}{2!} (kT)^2 & \dots \end{bmatrix} \begin{bmatrix} 1 \\ \gamma - T \frac{\gamma^2}{2} + T^2 \frac{\gamma^3}{3} - + \dots \\ \left(\gamma - T \frac{\gamma^2}{2} + T^2 \frac{\gamma^3}{3} - + \dots \right)^2 \\ \vdots \end{bmatrix}$$

where Taylor-series expansions of $\ln^i(1 + \gamma T)$ have been inserted according to (9), (10). By comparing coefficients on both sides of the above equation it is easy to see, that all terms on the righthand side including γ in a power greater than k must cancel each other. The proof follows from a further comparison with (11) which shows that $\mathbf{M}_{s\gamma}(N_y) \mathbf{\Gamma}$ includes exactly all terms of order up to N_y in the righthand side of the above equation. ■

The implication of the theorem is as follows. Application to (14) gives after a few simple manipulations

$$y(t, kT) = (1 + \delta T)^k y(t) = q^k y(t) = y(t + kT) \quad (20)$$

i.e. if perfect knowledge of the δ -derivatives of $y(t)$ is available (the deterministic case), then the predictor is exact at sampling instants when the predictor order is greater than or equal to the desired number of future predicted samples. Hence, in this case no information is wasted. However, the

predictor can still be used to predict inter-sample behavior and to extrapolate by predicting more than N_y samples ahead which will be exemplified later.

The δ -derivatives in (14) are unknown at time t (the δ -operator is anti-causal) and must therefore be estimated. This is achieved by a δ -domain version of time-derivative emulation. Consider the δ -domain ARMAX model

$$A(\delta)y(t) = B(\delta)u(t) + C(\delta)e(t) \quad (21)$$

where $\deg(A) = n$, $\deg(B) = m < n$ and $\deg(C) = n$, $C(\delta)$ has all roots inside the stability circle and $e(t)$ is a white noise signal. The system delay in samples is $d = n - m > 0$. The control criteria use a filtered version of the output:

$$\zeta(t) = \frac{P_n(\delta)}{P_d(\delta)}y(t) \quad (22)$$

where $\deg(P_n) \geq \deg(P_d)$. Introduce the notation

$$\zeta_k(t) = \delta^k \zeta(t) \quad (23)$$

By means of the two coupled Diophantine equations (suppressed arguments)

$$\delta^k P_n C = E_k P_d A + F_k \quad (24)$$

$$E_k B = G_k C + H_k \quad (25)$$

with the order restrictions $\deg(E_k) = k + n_{P_n} - n_{P_d}$, $\deg(F_k) = n_{P_d} + n - 1$, $\deg(G_k) = k + n_{P_n} - n_{P_d}$ and $\deg(H_k) = n - 1$ we can write

$$\zeta_k(t) = \underbrace{G_k u(t)}_{\text{future}} + \underbrace{\frac{H_k}{C} u(t) + \frac{F_k}{P_d C} y(t)}_{\text{past}} + \underbrace{E_k e(t)}_{\text{future}} \quad (26)$$

The emulation of $\zeta_k(t)$ is now achieved by taking expectation conditioned on information obtained at time $t - 1$. This gives the emulation

$$\zeta_k^*(t) = G_k u(t) + \zeta^0(t) \quad (27)$$

where

$$\zeta^0(t) = \frac{H_k}{C} u(t) + \frac{F_k}{P_d C} y(t) \quad (28)$$

In a matrix notation we can write

$$\zeta^* = \tilde{G} \tilde{u} + \zeta^0 \quad (29)$$

where

$$\zeta^* = [\zeta_0^*(t) \ \zeta_1(t) \ \dots \ \zeta_{N_y}(t)]^\top \quad (30)$$

$$\zeta^0 = [\zeta_0^0(t) \ \zeta_1^0(t) \ \dots \ \zeta_{N_y}^0(t)]^\top \quad (31)$$

$$\tilde{u} = [u(t) \ \delta u(t) \ \dots \ \delta^{(N_y + n_{P_n} - n_{P_d} - d)} u(t)]^\top \quad (32)$$

and row k of \tilde{G} is built up from the coefficients of G_k .

To reduce the dimension of \tilde{G} we impose the following constraint on the smoothness of the control strategy

$$\delta^k u(t) = 0 \quad \text{for } k > N_u \quad (33)$$

where the control order N_u satisfies

$$0 \leq N_u \leq N_y + n_{P_n} - n_{P_d} - d \quad (34)$$

By defining

$$u = [u(t) \ \delta u(t) \ \delta^2 u(t) \ \dots \ \delta^{N_u} u(t)]^\top \quad (35)$$

we can finally write the emulator based predictor as

$$\hat{\zeta}(t, \tau) = T_{N_y}(\tau) M_{s\gamma}(N_y) (G u + \zeta^0) \quad (36)$$

where G is obtained as the first $N_u + 1$ columns of \tilde{G} .

3. Control design

Two different criterion functions are considered

$$J_C = \int_{T_1}^{T_2} [\hat{\zeta}(t, \tau) - w(t, \tau)]^2 d\tau + \rho \int_0^{T_2 - T_1} [u(t, \tau)]^2 d\tau$$

$$J_Q = \sum_{i=N_1}^{N_2} [\hat{\zeta}(t, iT) - w(t, iT)]^2 + \rho \sum_{i=0}^{N_2 - N_1} [u(t, iT)]^2$$

The two resulting controllers will be referred to as the DCGPC (Delta-domain approximation of CGPC) and the DQGPC (Delta-domain approximation of q -domain GPC), respectively. Both criteria are subject to the constraint

$$\delta^k u(t) = 0 \quad \text{for } k > N_u \quad (37)$$

Using (36) and by writing the future reference and control signals as Taylor-series expansions similar to (14)

$$w(t, \tau) = T_{N_y}(\tau) M_{s\gamma}(N_y) w \quad (38)$$

$$u(t, \tau) = T_{N_u}(\tau) M_{s\gamma}(N_u) u \quad (39)$$

a unified criterion can be specified as

$$J_U = (G u + \zeta^0 - w)^\top W_y (G u + \zeta^0 - w) + \rho u^\top W_u u \quad (40)$$

where the weighting matrices are given by

$$W_y = \begin{cases} M_{s\gamma}^\top(N_y) \int_{T_1}^{T_2} T_{N_y}^\top(\tau) T_{N_y}(\tau) d\tau M_{s\gamma}(N_y) \\ M_{s\gamma}^\top(N_y) \sum_{i=N_1}^{N_2} T_{N_y}^\top(iT) T_{N_y}(iT) M_{s\gamma}(N_y) \end{cases}$$

$$W_u = \begin{cases} M_{s\gamma}^\top(N_u) \int_0^{T_2 - T_1} T_{N_u}^\top(\tau) T_{N_u}(\tau) d\tau M_{s\gamma}(N_u) \\ M_{s\gamma}^\top(N_u) \sum_{i=0}^{N_2 - N_1} T_{N_u}^\top(iT) T_{N_u}(iT) M_{s\gamma}(N_u) \end{cases}$$

with uppermost entry belonging to J_C in both cases. Notice the close analogy between the DCGPC case and the DQGPC case. It can be shown, that W_u is positive definite for $T > 0$ and $T_2 - T_1 > 0$ respectively $N_2 - N_1 \geq N_u$. The vector w is defined by

$$w = [w(t) \ \delta w(t) \ \delta^2 w(t) \ \dots \ \delta^{N_y} w(t)]^\top \quad (41)$$

Minimization of the unified quadratic criterion results in the optimal control vector

$$\begin{aligned} \mathbf{u} &= \left(\mathbf{G}^\top \mathbf{W}_y \mathbf{G} + \rho \mathbf{W}_u \right)^{-1} \mathbf{G}^\top \mathbf{W}_y (\mathbf{w} - \zeta^0) \\ &= \mathbf{K} (\mathbf{w} - \zeta^0) \end{aligned} \quad (42)$$

Note, that in the case $N_u = 0$ the matrix to be inverted is a scalar, which is highly attractive from a computational point of view. The controller is implemented following a receding horizon strategy and hence only the first element of \mathbf{u} need to be calculated:

$$u(t) = k^\top (\mathbf{w} - \zeta^0) \quad (43)$$

where $k^\top = [k_0 \ k_1 \ \dots \ k_{N_y}]$ is the first row of \mathbf{K} . The controllers can easily be rewritten to polynomial form

$$u(t) = \frac{Q(\delta)}{R(\delta)} w(t) - \frac{S(\delta)}{R(\delta)} y(t) \quad (44)$$

Using the results from the previous section and (42) it can be shown, that

$$R(\delta) = P_d(\delta) \left(C(\delta) + \sum_{i=0}^{N_y} k_i H_i(\delta) \right) \quad (45)$$

$$Q(\delta) = C(\delta) P_d(\delta) K(\delta) \quad (46)$$

$$S(\delta) = \sum_{i=0}^{N_y} k_i F_i(\delta) \quad (47)$$

with $\deg(R) = n_{P_d} + n$, $\deg(Q) = n_{P_d} + n + N_y$ and $\deg(S) = n_{P_d} + n - 1$. Note, that the transfer function $S(\delta)/R(\delta)$ is strictly proper which means, that the last value of the output $y(t)$ is not fed back. Note also, that the transfer function $Q(\delta)/R(\delta)$ is anti-causal, i.e. knowledge of future δ -derivatives of the reference signal $w(t)$ is necessary. While this information may be accessible in some applications, e.g. robotics, it may in other applications be necessary to make some assumptions on the future reference. A common assumption is constant future reference which can be obtained by choosing

$$\mathbf{w} = [w(t) \ 0 \ \dots \ 0]^\top \quad (48)$$

In this case the $Q(\delta)$ polynomial becomes

$$Q(\delta) = k_0 C(\delta) P_d(\delta) \quad (49)$$

with $\deg(Q) = n_{P_d} + n$ so that the transfer function $Q(\delta)/R(\delta)$ becomes exactly proper.

4. Simulation examples

4.1. The δ -domain emulator based predictor

To illustrate the prediction facilities of the new δ -domain emulator-based predictor some experiments have been performed on the third order stable system

$$\frac{B(s)}{A(s)} = \frac{3}{(s+3)(s^2+s+1)} \quad (50)$$

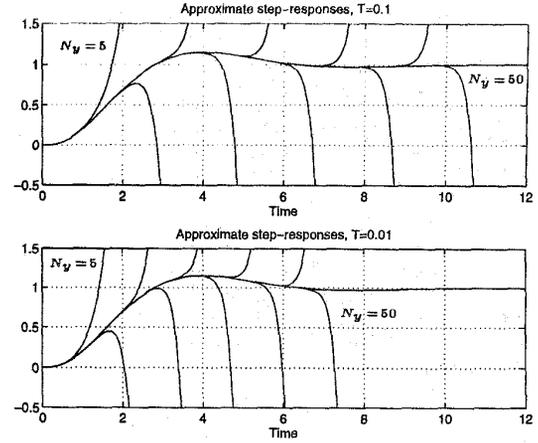


Figure 1: True and approximate step-responses with third order system and varying $N_y = 5, \dots, 50$ in steps of 5, $T = 0.1$ (top) and $T = 0.01$ (bottom).

As argued in [7] the precision of the predictor can be examined by comparing the true and approximate (predicted) step-responses of the open-loop system. In Figure 1 these responses are depicted for various predictor orders $5 \leq N_y \leq 50$ over the time-range $0 \leq t \leq 12$ with sampling periods of $T = 0.1$ and $T = 0.01$.

Clearly, the emulator-based predictor is in both cases capable of predicting the output much further than the "guaranteed" horizon $N_y T$, cf. Theorem 1. We note, that perhaps somewhat surprising the effective prediction horizon decreases when T is reduced. A possible explanation is, that $N_y \delta$ -derivatives contains information of the output in a longer absolute time range when T is large.

4.2. The DCGPC controller

To show, that the sampling period has very little influence on the closed-loop DCGPC poles consider the 3. order non-minimum phase system

$$\frac{B(s)}{A(s)} = \frac{-0.2s+1}{s(s^2+1)} \quad (51)$$

controlled by the DCGPC controller with design parameters

$$\begin{aligned} T_1 = 0, \quad T_2 = 5, \quad T = 0.0001, \dots, 0.5 \\ N_y = 30, \quad N_u = 3, \quad \rho = 0, \quad P_n/P_d = 1 \end{aligned} \quad (52)$$

The predictor order N_y has been chosen quite high so as not to affect the pole locations. The pole-locations are plotted in Figure 2. The poles have been converted to the s -domain using (8) since the δ -domain stability region depends on T which prevents a direct comparison of the δ -domain poles.

The figure shows, that the poles vary very little with the sampling period. Hence, even with quite large sampling periods the DCGPC controller is very close to the continuous-time limit obtained when $T \rightarrow 0$.

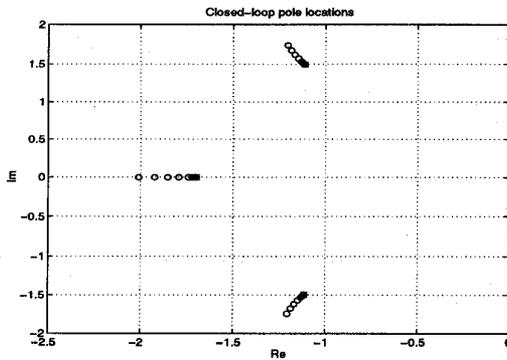


Figure 2: Closed-loop DCGPC poles (s -domain) for varying T .

4.3. The DQGPC controller

Consider the basic GPC criterion

$$J_{GPC} = E \left\{ \sum_{i=N_1}^{N_2} [y(t+iT) - w(t+iT)]^2 \right\} \quad (53)$$

$$= \text{const} + \sum_{i=N_1}^{N_2} [\hat{y}(t+iT) - w(t+iT)]^2 \quad (54)$$

where control weighting has been omitted and $\hat{y}(t+iT)$ is the MV-prediction of $y(t+iT)$ given information up to time t . The criterion is subject to the constraint

$$u(t+i) = u(t), \quad i > N_u \quad (55)$$

When considering this positional version of the GPC based on an ARMAX model it is clear, that the DQGPC approximates this controller when $N_u = 0$ and $P_n/P_d = 1$. To illustrate this some control designs have been made with the system (50) and design parameters

$$N_1 = 1, N_2 T = 10, T = 0.01, \dots, 0.5 \\ N_y = 10, N_u = 0, \rho = 0, P_n/P_d = 1 \quad (56)$$

Note, that the maximum absolute prediction horizon $N_2 T$ is kept constant so that N_2 varies with T . The closed-loop poles (transformed to the s -domain) for both the exact GPC and the approximating DQGPC are shown in Table 1.

The table shows, that when $N_y \geq N_2$ the two controllers are identical with the given precision. When $N_2 > N_y$ a very small discrepancy between the exact GPC and the DQGPC appears. However, the DQGPC is less computationally demanding since for instance at $T = 0.01$ only 10 δ -derivatives need to be emulated in the DQGPC whereas the exact GPC needs to predict 100 samples ahead. The extrapolating ability of the new δ -domain emulator-based predictor clearly gives a potential computational advantage when N_2 is large.

5. Conclusion

In this paper a new predictor based on discrete-time δ -domain models has been presented. The predictor uses ideas from the continuous-time emulator-based predictor

T	N_2	Exact GPC
0.01	100	$-3.4572, -1.6502 \pm 2.1329 i$
0.05	20	$-3.4707, -1.6507 \pm 2.1498 i$
0.1	10	$-3.4920, -1.6526 \pm 2.1768 i$
0.2	5	$-3.5529, -1.6590 \pm 2.2516 i$
0.5	2	$-3.9929, -1.6759 \pm 2.7130 i$
T	N_2	DQGPC
0.01	100	$-3.4595, -1.6506 \pm 2.1325 i$
0.05	20	$-3.4706, -1.6507 \pm 2.1498 i$
0.1	10	$-3.4920, -1.6526 \pm 2.1768 i$
0.2	5	$-3.5529, -1.6590 \pm 2.2516 i$
0.5	2	$-3.9929, -1.6759 \pm 2.7130 i$

Table 1: Closed-loop DQGPC and GPC poles (s -domain) for various sampling periods.

presented in [7]. The δ -derivatives are estimated in a way that closely resembles emulation of time-derivatives. The new predictor was in the deterministic case shown to be exact at sampling instants if the predictor order is chosen higher than or equal to the number of future predicted samples. In addition it has an ability to extrapolate which may give computational savings, as shown in the simulation examples.

Also two new control algorithms have been presented although treated in a unified manner. Simulations indicate, that the DCGPC algorithm closely approximates its continuous-time equivalent almost independently from the sampling period. Furthermore, the DQGPC algorithm has been shown to be closely related to usual q -domain GPC's based on optimal prediction. The DQGPC algorithm potentially offers large computational savings if the maximum cost horizon N_2 is large at the cost of very small deviations from the optimal solution.

References

- [1] Richard H. Middleton and Graham C. Goodwin. *Digital Control and Estimation: A Unified Approach*. Prentice Hall, Englewood Cliffs NJ, USA, 1990.
- [2] Michel Gevers and Gang Li. *Parametrizations in Control, Estimation and Filtering Problems*. Springer-Verlag, London, 1993.
- [3] Morten Rostgaard. *Modelling, Estimation and Control of Fast Sampled Dynamical Systems*. PhD thesis, Institute of Mathematical Modelling, Technical University of Denmark, 1995.
- [4] M. Rostgaard, N. K. Poulsen, and O. Ravn. General predictive control using the delta operator. In *Proceedings of 32nd IEEE Conference on Decision and Control*, pages 1769–1775, Texas, USA, 1993.
- [5] Morten Bach Lauritsen. Robust adaptive control. Master's thesis, IMSOR, Technical University of Denmark, 1994.
- [6] D. W. Clarke, C. Mohtadi, and P. S. Tuffs. Generalized predictive control—part i: The basic algorithm. *Automatica*, 23(2):137–148, 1987.
- [7] H. Demircioğlu and P. J. Gawthrop. Continuous-time generalized predictive control (CGPC). *Automatica*, 27(1):55–74, 1991.