Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Process Control 16 (2006) 255–264
JOURNAL OF PROCESS CONTROL
www.elsevier.com/locate/jprocont

Towards embedded model predictive control for System-on-a-Chip applications

Leonidas G. Bleris [a], Jesus Garcia [b], Mayuresh V. Kothare [c,*], Mark G. Arnold [b]

[a] Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18015, USA
[b] Department of Computer Science and Engineering, Lehigh University, Bethlehem, PA 18015, USA
[c] Department of Chemical Engineering, Lehigh University, Bethlehem, PA 18015, USA

Abstract

We propose a framework for embedding model predictive control for Systems-on-a-Chip applications. In order to allow the implementation of such a computationally expensive controller on chip, we propose reducing the precision of the microprocessor to the minimum while maintaining near optimal control performance. Taking advantage of the low precision, a logarithmic number system based microprocessor architecture is used, that allows the design of a reduced size processor, providing further energy and computational cost savings. The design parameters for this high-performance embedded controller are chosen using a combination of finite element method simulations and bit-accurate hardware emulations in a number of parametric tests. We provide the methodology for choosing the design parameters for two particular control problems; the temperature regulation in a wafer cross-section geometry, and the control of temperature in a non-isothermal fluid flow problem in a microdevice. Finally, we provide the microprocessor architecture details and estimates for the performance of the resulting embedded model predictive controller.

Keywords: Embedded model predictive control; Reduced precision microprocessors; Systems-on-a-Chip; Microchemical systems

1. Introduction

The realization of autonomous Systems-on-a-Chip (SoC) applications has been a very significant research area during the past decade. Moving on from simple prototype applications, new generations of more complex SoC are currently being developed for a variety of applications. The market for SoC is expanding [14] rapidly to areas like medicine and bioengineering (DNA and genetic code analysis and synthesis, drug delivery, diagnostics and imaging), information technologies, avionics and aerospace (nano and microscale actuators and sensors, smart reconfigurable geometry wings and blades, microgyroscopes), automotive systems and transportation (accelerometers), microreactors for in situ and on-demand chemical production. The functionality and performance of such a versatile SoC is directly related to the reliability and quality of the control logic used. For example processing DNA molecules at the microscale requires the fabrication of microfluidic devices capable of handling, mixing, thermal cycling and separating liquid microsamples. Another example is the application of closed-loop feedback control in medical devices. Recent advances in microelectronic technologies coupled with a better understanding of medical/physiological models have created an increased interest for the development of medical devices that can sense, think (carry out advanced algorithms) and act. There are numerous potential applications including: diabetes control, cardiac pacemakers and defibrillators, human

* Corresponding author. Tel.: +1 610 758 6654; fax: +1 610 758 5057.
E-mail address: mayuresh.kothare@lehigh.edu (M.V. Kothare).

0959-1524/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2005.06.006

immunodeficiency virus (HIV) control, anesthesia control, etc. Therefore efficient custom-made controllers have to be designed and integrated on chip.

In this paper we provide a framework for applying Model predictive control (MPC) and designing the microcontroller architecture that will allow optimal sensing–control–actuation performance for microchemical system applications [13,12]. Microchemical systems are a new generation of miniature chemical systems that carry out chemical reactions and separations in precisely fabricated three dimensional microreactor configurations in the size range of a few 100 s of microns. Our research effort is centered on eventually developing and embedding a controller in a catalytic reformer and separator microchemical system that can operate as a sustained source of hydrogen fuel for proton exchange membrane (PEM) fuel cells. This SoC will be used potentially as an alternative to conventional portable sources of electricity such as batteries for laptop computers and mobile phones due to its ability to provide an uninterrupted supply of electricity as long as a supply of methanol, water and heat can be provided.

The application of real-time embedded model predictive control for constrained systems with fast dynamics presents new technological challenges. Some initial research results have been reported in this direction. A work that addresses the issue of finding the programming procedure that results in the fastest implementation of the core calculations of model predictive control algorithms, amenable to parallel processing on a real-time multiprocessing system is provided in [9]. The combination of the new technology of field-programmable analog Arrays (FPAAs) with the more commonly used technology of field-programmable gate arrays (FPGAs) was proposed in [15], for the development of dynamically reconfigurable analog/digital hardware capable of handling MPC computation requirements. Some preliminary results on dynamic scheduling of model predictive controllers are provided in [11]. In this paper a feedback scheduling strategy for multiple MPCs is proposed, where the scheduler allocates CPU time to the tasks according to the current values of the cost functions. Since the MPC algorithm is iterative, the feedback scheduler may also abort a task prematurely to avoid excessive input–output latency. More recently a multiparametric programming method was proposed [2,16] to solve off-line the quadratic optimization problem associated with MPC. The constrained quadratic optimization is shown to be piecewise affine, using partitions of the state space determined by the constraints. The feedback laws are pre-computed and online calculations consist of table-lookup in memory and affine transformations. However, the upper bound [2] on the number of required regions is $N_r \leqslant \sum_{k=0}^{2^q-1} k! q^k$, where q is the number of constraints. For an input constrained problem, increasing the number of controlled variables (thus the number of constraints), yields prohibitive memory requirements for small size hardware implementations. Furthermore, the main result of the paper does not extend to non-quadratic objective functions in MPC.

Alternatively, we propose reducing the precision of a microprocessor to the minimum required by a particular application, while maintaining close to optimal control performance. Taking advantage of the low precision, a logarithmic number system (LNS) based microprocessor architecture is used, providing energy and computational cost savings. By using lower precision we are increasing the optimization speed while reducing the power consumption and the overall chip area. The power consumption in the arithmetic units is related to the number of switching transistors. Therefore by cutting down the transistor count we expect not only a lower cost for the embedded processor but also a measurable drop in the consumed power. A final advantage of reducing the size of the processor is that we can integrate the memory on the central processing unit.

The paper is organized as follows. In Section 2 we provide an introduction to model predictive control and the logarithmic number system arithmetic. The control problems of regulation of the temperature profiles across a wafer geometry and the temperature regulation of a non-isothermal microfluidic problem are provide in Section 3. In Section 4, we provide the Application Specific Instruction Processor (ASIP) architecture details and estimates on its performance. We conclude the paper with remarks on our presented research results and an analysis of unresolved issues that are currently under investigation.

2. Theoretical background

2.1. Model predictive control theory

Model predictive control [5] is also known as receding horizon control or moving horizon control. Generally controllers belonging to the MPC family are characterized by the following steps. Initially the future outputs are calculated at each sample interval over a predetermined horizon N, the prediction horizon, using the process model. These outputs $y(t+k|t)$ for $k = 1, \ldots, N$ depend up to the time t on the past inputs and on the future signals $u(t+k|t)$, $k = 0, \ldots, C-1$ which are those to be sent to the system. The next step is to calculate the set of future control moves by optimizing a determined criterion in order to keep the process as close as possible to a predefined reference trajectory. This criterion is usually (but not always) a quadratic function of the difference between the predicted output signal and the reference trajectory. In some cases the control moves $u(t+k|t)$ are included in the objective function in order

to minimize the control effort. Quadratic cost functions typically have the form:

$$J_P(k) = \sum_{k=0}^{P} \{[y(t+k|t) - y_{\text{ref}}]^2 + Ru(t+k|t)^2\} \quad (1)$$

$$|u(t+k|t)| \leqslant b, \quad k \geqslant 0 \quad (2)$$

where $y(t+k|t)$ are the predicted outputs, y_{ref} is the desired set reference output, $u(t+k|t)$ is the control sequence and R is a design parameter, the weighting on the control moves. The size of predicted outputs depends upon the choice of the prediction horizon N, and the size of the control sequence upon the control horizon C. The minimization of the objective in (1) is subject to input constraints given by the vector b.

The first control move $u(t|t)$, resulting from the optimization, is sent to the system while the rest are discarded. This is because at the next sampling instant the output is measured and the procedure is repeated with the new measured values so that we obtain past and current input values.

The basic model predictive control structure is given in Fig. 1. A model is used to predict the future plant outputs based on past and current values and on the calculated optimal future control moves. These actions are calculated by the optimization that also takes into account the constraints. There are many models used in different formulations of the model predictive control. One of the most popular is the step response model, which we can obtain from the measurement of the output when we excite the system with a step input. A similar model is the impulse response model, obtained by applying an impulse to the input. The transfer function model is another widely used model since it requires only a few parameters. Lastly, the state space model is also very useful since it can describe multivariable processes. In this paper we build the models using FEMLAB and we use the impulse responses to formulate the model predictive control problem.

2.2. Emulated optimization

The optimization is a fundamental part of MPC since it results in optimal control inputs for the process. The computational effort required in an embedded real-time MPC derives almost entirely from the optimization algorithm. Logically the choice of the optimization technique [7] is decisive to the performance of the controller. It has to be noted that from the architecture perspective the operations that are responsible for the majority of instructions that are executed in the processor are matrix inversions and abundant dot products. The algorithm that is implemented for the minimization of the cost function is a direct application of Newton's method. The constraints are incorporated in the cost function using barrier functions for the inequality constraints, and penalty functions for equality constraints; defined as

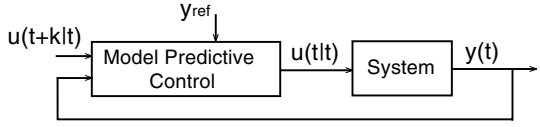

Fig. 1. Model predictive control block diagram.

$$d_i(u) = \mu_i(\mathbf{a}_i^{\mathrm{T}}u - \mathbf{b}_i)^2, \quad i \in E \quad (3)$$

$$d_i(u) = \mu_i \log(\mathbf{a}_i^{\mathrm{T}}u - \mathbf{b}_i), \quad i \in I \quad (4)$$

resulting in the unconstrained non-linear problem:

$$\underset{u}{\text{minimize}} \ f(u) = \frac{1}{2}u^{\mathrm{T}}\mathbf{G}u + \mathbf{g}^{\mathrm{T}}u + \sum_i d_i(u) \quad (5)$$

The problem of Eq. (5) can be solved numerically approximating $f(u)$ by a quadratic function around u, obtaining the gradient $\nabla f(u)$ and Hessian $\mathbf{H}(u)$, and iterating:

$$u^{(k+1)} = u^{(k)} - \mathbf{H}^{-1}(u^{(k)}) \cdot \nabla f(u^{(k)}) \quad (6)$$

The complexity of this algorithm is dominated by the computation of $\mathbf{H}^{-1}$, which requires $O(n^3)$ operations, where n is the number of variables in the optimization.

2.3. Logarithmic number system

One of the main advantages of floating point (FP) is its standardization, but for an architecture oriented towards embedded systems and therefore susceptible to being tailored for each particular application the need for novel approaches arises. The logarithmic number system represents the value of the real number X using a sign bit and the base-b logarithm of $|X|$, x, so that

$$x = \text{round}(\log_b|X|) \quad (7)$$

where x is a fixed-point signed binary number, consisting of K integer bits and F fraction bits [17]. The round( ) operation approximates the value of $\log_b|X|$ so that it is representable in $N = K + F$ bits. For the implementation we additionally need two sign bits; one for the exponent x and one for the real number X. The values for K and F define respectively the dynamic range (largest and closest to zero representable values), and the precision (distance between consecutive representable values) that are available.

The basic arithmetic operations can be computed as follows. Multiplication and division in LNS are implemented as simple fixed-point addition or subtraction, since for X, Y real positive numbers:

$$R_1 = XY \rightarrow \log_b(R_1) = \log_b(XY) \quad (8)$$

$$= \log_b(X) + \log_b(Y) = x + y = r_1 \quad (9)$$

$$R_2 = X/Y \rightarrow \log_b(R_2) = \log_b(X/Y) \quad (10)$$

$$= \log_b(X) - \log_b(Y) = x - y = r_2 \quad (11)$$

LNS is less effective for additions and subtractions. Leonelli realized around 1800 that

$$R_3 = X + Y \rightarrow \log_b(R_3) = \log_b\left(Y\left(1 + \frac{X}{Y}\right)\right) \tag{12}$$

$$R_4 = X - Y \rightarrow \log_b|R_4| = \log_b\left|Y\left(1 - \frac{X}{Y}\right)\right| \tag{13}$$

Defining $Z = X/Y$, and thus $z = x - y$, the functions $s_b(z)$ and $d_b(z)$ are defined as

$$\log_b(1 + Z) = \log_b(1 + b^z) = s_b(z) \tag{14}$$

$$\log_b|1 - Z| = \log_b|1 - b^z| = d_b(z) \tag{15}$$

and therefore:

$$r_3 = y + s_b(z) \tag{16}$$

$$r_4 = y + d_b(z) \tag{17}$$

This reduces the computation to performing a fixed-point subtraction $x - y$, evaluating a single-input (z) irrational function, $s_b(z)$ or $d_b(z)$, and performing a fixed-point addition of the result with one of the inputs, y. Both $s_b(z)$ and $d_b(z)$ can be stored in ROM tables. Techniques such as co-transformation [1], combined with multipartite tables [8] allow reducing the memory required for a given precision.

LNS provides a representation with numeric characteristics similar to those offered by FP [17]. The first reason for preferring LNS is the constant, as opposed to FP, relative error characteristic. FP gives higher worst-case error than LNS for the same word size. FP architectures can be scaled in a more efficient way for high precisions, but for low-to-moderate precisions (below 32 bits), LNS implementations offer an advantage in term of cost, power consumption and speed, that increases as the word size decreases.

3. Case studies

With the rapid development of computers and software tool capabilities, novel approaches can be used for the analysis of SoC. In order to choose appropriate design parameters for this high-performance embedded controller we utilize computational tools to simulate the plant and we emulate the microcontroller arithmetic operations. In this paper we use FEMLAB [6], a partial differential equation (PDE) solver. We use the ability of FEMLAB to export complex microgeometries designed under FEMLAB environment into MATLAB to analyze rigorously their dynamic behavior by solving problems such as heat transfer, convection–conduction and Navier–Stokes flows. Two control problems that are critical for the target microchemical system are provided. The control of the temperature distribution across a wafer geometry is initially examined. Secondly, we look into the problem of controlling the temperature of a fluid flowing in a microchannel and the flow velocity at the outlet of the microchannel by adjusting the inlet velocity and regulating the available distributed energy supply.

3.1. Control of temperature distribution across a wafer

Consider the geometry of Fig. 2(a) that illustrates a wafer with an array of resistive actuators placed on top. We examine the 2D cross-section (Fig. 2(b)) of this wafer. Nine actuators are placed on top and we measure the temperature of this geometry by placing nine sensors on the bottom of the wafer cross-section at the exact opposite location of the actuators. We are interested in heating and maintaining the temperature of the wafer at a predefined set-point, while constraining the heat supply. This is a significant control objective for our

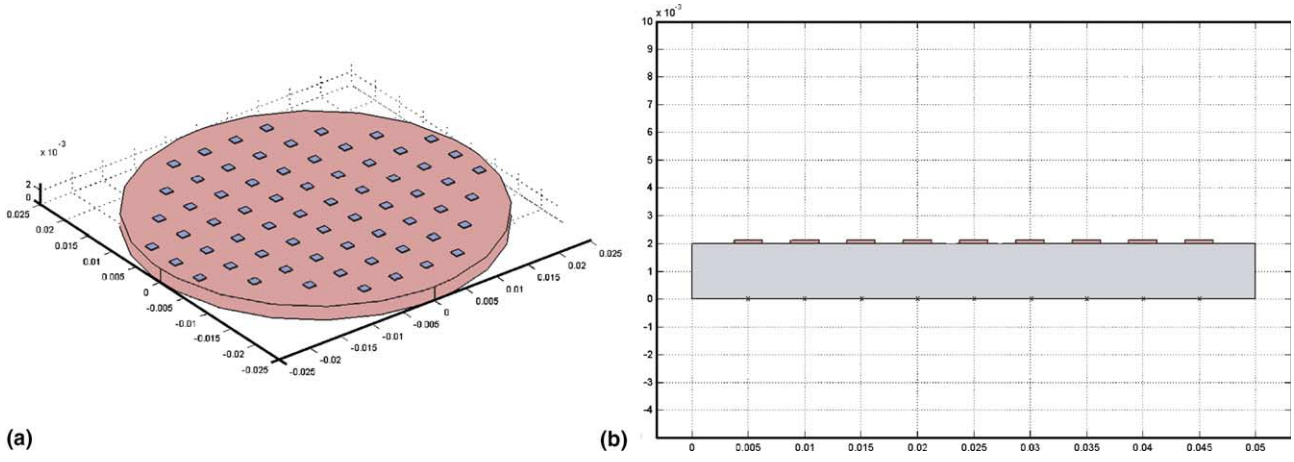

Fig. 2. (a) Wafer geometry and (b) cross-section geometry (dimensions in meters). The distributed heaters placed on top; indicated with × the locations of the distributed temperature sensors.

examined SoC application since the catalytic reformer and separator need to be heated to a specific temperature to ensure catalytic activation. Additionally the optimal working point for these two reactions is different; therefore, we need to maintain a temperature gradient within the wafer. The temperature distribution in the wafer is described by the time-dependent two-dimensional heat equation

$$\frac{\partial T(t,x,y)}{\partial t} = \frac{1}{a}\nabla^2 T(t,x,y) \tag{18}$$

with initial conditions $T(0,x,y) = 300$ K. Here $a = \rho C/\kappa$, κ is the thermal conductivity, ρ is the density and C is the heat capacity. The resistive actuator heat losses depend on the exposed surface $Q = hA(T - T_{\text{air}})$ where $T_{\text{air}} = 300$ K, the natural convection heat-transfer coefficient $h = 25$ W/m^2 K and A is the exposed heater area. With appropriate use of materials we have the ability [3] to control and maintain temperature gradients in the proposed geometry. The material used for the following FEM simulations is pyrex. We initially set the temperature set-point (T_{sp}) at the sensors placed at the boundaries at 370 K and for the rest at 390 K. The heat supply is constrained for each actuator at 5 W/m^2. We apply this problem on a 2D geometry in order to minimize to computational costs associated with the FEM simulations, but the proposed approach can be easily extended in 3D.

To estimate the performance of MPC using reduced precision, the basic arithmetic operations required by the optimization procedure were emulated using LNS. The design parameters of MPC, such as the weights on the barriers, the control moves, as well as the number representation K and F (integer and fractional part of the fixed-point exponents) were varied, allowing the selection of the desired compromise between precision reduction and response quality of the system. The control problem of reaching the set-point is solved initially using high precision; that is $K = 7$ (integer bits) and $F = 20$ (fraction bits) for different control horizons (C). The prediction horizon is fixed at 100 (with a sampling time of 1 s). We define the error of the controller performance using:

$$\text{Error} = \frac{\sum_{i=1}^{9} |T_{\text{sp}}(i) - T(i)|}{\sum_{i=1}^{9} (T_{\text{sp}}(i) - 300)} \times 100 \tag{19}$$

From Fig. 3 it is obvious that the performance of MPC is not satisfactory for $C = 4$ and it is closest to the set-point using $C = 8$. The total energy consumed using $C = 8$ was 567 J/m^2 while using $C = 6$ yields a 3.4% reduction. Additionally, since we are interested in minimizing the architectural complexity of the embedded controller we choose to use $C = 6$. In Fig. 4 we provide the simulation results. The objective now is to reduce further the implementation complexity, while maintaining a performance close to that of $K = 7$, $F = 20$ and $C = 6$.

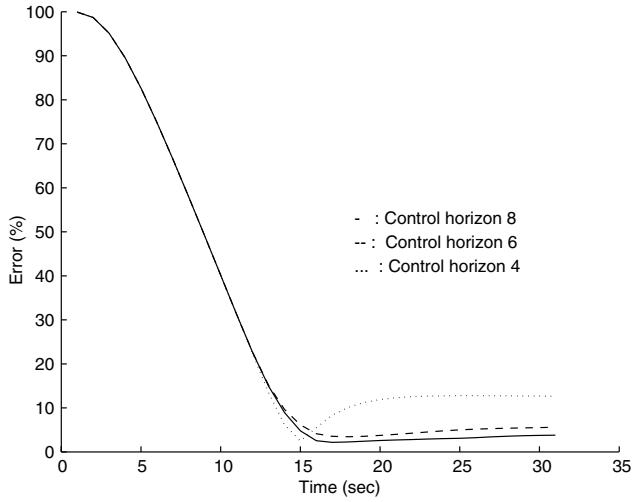

Fig. 3. Percentage of error using different control horizons.

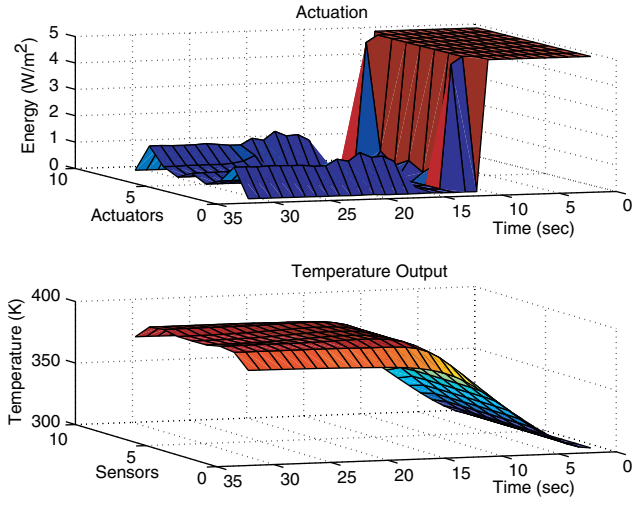

Fig. 4. Control variables and resulting temperature profile for $K = 7$, $F = 20$ and $C = 6$.

The next step for the proposed framework is to reduce the size of the integer part K. Results of the coupled system of the emulated model predictive microcontroller and the FEMLAB models show exactly the same performance for $K = 7, 6, 5$ and significantly worse for $K = 4$. Therefore $K = 5$ was chosen. Subsequently the fractional part F was gradually reduced and in Fig. 5 we provide the performance for two different cases. From Fig. 5 we notice that for $K = 5$, $F = 10$ and $C = 6$ the error increases significantly. This can be explained by the fact that the subtraction operation at low precisions can be inaccurate when the result is close to zero. As the plant output approaches the set-point, the values of the manipulated variables start oscillating (Fig. 6) to keep the output close to the set-point.

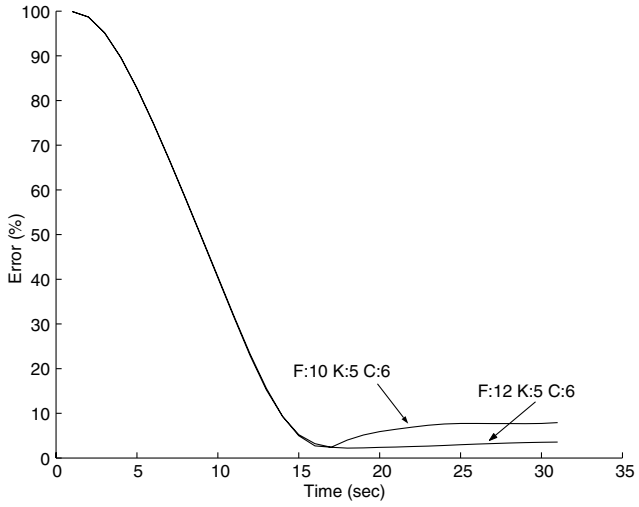

Fig. 5. Percentage of error reducing F (fraction bits).

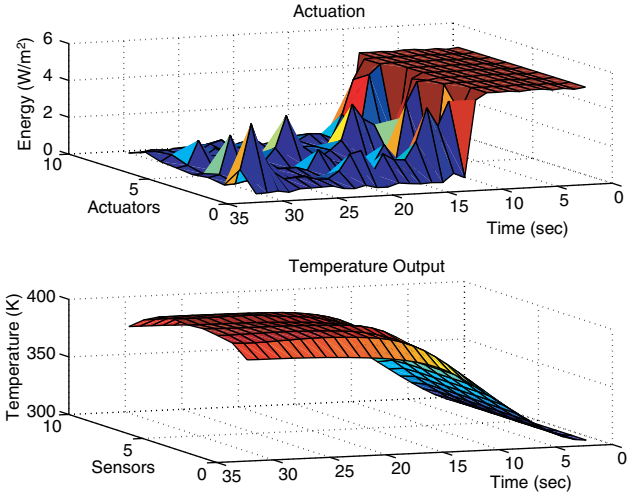

Fig. 6. Control variables and resulting temperature profile using $K = 5$, $F = 10$ and $C = 6$.

Furthermore this phenomenon creates added undesirable power consumption. To overcome this problem a hybrid MPC approach is under investigation, where the MPC switches to a state-space regulator close to the set-point. For the examined problem a decoupled proportional integral (PI) controller can also provide satisfactory results.

The preceding simulations show that for the examined problem, the minimum values for K and F were 5 and 12, respectively, which requires a 19-bit LNS representation after adding the two sign bits. We test this configuration using now a different set-point for the output temperature. The set-point is at 360 K for the first 4 sensors, the 5th sensor is free to take any value and the final 4 sensors at 400 K. From the simulation results of Fig. 7, we notice the successful performance of the proposed reduced-precision controller.

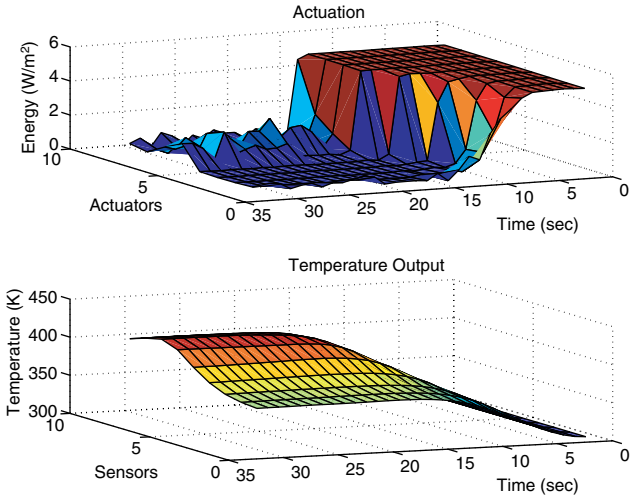

Fig. 7. Control variables and resulting temperature profile using $K = 5$, $F = 12$ and $C = 6$.

3.1.1. Comparison to a proportional integral controller

In this subsection we apply PI control for the temperature regulation across the wafer cross-section geometry. PI controllers are popular in dealing with control problems in microsystems; mostly due to the theoretical simplicity of the controller and the minimal implementation requirements. Due to the low thermal conductivity of pyrex we decouple the controller for each pair of sensor and actuator. The PI control consists of a term proportional to the error between the set-point and the state and a term proportional to the integral of this error, providing the update control variables (in our case the input heat supply E_i:

$$E_i = K_c(T_{\mathrm{sp}}(i) - T(i)) + K_i \int_0^t (T_{\mathrm{sp}}(i) - T(i)) \qquad (20)$$

where $i = 1, \ldots, M$ and M are the number of sensors/actuators. As illustrated in Fig. 8 the PI controller steers the temperature profile to the desired set-point within 35 s. The parameters K_c and K_i are both set at 0.01; a different tuning can result in faster response but with the undesirable temperature overshoot, and possibly violating the constraints. From the simulation results and from the error plot of Fig. 9 we notice that the use of a PI controller results in a slower response than that of MPC. Another drawback is that we made the restrictive assumption for most realistic control problems, that we can decouple the system for each pair of sensor–actuator.

3.2. Non-isothermal flow control problem

In this section we examine the coupled problem of microfluidic flow and convection–conduction in the 2D geometry of Fig. 10. The microchannel has depth of 3 mm and the arrows indicate the inlet, the outlet and

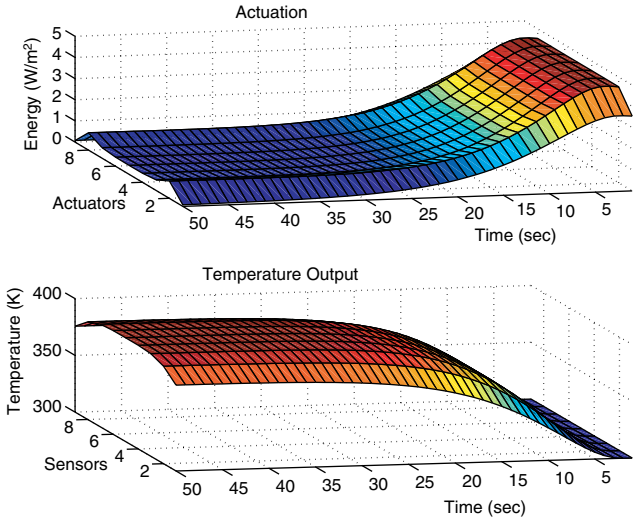

Fig. 8. Control variables and resulting temperature profile with PI control.

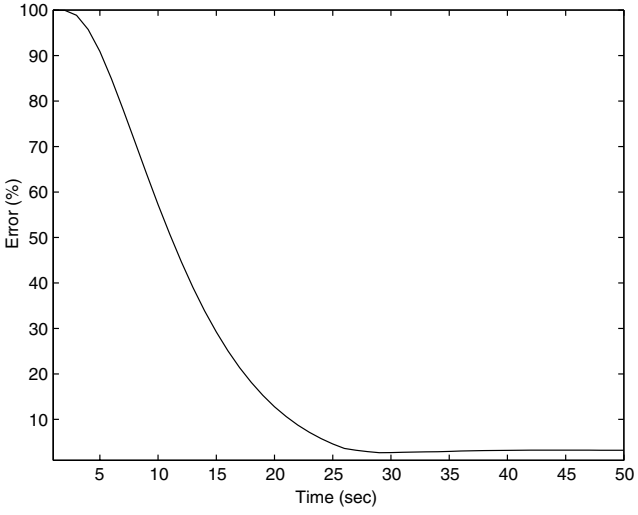

Fig. 9. Percentage of error using PI control.

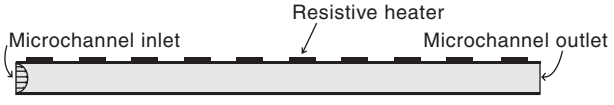

Fig. 10. Microchannel geometry.

the location of one of the 10 distributed resistive heaters. We measure the temperature by placing 10 sensors on the bottom of the geometry at the exact opposite location of the actuators.

The dynamic model of the flow can be described by the Navier–Stokes and the convection–conduction equations:

$$\rho\frac{\partial v}{\partial t} - \nabla\cdot\eta(\nabla v + (\nabla v)^{\mathrm{T}}) + \rho(v\cdot\nabla)v + \nabla p = 0 \qquad (21)$$

$$\nabla\cdot(\rho v) = 0 \qquad (22)$$

$$\rho C_p\frac{\partial T}{\partial t} + \nabla\cdot(-k\nabla T + \rho C_p T v) = 0 \qquad (23)$$

where k is the thermal conductivity and C_p the heat capacity. Furthermore we use the ideal gas law to relate the changes of temperature to density $\rho = \frac{pM}{RT}$, where R is the gas constant and M is the molar mass. For the Navier–Stokes equations we define the boundary conditions to be no-slip at the walls and straight out at the outlet. For the heat balance the temperature at the inlet is 300 K, at the top wall it is the energy provided by the resistive heaters, it is insulated at the bottom and at the outlet we assume that the transport is dominated by convection (the gradient of T perpendicular to the outlet is zero).

We are interested in heating and maintaining the temperature of the fluid to a predefined set-point, while constraining the heat supply. Moreover due to the changes of density, the velocity of the fluid increases upon heating [4]. We are therefore interested in controlling the velocity of the fluid at the outlet. Because of the fact that we have laminar flows in microscale (due to the low Reynolds number), we measure and control the maximum velocity which is at the middle of the channel. We solve the proposed control problem by decentralizing the system. Using proportional control the inlet velocity is adjusted to control the outlet velocity and MPC is applied for the regulation of the heat supply.

Because of the fact that the fluid enters the microchannel at 300 K, controlling the temperature profile becomes more important close to the inlet, especially for high velocities and elevated temperature set-point. We initially set the desired temperature at 350 K throughout the microchannel and the maximum velocity at the outlet at 2 mm/s. The simulation/emulations results of

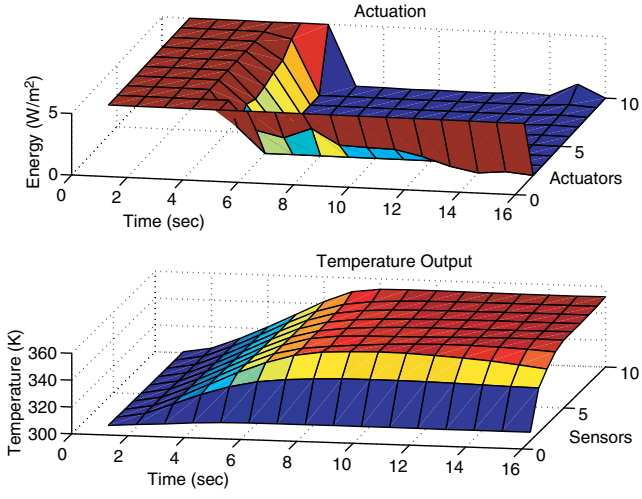

Fig. 11. Control variables and resulting temperature profile using $K = 7$, $F = 20$ and $C = 6$.

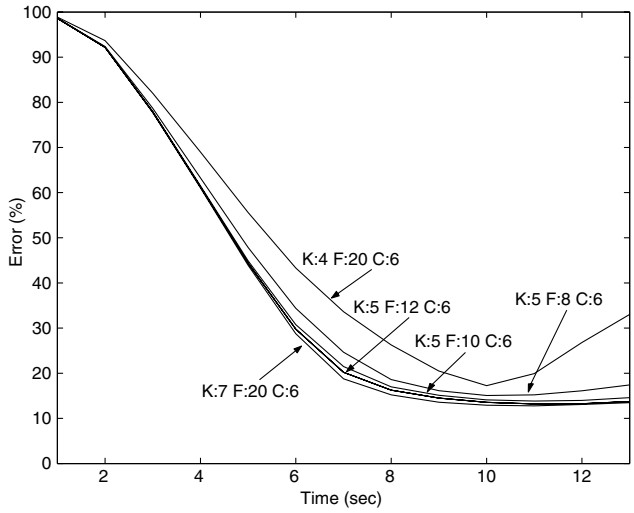

Fig. 12. Percentage of error using different design parameters.

applying MPC for controlling the temperature, with $K = 7$, $F = 20$, control horizon of 6 and prediction horizon 21, are illustrated in Fig. 11. We now simulate the system by gradually reducing the integer bits (K) and then the fraction bits (F) (Fig. 12). We are interested in calculating the minimum precision while maintaining the same control performance. From the simulation results it becomes apparent that we can reduce the controller size down to 17-bits without compromising the performance.

4. Proposed architecture of the ASIP

The requisites that an embedded MPC has to satisfy are: high speed, low power consumption, and low cost. The last two are related, since power consumption is related to the transistor count and therefore to the cost. The proposed architecture shown in Fig. 13 utilizes macrocell technology. Commercially available designs of macrocells such as microprocessor cores and embedded DRAM exist. By using LNS custom arithmetic units, realistic MPC problems can be solved online with a system that integrates every component on the same chip. Results in [8] show that LNS arithmetic units can be 50–30% smaller than equivalent FP units, for the precisions considered here. In Fig. 14 we have a comparison between the areas of the arithmetic units for LNS and FP, at different precisions.

Optimization is the dominant part in MPC algorithms; it requires a number of operations that grow with $O(n^3)$, where n is the product of the number of controlled variables times the control horizon. The optimization consists mainly of vectorial operations, that is matrix inversion and matrix–matrix multiplication (the most computationally intensive and responsible for the $O(n^3)$ complexity). Vector operations can be efficiently implemented using single-instruction multiple-data (SIMD) techniques. The most important aspects for a SIMD architecture are a fast pipeline in the arithmetic units and a high memory throughput. These are analyzed in the following paragraphs.

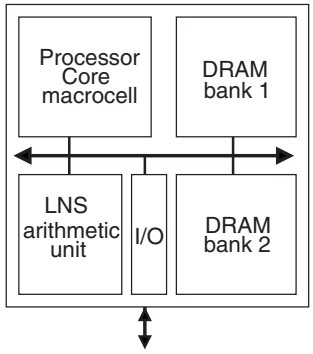

Fig. 13. Architecture block.

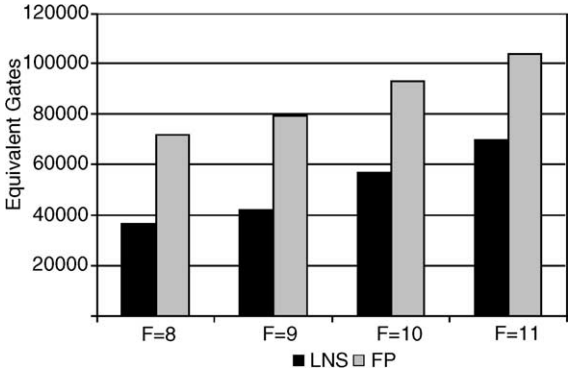

Fig. 14. Arithmetic logic unit (ALU) areas for LNS and FP at different precisions.

4.1. Pipelined data-path

From the data-path point of view, scalar–vector multiplication is very simple in LNS: a multiplier/divider is a simple fixed-point adder and the result sign logic. Vector–vector addition is more complicated, due to the larger latency of addition. However, LNS adder/subtracters can be efficiently pipelined. The latency will become slightly larger, but the throughput of a pipelined adder in a vector operation will be close to one operation/cycle (especially for large sized vectors). The very frequent dot products remain the main obstacle to achieve a throughput close to one operation/cycle for all computations. In Fig. 15 we illustrate the proposed pipelined data-path for all the above operations. Addition and subtraction are assumed to have a latency of 3 cycles; nevertheless the analysis that follows applies to any latency. A dot product issues operand pairs (from the operand buffers) at every cycle. The first pair is multiplied, and sent to the adder with the other input set to

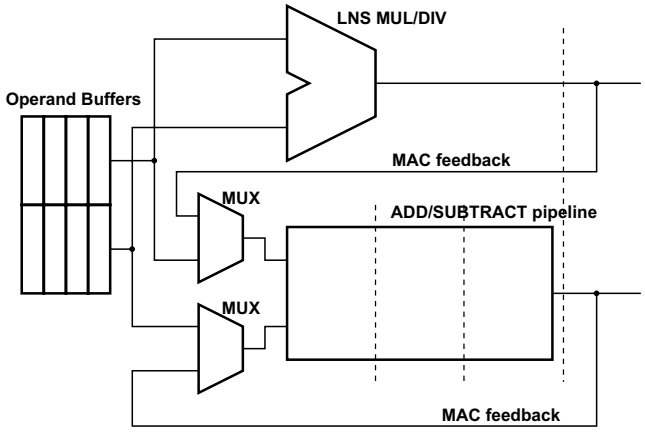

Fig. 15. Fully pipelined LNS MAC unit.

zero. This process is repeated for the second and third input pairs. When the fourth product is ready, the first addition (with zero, so it is simply the first product) is ready at the end of the addition pipeline. From this moment on, the results of the addition are routed as the second input to the adder, and three accumulations are carried on in parallel (in general, for a pipeline of p stages, p accumulations). As a result the pipeline remains fully occupied during the rest of the vector operation. In the end, the final p accumulated values have to be added together, requiring $p\lceil \log_2(p) \rceil$ extra cycles (aside of p cycles to fill the pipeline) to complete the Multiply and Accumulate (MAC) operation. For a dot product with vectors of length n, the overhead in the MAC unit is

$$\frac{p(1+\lceil \log_2(p) \rceil)}{n} \tag{24}$$

For an addition/subtraction pipeline of three stages, and a vector length of 100 operators, the overhead of this pipeline is 9%. Note that using FP, the MAC unit can be expected to have a slightly smaller overhead.

4.2. Performance of the ASIP and memory design

Potential exists for further increasing this throughput by replicating arithmetic units. However, with an estimated clock cycle at 5 ns, the proposed problem can be solved at sampling speeds as low as 0.032 s. To quantify the advantage of reducing the precision, estimations for both 64-bit FP and 16-bit LNS circuits show that for an arithmetic unit that computes addition, subtraction, multiplication and division, the size required is about 17 times larger for 64-bit FP, while the clock cycle is at least 3.23 times faster in 16-bit LNS. Having functional units that allow computing the majority of the operations with a throughput close to 1 operation/cycle, the other main obstacle to obtain high speed is an efficient memory access.

Fast access to the RAM is a major problem for most microprocessors [10]. This is addressed in general purpose processors using a small but fast memory cache; however this approach is not acceptable for a real-time application, due to the uncertainty that introduces in memory access time. Instead, we propose using a wide memory bus, which combined with the reduced size of data (almost 4-fold reduction compared with typical 64-bit floating point in MPC) allows high memory throughput. Since memory access is mainly sequential (reading and writing elements along a vector), paged memory access can be used. Another advantage of the reduced data size is the proportional reduction in data memory, which in the proposed case is close to 200 KBytes. Small and fast RAM technologies can be used, and even full integration of the RAM within the processor becomes possible.

5. Conclusions

A framework for embedding model predictive control logic on chip has been provided. Reducing the precision of the operations coupled with the use of a logarithmic number system arithmetic allows a very efficient implementation. We have emulated the performance of the reduced precision LNS-based MPC for two examined cases and we have provided brief microprocessor architecture details.

Future work includes obtaining theoretical bounds for convergence as a function of precision and formulating a detailed embedded system architecture. Also, we plan to study the incorporation of a hybrid logic in the controller to allow an early termination of the optimization algorithm so that further computational and energy savings are introduced.

Acknowledgements

Partial financial support for this research from the US National Science Foundation grant CTS-0134102 and the Pittsburgh Digital Greenhouse is gratefully acknowledged.

References

[1] M.G. Arnold, Improved cotransformation for LNS subtraction, IEEE International Symposium on Circuits and Systems 2 (May) (2002) 752–755.
[2] A. Bemporad, M. Morari, V. Dua, E.N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, Automatica 38 (1) (2002) 3–20.
[3] L.G. Bleris, M.V. Kothare, Model based control of temperature distribution in integrated microchemical systems, in: Proceedings of the 2003 American Control Conference, Denver, CO, June 2003, pp. 1308–1313.

[4] L.G. Bleris, M.V. Kothare, Proper orthogonal decomposition based control of fluid flow in microchemical systems, in: 2003 AIChE Annual Meeting, San Francisco, CA, November 2003.
[5] E.F. Camacho, C. Bordons, Model Predictive Control, Springer, New York, 1999.
[6] A.B. Comsol, M. A. Natick, FEMLAB Reference Manual, 2001.
[7] R. Fletcher, Practical Methods of Optimization, John Wiley & Sons, 1987.
[8] J.G. Garcia, M.G. Arnold, L.G. Bleris, M.V. Kothare, LNS architectures for embedded model predictive control processors, in: 2004 International Conference on Compilers, Architectures and Synthesis for Embedded Systems, Washington, DC, September 2004, pp. 79–84.
[9] G. Hassapis, Implementation of model predictive control using real-time multiprocessing computing, Microprocessors and Microsystems 27 (2003) 327–340.
[10] J.L. Hennessy, D.A. Patterson, Computer Architecture: A Quantitative Approach, Morgan Kaufmann, San Francisco, CA, 1996.
[11] D. Henriksson, A. Cervin, J. Akesson, K. Arzen, Feedback scheduling of model predictive controllers, in: Proceedings of the 8th IEEE Real-Time and Embedded Technology and Applications Symposium, San Jose, CA, September 2002.
[12] K.F. Jensen, Microchemical systems: status, challenges and opportunities, AIChE Journal 45 (10) (1999) 2051–2054.
[13] M.V. Kothare, A.V. Pattekar, K.A. Alfadhel, L.G. Bleris, S. Mukherjee, Microreactors for Efficient On-Chip Fuel Processing and Hydrogen Generation, in: Proceedings of the Conference on Nanofabrication: Technologies, Devices and Applications, Optics East, Philadelphia, PA, October 2004.
[14] S.E. Lyshevski, MEMS and NEMS, CRC Press, 2002.
[15] O.A. Palusinski, S. Vrudhula, L. Znamirowski, D. Humbert, Process control for microreactors, Chemical Engineering Progress (2001) 60–66.
[16] E.N. Pistikopoulos, On-line optimization via off-line optimization!—a guided tour to parametric programming and control, Invited Plenary Lecture at the 7th IFAC Symposium on Dynamics and Control of Process Systems (DYCOPS-7), Boston, MA, July 2004.
[17] E.E. Swartzlander, A.G. Alexopoulos, The sign/logarithm number system, IEEE Transactions on Computers 24 (12) (1975) 1238–1242.