Control Vector Parameterization with Sensitivity Based Refinement Applied to Baking Optimization
Proceedings of European Congress of Chemical Engineering (ECCE-6)
Copenhagen, 16-20 September 2007

Control Vector Parameterization with Sensitivity Based Refinement Applied to Baking Optimization

Hadiyanto,[a] D.C. Esveld[b,] R.M. Boom[b] , G. van Straten[a] and A.J.B. van Boxtel[a]

a. *Systems and Control Group, Wageningen University, PO BOX 17, 6700 AA Wageningen, The Netherlands: Email: hady.hadiyanto@gmail.com*
b. *Food Process Engineering Group, Wageningen University, PO BOX 8129, 6700 EV Wageningen, The Netherlands*

Abstract

In bakery production product quality attributes as crispness, brownness, crumb and water content are developed by the transformations that occur during baking and which are initiated by heating. A quality driven procedure requires process optimization to improve bakery production and to find operational procedures for new products. Control vector parameterization (CVP) is an effective method for the optimization procedure. However, for accurate optimization with a large number of parameters (representing the control vector), CVP optimization takes a long time for computation. In this work, an improved method for direct dynamic optimization using CVP is presented. The method uses a sensitivity based step size refinement for the selection of control input parameters. The optimization starts with a coarse discretization level for the control input in time. In successive iterations the step size was refined for the parameters for which the performance index has a sensitivity value above a threshold value. With this selection, optimization is continued for a selected group of input parameters while the other non sensitive parameters (below threshold) are kept constant. Increasing the threshold value lowers the computation time, however the obtained performance index becomes less. A threshold value in the range of 10-20% of the mean sensitivity satisfies well. The method gives a better solution for a lower computation effort than single run optimization with a large number of parameters or refinement procedures without selection.

Key words: Baking, optimal operation strategy, baking model, product quality, optimization.

1. Introduction

Quality driven process design can help to meet the challenges of the food industry to produce high quality products. Moreover, the approach also can create flexibility to produce a wide range of products. In quality driven process design, one starts with a process model, which describes the conversion process from ingredients and process conditions, to the product. Next, the product specifications are translated to an objective function. An optimization procedure is applied to find the required product treatment as a function of time and time dependent process conditions. Finally, the treatments are translated into processing equipment (Hadiyanto et al 2007b, Garcia et al 2006).

To solve the optimization problem for the processing time indirect and direct methods for dynamic optimization are available. Indirect methods are based on the calculus of variations and use adjoint variables. It follows from the calculus of variations that optimal conditions have been obtained when the derivative of the Hamiltonian with respect to the inputs equals to zero for any point at the input trajectories (Bryson and Ho, 1975). Therefore indirect methods require the computation of the gradient and a search for the control variables trajectories for which the gradient is zero. Betts and Huffman (1998) mentioned two main drawbacks for this approach. First, the necessary conditions for optimization have to be defined and for complicated nonlinear dynamic system this can be quite daunting task. Second, the region of convergence may be surprisingly small, especially when the adjoint variables values do not have a clear physical meaning.

For direct methods the dynamic optimization problem is transformed into a nonlinear programming problem. The main advantage is that there is no requirement to satisfy the necessary conditions for the Hamiltonian function or to use adjoint variables. The control variables are adjusted and optimize the objective function directly. A well known direct method parameterizes the input trajectory over the time interval; this approach is named control vector parameterization (CVP) (Betts and Huffman, 1998).

Both methods (indirect and direct) have been applied for baking processes (Hadiyanto et al, 2007b) . Optimization resulted in optimum heating trajectories which can be translated into design for unit operations. The direct method is based on a low discretization level of the control input for heating and cooling. As a consequence, the results obtained with the direct method were of less quality compared to that of the indirect method. Proper choice of the discretization level is a point of concern. Low numbers may not yield optimal results, while a high number mostly may end in local minima and an input trajectory with strong switching values (see Roubos et. al, 1999).

The computational time required for direct methods increases significantly with the number of parameters. In recent years a number of methods to reduce computational time of large-scale optimization problems were proposed; for example *refinement of control input* (Binder et al, 2000; Schlegel et al, 2005) or *successive re-optimization* (Garcia et al, 2006). These refinement methods start the optimization with a rough grid (a few parameters) and subsequently the grid is refined to increase the resolution of the control inputs (Binder et al, 2000 and Schlegel et al, 2005). After some refinement iterations a smooth grid is obtained for an acceptable computation time. In the approach of Garcia et al

(2005) the refinement is applied to all positions in the time grid by halving the step size from previous refinement iteration until the stopping criteria are fulfilled. However, it must be noted that not all parameters have significant effect on the improvement of the objective function. Therefore, one can reduce the necessary computation time by applying the step size refinement only at points in the time grid with enough sensitivity. This paper illustrates the use of the sensitivity based refinement method for the design of optimal baking operations.

2. Dynamic Optimization

2.1 Problem formulation

Dynamic optimization, also known as open loop optimal control, computes a set of control variables as a function of time that minimizes or maximizes an performance index. The performance index (J) composed from the terminal (Φ) and running cost (L) is optimized within the constraints of the process ($\dot{x} = f(t,x,u)$), and the defined lower and upper bounds of input variables (u):

$$\begin{aligned} &\min_{u,p,tf} J = \phi(x(t_f)) + \int L(x,t,u)\, dt \\ &st: \quad \dot{x} = f(t,x,u) \quad [t_o, t_f] \quad x(0) = x_0 \\ &\qquad u^L \le u \le u^U \qquad [t_o, t_f] \end{aligned} \tag{1}$$

2.2. Control Vector parameterization

For direct dynamic optimization the optimal control problem is transformed into a Non Linear Programming (NLP) problem. Control vector parameterization implies that the control input is discretized and approximated by a basis function with a limited number of parameters. The state variables in the process model remain in the form of continuous differential equations (Goh and Teo, 1988) (Figure 1). These differential equations are solved by forward integration and for the endpoint ($t=t_f$) the performance index is evaluated and optimized over a number of iterations.

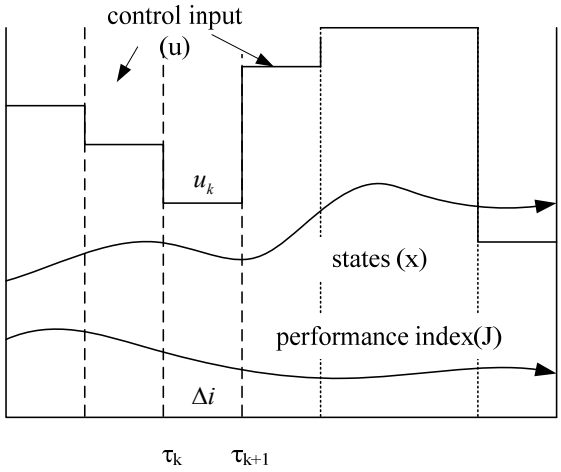

Figure 1. Example of a piece wise constant discretized control input and continuous trajectories of the states and the development of the performance index.

Mostly a low order B-spline function (for example piece-wise constant or piece-wise linear functions) is used to represent the control inputs. Polynomial functions such as Chebychev polynomials can also be used.

For piece-wise constant functions the control input is formulated as:

$$u(t) = u_k, \quad t \in \left[\tau_k \quad \tau_{k+1}\right] \quad for\ k = 0,1,..N-2 \tag{2}$$

2.3. Refinement Procedure

A fine grid for the discretization of the control vector in time improves the quality of the control strategy, but it has significant effect on the computational effort. Therefore, to limit the computational effort, we propose to start with optimisation by starting with a low number of parameters (coarse grid). When this optimization has reached a plateau, a step size refinement is applied for a next iteration to achieve better performance. Such refinement procedure has been considered important in the improvement of direct optimization methods. Binder (2000) and Schlegel et al, (2005) used a local resolution based analysis to point out which control parameter needs to be refined further while Balsa Canto et al (2003) and Garcia et al (2006) applied grid refinement throughout the trajectory.

However, for process optimization, we found that there are several intervals where adjustment of the control parameter has no significant effect on the improvement of the performance index. These intervals can be excluded from further optimization. The selection uses a threshold value for the sensitivity (ε_s) which separates the control parameters into two groups: u^{opt}, with sensitivity above the threshold value and which will

be considered for refinement and further optimization, and u^{const}, with a sensitivity below the threshold value and which are excluded from further optimization. Figure 2 illustrates the selection of control input based on its sensitivity.

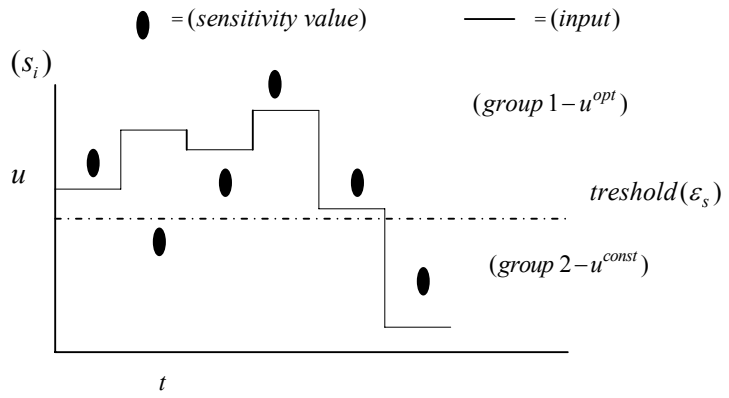

Figure 2. Selection of parameters for refinement based on the sensitivity

The sensitivity (s_i) of each parameter input is given by:

$$s_k = \frac{\partial J}{\partial u_k} \tag{3}$$

$$\varepsilon_s = r_\varepsilon . \bar{s}_k \tag{4}$$

and the selection by

$$\begin{aligned} u_k^\ell (s_k >= \varepsilon_s) &\in u_k^{\ell,opt} \\ u_k^\ell (s_k < \varepsilon_s) &\in u_k^{\ell,const} \end{aligned} \tag{5}$$

with s_k the sensitivity of parameter with index k, and ε_s the threshold value which fraction (r_ε) of the average sensitivity $(\bar{s}_k)$. In this work the sensitivities are numerically calculated by small perturbations for each individual parameter with $\delta u = 10^{-6}$.

The threshold value is chosen such that the less sensitive parameters are separated from the parameters to be optimized. Here, the threshold value is linked to the mean sensitivity by multiplication with a proportionality factor (r_ε). The proportionality factor value is an indicator for the range below mean sensitivity; $r_\varepsilon = 0$ means that all input parameters are above the threshold value and therefore they are always refined and further optimized. Using the value $r_\varepsilon = 0$ corresponds to the work of Balsa-Canto (2003) and Garcia et al (2006). By increasing the r_ε value, more parameters will be transferred to the second group that is not optimized further.

The optimization procedure is given by the following pseudo-algorithm. In the first step, the initial number of control parameter input (N_u) and their values (u^0) are defined together with the threshold parameter (r_ε), stopping criteria of optimization, and the maximum

number of refinement(l_{max}). For the first iteration (*l=1)*, all control parameters are above the threshold value and therefore they are all parameters to be optimized ($u_k^{0,opt}$). The optimization is performed for this set of parameters and at the end the obtained parameters are evaluated for their sensitivity to performance index (see equations 3 and 4). After grouping by using the threshold value and sensitivity values, a new set of input parameters is obtained by doubling the number of parameters from previous iteration $N_u^{\ell+1} = 2N_u^{\ell}$. The new set of parameters optimized and will be subject for a following refinement. The procedure is repeated until l_{max} is reached.

Pseudo Algorithm

Choose number of input parameters, initial control profiles and final time: N_u^{ℓ}, u^0, t_f

Specify tolerances, and maximum number of refinement : *TolMesh,TolFun, TolX,* r_s, l_{max}

For ℓ=1,.. $\ell_{\max}$

$u_k^0 = \left[u_k^{0,opt} \right]$ and $u_k^{0,const} = [\ ]$ (*for initial iteration*)

Do optimization problem with initial guess u_k^0,

→ store the optimal solution $u_k^{*,\ell}, J^{*,\ell}$

if $\ell < \ell_{\max}$ **t** hen

Calculate s_i *and* ε_s

(i) $u_k^{*,\ell}\left(s_k^{\ell} > \varepsilon_s\right) \rightarrow u_k^{\ell,opt}$

(ii) $u^{*,\ell}{}_k\left(s_k^{\ell} < \varepsilon_s\right) \rightarrow u_k^{\ell,const}$

→refine : $u_k^{\ell+1,opt} = \text{interp}(u_k^{\ell,opt}, \tau_k^{l+1})$,

$u_k^{\ell+1} = u_k^{\ell+1,opt}$, $u_k^{l+1,const} = u_k^{l,const}$

Else

Exit

end if

end for

→ optimal solution: $J_k^*, u_k^{*,\ell}$

For optimization, a direct search by using *Patternsearch* from Matlabs' optimization toolbox is applied. The method is normally used for highly nonlinear functions and if other direct gradient-based methods are not reliable anymore. *Patternsearch* operates from a set of points that form a pattern and it does not require derivatives to determine the descent direction. The pattern is reflected, expanded or shrunk, depending on whether any point within the pattern has a lower objective function value than the current point. The stopping criteria for the procedure are related to those characteristics. If the progress in optimization, expressed in terms of changes in the objective function (*TolFun*), and in changes of the mesh (*TolMesh*), and the changes in the parameters (*TolX*) is below the values as given in Table 1, the optimization ends.

Table 1 Setting of stopping criteria of *patternsearch*

Options	value
TolMesh	10^{-4}

TolX	10^{-5}
TolFun	10^{-5}
SearchMethod	*Positive basis Np1*
Mesh Contraction	*2*
Mesh refinement	*0.3*

2.4. Evaluation of the procedure on a reference process

For evaluation of the procedure a reference case on the optimal production of protein in a fed-batch reactor is used. This case was originally formulated by Park and Ramirez (1988). The objective of this case is to maximize the secreted heterologous protein by a yeast strain in a fed-batch culture. The model and its description are given in the work of Park and Ramirez (1988) and Balsa-Canto et al (2003).

Luus (1995) applied dynamic programming while Banga et al (1998) used control vector parameterization without refinement to solve this optimization problem. The attained performance index values were J=32.686 and J=32.562, respectively. To test the effect of refinement to this particular case, we first did a single run optimization (i.e. without refinement) with 40 parameters. The result of this optimization (Figure 3) shows strong variations in values of the succeeding parameters which is the result of local optima of the solution. The calculation time was 35 minutes and the obtained performance index J=32.7297

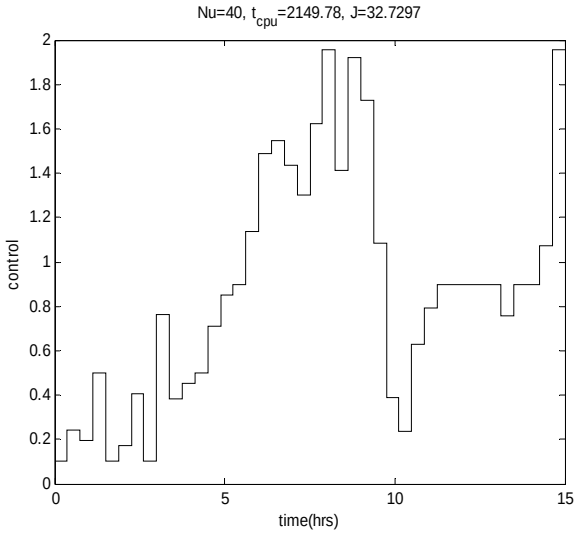

Figure 3 Single run optimization for optimal production of protein with a control vector parameterization by 40 parameters.

In the refinement method the choice for the threshold value is important. Its value has effect on the obtained result and the parameters used during the refinement iterations. The threshold is a fraction (r_ε) of the mean current sensitivity. Figure 4 gives the obtained value of the performance index and the final number of optimization parameters for threshold

factors *(r_ε)* varying from 0 to 1. Increasing threshold factors reduce the number of parameters for optimization (u^{opt}) and consequently lower computation time, but at the same time the final obtained performance index is reduced, meaning that the optimum solution is not attained. The threshold of r_ε =0, which mean all parameters are optimized, could give better performance index, however the computation time is high. Therefore, the choice for the threshold factor is recommended in the range r_ε =0.1-0.2.

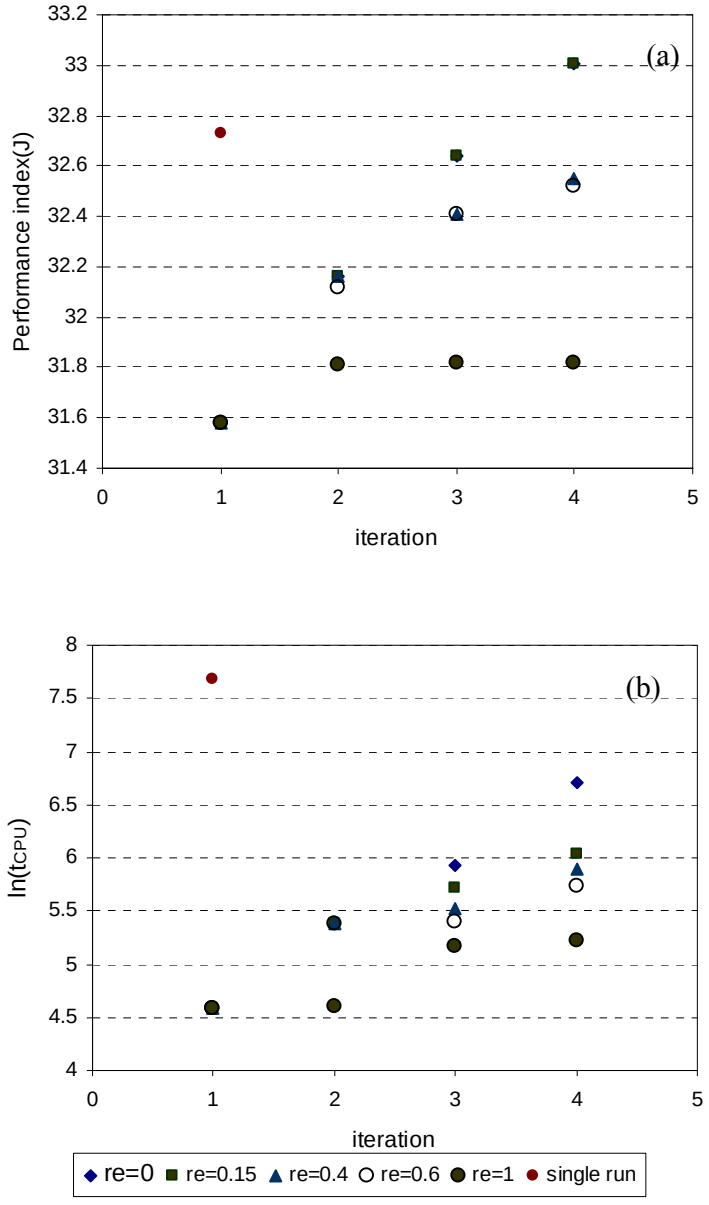

Figure 4. The effects of threshold factor *(r_ε)* variation to the final obtained value of the performance index (a) and the required computation time (logarithmic scale) (b)

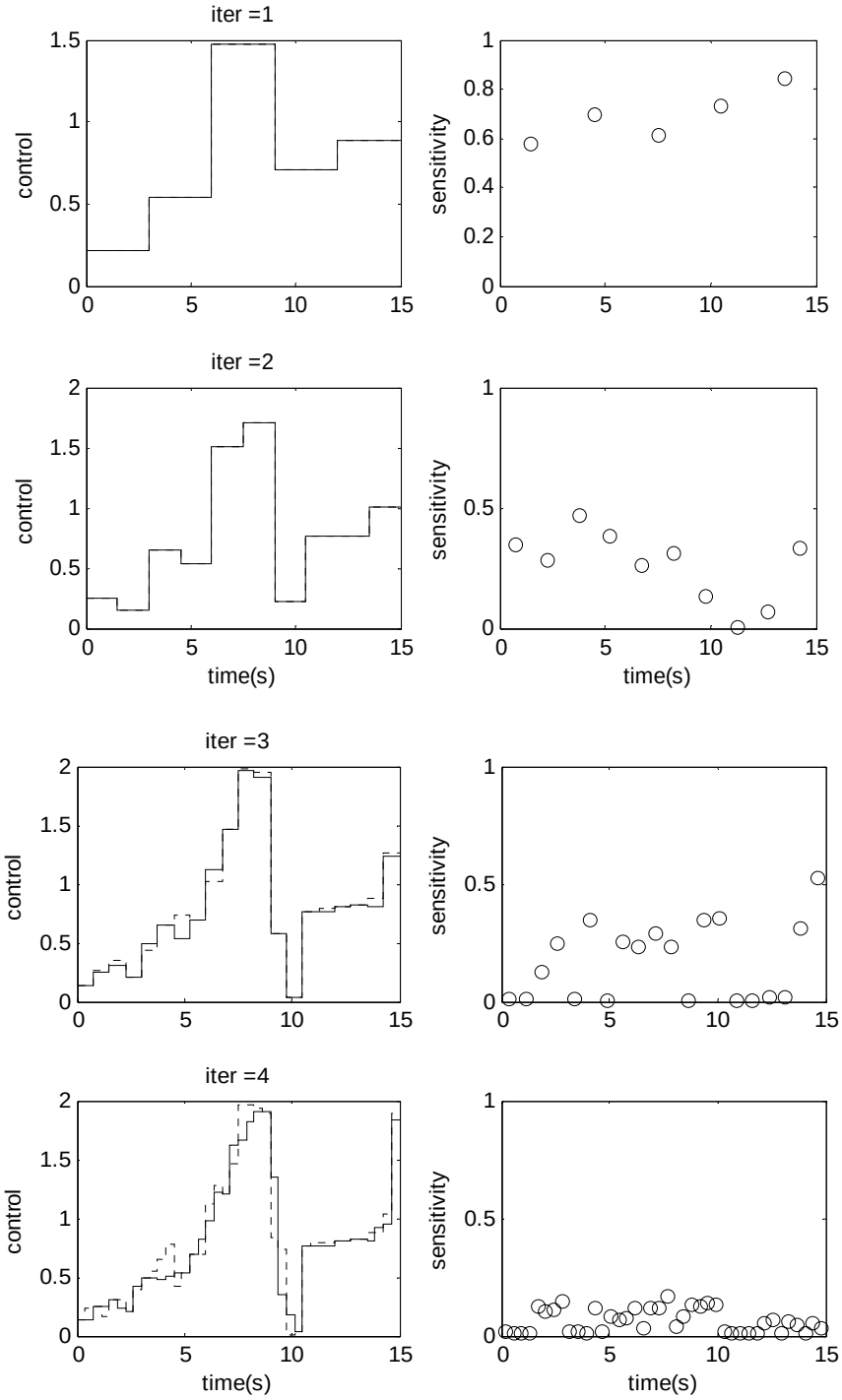

Figure 5 Trajectories for optimal production of protein by using step size refinement with (r_ε=0,--) and (r_ε=0.15,-) from the mean sensitivity value.

Figure 5 shows the development of the input trajectories and the sensitivity values for (r_ε)=0 and 0.15 respectively. The first refinement iteration started with 5 parameters and after optimization the sensitivity of each parameter is still above the threshold value. Thus, all input parameters are refined and the number of parameters for the second refinement is doubled to 10. After optimization in the second refinement iteration, the sensitivity values are evaluated and now there is only one parameter with sensitivity below the threshold value. Therefore, in the third refinement step, the control input to be optimized (u^{opt}) has 18 parameters and one parameter is not optimized further. During the fourth iteration there are only 22 parameters to be optimized which make the computation time is less than full optimization and the result after this step is given in the last graph of Figure 5.

Table 2. Computation time and performance index for optimal protein production optimization

Iteration	Refinement on all CVP-points (r_ε=0)			Refinement with threshold value (r_ε=0.15)			Single run	
	u^{opt}	J	t_{cpu} (s)	u^{opt}	J	t_{cpu} (s)	u^{opt}	J
1	5	31.5805	98	5	31.5805	98	40	32.7297
2	10	32.1610	218	10	32.1612	218		
3	20	32.6412	376	18	32.6392	303		
4	40	33.0028	813	22	33.0032	451		
Total t_{cpu}			*1506*			*1071*		*2149*

u^{opt} : number of parameters for optimization in this step, J: performance index value, t_{cpu}= computation time(s). Computation time on Intel Pentium M processor 1.40 GHz ,Matlab 7.0.

The results in Table 2 show that the required computational time for the refinement based on threshold sensitivity method (r_ε=0.15) is favourable compared to the refinement with no threshold value (r_ε=0) as was used by Balsa-Canto et al., (2003) and Garcia et al., (2005). However, both refinements in Table 2 perform better than the single run optimization (40 parameters, 2149 seconds) in terms of the performance index and computation time. Furthermore, comparing to previous studies (Banga et al, 2003; Luus, 1995) an interesting improvement for the performance index is realized.

3. Application to baking process

In section 2.4 the method has been tested to a standard problem from literature in which the objective function was maximized, and now the method will be applied to bakery production (baking process) optimization, The general purpose of baking optimization is to minimize the deviation of final qualities from the aimed values.

3.1 Formulation of baking optimization problem

The objective of baking optimization is to find optimal heating strategies that result in the specified final product qualities. Baking can be performed by applying different heating inputs as: convective, radiation and microwave heating. Each heating input has a different

role in the improvement of baking performance. Convective heating is the most applied type; heat is transferred to the product surface and then penetrates into product by convection and conduction. Microwave heating generates heat inside the product. Radiation (usually infrared) directly heats the upper layer of the product, by exciting rotational/vibration modes in the present molecules. Hadiyanto et al., (2007b) showed that depending on the required final product quality, different optimal heating strategies can be found by optimization.

For optimization the required final qualities have to be translated into a performance index. The following formulation is used to express the performance index.

$$\min\ J = \sum_{i=1}^{N} w_i.(x_i(t_f) - x_{s,i})^2 \qquad (6)$$

where $x_{s,i}$ represent the setting values of the states (qualities) at the end of baking time, and w_i are weighting factors for each product quality. The results in this work are based on the setting values and weight factors as given in Table 3. Please note that while in the validation example of section 2.4 the objective function was maximized, we will here minimize the objective function. This does not give any differences in results or method.

Table 3. Applied setting values for the final product qualities and weight factors

Optimized quality (x)	**weight factor (w_i)**	**setting value (x_s)**
brownness surface	1 [-]	0.8
crispness surface	10 [-]	0.65
water content centre [kg/kg]	10 [(kg/kg)$^{-2}$]	0.38
temperature (surface)	0.0001 [(°C)^2]	25°C

3.2 Baking model development and assumptions

The baking model concerns a series of three sequential processes:

1. Heat and mass transfer of liquid water, water vapour and CO_2,
2. Product transformations
3. Product quality development

The full model was presented in previous work (Hadiyanto et al., 2007a); here the main equations are briefly discussed and presented in Appendix 1 (Table A1, A2 and A3). Discretization of 1-D in 10 segments was found to be satisfactory. This resulted in a system with 108 ordinary differential equations (ODEs) which were solved with integration procedures for stiff sets of ODEs (Matlabs' ode15s or ode23s).

4.1. Heat and mass transfer

The heat and mass balances for liquid water, water vapour and CO_2 gas (eqs. A1-A4) follow from the laws for mass and energy conservation and include water evaporation, heat conduction, and heat fluxes due to internal convection during baking. The local changes of liquid water in the product are result of the diffusion and the evaporation rate (I_v). For additional equations see appendix 1 tables A1 and A2.

4.2. State transformations

The product extension (*e*) gives the change of size (height) compared to the initial height of the product (equation A5). The extension of height is based on *Kelvins-Voight'* model for visco-elastic systems. Protein thermosetting reactions and starch state transformation from crystalline state into the gel state and reverse are the main transformations for product texture. Protein thermosetting reactions only solidify the network. Therefore, starch gelatinization and retrogradation are the main transformations relevant for textural properties. The changes of the degree of gelatinization are equal to the gelatinization rate (Zanoni et al., 1995) minus the retrogradation rate (equation A6). Here is α_{max} the maximum attainable degree of gelatinization which is a function of the initial composition of the product, i.e. content of water (W), starch (S) and other water binding components (C) in dough (Hadiyanto et al., 2007a).

Gelatinization occurs at higher temperatures and is faster than retrogradation. Therefore gelatinization takes place during baking and retrogradation during storage (staling of product). Additional equations are given in appendix 1 tables A2 and A3.

Browning of bakery products is mainly caused by the Maillard reaction which forms melanoidins as colouring compounds (eq A7). These reactions are zero order (van Boekel, 2001) and the reaction rate depends on temperature and water content in the product (see also Table 4b)

4.3. Product quality model

Crumb (i.e. the open network structure in the centre of bread) is linked to the degree of starch gelatinization (Hadiyanto et al., 2007a). For a range of Dutch bakery products the relation between crumb and degree of gelatinization given by equation A8 is found.

Crispness and softness of bakery products are linked to the difference between the current product temperature (T_r) and the glass transition temperature of the product (T_g). A product is crisp when T_r-T_g<*0* (equation A9). The glass transition temperature is a function of the product composition (see Table A3). Softness is a combined function of T_r-T_g and the degree of gelatinisation. Products are soft for T_r-T_g >*0* but softness requires a minimum value of the degree of gelatinization above 0.3 (equation A10).
The relation between brownness and the amount of melanoidins (*me*) is given by equation A11.

5. Results and Discussion

5.1. Case 1: Baking using convective heating

We first discuss baking with convective heating only. The set points for the quality attributes are given in Table 3. A single run optimization (l = 1) with 45 parameters and 120°C as initial value for all parameters resulted in a computation time of around 5 hours. The obtained performance index is J = 0.0071432. The optimization result (Figure 6) shows several irregular peaks of the control input which are not expected for a continuous process. This phenomenon is common for CVP with a large number of parameters and illustrates that the final result is in a local minimum.

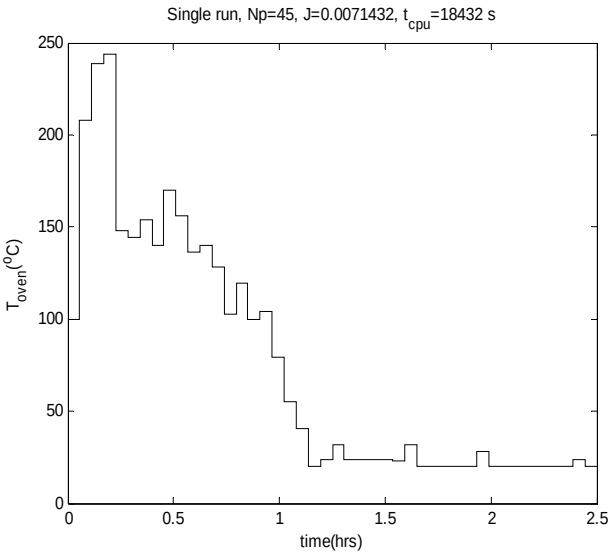

Figure 6 Single run optimization for convective baking with an input represented by 45 parameters

The procedure with refinement is started with 6 parameters and 120°C as initial values for the parameters. Two cases are considered, with $r_\varepsilon = 0$ and 0.2, respectively. Figure 7 gives the development of the input trajectories for convective heating and the sensitivity values used for refinement for the case $r_\varepsilon = 0.2$. For both cases the trajectories fall almost together and have a much more regular form than for the single run optimization. The convergence to optimal solution is illustrated by the decrease of the sensitivity values at each refinement.

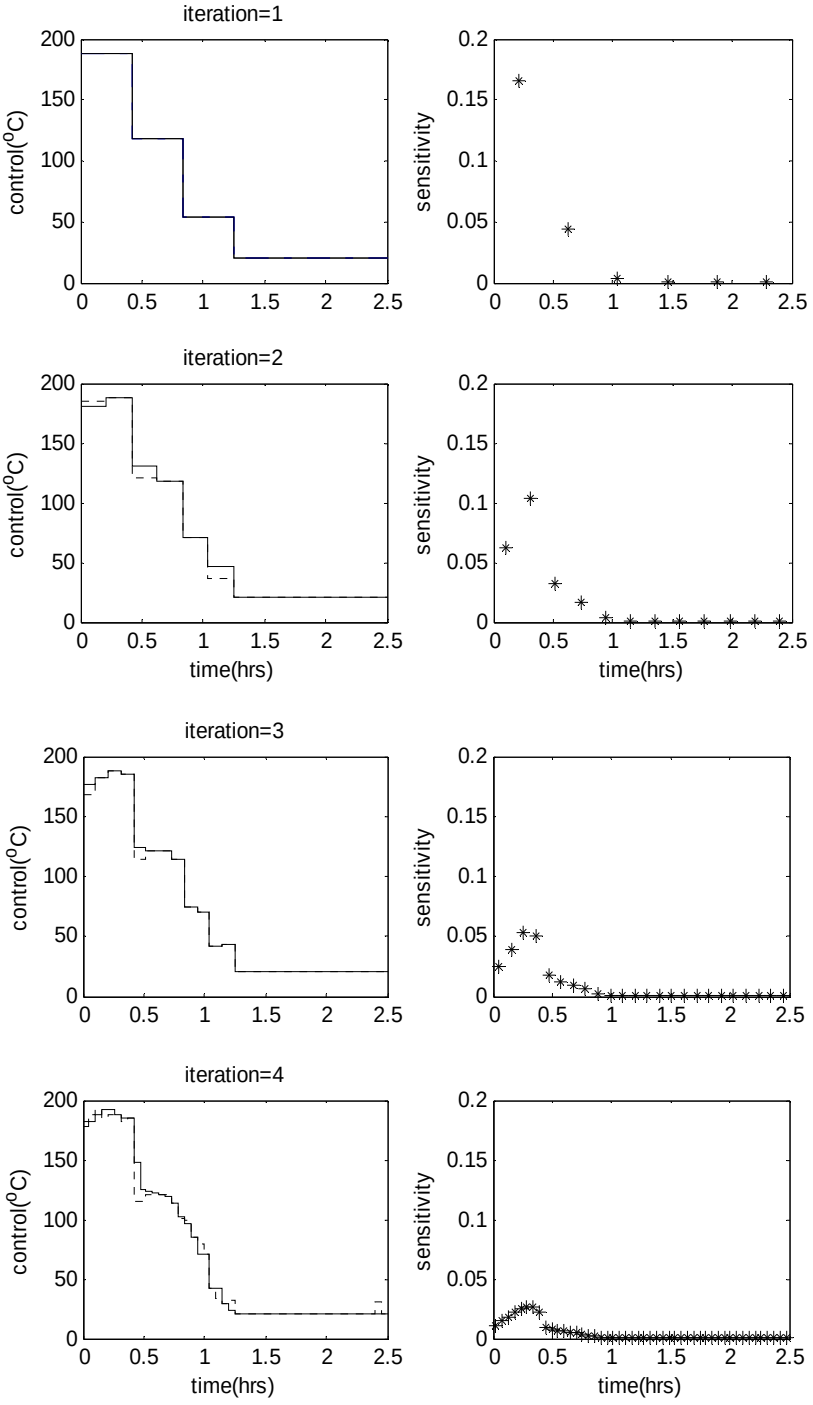

Figure 7 Convective heating baking operation. Succeeding iterations for the optimization procedure with refinement for iteration 1 to 4. (–) with threshold value $r_e = 0.2$, (--) without threshold value ($r_e = 0$).

For a factor $r_\varepsilon=0.2$, similar trajectories are obtained as for $r_\varepsilon=0$ while the computation is almost halved (Table 4). The computation time is also about 4 times faster than that for a single run optimization. Moreover, compared to single run optimization the refinement procedure with a threshold factor $r_\varepsilon=0.2$ gives a clear improvement of the performance index.

Table 4. Computational time and performance index for three different methods

Iteration	Refinement on all CVP-points ($r_\varepsilon=0$)			Refinement with threshold value ($r_\varepsilon=0.2$)			Single run	
	u^{opt}	J	t_{cpu} (s)	u^{opt}	J	t_{cpu} (s)	u^{opt}	J
1	6	0.00718145	382	6	0.00718145	382	45	0.0071432
2	12	0.00697551	869	6	0.00697542	537		
3	24	0.00692146	1718	12	0.00692135	1078		
4	48	0.00686000	4941	24	0.00684551	2245		
Total t_{cpu}			*7911*			*4243*		*18432*

u^{opt} : number of parameters for optimization in this step, J: performance index value, t_{cpu}= computation time(s). Computation time on Intel Pentium M processor 1.40 GHz using Matlab 7.0.

5.2. Case 2: Baking with multi-heating inputs

Multi-heating baking is another application with setting values as given in table 3. The applied heating sources are convective heating, characterized by the oven temperature (T_{oven}), radiation, characterized by the temperature of the radiating element (T_{rad}) and microwave power (P_{mw}). The use of the three heating sources makes the baking system more flexible and can result in a better achievement of the product quality goals (Hadiyanto et al, 2007b).

First, the multi-heating system is optimized by a single run optimization and applying 45 parameters for each input, which results in a total of 135 of parameters for optimization (u^{opt}). Trajectories are presented in Figure 8. The single run optimization results in a performance index of J = 0.003725 and required about 11 hours of computation time. The trajectories are more irregular than for the single input optimization and are not very intuitive for a continuous process.

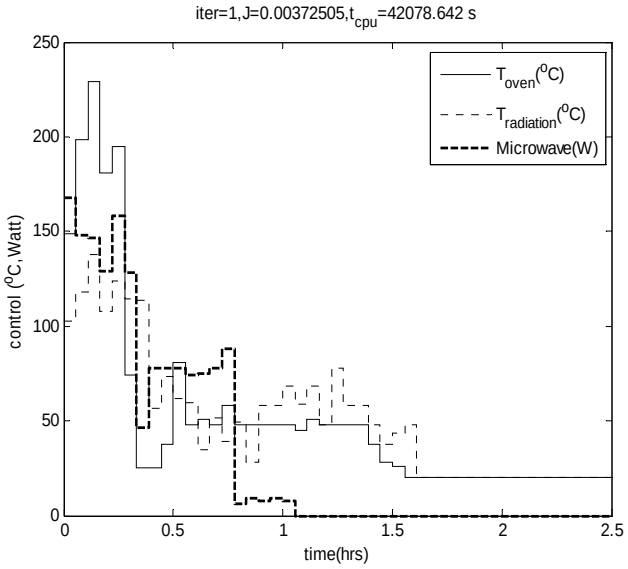

Figure 8. Single run optimization for multi-heating baking. Each input is represented by 45 parameters.

In the case with only convective heating, we concluded that the refinement procedure with r_e = 0.2 performed best. Thus we only apply this procedure here. The results for the step size refinement are given in Figure 9 and Table 5.

Figure 9 shows the development of the trajectories and the sensitivity values for four succeeding refinement steps. At the start, each input is represented by 6 parameters which results in a total of 18 parameters. This set of parameters can be optimized quickly. In the next refinement iteration only 6 input parameters with sensitivity above the threshold value are optimized (18 for total). This strongly reduces the computation time compared to a procedure in which all parameters are refined (36 parameters).

Table 5 shows that finally 50% of the control parameters are selected for u^{opt}. Control parameters in the first 1.25 hour are the most sensitive (u^{opt}), while the rest are not sensitive and are not varied. Compared to the single run optimization, the refinement method gives a much better performance index (80% improved) and requires much less computation time (a factor 6 less). Moreover the trajectories have a more regular form that complies to the continuous nature of the process.

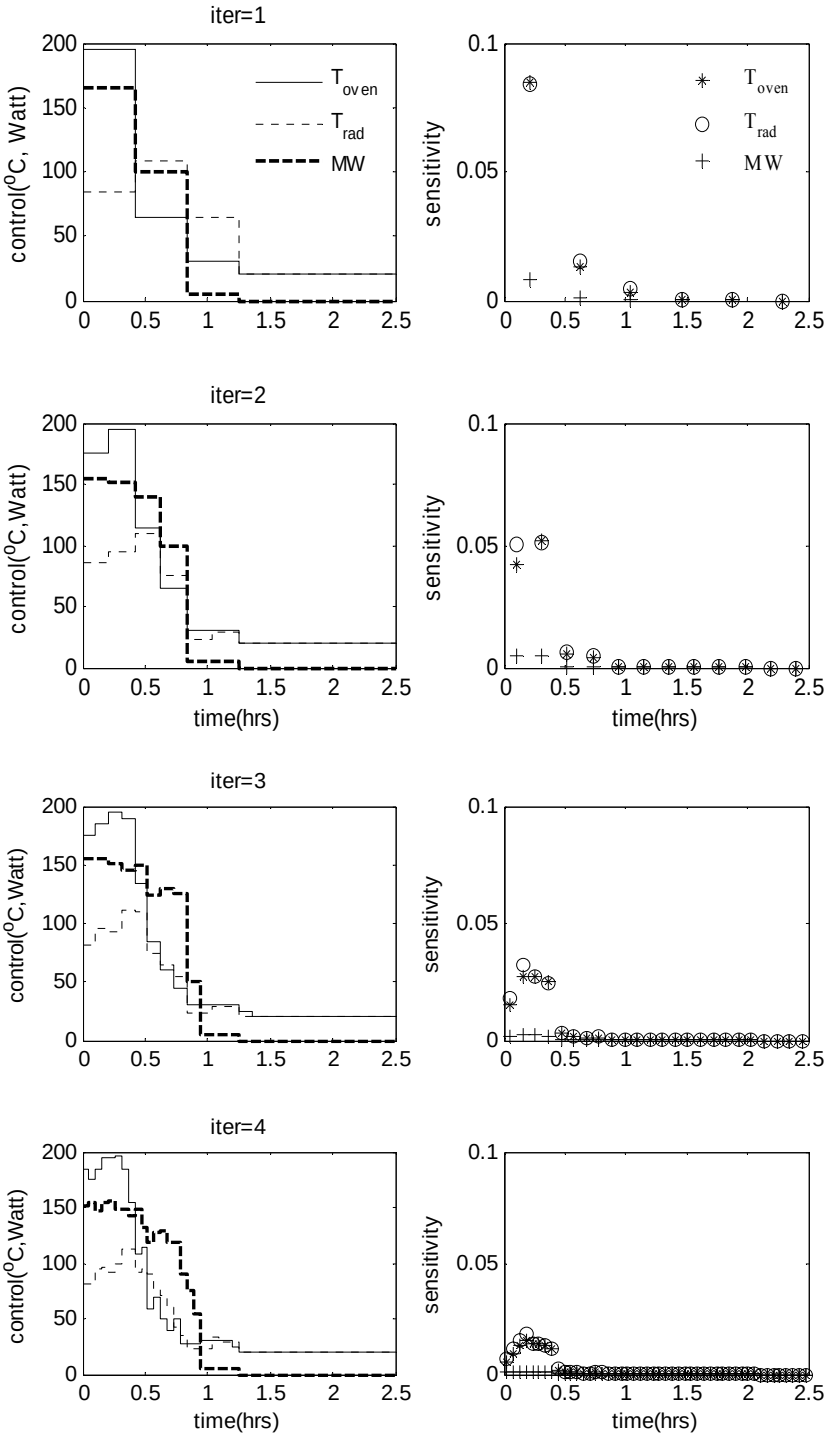

Figure 9. Succeeding iterations optimization procedure for multi-heating baking operation with refinement for iteration 1 to 4 and threshold factor r_{ε}=0.2.

Table 5. Results of optimization with refinement for multi-heating baking process. The optimization is performed with threshold factor r_ε=0.2 and compared to single run optimization.

Iter (l)	Refinement with r_ε=0.2			Single run	
	u^{opt}	J	t_{CPU} (s)	u^{opt}	J
1	18	0.00370194	785	45	0.0071432
2	18	0.00205799	1264		
3	36	0.00156933	2057		
4	76	0.00144551	3245		
Total t_{cpu}			*7352*		*42078*

6. Conclusion

The improvement of control vector parameterization in optimization by using sensitivity functions has been presented. Starting with a low number of parameters, the refinement method showed a significant reduction of computation time while the achieved performance index was still equal as compared to full control vector parameterization. The refinement method used a threshold sensitivity to group input parameters. The reduction of the number of input parameters to be optimized (above threshold value) resulted in lower computational effort, and in this work it was found that the recommended threshold value is in the range of 10-20% of the mean sensitivity.

Keeping computational time within limits becomes critical for larger complex systems. A significant reduction of the computation time was achieved with convective and multi-heating baking, including a heating and cooling period. The control parameters in the cooling period have minimal effect on the performance index and therefore these parameters were hardly considered in the optimisation procedure. The optimization focused itself on the heating period, by which the performance index is the most affected. Refinement gives a good estimate of the optimal process.

The proposed control vector parameterization method has two-fold benefit. First, it is useful for optimization of the design and operation of large process systems, but also identifies critical control points in the production process, yielding more insight in what factors determine the quality of the product. For example, the layout and operation of complete or parts of complex food production systems, or even supply chains with long periods of cooling where product quality is not very sensitive to the settings, as long as they stay in a certain (low) region. Secondly, the method identifies the parameters that are most important to the product quality. This critical control point analysis will facilitate further study and improvement of the production system.

Acknowledgement

This work is financed by the Graduate School VLAG, the Netherlands. The authors acknowledge Dr. Hans Stigter (Wageningen University) for his highly valuable suggestions and discussions for this work.

Annotations

Symbol	Description	Unit
a_w	Water activity	[-]
C	Non-starch water binding components	$kg.kg^{-1}$
D_v	Gas diffusivity	$m^2.s^{-1}$
D_w	Liquid diffusivity	$m^2.s^{-1}$
e	Relative extension of height	[-]
f	Fusion factor	[-]
E_a	Activation energy	$J.mol^{-1}$
G	Retrogradation rate constant	s^{-1}
G_0	Reference value for retrogradation rate	s^{-1}
H_c	Enthalpy of water vapour	$J.kg^{-1}$
H_v	Enthalpy of CO_2 gas	$J.kg^{-1}$
h_c	Convective heat transfer coefficient	$Wm^{-2}K^{-1}$
h_v	Mass transfer coefficient	$m.s^{-1}$
J	Performance index	[-]
k	Thermal conductivity of product	$Wm^{-1}K^{-1}$
K_g	Constant	[-]
k_{me}	Reaction rate constant for Maillard reaction	s^{-1}
k_{soft}	Constant for gelatinization	[-]
k_{gel}	Rate constant for gelatinization	[-]
k_{retro}	Rate constant for retrogradation	[-]
m_v	Mass flux of water vapour	$kg.m^{-2}s^{-1}$
me	melanoidines	[-]
m_c	Mass flux of CO_2 gas	$kg.m^{-2}s^{-1}$
M_w	Molecular weight of water	$kg.kmol^{-1}$
Nu	Total number of input parameter	[-]
P	Total pressure	Pa
$P_{0,r}$	Incident power of microwave	$Watt.m^{-3}$
P_{mw}	Microwave power	Watt
$p_{v,sat}$	Saturated pressure of water vapour	Pa
R_g	Gas constant	$J.mol^{-1}K^{-1}$
R	Height of product	[m]
R_{CO2}	CO_2 production rate	$kg.m^{-2}.s^{-1}$
S	Sugar content	$kg.kg^{-1}$
t	Time	s
T_∞	Hypothetical temperature	K
To	Initial dough temperature	K
U^*	Activation energy for recrystallization	$J.mol^{-1}$
V_c	CO_2 gas concentration	$kg.kg^{-1}$
V_v	Water vapour	$kg.kg^{-1}$
W	Water content	$kg.kg^{-1}$
W_o	Initial water content	$kg.kg^{-1}$

Z	Starch content	$kg.kg^{-1}$
T_g	Glass transition temperature	K
T_m	Melting temperature	K
S/Z	Ratio sugar and starch	[-]
s	Sensitivity	
u^{opt}	Input parameter subjected for optimized group	
u^{const}	Input parameter subjected for constant group	
r_ε	Factor of sensitivity threshold	[-]
l_{max}	Maximum refinement iteration	[-]
t_f	Final time of processes	[hrs]

Greek letters

α	Total degree of starch gelatinization	[-]
α_{max}	Maximum attainable degree of gelatinization	[-]
$\alpha_{,mw}$	Attenuation factor	$[m^{-1}]$
λ	Evaporation heat	$J.kg^{-1}$
ε	Porosity	[-]
ε_s	The threshold of sensitivity	[-]
υ	Kinematic viscosity	$m^2.s^{-1}$

References

Balsa-Canto, E., Banga, J.R., Alonso, A.A., and Vassiliadis,V.S. (2001).Dynamic Optimization of Chemical and Biochemical Processes Using restricted second order information. *Computer and Chemical Engineering 25*, 539-546.

Banga, J.R., Balsa-Canto, E., Moles, C.G., and Alosno, A.A. (2003). Improving food processing using modern optimization methods. *Trends in Food Science & Technology, 14(4), 131-144*

Betts, J., T., and Huffman W.,P., (1998), Mesh refinement in direct transcription methods for optimal control, *Optimal Control Applications and Methods,19(1):1-21*

Binder, T., Cruse, A., Cruz Villar,C.A and Marquardt,W. (2000). Dynamic optimization using a wavelet based adaptive control vector parameterization strategy. *Computers & Chemical Engineering,* ***24***(2-7): 1201.

Bryson, A.E. (1999). Dynamic optimization, Editorial Addison Wesley Longman, Inc, Menlo Park, CA.

Bryson, A.E., and Ho, Y.C., (1975). Applied Optimal Control Optimization, Estimation and Control. Hemisphere Publication Company, New York

Garcia, M.S.G, Balsa-Canto,E., Alonso,A.A. and Banga,J.R. (2006). Computing optimal operating policies for the food industry. J*ournal of Food Engineering, 74(1),13-23*

Goh,C.J. and Teo,K.L.(1988).Control parameterization: A unified approach to optimal control problems with general constraints. *Automatica*,24(1): 3-18

Hadiyanto, A. Asselman, G. van Straten, R.M. Boom, D.C. Esveld and A.J.B. van Boxtel, (2007a), Quality prediction of bakery products in the initial phase of process design. *Innovative Food Science & Emerging Technologies*, 8(2),131-144

Hadiyanto, D.C. Esveld, R.M. Boom, G. van Straten,and A.J.B. van Boxtel, (2007b), Quality driven design of baking operation using dynamic optimization, *submitted*

Luus, R. Sensitivity of control policy on yield of a fed-batch reactor. *IASTED International Conference on Modelling and Simulation,* Pittsburgh, PA, 1995.

Park, S. and Ramirez, W.F., (1988). Optimal production of secreted protein in fed-batch reactors. *AIChE Journal* **34** 8, pp. 1550–1558.

Roubos, J.A., Straten, G van and A.J.B. van Boxtel, A.J.B. (1999). An evolutionary strategy for fed-batch bioreactor optimization: Concepts and performance. *Journal of Biotechnology* **67:** 173–187

Schlegel, M., Stockmann, K. , Binder,T., and Marquardt,W. (2005). Dynamic optimization using adaptive control vector parameterization. *Computers & Chemical Engineering,* 29(8): 1731.

Zhang and Datta. (2005). Mathematical modelling of bread baking process. *Journal of Food Engineering,* 75, 78-89

Appendix 1

Table A1. Main equations for the baking model renumber the equations

Laws of conservation		
Energy	$\rho c_p \frac{\partial T(r,t)}{\partial t} + \frac{\rho\lambda}{1+e(r,t)} \frac{\partial e(r,t)}{\partial t} = k \frac{\partial^2 T(r,t)}{\partial r^2} - \lambda I_v(r,t) - \frac{\partial (m_v H_v)}{\partial r} - \frac{\partial (m_c H_c)}{\partial r} + Q(r,t)$	(A1)
Liquid water	$\rho \frac{\partial W(r,t)}{\partial t} + \frac{\rho W(r,t)}{1+e(r,t)} \frac{\partial e(r,t)}{\partial t} = \frac{\partial}{\partial r}(D_w \frac{\partial W}{\partial r}) - I_v$	(A2)
Water vapour	$\rho \frac{\partial V(r,t)}{\partial t} + \frac{\rho V(r,t)}{1+e(r,t)} \frac{\partial e(r,t)}{\partial t} = \frac{\partial}{\partial r}(D_{vc} \frac{\partial V}{\partial r} - m_v) + I_v$	(A3)
CO_2	$\rho \frac{\partial V_c(r,t)}{\partial t} + \frac{\rho V_c(r,t)}{1+e(r,t)} \frac{\partial e(r,t)}{\partial t} = \frac{\partial}{\partial r}(D_{vc} \frac{\partial Vc}{\partial r} - m_c) + I_c$	(A4)
State transformations		
Product extension	$\eta \frac{de(r,t)}{dt} + Ee(r,t) = P - P_{atm}$	(A5)
Degree of gelatinization	$\frac{d\alpha(r,t)}{dt} = k_{gel}(\alpha_{\max} - \alpha(r,t)) - k_{retro}\alpha(r,t)$	(A6)
Melanoidine formation	$\frac{dm_e(r,t)}{dt} = k_{me}(r,t)$ (Maillard reaction)	(A7)
Quality attributes		
	$crumb(r,t) = \begin{cases} 0 \quad if \quad \alpha(r,t) = 0 \\ 2\alpha(r,t) \;\; if \quad \alpha(r,t) \le 0.5 \\ 1 \;\; if \quad \alpha(r,t) > 0.5 \end{cases}$	(A8)
	$crispness(r,t) = -\frac{0.0067.(T_r - T_g(r,t))}{1+\exp(3.(T_r - T_g(r,t)))}$	(A9)
	$softness(r,t) = -k_{soft} \frac{0.01.\exp(3(T_r - T_g(r,t)))}{1+\exp(3.(T_r - T_g(r,t)))}$	(A10)
	$brownness(r,t) = 1 - (1 - m_e(0))e^{-0.23(m_e(r,t))}$	(A11)

Table A2. Additional equations for the baking model

Equation	
A1-A4	$m_v = -\frac{\kappa}{\nu}\frac{V(r,t)}{V(r,t)+V_c(r,t)}\frac{\partial P}{\partial r}$
A1-A4	$m_c = -\frac{\kappa}{\nu}\frac{V_c(r,t)}{V(r,t)+V_c(r,t)}\frac{\partial P}{\partial r}$
A5	$P = p_v + p_c$
A1	$Q(r,t) = \frac{2\alpha_{mw}RP_{0,r}}{r}\exp\left(-2\alpha_{mw}(R-r)\right),\ P_0, r = \frac{P_{mw}}{2\pi R(L+R)}$
A1-A4,A7	$a_w = \frac{1.05W(r,t)}{0.09+W(r,t)}$
A6	$k_{gel} = 2.8.10^{19}\exp(\frac{-139000}{RX_1(r,t)})$
A7	$k_{retro} = G_0.\exp\left[\frac{-U^*}{R(T(r,t)-T_\infty)}\right]\exp\left[\frac{-K_g}{T(r,t).\Delta T.f}\right]\quad if\quad X_1 < 298^o K$
A7,A9,A10	$\alpha_{\max} = \begin{cases} 0 & if\quad W < 0.5(S+C) \\ \frac{(W-0.5S-0.5C)}{S} & if\ 0.5(S+C) < W < 0.5(3S+C) \\ 1 & if\quad 0.5(3S+C) < W \end{cases}$
A7	$k_{me}(r,t) = 4.9\times10^{-3}\times\frac{\exp(9a_w)}{2\times10^3+\exp(11.3a_w)}\exp\left[\frac{-E_a}{R}\left(\frac{1}{T(r,t)}-\frac{1}{363}\right)\right]$
A10	$k_{soft} = (\frac{-3}{7}+\frac{10}{7}\alpha_{\max}).\frac{\exp(500.(\alpha_{\max}-0.3))}{1+\exp(500.(\alpha_{\max}-0.3))}$

Initial and boundary condition for heat and mass transfer :

Surface $-k\left.\frac{\partial T}{\partial r}\right|_{r=R} = h_c(T_{oven}-T(R,t)) - \lambda\rho D_w\left.\frac{\partial W}{\partial r}\right|_{r=R} + F\varepsilon(T_r^4 - T(R,t)^4)$

$-D_v\left.\frac{\partial V}{\partial r}\right|_{r=R} = h_v(V_{ext}-V(R,t))$

centre $-k\left.\frac{\partial T}{\partial r}\right|_{r=0} = -D_v\left.\frac{\partial V}{\partial r}\right|_{r=0} = 0$

Initial *T(0,t)=To, W(0,t)=Wo;*

Table A3. Equation for glass transition and melting temperature

$$T_{g/m} = p_1 + p_2(S/Z) + p_3 W + p_4(S/Z)W + p_5(S/Z)^2 + p_6(W)^2 + p_7(S/Z)^2 W^2$$

Parameters	p_1	p_2	p_3	p_4	p_5	p_6	p_7
T_g	457.10	-396.32	-853.21	716.76	430.27	778.44	-1424.71
T_m	472.69	-180.90	-519.97	419.63	124.46	471.87	-749.88