further delayed with the increase of the concentration of surfactant. The delay was explained in terms of interface movement and a double-layer repulsion. Therefore, the increase of the C_{IV} value is due to the decrease of coalescence efficiency in the presence of a surfactant whose effect was not considered in the coalescence efficiency model (eq 9).

Conclusions

A population balance equation based on the breakage and coalescence models proposed by Coulaloglou and Tavlarides (1977) and Sovová (1981) was solved numerically to generate theoretical drop size distributions, which were compared with the experimental results obtained from the liquid–liquid dispersions in agitated vessels. In an attempt to choose a model which would best fit our experimental results, several alternatives (Table III) have been considered.

1. The theoretical drop size distribution predicted by employing the two alternative models for the breakage frequency (eq 4 and 5) provided an almost identical curve.
2. When the number of the daughter droplets was assumed to be 7, the predicted distribution curve gave better fit of the experimental data than when it was 2. This finding agrees with the experimental observation made by Konno et al. (1983).
3. The predicted drop size distributions according to the film drainage model or the impact of the colliding drop model were in the same drop size range, but the latter model results in a narrower distribution than the former.
4. Any combination of the models tested (Table III) could predict the drop size distribution reasonably well. However, one parameter out of four in the models has to be adjusted to fit the data for the liquid–liquid system without surfactant. For the system with a surfactant, two out of four parameters have to be adjusted. More parameters may need to be adjusted to fit the data for a wider range of operating conditions than tested. Therefore, more studies are needed to formulate more realistic and reliable models for the breakage and coalescence phenomena of liquid–liquid dispersion.

Nomenclature

$A(v)$ = probability density of droplet size v in vessel
C_I–C_{IV} = constants
d_{max} = size of the largest stable drop in the dispersion, cm
d_p = droplet diameter, cm
d_{32} = Sauter mean drop diameter, cm
D = impeller diameter, cm
$g(v)$ = breakage frequency of drops of having volume v, s^{-1}
$h(v,v')$ = collision frequency of drops of volume v and v', s^{-1}
n = impeller speed, rps
N = total number of drops at time t
T = tank diameter, cm
v = volume of droplet, cm^3
$\bar{v}$ = average volume of droplet, cm^3

Greek Symbols

$\beta(v',v)$ = number fraction of droplets with volume v to $v + dv$ formed by breakage of a drop of volume v'
$\lambda(v,v')$ = coalescence efficiency of drops of volume v with drops of volume v'
μ = viscosity, (dyn s)/cm^2
$\nu(v)$ = number of drops formed per breakage of drop of volume v
ρ = density, g/cm^3
σ = interfacial tension, dyn/cm
σ_v^2 = variance
ϕ = fraction of the dispersed phase
$\omega(v)$ = coalescence frequency out of v to $v + dv$, s^{-1}

Subscript

d = dispersed phase

Literature Cited

Chatzi, E. M. S. Thesis, Cleveland State University, Cleveland, OH, 1983.
Coulaloglou, C. A. Ph.D. Thesis, Illinois Institute of Technology, Chicago, 1975.
Coulaloglou, C. A.; Tavlarides, L. L. *Chem. Eng. Sci.* **1977**, *32*, 1289.
Curl, R. L. *AIChE J.* **1963**, *9*, 175.
Hodgson, T. D.; Lee, J. C. *J. Colloid Interface Sci.* **1969**, *30*, 94.
Hong, P. O.; Lee, J. M. *Ind. Eng. Chem. Process Des. Dev.* **1983**, *22*, 130.
Hong, P. O.; Lee, J. M. *Ind. Eng. Chem. Process Des. Dev.* **1985**, *24*, 868.
Konno, M.; Aoki, M.; Saito, S. *J. Chem. Eng. Jpn.* **1983**, *16*, 312.
Lee, J. M.; Soong, Y. *Ind. Eng. Chem. Process Des. Dev.* **1985**, *24*, 118.
Rietema, K. *Adv. Chem. Eng.* **1964**, *5*, 237.
Sovová, H. *Chem. Eng. Sci.* **1981**, *36*, 1567.
Valentas, K. J.; Amundson, N. R. *Ind. Eng. Chem. Fundam.* **1966**, *5*, 533.
Valentas, K. J.; Bilous, O.; Amundson, N. R. *Ind. Eng. Chem. Fundam.* **1966**, *5*, 271.

Received for review September 11, 1986
Revised manuscript received July 20, 1987
Accepted August 5, 1987

Bilinear Model Predictive Control

Yeong K. Yeo† and Dennis C. Williams*
Department of Chemical Engineering, Auburn University, Auburn, Alabama 36849

Bilinear model predictive control is defined for single-input–single-output systems. Offset compensation is provided to correct for the effects of unmeasured disturbances and model inaccuracies. Controller tuning is accomplished through the adjustment of a single parameter. The control algorithm is explicit, which allows implementation without requiring large computer memory or rapid computational speed. A filter with a single parameter is used to correct for the effects of an incorrect model.

Model predictive control is the most important development in chemical process control in the last several years. Model predictive controllers were developed independently at ANDREA/GERBOIS in France as Model Predictive Heuristic control (MPHC) (Richalet et al., 1978; Richalet, 1980) and at Shell Oil Company in the United States as Dynamic Matrix Control (DMC) (Cutler and Ramaker, 1980). Many successful applications have been reported, including a furnace (Cutler and Ramaker, 1980),

*Present address: HIMONT Research and Development Center, Wilmington, DE 19808.
†Present address: Division of Chemical Engineering and Polymer Technology, KAIST, Dong Dae Mun, Seoul, Korea.

0888-5885/87/2626-2267$01.50/0

a catalytic cracking unit (Richalet et al., 1978), a vinyl chloride production plant (Lebourgeois, 1980), a petroleum crude distillation unit (Engrand, 1980), and a steam generator (Mehra and Eterno, 1980). These applications include multivariable plants, time delays, and nonminimum phase behavior.

The reported successes of MPHC and DMC, which were developed on a heuristic basis, motivated the development of a theoretical basis of model predictive control. These results demonstrated the robustness of model predictive control, stability properties, tuning guidelines, and the relationships to linear quadratic optimal control, the Smith predictor, classical feedback control, statistical parameter estimation, deadbeat control, and Dahlin's algorithm (Astrom, 1980; Callier and Desoer, 1982; Desoer and Chen, 1981; Garcia and Morari, 1982, 1985a,b; Marchetti et al., 1983; Martin, 1981; Mehra and Rouhani, 1980; Ogunnaike, 1983; Reid et al., 1979, 1980, 1981; Rouhani and Mehra, 1982; Zames, 1981). As well as providing better regulatory behavior, model predictive control is useful for plant design and synthesis of control systems (Morari, 1983). The structure of model predictive control makes more apparent the limitations on achievable control quality which are inherent in the process and not due to the choice of control strategy.

MPHC, DMC, and other model predictive controllers use a linear plant model. This allows for an explicit control algorithm and also for the development of theorems which provide guidelines for controller tuning. Since most chemical plants are nonlinear, these advantages come at the cost of an inaccurate plant model. The robustness of model predictive control compensates for this inaccuracy. Robustness can be further enhanced by the addition of a filter on the input signal to the controller (Garcia and Morari, 1985a). The filter has a single parameter, which provides simple tuning and indicates when the linear model may need to be modified. Another approach to the problem of inaccurate plant models is to use nonlinear models within a model predictive control structure. Here we use a bilinear plant model.

Bilinear models have several advantages. They arise naturally in many chemical processes, where mass and energy balances contain products of flows, which are commonly used as manipulated variables, and temperatures or concentrations, which are commonly the controlled variables in a process. Bilinear models are more tractable mathematically than general nonlinear models. As an approximation to a general nonlinear plant, the bilinear model can provide a more accurate representation than the linear model. For a general nonlinear plant in which the control appears linearly, a dynamically equivalent bilinear model can be found (Lo, 1975).

Controller Design

The model predictive controller structure is shown in Figure 1. For convenience, we have used the notation and definitions of Internal Model Control (IMC) (Garcia and Morari, 1982, 1985a,b) where possible. IMC contains MPHC and DMC as special cases.

The plant to be controlled is assumed to be described by a discrete, bilinear model of the form

$$y^m = \sum_{i=1}^{m}\{\hat{a}_i y_{k-i} + \hat{b}_i y_{k-i} u_{k-i-\tau} + \hat{c}_i u_{k-i-\tau}\} \quad (1)$$

We use y_{k-i}, $i \geq 0$, to indicate the present or a past measured output, y^m to indicate the output of (1), and $\hat{y}_{k+i}$, $i \geq 1$ (defined below), to indicate a prediction of a future output. y^m is the prediction of the output at the present time, k, under the assumption of no disturbance.

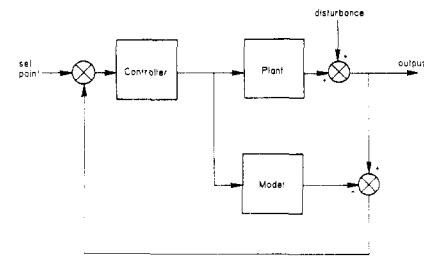

Figure 1. Model predictive control structure (Garcia and Morari, 1982).

The control algorithm is designed as the solution of the following optimization problem. At each sampling time, the next M control inputs are chosen to minimize the objective function

$$\Psi = \sum_{j=1}^{P}\{\gamma_j^2[s_{k+\tau+j} - \hat{y}_{k+\tau+j}]^2 + \beta_j^2 u_{k+j-1}^2\} \quad (2)$$

Because the present input u_k affects no outputs prior to $y_{k+\tau+1}$, the earliest set point considered in the objective function is $s_{k+\tau+1}$. This ensures a realizable control algorithm. The predicted values of the output are calculated recursively from model (1), and the estimated disturbance, $\hat{d}_k$, i.e.,

$$\hat{d}_k = y_k - y^m \quad (3)$$

$$\hat{y}_{k+1} = \sum_{i=1}^{m}\{\hat{a}_i y_{k+1-i} + \hat{b}_i y_{k+1-i} u_{k+1-i-\tau} + \hat{c}_i u_{k+1-i-\tau}\} + \hat{d}_k \quad (4)$$

$$\hat{y}_{k+j} = \sum_{i=1}^{j-1}\{\hat{a}_i \hat{y}_{k+j-i} + \hat{b}_i \hat{y}_{k+j-i} u_{k+j-i-\tau} + \hat{c}_i u_{k+j-i-\tau}\} + \sum_{i=j}^{m}\{\hat{a}_i y_{k+j-i} + \hat{b}_i y_{k+j-i} u_{k+j-i-\tau} + \hat{c}_i u_{k+j-i-\tau}\} + \hat{d}_k \quad (5)$$

for $j = 2, 3, \ldots, m$ and

$$\hat{y}_{k+j} = \sum_{i=1}^{m}\{\hat{a}_i \hat{y}_{k+j-i} + \hat{b}_i \hat{y}_{k+j-i} u_{k+j-i-\tau} + \hat{c}_i u_{k+j-i-\tau}\} + \hat{d}_k \quad (6)$$

for $j = m+1, m+2, \ldots, P$.

The optimization problem is nonlinear, and an explicit equation for the control input cannot be obtained. For the special case $P = M$, the following algebraic transformation allows an explicit calculation of the present control input, u_k. Define

$$\mathbf{A} = \begin{bmatrix} \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \cdots & \hat{a}_m \\ \hat{a}_2 & \hat{a}_3 & \hat{a}_4 & \cdots & 0 \\ \hat{a}_3 & \hat{a}_4 & & \cdots & 0 \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ \hat{a}_m & 0 & \cdots & & 0 \end{bmatrix} \quad (7)$$

$\mathbf{B}$ and $\mathbf{C}$ are similarly defined matrices containing the model coefficients $\hat{b}_i$ and $\hat{c}_i$:

$$\mathbf{Y}_k = [\hat{y}_{k+\tau-1}\ \hat{y}_{k+\tau-2} \cdots \hat{y}_{k+\tau-m}]^T \quad (8)$$

$$\mathbf{X}_k = [\hat{y}_{k+\tau-1}u_{k-1}\ \hat{y}_{k+\tau-2}u_{k-2} \cdots \hat{y}_{k+\tau-m}u_{k-m}]^T \quad (9)$$

$$\mathbf{U}_k = [u_{k-1}\ u_{k-2} \cdots u_{k-m}]^T \quad (10)$$

$$\alpha_k = \mathbf{A}\mathbf{Y}_k + \mathbf{B}\mathbf{X}_k + \mathbf{C}\mathbf{U}_k \quad (11)$$

where $\boldsymbol{g}_j$ = vector of length m with elements

$$(\boldsymbol{g}_j)_i = (0,\ i \neq j;\ 1,\ i=j) \tag{12}$$

$$q_k = \boldsymbol{g}_1 \alpha_k + \hat{d}_k \tag{13}$$

$$q_{j+k} = \sum_{i=1}^{j} q_{k+i-1}\hat{a}_{j-i+1} + \boldsymbol{g}^T{}_{j+1}\alpha_k + d_k \quad 1 \le j < m \tag{14}$$

$$q_{j+k} = \sum_{i-1}^{m} q_{k+j-i}\hat{a}_i + \hat{d}_k \qquad j \ge m \tag{15}$$

$$h_{k+1} = \hat{b}_1 q_k + \hat{c}_1 \tag{16}$$

$$h_{k+i} = (\hat{b}_1)^{i-1} h_1 \tag{17}$$

In (8) and (9), the predicted outputs $\hat{y}_{k+\tau-i}$ are replaced with the measured outputs $y_{k+\tau-i}$ for $i \ge \tau$. With these definitions, model (1) can be rewritten as a linear equation

$$\hat{y}_{k+\tau+i} = h_{k+i} m_{k+i} + q_{k+i} \tag{18}$$

where the new input variables m_{k+i} are complicated nonlinear functions of the actual inputs $u_k, u_{k+1}, \ldots, u_{k+M-1}$. No convenient formula for m has been found. They must be expressed recursively (Yeo, 1986); however, as seen below, the only equation needed is

$$m_{k+1} = u_k \tag{19}$$

If we define

$$\epsilon = [(s_{k+\tau+1} - q_{k+1}) \ \ldots \ (s_{k+\tau+P} - q_{k+P})]^T \tag{20}$$

$$\mathbf{M} = [m_{k+1}\ m_{k+2} \ \ldots \ m_{k+P}]^T \tag{21}$$

$$\mathbf{H} = \text{diag}\ [h_{k+1}\ h_{k+2} \ \ldots \ h_{k+P}] \tag{22}$$

$$\boldsymbol{\Gamma} = \text{diag}\ [\gamma_1{}^2\ \gamma_2{}^2 \ \ldots \ \gamma_P{}^2] \tag{23}$$

$$\beta = \text{diag}\ [\beta_1{}^2\ \beta_2{}^2 \ \ldots \ \beta_P{}^2] \tag{24}$$

The objective function (2) becomes

$$\Psi = [\epsilon - \mathbf{HM}]^T \boldsymbol{\Gamma} [\epsilon - \mathbf{HM}] + \mathbf{M}^T \beta \mathbf{M} \tag{25}$$

The solution is then

$$\mathbf{M} = [\mathbf{H}^T \boldsymbol{\Gamma} \mathbf{H} + \beta]^{-1} \mathbf{H}^T \boldsymbol{\Gamma} \epsilon \tag{26}$$

It is easy to show that the solution $\mathbf{M}$ of the optimization problem (25) is also a solution of the original problem (2) (Yeo, 1986).

The problem is formulated to select a sequence of future control inputs. In practice a new sequence is calculated at each sampling instant in order to use the most recent set points and disturbance estimates. Therefore, it is necessary to calculate explicitly only $u_k = m_{k+1}$.

After appropriate algebraic manipulation, the control algorithm can be put into the explicit form

$$u_k = \{\gamma_1{}^2(\hat{b}_1\hat{y}_{k+\tau} + \hat{c}_1)[s_{k+\tau+1} - (y_k - y^m) - \hat{a}_1\hat{y}_{k+\tau} - \sum_{i=2}^{m}(\hat{a}_i\hat{y}_{k+\tau+1-i} + \hat{b}_i\hat{y}_{k+\tau+1-i}u_{k+1-i} + \hat{c}_i u_{k+1-i})]\}/\{\gamma_1{}^2(\hat{b}_1\hat{y}_{k+\tau} + \hat{c}_1)^2 + \beta_1{}^2\} \tag{27}$$

The predicted outputs, $y_{k+\tau+1-i}$, $i < \tau$, are calculated from (3)–(6). In (27) and (28) below, the predicted outputs $\hat{y}_{k+\tau-1-i}$ are replaced with the measured outputs $y_{k+\tau-1-i}$ for $i \ge \tau$. Notice that the only tuning parameters which appear in the control algorithm are γ_1 and β_1. This is a consequence of requiring $P = M$ and occurs in IMC as well as here. The algorithm can be further simplified if the original objective function is divided by $\gamma_1{}^2$. The control algorithm becomes

$$u_k = \{(\hat{b}_1\hat{y}_{k+\tau} + \hat{c}_1)[s_{k+\tau+1} - (y_k - y^m) - \hat{a}_1\hat{y}_{k+\tau} - \sum_{i=2}^{m}(\hat{a}_i\hat{y}_{k+\tau+1-i} + \hat{b}_i\hat{y}_{k+\tau+1-i}u_{k+1-i} + \hat{c}_i u_{k+1-i})]\}/\{(\hat{b}_1\hat{y}_{k+\tau} + \hat{c}_1)^2 + \beta\} \tag{28}$$

where

$$\beta = (\beta_1{}^2/\gamma_1{}^2) \ge 0 \tag{29}$$

is the single tuning parameter.

The final result is that the control structure and predictive nature of the control algorithm of linear model predictive control has been retained in bilinear model predictive control without increasing the computer speed or memory requirements. This has come at a cost of reduced flexibility in that the number of tuning parameters is reduced from $2 + M + P$ to 1. (This assumes that the sampling time is fixed and not available as a tuning parameter.) This reduction in parameters also provides an advantage in that tuning is much simpler.

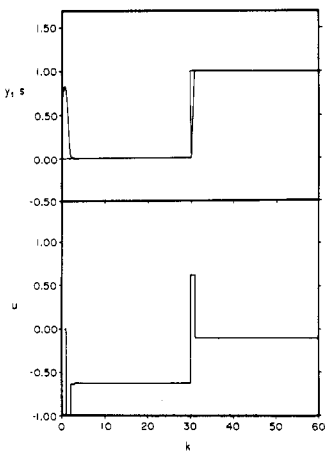

Figure 2. Deadbeat control of first-order plant.

Tuning

The effect of the tuning parameter β can be seen by examination of the objective function (2). If $\beta = 0$ ($\beta_1{}^2 = 0$), no penalty is assessed for the control input; hence, the control algorithm will be the inverse of the plant model. If the model is perfect, this will yield deadbeat control of the sampled outputs. If the plant model is nonminimum phase, the control algorithm will be unstable. Even if the control algorithm is stable, deadbeat control often requires excessive control action and gives strong oscillations in the output between sampling instants. As β is increased from zero, more weight is placed upon the control action. This decreases the control action. The effect of tuning is illustrated in the following examples.

In each of the examples in the paper, the system is initially at steady state ($y_s = 0$, $u_s = 0$) with no disturbance. At $k = 0$, a constant disturbance ($d_k = 0.5$) is introduced. At $k = 30$, a step change in set point (to $s_k = 1.0$) is introduced. The control input is constrained such that $|u_k| \le 1.0$.

Example 1. The plant is

$$y_k = 0.6y_{k-1} + 0.2y_{k-1}u_{k-1} + 0.8u_{k-1} + d_k \tag{30}$$

A perfect model is used for control. Figure 2 shows the results for $\beta = 0$ (deadbeat control). Since the plant is first-order, deadbeat control is satisfactory.

Example 2. The plant is

$$y_k = 0.6y_{k-1} + 0.3y_{k-2} + 0.3y_{k-1}u_{k-1} + 0.2y_{k-2}u_{k-2} + 0.5u_{k-1} + 0.4u_{k-2} + d_k \tag{31}$$

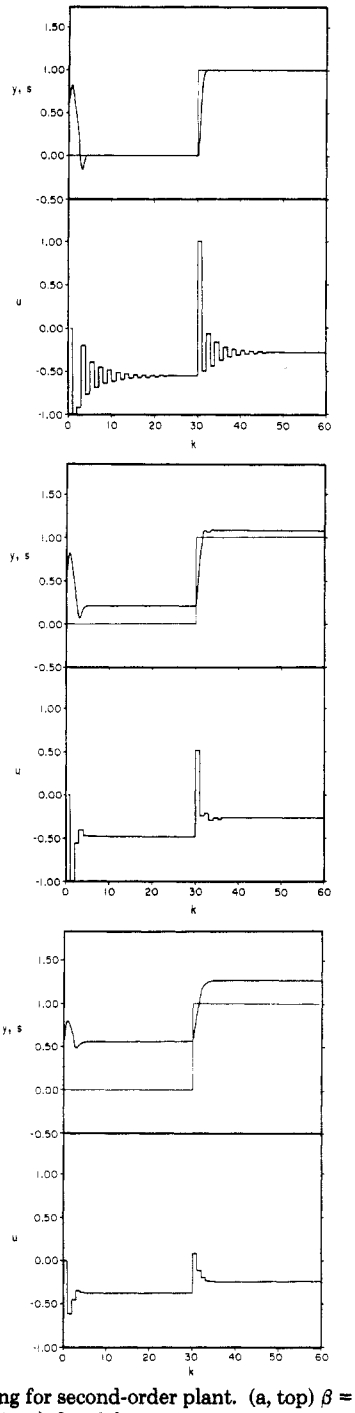

Figure 3. Tuning for second-order plant. (a, top) $\beta = 0$. (b, middle) $\beta = 0.5$. (c, bottom) $\beta = 1.0$.

A perfect model is used for control. Figure 3 shows the results for $\beta = 0$, 0.5, and 1.0. Deadbeat control is satisfactory for the output but requires strong oscillations in input. As β is increased, the input oscillations are reduced, but offset is introduced and increased.

Example 3. The plant is

$$y_k = 0.6y_{k-1} + 0.4y_{k-2} + 0.2y_{k-1}u_{k-1} + 0.3y_{k-2}u_{k-2} + 0.4u_{k-1} + 0.5u_{k-2} + d_k \quad (32)$$

A perfect model is used for control. Figure 4 shows the results for $\beta = 0$, 0.2, and 0.5. The plant inverse is unstable; therefore, deadbeat control, which uses the plant inverse for the control algorithm, is unstable. As β is increased, the system is stabilized, but offset is introduced.

Offset Compensation

In the previous examples, the tuning parameter β was

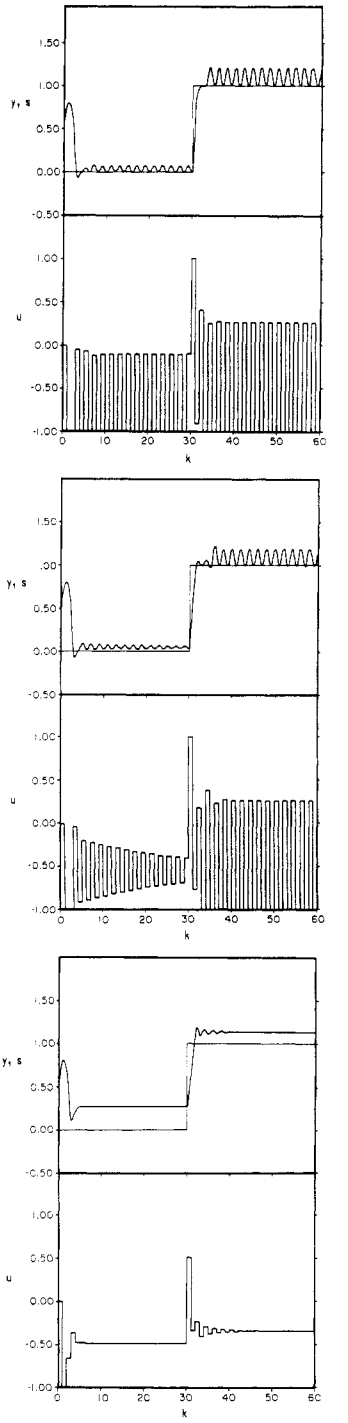

Figure 4. Tuning for plant with unstable inverse. (a, top) $\beta = 0$. (b, middle) $\beta = 0.2$. (c, bottom) $\beta = 0.5$.

first increased to produce a stable control algorithm and then further increased to produce a less oscillatory response. The more desirable transient response came at the cost of steady-state offset. Here we show that steady-state offset always occurs if $\beta \neq 0$ and define offset compensation to eliminate offset if $\beta \neq 0$. We define

$$\hat{A} = \sum_{i=1}^{m} \hat{a}_i \qquad \hat{B} = \sum_{i=1}^{m} \hat{b}_i \qquad \hat{C} = \sum_{i=1}^{m} \hat{c}_i \quad (33)$$

At steady state, (28) becomes

$$u_s = (\hat{b}_1\hat{y}_s + \hat{c}_1)\{s_s - (y_s - y_s^m) - \hat{A}\hat{y}_s - (\hat{B} - \hat{b}_1)\hat{y}_s u_s - (\hat{C} - \hat{c}_1)u_s\}/\{(\hat{b}_1\hat{y}_s + \hat{c}_1)^2 + \beta\} \quad (34)$$

and (1) becomes

$$y_s^m = \hat{A}y_s + \hat{B}y_s u_s + \hat{C}u_s \quad (35)$$

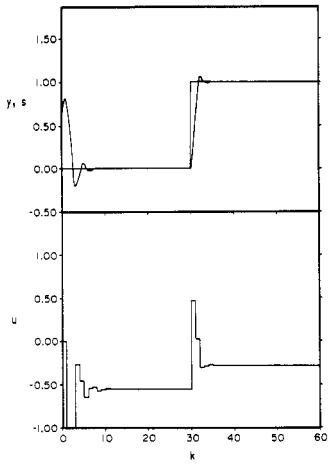

Figure 5. Offset compensation for second-order plant with $\beta = 1.0$.

Recursive solution of (3)–(6) shows that at steady state $y_{k+1} = y_{k+2} = ... = y_{k+\tau} = \hat{y}_s = y_s$. After substitution of (35) into (34) and rearrangement, the steady-state error can be written as

$$s_s - y_s = \beta u_s/(\hat{b}_1 y_s + \hat{c}_1) \quad (36)$$

Therefore, in general there is no offset for $\beta = 0$, and there is offset for $\beta \neq 0$. We have already seen that it is not desirable always to choose $\beta = 0$. Thus, we introduce offset compensation.

The control algorithm (28) is modified with the introduction of an offset compensation parameter ζ:

$$u_k = \{(\hat{b}_1\hat{y}_{k+\tau} + \hat{c}_1)[\zeta s_{k+\tau+1} - \zeta(y_k - y^m) - (\hat{a}_1 - \hat{A} + \zeta\hat{A})\hat{y}_{k+\tau} - \sum_{i=2}^{m}(\hat{a}_i\hat{y}_{k+\tau+1-i} + \hat{b}_i\hat{y}_{k+\tau+1-i}u_{k+1-i} + \hat{c}_i u_{k+1-i})]\}/\{(\hat{b}_1\hat{y}_{k+\tau} + \hat{c}_1)^2 + \beta\} \quad (37)$$

At steady state, (37) becomes

$$u_s = (\hat{b}_1 y_s + \hat{c}_1)\{\zeta s_s - \zeta(y_s - y_s^m) - \zeta\hat{A}y_s - (\hat{B} - \hat{b}_1)y_s u_s - (\hat{C} - \hat{c}_1)u_s\}/\{(\hat{b}_1 y_s + \hat{c}_1)^2 + \beta\} \quad (38)$$

Substitution of (35) into (38) and rearrangement yield a steady-state error of

$$s_s - y_s = \{1 - \zeta + \beta/[(\hat{b}_1 y_s + \hat{c}_1)(\hat{B}y_s + \hat{C})]\}(\hat{B}y_s + \hat{C})u_s/\zeta \quad (39)$$

Therefore, offset compensation can be achieved for

$$\zeta = 1 + \beta/\{(\hat{b}_1 s_s + \hat{c}_1)(\hat{B}s_s + \hat{C})\} \quad (40)$$

If $\beta = 0$, $\zeta = 1$, as expected. If $b_i = 0$ for all i, i.e., the model is linear, ζ becomes a constant independent of the steady state, as expected. If $b_i = 0$ and $c_i = 0$ for all i, i.e., the model is an impulse model of a linear plant as is used in linear model predictive control, (40) reduces to the identical offset compensator introduced by Garcia and Morari (1982) in IMC. The offset compensator yields a cubic steady-state equation for a bilinear process. We have yet been unable to show that the no offset solution is the only reachable one. In all simulation testing thus far, no offset has been obtained.

Figures 5 and 6 show the effect of the offset compensator for examples 2 and 3, respectively. The offset is eliminated. The dynamic response is also affected. Comparison with Figures 3c and 4c shows that the response is more oscillatory with the compensator. This effect is shown below to be more important when the model is not perfect.

Incorrect Models

In the previous examples, a perfect model was used in order to demonstrate the effects of the tuning parameter and offset compensator clearly. Here we show that the control deteriorates if the model is incorrect. A simple

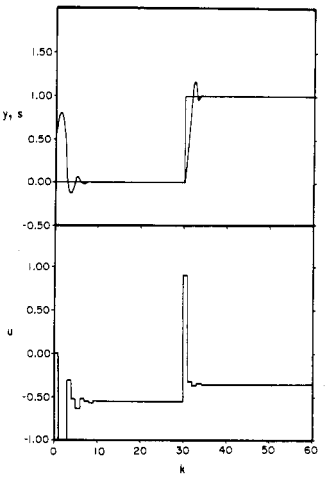

Figure 6. Offset compensation for plant with unstable inverse with $\beta = 0.5$.

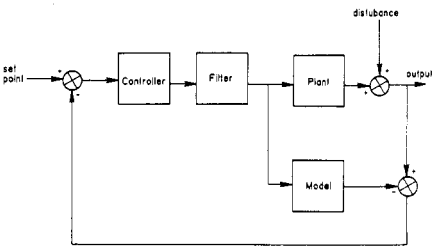

Figure 7. Model predictive control with filter.

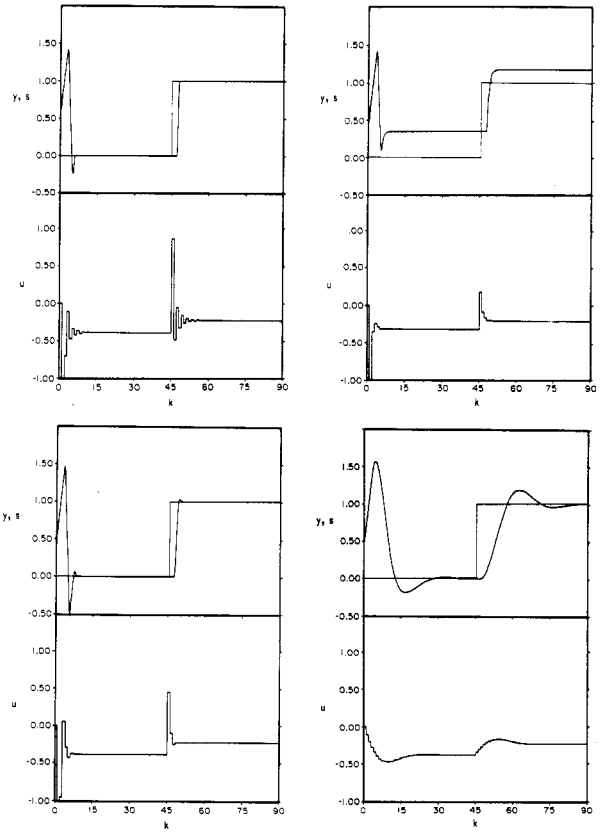

Figure 8. Tuning, filter, and offset compensation for perfect model. (a, top left) $\beta = 0$, no filter, no offset compensation. (b, top right) $\beta = 1.0$, no filter, no offset compensation. (c, bottom left) $\beta = 1.0$ with offset compensation and no filter. (d, bottom right) $\beta = 1.0$ with filter and offset compensation.

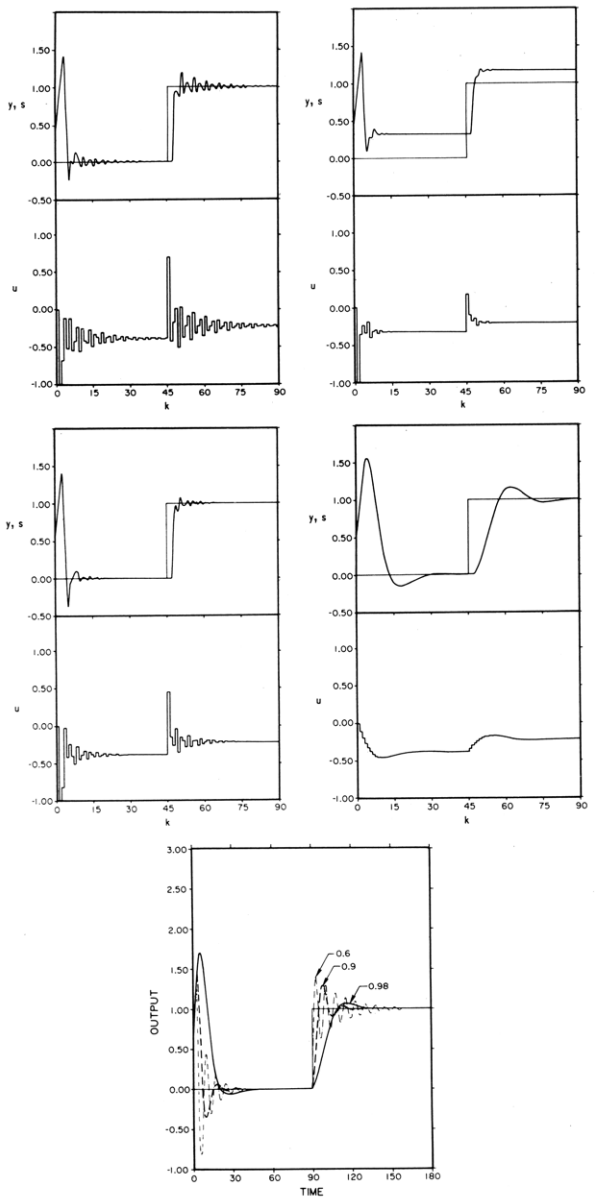

Figure 9. Tuning, filter, and offset compensation for incorrect model parameters. (a, top left) $\beta = 0$, no filter, no offset compensation. (b, top right) $\beta = 1.0$, no filter, no offset compensation. (c, middle left) $\beta = 1.0$, offset compensation, no filter. (d, middle right) $\beta = 1.0$, filter, offset compensation. (e, bottom) $\beta = 1.0$, offset compensation, filter parameters 0.6, 0.9, and 0.98.

filter is used to correct for the effect of model inaccuracy (Figure 7). The equation for the filter is

$$u_k = (1 - \alpha)u_k^* + \alpha u_{k-1} \tag{41}$$

where u_k^* is the unfiltered input from the control algorithm (37).

Example 4. The plant is

$$y_k = 0.6y_{k-1} + 0.3y_{k-2} + 0.3y_{k-1}u_{k-3} + 0.2y_{k-2}u_{k-4} + 0.8u_{k-3} + 0.5u_{k-4} + d_k \tag{42}$$

Because the model inaccuracy slows the speed of the system response, the step change in set point is introduced at $k = 45$ or $k = 90$; otherwise, the conditions are the same as in the previous examples. The filter parameter in all cases except case 2 is 0.95. No attempt was made to optimize the response through adjustment of the filter parameter. For case 2, several values of α were used to illustrate the effect of the filter parameter.

Case 1: Perfect Model. This case is included for comparison. The results are shown in Figure 8. The system is easy to control with the perfect model.

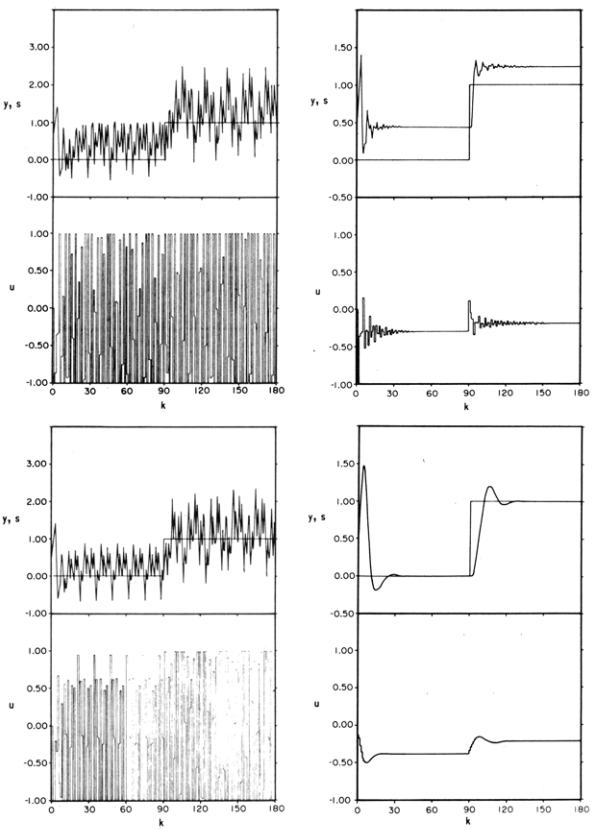

Figure 10. Tuning, filter, and offset compensation for incorrect model order. (a, top left) $\beta = 0$, no filter, no offset compensation. (b, top right) $\beta = 1.2$, no filter, no offset compensation. (c, bottom left) $\beta = 2.5$, offset compensation, no filter. (d, bottom right) $\beta = 1.2$, filter, offset compensation.

Case 2: Incorrect Parameters. The model is

$$y^m = 0.7y_{k-1} + 0.21y_{k-2} + 0.25y_{k-1}u_{k-3} + 0.12y_{k-2}u_{k-4} + 0.91u_{k-3} + 0.4u_{k-4} \tag{43}$$

The results are shown in Figure 9. The perfect controller ($\beta = 0$, $\alpha = 0$) is seen to be very sensitive to the incorrect model (compare Figures 8a and 9a). An increase in β smooths the response (Figure 9b). The response with the offset compensator is much more oscillatory and takes much longer to reach steady state with the incorrect model (compare Figures 8c and 9c). The use of the filter smooths the response (Figure 9d), but it does not take significantly longer to reach steady state than it does when the compensator is used without the filter. Figure 9e illustrates the effect of the filter parameter. As α is increased, the response becomes more damped and more sluggish.

Case 3: Incorrect Order. The model is

$$y^m = 0.94y_{k-1} + 0.23y_{k-1}u_{k-3} + 0.88u_{k-3} \tag{44}$$

The results are shown in Figure 10. The mismatch between the plant and the model is so large that the perfect controller is unstable. The system is stabilized by an increase in β, but the introduction of the offset compensator destabilizes the system even though a larger β is used. When the filter is used with the offset compensator, the response is stabilized.

Case 4: Incorrect Time Delay. The model is

$$y^m = 0.6y_{k-1} + 0.3y_{k-2} + 0.3y_{k-1}u_{k-2} + 0.2y_{k-2}u_{k-3} + 0.8u_{k-2} + 0.5u_{k-3} \tag{45}$$

The results are shown in Figure 11. The results are similar

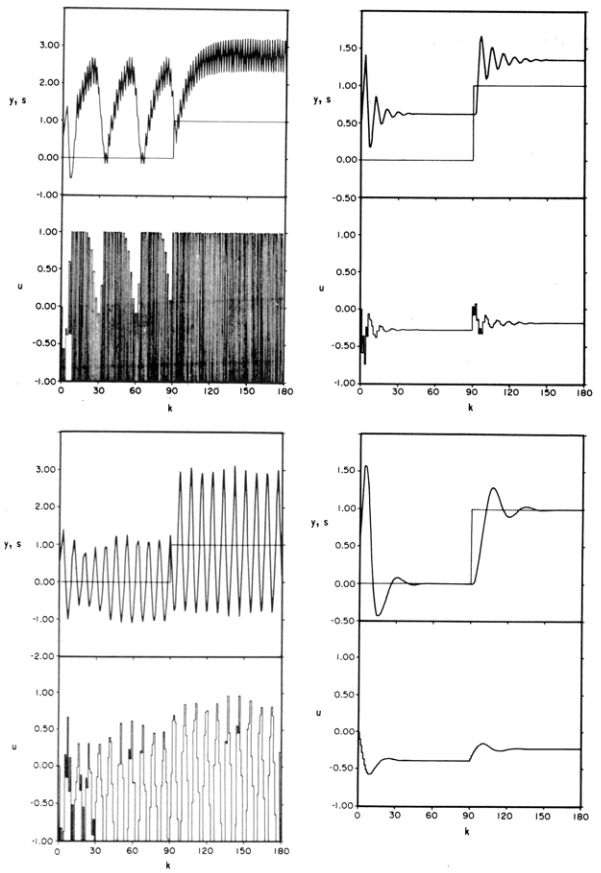

Figure 11. Tuning, filter, and offset compensation for incorrect model time delay. (a, top left) $\beta = 0$, no filter, no offset compensation. (b, top right) $\beta = 1.5$, no filter, no offset compensation. (c, bottom left) $\beta = 1.5$, offset compensation, no filter. (d, bottom right) $\beta = 1.5$, filter, offset compensation.

to those for the case with incorrect model order.

Tuning with Incorrect Model

The examples suggest the following two-step tuning procedure be used.

(1) The control algorithm parameter β should be selected based on the assumption that the model is perfect. This selection can be done by off-line simulation.

(2) The filter parameter α should be selected to correct for the model inaccuracies. This selection must be done on-line. The need for a large value of α indicates that identification of a new model may be warranted.

Constraints

In this work, we have treated input constraint violations by simply setting the input equal to the constraint. This has been shown to induce limit cycles for some nonlinear systems (Chang and Chen, 1984). For linear model predictive control system stability is maintained as long as the constrained input is used in the model (1) and prediction equations (3)–(6) (Garcia and Morari, 1985a; Mehra et al., 1980). We have been unable to prove that stability is also maintained in the bilinear case, but we have been unable to induce limit cycles in simulation.

Feedforward Control

If it is possible to measure some or all of the disturbances, the predictive model controller structure allows the addition of a feedforward path naturally (Figure 12). If the dynamics between the measured disturbances and the outputs are known, these may also be incorporated into feedforward control.

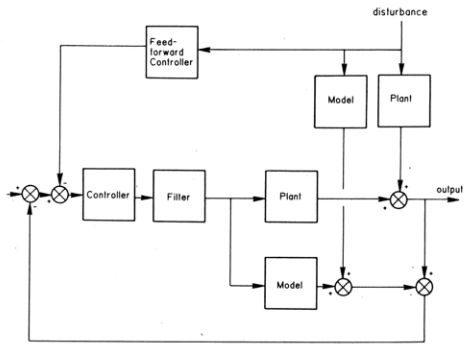

Figure 12. Feedforward model predictive control (Garcia and Morari, 1985a).

Summary

Bilinear model predictive control has been developed for single-input–single-output systems. The controller uses a bilinear plant model, which makes possible a greater range of accurate representation of the plant than is possible with a linear model. The transparent structure of linear model predictive control is retained with an explicit control algorithm. Offset compensation is provided to correct for the effects of unmeasured disturbances and model inaccuracies at steady state. A filter is used to correct for the dynamic effects of model inaccuracies. Controller tuning is simple through the two-step adjustment of two parameters. We are currently developing adaptive and multivariable versions of bilinear model predictive control.

Acknowledgment

We acknowledge the support of this work by the Auburn University Engineering Experiment Station and the Auburn University Advanced Manufacturing Technology Center.

Literature Cited

Astrom, K. J. *Automatica* **1980**, *16*, 313–315.
Callier, F. M.; Desoer, C. A. *Multivariable Feedback Systems*; Springer-Verlag: New York, 1982.
Chang, H.-C.; Chen, L.-H. *Chem. Eng. Sci.* **1984**, *39*, 1127–1142.
Cutler, C. L.; Ramaker, B. L. "Dynamic Matrix Control—A Computer Control Algorithm", Joint Automatic Control Conference, San Francisco, 1980, paper WP5-B.
Desoer, C. A.; Chen, M. J. *IEEE Trans. Autom. Control* **1981**, *AC-26*, 408–415.
Engrand, J. C. "Applications of Multivariable Control in a Refinery and Implementation of a Dedicated Minicomputer", Joint Automatic Control Conference, San Francisco, 1980, paper FA9-D.
Garcia, C. E.; Morari, M. *Ind. Eng. Chem. Process Des. Dev.* **1982**, *21*, 308–323.
Garcia, C. E.; Morari, M. *Ind. Eng. Chem. Process Des. Dev.* **1985a**, *24*, 472–484.
Garcia, C. E.; Morari, M. *Ind. Eng. Chem. Process Des. Dev.* **1985b**, *24*, 484–494.
Lebourgeois, F. "IDCOM Applications and Experiences on a PVC Production Plant", Joint Automatic Control Conference, San Francisco, 1980, paper FA9-C.
Lo, J. T.-H. *SIAM J. Control* **1975**, *13*, 879–885.
Marchetti, J. L.; Mellichamp, D. A.; Seborg, D. E. *Ind. Eng. Chem. Process Des. Dev.* **1983**, *13*, 879–885.
Martin, G. D. *AIChE J.* **1981**, *27*, 748–753.
Mehra, R. K.; Eterno, J. S. "Model Algorithmic Control for Electric Power Plant", IEEE Conference on Decision and Control, Albuquerque, 1980, pp 280–285.
Mehra, R. K.; Rouhani, R. "Theoretical Considerations on Model Algorithmic Control for Nonminimum Phase Systems", Joint Automatic Control Conference, San Francisco, 1980, paper TA8-B.
Mehra, R. K.; Rouhani, R.; Praly, L. "New Theoretical Developments in Multivariable Predictive Algorithmic Control", Joint Automatic Control Conference, San Francisco, 1980, paper FA9-B.

Morari, M. *Chem. Eng. Sci.* **1983**, *38*, 1881-1891.
Ogunnaike, B. A. "A Statistical Appreciation of Dynamic Matrix Control", American Control Conference, San Francisco, 1983, pp 1126-1131.
Reid, J. G.; Chafin, D. A.; Silverthorn, J. T. "Output Predictive Algorithmic Control: Precision Tracking with Application to Terrain Following", Joint Automatic Control Conference, San Francisco, 1980, paper FA-9F.
Reid, J. G.; Chafin, D. E.; Silverthorn, J. T. *J. Guidance Control* **1981**, *4*, 502-509.
Reid, J. G.; Mehra, R. K.; Kirkwood, E. "Robustness Properties of Output Predictive Deadbeat Control: SISO Case", IEEE Conference on Decision and Control, Fort Lauderdale, 1979, pp 307-314.
Richalet, J. "General Principles of Scenario Predictive Control Techniques", Joint Automatic Control Conference, San Francisco, 1980, paper FA9-A.
Richalet, J. A.; Rault, A.; Testud, J. D.; Papon, J. *Automatica* **1978**, *14*, 413-428.
Rouhani, R.; Mehra, R. K. *Automatica* **1982**, *18*, 401-414.
Yeo, Y. K. "Adaptive and Non-adaptive Bilinear Model Predictive Control", Ph.D. Dissertation, Auburn University, Auburn, AL, 1986.
Zames, G. *IEEE Trans. Autom. Control* **1981**, *AC-26*, 301-320.

Received for review October 15, 1985
Revised manuscript received July 13, 1987
Accepted August 19, 1987

A Modified UNIFAC Group-Contribution Model for Prediction of Phase Equilibria and Heats of Mixing

Bent L. Larsen
DECHEMA, Frankfurt (M), BRD

Peter Rasmussen and Aage Fredenslund*
Instituttet for Kemiteknik, The Technical University of Denmark, DK-2800 Lyngby, Denmark

The Modified UNIFAC model for predicting activity coefficients presented in this work is based on the well-known UNIFAC model. Two changes are introduced in Modified UNIFAC: (1) the group-interaction parameters have been made temperature-dependent and (2) the combinatorial term is slightly modified. Group-interaction parameters have been determined for 21 different main groups. It is shown that Modified UNIFAC gives somewhat better predictions of vapor-liquid equilibria than does UNIFAC, while the predictions of excess enthalpies are much improved. Hence, Modified UNIFAC has a better built-in temperature dependence than UNIFAC. Modified UNIFAC gives in general predictions of liquid-liquid equilibria (LLE) of the same quality as the original UNIFAC with LLE-based parameters. A Modified UNIQUAC model corresponding to the Modified UNIFAC model is also presented.

The UNIFAC group-contribution method is widely applied to the prediction of liquid-phase activity coefficients in nonelectrolyte, nonpolymeric mixtures at low to moderate pressures and at temperatures between 275 and 425 K. The parameters needed for the use of UNIFAC are group volumes (R_k), group surface areas (Q_k), and group-interaction parameters (a_{mn} and a_{nm}). Extensive tables with revised and updated values for 44 commonly needed groups are published in collaboration between University of Dortmund (BRD) and Technical University of Denmark (DK); see Gmehling et al. (1982), Macedo et al. (1983), and Tiegs et al. (1987).

The uses and shortcomings of UNIFAC are reviewed by Fredenslund and Rasmussen (1985).

One of the shortcomings of UNIFAC is that the built-in temperature dependence is not good enough for simultaneous prediction of vapor-liquid equilibria (VLE) and excess enthalpies (H^E).

This paper presents a Modified UNIFAC model which permits simultaneous representation and prediction of VLE and H^E through the introduction of temperature-dependent group-interaction parameters. In addition, the combinatorial term is modified according to the ideas of Kikic et al. (1980).

The Modified UNIQUAC and UNIFAC Models

As in the UNIQUAC (Abrams and Prausnitz, 1975) and UNIFAC (Fredenslund et al., 1977) models, the excess Gibbs function is calculated as a sum of a *combinatorial* and a *residual* contribution:

$$g^E = g_c^E + g_r^E \tag{1}$$

The Combinatorial Term. It is assumed that for mixtures containing alkanes only, the residual excess Gibbs function is zero. Thus, mixtures containing different alkanes are described by the combinatorial term only. We must then seek an expression for the combinatorial term, which can describe as well as possible the activity coefficients of alkanes in mixtures with other alkanes.

Combinatorial expressions can be obtained from statistical mechanical arguments. The Flory-Huggins combinatorial can be written as

$$\ln \gamma_i^c = \ln\left(\frac{\Phi_i}{x_i}\right) + 1 - \frac{\Phi_i}{x_i} \tag{2}$$

while the Staverman-Guggenheim combinatorial, which is used in UNIQUAC and UNIFAC, is

$$\ln \gamma_i^c = \ln\left(\frac{\Phi_i}{x_i}\right) + 1 - \frac{\Phi_i}{x_i} - \frac{z}{2}q_i\left(\ln\left(\frac{\Phi_i}{\theta_i}\right) + 1 - \frac{\Phi_i}{\theta_i}\right) \tag{3}$$

The molecular volume fraction (Φ_i) and surface area fractions (θ_i) are defined in terms of the segment numbers (r) and contact number parameters [$(z/2)q$] as

$$\Phi_i = \frac{x_i r_i}{\sum_j x_j r_j} \quad \text{and} \quad \theta_i = \frac{x_i \frac{z}{2} q_i}{\sum_j x_j \frac{z}{2} q_j} \tag{4}$$

0888-5885/87/2626-2274$01.50/0