RLS-BASED ADAPTIVE CONTROL OF A BIODIESEL TRANSESTERIFICATION REACTOR

HO YONG KUEN\textsuperscript{a,1}, FAROUQ S. MJALLI\textsuperscript{b}, YEOH HAK KOON\textsuperscript{a}

\textsuperscript{a}Chemical Engineering Dept., Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia
\textsuperscript{b}Petroleum & Chemical Engineering Department, Sultan Qaboos University, Muscat 123, Oman

ABSTRACT

Proper control of biodiesel reactors poses a number of challenges. These arise from the presence of multiple chemical reactions, the complex heat and mass transfer characteristics, and the highly nonlinear dynamics. In this work, a new adaptive control scheme was implemented to automatically tune the controller based on the most recent updated process dynamics. This scheme demonstrates the powerful integration of an online process modeling tool (Recursive Least Squares, RLS) into a renowned and yet simple model-based controller design method (Internal Model Control, IMC). Two adaptive control loops were designed, in which the sampling time of the RLS algorithm and the IMC closed loop time constant were determined by constrained optimization with the genetic algorithm. Comparison with conventional PID controllers revealed the superiority of the new adaptive control scheme in set point tracking. Good disturbance rejection properties were also demonstrated by the new adaptive control scheme. The results attained in this work demonstrate that a good adaptive control scheme can be implemented for the biodiesel transesterification reactors with minimal knowledge about the process model.

Keywords: Adaptive controller, Biodiesel, Internal Model Control, Recursive Least Squares, Transesterification reactor

\textsuperscript{1}Corresponding author, vongkuen.ho@gmail.com, Tel: +6012-631-3536
1. INTRODUCTION

The transesterification reactor is regarded as the heart of the entire production process in any biodiesel production facility. The quality of the biodiesel produced is closely related to the performance of these reactors. However, the operation of biodiesel reactors entails many practical difficulties in meeting product specifications and operation limits\cite{1}. Moreover, nonlinearities arise from fluctuations of reactant concentration, reactant temperature, coolant temperature, ambient temperature, instrumentation noise or miss-calibration. Hence, a fundamental control problem arises in selecting a process control strategy to handle significant process changes which can occur but cannot be measured or anticipated\cite{2}. Under circumstances where process uncertainties are huge, the conventional solution is to implement a fixed controller tuning for the worst case scenario. However, controller tuning which caters to the worst case scenario might not be suitable for nominal operating conditions. For instance, controllers tuned based on the event of a stirrer failure might not be viable for normal operations, although stirrer failure remains a possible operating event. In view of these, an advanced control scheme which can cater to unexpected operating events must be designed and implemented to optimize the performance of biodiesel reactors.

Among the many good control algorithms available to date, adaptive control schemes offer systematic and flexible approaches for dealing with uncertainties, nonlinearities, and time-varying process parameters. Within the multitude of adaptive control schemes, the self-tuning approach has received the most attention in the past decades\cite{3-5}. In this approach, the Recursive Least Squares (RLS) algorithm is usually used online to model the process, followed by re-tuning of the controller based on the estimated process model parameters. The ability of RLS in processing real time data to produce sequential updates of process model parameters is crucial to the implementation of adaptive control. Many successful implementations of RLS-based adaptive control for chemical reactors and some difficult processes are available in the literature\cite{6-20}.

Although RLS-based adaptive control strategies were implemented in many applications of chemical reactors, little work was done specifically on the control of transesterification reactors. In the previous work of Mjalli et al.\cite{1}, a rigorous mechanistic model for the transesterification reactor was developed and validated with experimental data. The validated process model was utilized to control the process using an implementation of a Multiple Model Adaptive Control (MMAC) scheme, with Internal Model Control (IMC) being the method of controller design. However, this approach is limited by the amount of local linear models available to the controller. Moreover, a huge amount of local linear models, hence considerable effort, are required to accurately represent a process that exhibits extreme nonlinearity. Following this, Mjalli et al.\cite{21} proposed an online recursive modeling strategy to address the
issue of cumbersome offline modeling involved in the MMAC scheme. In this case, a black-box Artificial Neural Networks (ANN) modeling technique was used to model the process offline. Instantaneous linearization algorithm was then applied to the developed offline model to produce its linear equivalent which can be used further in controller design. This modeling approach, however, is not fully automated as it involves the development of an offline model using the ANN, of which the procedures are not widely known to a typical process and control engineer. Hence, the motivation of this work is to alleviate the above mentioned practical implementation issues by focusing on the synthesis and design of a fully automated RLS-based adaptive controller, while selecting the simple IMC as the main controller structure.

The essence of this work is the synergistic combination of the modeling capability of the RLS algorithm and the well known IMC method of controller design. The RLS is used to model the process online, and the IMC controller settings will be updated according to the most recent description of the process dynamics. With this, the controller will be able to adapt to operational variations and unexpected events in the biodiesel reactor. In short, this work contributes to the design and implementation of a more practical self-tuning adaptive controller with lesser user intervention specifically on the biodiesel reactor, where such application was not considered previously. As this work focuses on the implementation of the RLS-based adaptive controller, readers interested in the modeling of the transesterification reactor are referred to the work of Mjalli et al.\cite{1} for details. In the following sections, necessary equations on the RLS algorithm and the IMC method for controller design are discussed, followed by a comparison of the performance of different RLS algorithms in identifying time varying system. The suitability of the RLS algorithm in identifying the process model parameters in a transesterification reactor are also demonstrated, prior to coupling with a controller in closed loop. The RLS-based adaptive IMC controller is finally tuned and tested for its performance. Comparisons of its performance with that of a conventional PID controller are also included.

2. ONLINE PROCESS MODELING

For a particular adaptive control scheme to be successful, process modeling plays a key role in capturing the varying dynamics of the system as the operational process parameters move from one region to another. For this purpose, recursive system identification methods are used to estimate the process model parameters, given the input and output data.

Ljung and Söderström\cite{22} provided an in-depth review of different recursive system identification methods. Due to its simplicity and rapid convergence when properly applied, the RLS algorithm is regarded as the most popular online parameter estimation technique used in
adaptive control\cite{4}. The RLS algorithm is based on the minimization of the sum of squared prediction errors, where estimated process model parameters are improved progressively with each new process data acquired.

Consider a Single Input Single Output (SISO) process described by the general higher order Auto-Regressive eXogenous (ARX) model:

$$y(k) = z^T(k)\theta(k) + \xi(k)$$

(1)

where the regressor vector $z$ and the true system parameter vector $\theta$ are defined as:

$$z^T(k) = [y(k-1), y(k-2), \ldots, y(k-n_a), u(k-1), u(k-2), \ldots, u(k-n_b), 1]$$

$$\theta^T(k) = [a_1, a_2, \ldots, a_{n_a}, b_1, b_2, \ldots, b_{n_b}, d]$$

(2)

In Eqns. (1) and (2), $y$ is the process output, $u$ is the process input, $\xi$ is a stochastic noise variable (random variable with normal distribution and zero mean), and $k$ is a nonnegative integer which denotes the sampling instant, $k = 0, 1, 2, \ldots$ In Eqn. (2), $n_a$ and $n_b$ are known positive integers.

The conventional form of the RLS algorithm\cite{22} is given in Eqns. (3) - (6):

$$L(k+1) = \frac{P(k)z(k+1)}{1 + z^T(k+1)P(k)z(k+1)}$$

(3)

$$P(k+1) = P(k) - L(k+1)z^T(k+1)P(k)$$

(4)

$$\epsilon(k+1) = y(k+1) - z^T(k+1)\hat{\theta}(k)$$

(5)

$$\hat{\theta}(k+1) = \hat{\theta}(k) + L(k+1)\epsilon(k+1)$$

(6)

where $L$ is the Kalman gain, $P$ is called the covariance matrix of the prediction error and $\epsilon$ is the prediction error. Eqns. (3) - (6) are computed recursively to predict the process model parameters, $\hat{\theta}$, based on the minimization of the objective function, $J$\cite{22, 23}:

$$J(k) = \sum_{i=1}^{k} \epsilon_i^2$$

(7)
However, the form of algorithm and objective function given in Eqns. (3) - (7) is best suited for linear systems only, and when implemented for the purpose of tracking systems with time-varying parameters, the algorithm loses its sensitivity in the long run. One practical way to maintain the adaptation quality of the RLS algorithm is to emphasize more on recent measurements by an exponential weighting factor $\lambda \in (0, 1]$ (also called the ‘forgetting factor’) in the objective function:

$$J(k) = \sum_{i=1}^{k} \lambda^{k-i} \varepsilon_i^2$$  \hspace{1cm} (8)

When $\lambda = 1$, no exponential forgetting is being implemented, and hence, Eqn. (8) reduces to Eqn. (7). With exponential forgetting, Eqns. (3) - (4) become:

$$L(k+1) = \frac{P(k)z(k+1)}{\lambda(k+1) + z^T(k+1)P(k)z(k+1)}$$  \hspace{1cm} (9)

$$P(k+1) = \frac{1}{\lambda(k+1)} \left[ P(k) - L(k+1)z^T(k+1)P(k) \right]$$  \hspace{1cm} (10)

Previous work investigated how the forgetting factor should be effectively varied[24-27]. Among these, Park et al.[27] proposed the following novel relationship between the forgetting factor and the prediction error:

$$\lambda(k) = \lambda_{\text{min}} + \left(1 - \lambda_{\text{min}}\right) \cdot 2^{-\text{NINT}[\rho \varepsilon^2(k)]}$$  \hspace{1cm} (11)

where $\lambda_{\text{min}}$ is the minimum value of the forgetting factor, $\rho$ is a design parameter, and $\text{NINT}[\cdot]$ is defined as the nearest integer to $[\cdot]$. The idea behind Eqn. (11) is that the magnitude of the prediction error serves as an indication of changes in the process model parameters, thus when the prediction error increases, it is then necessary to initiate the data discounting mechanism. Park et al.[27] showed that by incorporating Eqn. (11), the RLS algorithm performed better than that of Fortescue et al.[24] and Salgado et al.[26], with smaller miss-adjustment errors in the steady state parameters. For ease of reference, Eqns. (5) & (6), (9) - (11) are referred to as the Exponential Weighting Recursive Least Squares (EWRLS), following Park et al.[27].

Another approach of identifying time-varying system parameters is to forget in the direction where new information is coming[28]. Such an approach is demonstrated by the Exponential and Directional Forgetting (EDF) algorithm[29] and interested readers can refer to the
work of Kulhavý and Kárný for details. In addition to the algorithm proposed by Kulhavý and Kárný, Bittanti et al.\textsuperscript{[30]} suggested a correction factor to the equation of covariance update to improve the tracking performance of the EDF algorithm. The EDF algorithm with corrections suggested by Bittanti is shown here for completeness:

\begin{align}
    r(k+1) &= z^T(k+1) P(k) z(k+1) \\
    L(k+1) &= \frac{P(k) z(k+1)}{1 + r(k+1)} \\
    \beta(k+1) &= \begin{cases} 
            \lambda(k+1) - \frac{1 - \lambda(k+1)}{r(k+1)} & \text{if } r(k+1) > 0 \\
            1 & \text{if } r(k+1) = 0 
        \end{cases} \\
    P(k+1) &= P(k) - \frac{P(k) z(k+1) z^T(k+1) P(k)}{\beta(k+1) + r(k+1)} + \delta I
\end{align}

(12) \quad (13) \quad (14) \quad (15)

\begin{align}
    \varepsilon(k+1) &= y(k+1) - z^T(k+1) \hat{\theta}(k) \\
    \hat{\theta}(k+1) &= \hat{\theta}(k) + L(k+1) \varepsilon(k+1)
\end{align}

(16) \quad (17)

where \( I \) is the identity matrix and \( \delta \in [0, 0.01] \) is the Bittanti factor which is added to Eqn. (15) to ensure the exponential convergence of the EDF algorithm.

As in all RLS implementations, in order to avoid negativity of the covariance matrix \( P \) due to numerical instability\textsuperscript{[31]}, the covariance matrix \( P \) is decomposed as follows:

\[ P(k) = U(k) D(k) U(k)^T \]

(18)

where \( U \) is an upper triangular matrix with ones in the diagonal and \( D \) is a diagonal matrix. Bierman\textsuperscript{[32]} demonstrated that recursive updates of the covariance matrix could be accomplished through the factorized components of Eqn. (18).

The performance of the EWRLS algorithm together with Bierman’s \( \text{UDU}^T \) implementation will be compared with the EDF algorithm later in Section 4.
3. ADAPTIVE CONTROL STRATEGY

For linear process models, IMC has been shown to have good performance properties against disturbances and model mismatch\cite{33}. A major attraction of IMC is that it can be implemented in an equivalent structure of conventional PID controller\cite{34}. Furthermore, the computational burden of implementing IMC is trivial compared to other model-based control design methods.

![Block diagram of the RLS-IMC adaptive control strategy](image)

Figure 1. Block diagram of the RLS-IMC adaptive control strategy: $u =$ process input, $y =$ process output, $d_1$ and $d_2 =$ disturbances, $y_{sp} =$ set point, $\theta_p =$ estimated process parameters, and $\theta_c =$ controller settings

In designing the controller, here the IMC method developed by Garcia and Morari\cite{35} was coupled with the RLS algorithm to include auto-tuning of the controller settings based on identified process model parameters. The essential feature of this adaptive scheme is illustrated in Figure 1, where $u$ is the process input, $y$ is the output, $d_1$ and $d_2$ are disturbances, and $y_{sp}$ is the controller set point. Two different loops were designed in this adaptive strategy, one responsible for online system identification whereas the other was used for feedback control. The process model parameters, $\theta_p$, for the current conditions were estimated by the EWRLS and will be utilized in controller design to calculate the updated controller settings, $\theta_c$. This combination of EWRLS-IMC will produce a controller that is capable of functioning beyond the nominal operating point, where any unexpected change in the dynamics of the transesterification reactor can be handled in real time by the adaptive controller.
To design the controller based on the principles of IMC, a suitable model for the process is required. The literature reports chiefly the use of the “first order plus dead time (FOPDT)” model as an approximate process model. However, in this case, no knowledge about the process dead time and its characteristics are available \textit{a priori}. Known techniques for online dead time estimation\cite{6, 36, 37} introduce additional computational complexities; whereas offline dead time estimation techniques are cumbersome if the process has a time-varying delay. One way to handle such a situation is to increase the order of the model\cite{4}. This has to be balanced by considerations that too high a model order introduces additional computational complexity, while too low a model order describes the process dynamics with dead time poorly. Considering these trade-offs, a relatively simple second order process model, as shown by Eqn. (19), was employed to represent the transesterification dynamics here.

\begin{equation}
G(s) = \frac{K_p}{(\tau_1 s + 1)(\tau_2 s + 1)} \quad (19)
\end{equation}

with $K_p$ being the gain of the process, and $\tau_1, \tau_2$ the process time constants.

For the model of Eqn. (19), the corresponding IMC-based PID controller settings\cite{34} are:

\begin{equation}
K_c = \frac{\tau_1 + \tau_2}{K_p \tau_c} \quad \tau_I = \tau_1 + \tau_2 \quad \tau_D = \frac{\tau_1 \tau_2}{\tau_1 + \tau_2} \quad (20)
\end{equation}

where $K_c$, $\tau_I$, $\tau_D$ are the controller gain, integral reset, and derivative time constants, respectively. $\tau_c$ is the IMC closed loop time constant, which serves as a design parameter (more below).

Since the EWRLS used in this work identifies the process in the discrete time domain, whereas the existing IMC-based PID settings are in the continuous time domain, a model conversion from the discrete time domain to the continuous time domain is necessary. Although there are versions of RLS algorithms which are able to estimate directly the process model parameters in the continuous time domain\cite{38}, in this work, converting the model parameters from the discrete to continuous time domain is preferred. In tandem with the process model chosen for IMC, the EWRLS algorithm identifies the process as a second order discrete time model. This conversion to the continuous time domain\cite{39} is given by Eqns. (21) to (24):

\begin{equation}
G(z) = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (21)
\end{equation}
\[ K_p = \frac{b_1 + b_2}{1 + a_1 + a_2} \]  

(22)

\[ \tau_1 = \frac{t_s}{\log\left(-0.5a_1 + 0.5\left(a_1^2 - 4a_2\right)^{0.5}/a_2\right)} \]  

(23)

\[ \tau_2 = \frac{t_s}{\log\left(-0.5a_1 - 0.5\left(a_1^2 - 4a_2\right)^{0.5}/a_2\right)} \]  

(24)

where \(a_1, a_2, b_1, b_2\) are the model parameters estimated by the EWRLS algorithm, and \(t_s\) is the sampling time. Eqn. (21) can be obtained from the z-transform of Eqn. (1) when \(n_a = n_b = 2\). Eqns. (20), (22) - (24) were directly involved in calculating the controller settings at every time step.

Arising from Eqns. (20) & (23) - (24) above are two important parameters, viz. the IMC closed loop time constant, \(\tau_c\) and the sampling time, \(t_s\). A rule of thumb is to select \(\tau_c\) to be twice as fast as the open loop response time\(^{[34, 40]}\), and \(t_s\) that is small enough to adequately describe the dynamics of the process\(^{[31, 41, 42]}\). While inappropriate values of \(\tau_c\) will result in a controller that is either too sluggish or aggressive, poor choices of \(t_s\) (i.e. too large or too small) will result in poor description of the process dynamics or unnecessary increase in computational complexity.

In the design of this adaptive controller, rule of thumb for \(\tau_c\) and \(t_s\) were used as initial values, followed by constrained optimization via the Simulink\textsuperscript{®} Response Optimization\textsuperscript{™} software version 3.1.3 (signal constraint block) to optimize the responses of the control loops by manipulating \(t_s\) and \(\tau_c\). Genetic algorithm was selected as the search method in this optimization, while leaving the remaining settings as default. For ease of reference, the values of the parameters involved in the genetic algorithm are given here: initial population = 20, elite count = 2, crossover fraction = 0.8, migration direction = ‘forward’, migration fraction = 0.2 and maximum generation = 100. The optimization constraints for the process responses were taken as \(\pm 5\%\) of the controllers’ set point profiles with the aim of minimizing overshoot in the process variables, attaining a reasonable rise time and yet avoiding responses that are too sluggish.

Although new sets of controller settings are calculated at every time step, however, a successful adaptive controller does not require that the controller settings be updated at every
It is critical to filter off unrealistic estimates of the process parameters by the EWRLS algorithm and refrain from updating the controller\cite{4}. The easiest way is to selectively update the controller settings only when the poles of the identified model lie within the unit circle, implying stable dynamics.

The sequence of recursive closed loop calculations in this adaptive control scheme is as follows:

a) Online process modeling by EWRLS algorithm  
b) Model parameter conversion from discrete to continuous time domain  
c) Calculation of controller settings based on the IMC method  
d) Evaluation of identified model poles  
e) Update controller settings if stable poles are obtained

It is important to note that as in all adaptive control, updating of the controller settings does not take place from the start-up of the reactor since the EWRLS algorithm needs to collect sufficient process data before reaching a satisfactory level of performance. It is also crucial, at the beginning of implementation, to only activate EWRLS when the inputs and outputs of the process are at steady state. Based on process experience and also intuitively, the reactor temperature loop has a slower open loop dynamics compared to the methyl ester concentration loop. Hence, to account for the time needed for both loops to reach their initial steady state and the time needed for the RLS algorithm to correctly estimate the process model parameters, ‘safe values’ of time period (i.e. 2000 s for the reactor temperature loop and 1000 s for the methyl ester concentration loop) must elapse before the adaptive controllers for both loops are turned on. Within these time periods, conventional controllers with fixed tuning parameters were used instead for both loops. The choice of the conventional controllers prior to the activation of the adaptive control scheme is a matter of preference. Here the conventional controllers were designed using the IMC approach based on step test data at the nominal operating point.

4. PERFORMANCE OF EWRLS IN MODEL IDENTIFICATION

Before applying the EWRLS to the transesterification reactor, a comparison of the tracking ability was made between the conventional RLS algorithm [Eqns. (3)-(6)], the EWRLS algorithm, and the EDF algorithm. In this comparison, a hypothetical discrete time transfer function with ten parameters to be identified was introduced to investigate the performance of each algorithm under situations with time-varying system parameters, following the approach of Park et al.\cite{27}.
To simulate a time-varying environment, the system parameters were changed at the 70th time step to Eqn. (26).

\[
G(z) = \frac{0.6z^{-1} + 0.7z^{-2} + 0.8z^{-3} + 0.9z^{-4} + z^{-5}}{1 + 0.1z^{-1} + 0.2z^{-2} + 0.3z^{-3} + 0.4z^{-4} + 0.5z^{-5}}
\]  \tag{25}

As in Park et al.\cite{27}, white Gaussian noise with zero mean and variance of 0.01 was added to the output of the system to simulate a noisy environment, while the same noise type with a variance of 0.1 was used as an input to the system. The values of minimum forgetting factor, \( \lambda_{\text{min}} \) and design parameter, \( \rho \) selected by Park et al.\cite{27} were 0.7 and 5 respectively, and were retained in this simulation.

![Figure 2](image_url)

**Figure 2.** The performance of three parameter estimation algorithms in tracking changes in parameters at the 70th time step: EDF is Exponential and Directional Forgetting algorithm, EWRLS is Exponential Weighting Recursive Least Squares algorithm, and RLS is Conventional Recursive Least Squares algorithm.

It is worth noting that the fifth order model used in this section resembles a challenging system identification problem as the size of the parameter estimation (\( i.e. \) 10 parameters) is much
larger than the actual number of parameters to be estimated for the biodiesel reactor (i.e. 4 parameters). Hence, if any version of the RLS algorithms passes the test of a fifth order parameter estimation, it should perform well in estimating a lower order process. Figure 2 compares the Euclidean norm of the error in parameter estimation, \( L_2 = \| \theta - \hat{\theta}(k) \|_2 \) for 500 time steps, where \( \theta \) is the true system parameter vector. As can be inferred from the figure, the convergence speed of EWRLS and EDF are similar. Given time, all three algorithms managed to estimate the system parameters to a norm of 0.1. It is however, worth noting, that the EDF algorithm exhibits fluctuating norm after the 70\(^{th}\) time step where the system parameters were varied, which indicates that the EDF algorithm does not give stable predictions of time-varying system parameters, as opposed to predictions given by the EWRLS algorithm. The conventional RLS algorithm was the slowest to converge, suggesting that it is not a suitable algorithm to track time-varying systems. Hence, the EWRLS, which produced more stable parameter estimates, was a more practical algorithm to be implemented in this study.

Table 1: Nominal operating conditions of the biodiesel reactor\(^{[1]}\)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reactor temperature</td>
<td>( T )</td>
<td>60</td>
<td>(^{\circ} )C</td>
</tr>
<tr>
<td>Reactor pressure</td>
<td>( P_r )</td>
<td>1</td>
<td>atm</td>
</tr>
<tr>
<td>Reactant flow rate</td>
<td>( F_o )</td>
<td>0.119</td>
<td>( m^3/s )</td>
</tr>
<tr>
<td>Coolant flow rate</td>
<td>( F_c )</td>
<td>0.00268</td>
<td>( m^3/s )</td>
</tr>
<tr>
<td>Stirrer rotational speed</td>
<td>( N )</td>
<td>6</td>
<td>rps</td>
</tr>
<tr>
<td>Initial concentration of triglyceride</td>
<td>( C_{TG_o} )</td>
<td>1.11</td>
<td>kmol/m(^3)</td>
</tr>
<tr>
<td>Initial concentration of methanol</td>
<td>( C_{A_o} )</td>
<td>21.499</td>
<td>kmol/m(^3)</td>
</tr>
</tbody>
</table>

Using the EWRLS algorithm, an open loop system identification was carried out on the transesterification reactor using equations (21) to (24) as described earlier. The nominal operating conditions of the transesterification reactor were identical to those used previously by Mjalli et al.\(^{[1]}\) and are summarized in Table 1. In the same work, using the Relative Gain Array (RGA), it was also shown that the appropriate control loop pairings are the reactor temperature (\( T \)) with the coolant flow rate (\( F_c \)) for the temperature loop and the product concentration (\( C_{ME} \)) with reactant flow rate (\( F_o \)) for the concentration loop. These pairings were retained in this work. The sampling times for the temperature and the concentration loops were chosen to be 40 seconds and 10 seconds respectively based on rule of thumb and process experience.
Figure 3. Open loop prediction errors for the reactor temperature and the product concentration loops: A 10% increase in both the manipulated variables $F_c$ (coolant flow rate) and $F_o$ (reactant flow rate) were imposed at $t = 2000$s on separate runs, hence shown here are the superposition of these individual runs.

Figure 3 shows the prediction error [Eqn. (5)] when each individual loop is perturbed from its nominal operating point at 2000 seconds. When the process is perturbed from its nominal operating point at 2000 seconds, it moves into another region of operation, depending on the level of perturbation and the degree of nonlinearity of the process. When this happens, it is equivalent to having the process model parameters varied, and hence a change in the prediction error occurs (as can be inferred from the figure). However, as the EWRLS algorithm captures more input and output data from the changed process, the algorithm will bring the prediction error back towards zero. This capability is further illustrated by Figure 4, which shows the open loop temporal evolution of the estimated process model parameters for both the reactor temperature and the product concentration loops when the system was perturbed from its nominal operating point. The estimated process model parameters for the temperature loop stabilized after 6 to 14 sampling time, while those for the concentration loop stabilized faster in less than 8 sampling time.
Figure 4. (i) Reactor temperature and (ii) product concentration open loop adaptation of 2\textsuperscript{nd} order discrete time model parameters (Eqn. 14) for a 10\% increase in coolant flow rate ($F_c$) and reactant flow rate ($F_o$) respectively at 2000 s: Here, the sampling times used for the reactor temperature loop and product concentration loop were 40 s and 10 s respectively.
The results above showed that EWRLS was not only capable of identifying a hypothetical high order model, but was also equally adept at identifying a validated second order model for a transesterification reactor. Such system identification capability could then be coupled to a process controller, as investigated next.

5. PERFORMANCE OF THE ADAPTIVE CONTROL SYSTEM

As alluded to in Section 3, the second order discrete time process model parameters identified by the EWRLS must be transformed into the continuous time model to enable the use of IMC’s controller design method in an equivalent PID structure. This involves one critical parameter, which is the sampling time $t_s$. Following model conversion, it is also important to ensure that the controller moves are not overly aggressive, in which case the IMC closed loop time constant $\tau_c$ plays an important role. Both the values of $t_s$ and $\tau_c$ were optimized here by a genetic algorithm. Figure 5 and Figure 6 compare the performances using the initial and the optimized values of $t_s$ and $\tau_c$ in the reactor temperature and the product concentration loops. These profiles are representative of the behavior for other set point changes of comparable magnitude. This is expected as the optimization using the genetic algorithm had included a variety of set point changes within the normal operating region of a transesterification reactor. The optimized values of $t_s$ and $\tau_c$ are summarized in Table 2.

![Figure 5](image-url)

**Figure 5.** Effect of the sampling time $t_s$ and the IMC closed loop time constant $\tau_c$ on the closed loop reactor temperature response to a set point change at $t = 3000$ s. The first pair $[t_s, \tau_c] = [40$ s, $100$ s] was the initial guess, the second was the optimized values.
Figure 6. Effect of the sampling time $t_s$ and the IMC closed loop time constant $\tau_c$ on the closed loop product concentration response to a set point change at $t = 6000$ s. The first pair $[t_s, \tau_c] = [10 \text{ s}, 75 \text{ s}]$ was the initial guess, the second was the optimized values.

Table 2. Optimized values of the adaptive controller parameters for the transesterification reactor

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Reactor Temperature Loop</th>
<th>Product concentration Loop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling Time, $t_s$ (s)</td>
<td>55.73</td>
<td>19.22</td>
</tr>
<tr>
<td>IMC Closed Loop Time Constant, $\tau_c$ (s)</td>
<td>79.38</td>
<td>92.48</td>
</tr>
</tbody>
</table>

The performance of this optimized adaptive control scheme was compared with the conventional PID control tuned using the Ziegler-Nichols (Z-N) open loop method\[^{38}\]. Figure 7 and Figure 8 show the performances of these controllers for a change in the set point. Changes of set points of different magnitudes as well as in the opposite directions were also investigated (not shown), and the outcomes were typical of Figures 7 and 8. As shown in Tables 3 and 4, generally the adaptive controllers showed superior performance in achieving lower overshoots and faster settling times for both the reactor temperature and methyl ester concentration loops. Although the rise time achieved by the adaptive controller for the reactor temperature loop was slightly slower than the conventional PID controller, nonetheless, the performance of the adaptive controller was in general better than the conventional PID controller in view of the negligible overshoot and the fast settling time achieved. Furthermore, the assessment for the methyl ester concentration loop revealed the strength of the adaptive controller in all performance indicators over the conventional PID controller.
Figure 7. Comparison of the performance between the optimized adaptive controller ($t_r = 55.73 \text{ s}$, $\tau_c = 79.38 \text{ s}$) and the conventional PID controller for the reactor temperature loop.

Figure 8. Comparison of the performance between the optimized adaptive controller ($t_r = 19.22 \text{ s}$, $\tau_c = 92.48 \text{ s}$) and the conventional PID controller for the product concentration loop.

Table 3. Comparison of the performance between the adaptive controller and the conventional PID controller for the reactor temperature loop. The corresponding temporal evolution is given in Figure 7.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Conventional PID</th>
<th>Adaptive Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overshoot</td>
<td>0.25 K</td>
<td>Negligible</td>
</tr>
<tr>
<td>Rise time</td>
<td>170 s</td>
<td>270 s</td>
</tr>
<tr>
<td>Settling time</td>
<td>640 s</td>
<td>270 s</td>
</tr>
</tbody>
</table>
Table 4. Comparison of the performance between the adaptive controller and the conventional PID controller for the methyl ester concentration loop. The corresponding temporal evolution is given in Figure 8.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Conventional PID</th>
<th>Adaptive Controller</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overshoot</td>
<td>0.3 kmol/m³</td>
<td>0.01 kmol/m³</td>
</tr>
<tr>
<td>Rise time</td>
<td>250 s</td>
<td>200 s</td>
</tr>
<tr>
<td>Settling time</td>
<td>750 s</td>
<td>380 s</td>
</tr>
</tbody>
</table>

The performance of the optimized adaptive controller was further examined under successive changes in the set points in both directions. For generality, the step sizes and the duration between set point changes were all random and arbitrary, but within the operating range of the transesterification reactor. The results of one such series of set point changes are depicted in Figure 9 and Figure 10. From both figures, there was no sign of deterioration in the tracking performance of the adaptive controller, even though the set points were varied from one operating region to another. Hence, the adaptive controller demonstrated good adaptation properties in handling nonlinear operational regions.

Figure 9. Closed loop response of the reactor temperature loop using the optimized adaptive controller ($\tau_r = 55.73$ s, $\tau_c = 79.38$ s), under successive, random set point changes.
However, a good controller must not only exhibit good performance characteristics in set point tracking, but it should also be able to reject disturbances as well. In this reactor, the effects of process disturbances, *viz.* triglyceride (reactant) concentration ($C_{\text{TGO}}$), feed temperature ($T_o$), coolant inlet temperature ($T_{\text{Co}}$), and stirrer rotational speed ($N$) were evaluated. These variables were included in the comprehensive model of the transesterification reactor in the previous study\cite{1}, hence no further model development is necessary here.

Figure 10. Closed loop response of the product concentration loop using the optimized adaptive controller ($\tau_c = 19.22$ s, $\tau = 92.48$ s), under successive, random set point changes.

Figure 11. Reactor temperature ($T$) and product concentration ($C_{\text{ME}}$) profiles for 5% increment in the nominal values of disturbance variables: triglyceride concentrations ($C_{\text{TGO}}$, ---), feed temperature ($T_o$, ---), coolant inlet temperature ($T_{\text{Co}}$, ...), and stirrer rotational speed ($N$, - - - - - -) at $t = 8000$ s. These variables were changed one at a time, hence shown here are superposition of four separate runs. Although both $C_{\text{TGO}}$ and $T_o$ had the strongest effects, their extent of influence on $T$ and $C_{\text{ME}}$ differs. Overall, the effects of changes in all four variables on $T$ and $C_{\text{ME}}$ are small.
A five percent increment in the nominal values of each disturbance variable was introduced, and the adaptive controllers were able to bring the process response back to the set point in a sufficiently short amount of time, with minimal overshoot, as illustrated by Figure 11. For the temperature loop, the adaptive controller was able to bring the temperature of the reactor back to the set point in less than 1000 s with a maximum overshoot of less than 0.4°C. For the more sluggish product concentration loop, the methyl ester concentration was brought back to the set point in less than 2000 s with a maximum overshoot of 0.01 kmol/m³. The results also showed that the feed temperature has the highest effect on the reactor temperature whereas the triglycerides concentration has the largest effect on the product concentration. The stirrer speed has negligible effect on both loops. In any case, a 5% rise in any of the four variables had a marginal effect on T and C_ME, and the adaptive controller brought the system back to the desired operating point.

6. CONCLUSION

By using the EWRLS algorithm, this work extends the basic idea of having multiple predetermined linear models in the MMAC strategy to an online model-based adaptive control. With this, the dynamics of the process at any operating point is captured continuously and the controller settings are altered based on the most recent process dynamics. In MMAC terms, this is equivalent to having huge model banks with an infinite amount of local linear models, but without the penalty of exorbitant effort in constructing those models. Furthermore, the adaptive control strategy advocated in this work reduces user intervention in the design of the adaptive controller, fully exploiting the online modeling capability of the EWRLS for self-tuning purposes.

A decentralized two-input two-output adaptive control scheme was implemented on a validated process model for a transesterification reactor. With an optimized set of sampling time and IMC closed loop time constant, the performance of the adaptive controller was superior to that of a conventional Z-N-tuned PID controller. The adaptive controller is not only stable, but also tracks set points more efficiently with minimal overshoots and shorter settling times. Moreover, it exhibits good disturbance rejection characteristics.

The downside of this approach, indeed for all adaptive control schemes, is the added computational complexity compared to the conventional PID controllers. For critical applications in which the process characteristics change over the operating region, it is nonetheless justifiable. To this end, this work had demonstrated the efficacy of the EWRLS-
IMC architecture on a validated nonlinear process model. The next course of action is to realize it in the laboratory. It remains to be seen if the scheme could handle implementation challenges such as the effects of real noise, real time actuation of sluggish control valves, additional nonlinearities that are not captured by the simplistic model etc.
7. SYMBOLS USED

C  [kmol/m$^3$] concentration
$D$  [-] diagonal matrix in UDU$^T$ factorization
F  [m$^3$/s] volumetric flow rate in reactor
$J$  [-] objective function
$K_c$  [-] controller proportional gain
$K_p$  [-] process gain
$L$  [-] Kalman gain
$L_2$  [-] Euclidean norm of parameter estimation error
$N$  rps stirrer rotational speed
$P$  [-] covariance matrix of estimation error
$P_r$ atm reactor pressure
T  [K] reactor temperature
$U$  [-] upper triangular matrix in UDU$^T$ factorization
$a_i$  [-] the i-th denominator parameter
$b_i$  [-] the i-th numerator parameter
$d_i$  [-] the i-th disturbance variable
$e$  [-] feedback error
$ts$  [s] sampling time
$u$  [-] process input
$y$  [-] process output
$y_{sp}$  [-] set point
$z$  [-] regressor vector
GREEK SYMBOLS

\( \varepsilon \) [-] prediction error
\( \theta \) [-] vector of true process parameters
\( \theta_c \) [-] vector of controller tuning parameters
\( \theta_p \) [-] vector of modeled process parameters
\( \lambda \) [-] forgetting factor
\( \lambda_{\text{min}} \) [-] minimum forgetting factor
\( \xi \) [-] stochastic noise variable
\( \rho \) [-] EWRLS design parameter
\( \tau_c \) [s] IMC closed loop time constant
\( \tau_D \) [s] derivative time constant
\( \tau_i \) [s] the i-th process time constant
\( \tau_I \) [s] integral reset

SUBSCRIPTS

C coolant
ME methyl ester
TG triglyceride
o inlet/initial condition
8. REFERENCES


