Applied Intelligent Control - Control of Automotive Paint Process

Dimitar Filev
Ford Motor Company
24500 Glendale Av., Detroit MI 49239
E-mail: dfilev@ford.com

Abstract - We present an intelligent control algorithm that is targeted to process control of the steady state of a class of industrial MIMO nonlinear systems. The algorithm, called the RBIC Intelligent Control Algorithm, combines the conventional indirect adaptive control approach with a Rule Base of Initial Conditions (RBIC) - an intelligent tool improving the conventional indirect adaptive algorithm in the presence of large disturbances and multiple operating modes. The RBIC operates as an associative memory that periodically reinitializes the indirect adaptive control algorithm by using a fuzzy reasoning inference mechanism. We discuss the main features and application aspects of the RBIC Intelligent Control Algorithm. We also demonstrate one large scale process control application of the RBIC Intelligent Control Algorithm as the main component of Ford Motor Company's Integrated Paint Quality Control System - an industry first automotive paint process control system.

Key Words: Industrial Applications, Intelligent Control, Fuzzy Rules

I. INTRODUCTION

A wide class of industrial process control problems deal with control of nonlinear systems represented by steady state models. Generally, systems of this type are approximated by mappings between groups of adjustable process parameters (inputs) and the effected measurable process characteristics (outputs):

$$y(k) = S(u(k)) \quad (1)$$

where the MIMIO nonlinear plant model S is assumed to be unknown but smooth. The objective is to control the output y as close as possible to a given set point y_d, i.e. to minimize the error:

$$e(k) = (y(k) - y_d)^T (y(k) - y_d) \quad (2)$$

in the presence of measurement noise, unmeasured disturbances, process and equipment constraints, and multiple operating modes. The problem of indirect adaptive control of static MIMO nonlinear systems was considered in [1-4], where several schemes for model identification and feedback control were analyzed and compared. In parallel, the concept of a Rule Base of Initial Conditions (RBIC) as an intelligent tool for supporting the conventional indirect adaptive control techniques was developed. The RBIC functions as a long-term associative memory - it summarizes essential information, which is forgotten by the conventional adaptive algorithm and guides the adaptation process by providing a set of "default" models and controls. By merging the conventional indirect adaptive control [5] with the RBIC concept a new efficient, reliable and easy-to-implement intelligent control algorithm, called the RBIC Intelligent Control Algorithm, was derived. Further details of its advantages and its relationship to the class of multiple model switching control algorithms proposed by Narendra et al. [6] are provided in [7]. Benefits of using RBIC Intelligent Control Algorithm versus the conventional indirect adaptive control and neural network based adaptive control were demonstrated in [3] and [4].

The first major industrial application of the RBIC Intelligent Control Algorithm was in the Integrated Paint Quality Control System – an industry first feedback control system for paint process control that was recently developed and implemented at Ford Motor Company. In this paper we focus on the application aspects of the RBIC based intelligent control algorithm. We also present the results of implementation of the Integrated Paint Quality Control System where the RBIC Intelligent Control Algorithm is a critical component.

II. THE RBIC INTELLIGENT CONTROL ALGORITHM

The RBIC Intelligent Control Algorithm consists of three main modules - a Model Updating Module, a Control Updating Module and a Rule Base of Initial Conditions (RBIC) Module.

We consider a linearized time varying approximation of the nonlinear input-output mapping S:

$$\Delta y(k) = J(k)\ \Delta u(k) \quad (3)$$

where

$\Delta u(k) = u(k) - u(k-1)$, $\Delta y(k) = y(k) - y(k-1)$,
$\Delta u \in R^r$, $\Delta y \in R^q$, and $J(k) = \delta y_s / \delta u_j$, j=[1, r], s=[1, q].

and J(k) is the Jacobian matrix calculated by linearization of (1) around the operating point (u(k), y(k)).

The Model Updating Module recursively updates a linearized Jacobian model (3) by using two optional learning strategies – the Least Mean Square (LMS, Widrow-Hoff rule) and the Recursive Least Square (RLS, Kalman Filter) algorithm.

The Control Updating Module calculates a control update minimizing the error (2) subject to the updated plant model and control constraints. It includes two optional algorithms – a modified version of the Levenberg-Marquardt algorithm combined with saturation, and an on-line version of the Constraint Linear Least Squares (LSQLIN) optimization algorithm [2].

The RBIC module, handles the multiple operating modes and disturbances. It summarizes previous control performance by automatic generation of rules that map prototypical operating points with the models associated with them. It also provides a fuzzy logic based inference mechanism to estimate the model that relates to the current operating point by using the information contained in the rules. This estimated model is used to periodically reinitialize the learning algorithm of the first control subsystem in the case of poor control performance. Below we discuss the three main modules of the RBIC Intelligent Control Algorithm.

III. MODEL UPDATING MODULE

We apply two common methods for learning linear models - the LMS method (Widrow-Hoff rule) and the Recursive Least Square method (RLS, Kalman filter) - to on-line estimate the linearized (Jacobian) model of the plant (1).

LMS Model Update

According to the LMS rule we obtain the following expression for updating the Jacobian:

$$\hat{J}(k) = \hat{J}(k-1) + \alpha\,(\Delta y(k) - \hat{J}(k-1)\,\Delta u(k))\;\Delta u^T(k) \,/\, (\Delta u^T(k))\;\Delta u(k)) \quad (4)$$

where $0<\alpha<2$ is the learning rate.

RLS Model Update

To apply the Kalman filter for learning the Jacobian we assume that the MIMO linearized system (3) is decomposed into q MISO subsystems $\Delta y_j = J^T_j(k)\ \Delta u(k)$, $j=[1,q]$, where J^T_j is the jth row of the Jacobian **J**. Each of the MISO systems maps all inputs to one of the outputs. Jacobian changes in time are captured by the dynamic systems:

$$J_j(k+1) = J_j(k) + w_j(k) \quad (5i)$$

$$\Delta y_j(k) = \Delta u^T(k)\, J_j(k) + v_j(k),\ j=[1, q] \quad (5ii)$$

where vector $w_j(k)$ represents the imprecision of the linearized model (3) with covariance $Q_j = E\{w^T_j(k)\, w_j(k)\}$ and $v_j(k)$ is measurement noise with zero mean and variance $R_i = E\{v_i^2(k)\}$. By applying the Kalman filter to each of the MISO subsystems (5) we obtain an expression for recursive updating of the rows of the Jacobian estimate $\hat{J}_j$:

$$\hat{J}^T_j(k) = \hat{J}^T_j(k-1) + L_j(k-1)\,(\Delta y_j(k) - \hat{J}^T_j(k-1)\,\Delta u(k)) \quad (6i)$$

$$L_j(k-1) = P_j(k-1)\,\Delta u(k)\,(R_j + \Delta u^T(k)\,P_j(k-1)\,\Delta u(k))^{-1} \quad (6ii)$$

$$P_j(k-1) = L_j(k-1)\,\Delta u^T(k)\,P_j(k-1) + Q_j \quad (6ii)$$

Matrix Q_j in (6ii) represents the drift factor that is analogous to the forgetting factor [8] and can be estimated from the expected changes in the Jacobian. The advantage of using the drift factor versus the exponentially forgetting factor is in the cases when the system is not excited. It forces the covariance matrix P_j (which essentially controls the variable learning rate of the Kalman filter) to grow linearly rather than exponentially.

IV. CONTROL UPDATING MODULE

We use two alternative control strategies – the Levenberg-Marquardt update and the Constraint Linear Least Squares (LSQLIN) - to calculate the next control update based on the updated Jacobian $\hat{J}(k)$.

Modified Levenberg-Marquardt Based Control Update

$$u(k+1) = u(k) + K(k)\,(y_d - y(k)) \quad (7i)$$

$$K(k) = \hat{J}^T(k)\,G\,(\rho I + \hat{J}(k)\,\hat{J}^T(k))^{-1} \quad (7ii)$$

$$u(k+1) = sat(u(k) + \hat{J}^+(k)\,(y_d - y(k))) \quad (7iii)$$

where ρ is a positive constant, I is an identity matrix of compatible size, and diagonal matrix G represents the gain of the controller. Saturation function (7iii) holds the control vector within its feasibility region.

<u>Constraint Linear Least Squares (LSQLIN)</u>

$$u(k) = argmin_U\,(\|y_d - \hat{y}(k)\|^2 + \beta\,\|u(k) - u(k-1)\|^2) \quad (8i)$$

s. t.

$$\hat{y}(k) = y(k-1) + \hat{J}(k)\,(u(k) - u(k-1)) \quad (8ii)$$

where parameter β represents the weight of the control vector update in the cost function. The penalty on the control change (8i) is not formally required, but it plays an important role for improving the robustness of the controller.

By combining the options of the two modules we obtain four alternative versions of the RBIC Intelligent Control Algorithm.

V. RULE-BASE OF INITIAL CONDITONS (RBIC)

The learned linearized model $\hat{J}(k)$ is relevant only in the vicinity of the operating point. Plant non-linearity or/and disturbances, and multiple operating modes may cause change of the operating point. If the change of the operating point is small the learning rule (LMS or RLS) gradually changes the model, which results in gradual adjustment of the control - the control algorithm adapts to the change. If the change is substantial, the learning algorithm will drastically update the model leading to possible instability of the feedback control system.

The RBIC essentially stores operating points and models by summarizing them into rules if the control system demonstrates satisfactory performance. It also guides the indirect adaptive control algorithm in the cases when it performs unsatisfactorily by helping to estimate the Jacobian model that relates to the current operating point.

The RBIC includes m rules of the format:

IF x(k) is *close to* x_i*
THEN $\tilde{J}$ (k) is J_i* and $\tilde{u} = u_i$*, *i=[1, m]* (9)

where $x(k) = (u'(k), y'(k))'$ is the current operating point, $x_i^* = (u_i^*, y_i^*)$ is a prototypical operating point at which the Jacobian model is J_i^*. Each rule (9) defines a model J_i^* and a default initial control u_i^* that are associated with a certain prototypical operating point x_i^*. A rule of the RBIC can be regarded as a representation of a cluster of operating points that are close in the input-output space and centered around x_i^*. At these operating points the Jacobian model of the plant is approximated by J_i^*. We assume that the degree of closeness (membership grade) of the current operating point x(k) to the prototypical operating points x_i^* , i=[1, m] is captured by the Gaussian:

$$\tau_i = exp(-(x(k) - x_i^*)' (x(k) - x_i^*) / \sigma^2), i=[1, m] \quad (10)$$

where σ is a smoothing parameter. The output of the rule-base (9) is obtained through interpolation between the rule consequents based on the degree of matching the antecedents by the current operating point x(k):

$$\tilde{J}(k) = \sum_{i=1}^{m} \tau_i J_i^* / (\sum_{i=1}^{m} \tau_i); \quad (11)$$

$$\tilde{u}(k) = \sum_{i=1}^{m} \tau_i u_i^* / (\sum_{i=1}^{m} \tau_i) \quad (12)$$

Control performance is quantified in terms of the measure:

$$E(k) = \sum_{j=k-h+1}^{k} (y(j) - y_d)' (y(j) - y_d) \quad (13)$$

where h is a parameter defining a moving window of the last h output values that are used to calculate the performance measure (13). In the case of unsatisfactory control performance, i.e. $E(k) \geq e_L$, the model $\tilde{J}$ (k) inferred by the RBIC reinitializes the LMS (4) or RLS (6) algorithm and the inferred control $\tilde{u}$ (k) provides a default control vector. *Vice versa*, since the RBIC summarizes successful control performance we use these operating point x(k), and model $\hat{J}$ (k) that are associated with performance measure value below the threshold, i.e. $E(k) < e_l$, for updating the rule base. We call the pair (x(k), $\hat{J}$ (k)) that meets this criterion the *eligible pattern*. The algorithm for learning RBIC uses the eligible patterns to update the rule prototypes x_i^* and J_i^* based on a combination between the regression NN learning algorithm [10] and the K-Nearest Neighbor algorithm.

1. The first eligible pattern (x(k), $\hat{J}$ (k)) establishes the parameters of the first rule:

$m = 1$; $x_m^* := x(k)$; $J_m^* = \hat{J}(k)$; $N_m := 1$

where N_i is counter of the operating points that are bundled by the rule.

2. Assume that there are already established m rules with parameters x_i^*, J_i^*, N_i, i = [1, m], and (x(k+p), $\hat{J}$ (k+p)) is the s-th eligible pattern. Let $j = arg\ min_i \|x(k+p) - x_i^*\|$, i=[1, m].

i. If $\|x(k+p) - x_j^*\| < \varepsilon$, the s-th eligible pattern updates the consequent of the closest j-th rule as follows:

$J_j^* = (N_j J_j^* + \hat{J}(k+p)) / (N_j + 1)$; $N_j := N_j + 1$

ii. If $min \|x(k+p) - x_i^*\| \geq \varepsilon$ the s-th eligible pattern (x(k+p), $\hat{J}$ (k+p)) creates a new (m+1)-th rule, with parameters as follows:

$m := m + 1$; $x_m^* := x(k+p)$; $J_m^* := \hat{J}(k+p)$; $N_m := 1$

Parameter ε controls the ability of the RBIC to summarize data. For small ε more rules are created and for very small ε almost every operating point and the model associated with it form a separate rule. On the contrary, for large ε more eligible patterns are used to determine the parameters of a rule and the ability of the RBIC to summarize increases.

One simplified form of the RBIC contains a single *default rule* that contains the vector of operating conditions and the Jacobian corresponding to the best past performance of the control algorithm in terms of the performance measure (13). It contains a single model that can initialize the learning algorithm. This default rule reflects the well-known engineering practice to copy the past best parameters of a regulator for further use if a better control performance cannot be accomplished.

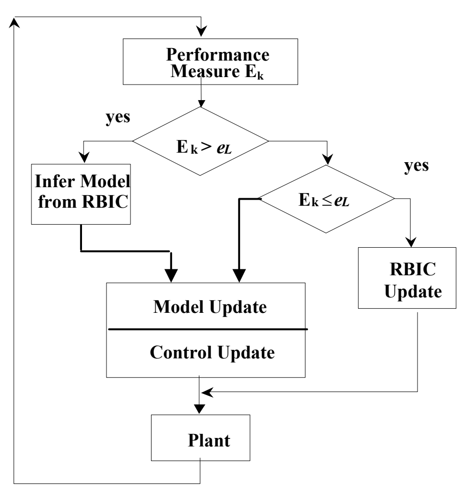

Figure 1. RBIC Intelligent Control Algorithm

The block-diagram in Figure 1 illustrates the modular structure of the RBIC Intelligent Control Algorithm. By combining the two optional model learning modules and the two control updating modules with the RBIC module we arrive at four alternative versions of the algorithm.

The RBIC Intelligent Control Algorithm is the main component of the Integrated Paint Quality Control System that was developed and implemented recently at Ford Motor Company. In the next section we discuss the application of the RBIC Intelligent Control Algorithm to paint process control.

VI. INTEGRATED PAINT QUALITY CONTROL SYSTEM AS A PROCESS CONTROL TOOL

Paint quality, durability, and color matching are critical for customer satisfaction and require significant process control efforts. Generally, paint film thickness is considered one of the main factors that determine overall paint quality. Typically, no more than 2-3 vehicles per shift are measured off-line and this information is used to make decisions on adjusting the paint applicator parameters. This film thickness data is not enough to control and optimize process performance taking into account the variety of possible combinations of colors, styles, and paint equipment needed to paint a vehicle. The need for paint process control as a tool for improving paint quality and optimizing material usage has been well recognized in the automotive industry. Ford Motor Company's Integrated Paint Quality Control System (IPQCS) is an automatic spray booth control system that integrates paint applicators and paint film thickness measurements into a closed loop feedback control system which reduces paint process variability and produces consistent, high quality paint finishes.

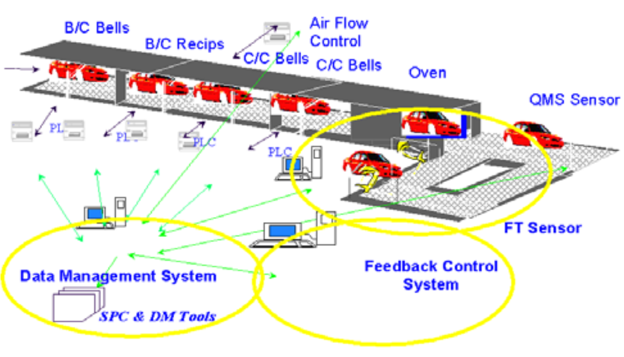

Figure 2. Integrated Paint Quality Control System (IPQCS).

Figure 2 shows a typical automotive paint facility that is controlled by the Integrated Paint Quality Control System. In automotive assembly plants three layers of paint (prime, base and clear) are applied by groups of robotized PLC controlled paint applicators (bells and guns) that are located in one prime and two enamel paint booths. The amount of paint applied by each applicator depends on many factors but the most critical ones are the fluid flow and the shaping/atomizing air (the air dispersing the paint that is sprayed by the applicator). These two parameters have different values when the paint is applied to different panels (zones) along the vehicle body resulting in a set of 10 control variables (for an average number of five zones). Thus for a typical paint booth with 9 applicators the spatial distribution of the paint on the vehicle body is determined by 90 control variables taking different values for each color, style, and layer. The main process quality characteristic, the film thickness, is measured by a contact film thickness sensor AutoPelt - robotized version of the High Frequency Ultrasonic Contact Sensor [10] (Figure 2). It scans the paint film thickness distribution in a semi on-line mode, i.e. selected vehicle is moved off-line to a conveyor cell where a sequence of simultaneous measurements of the e-coat, prime coat, base coat and clear coat layers in 90 -100 locations is taken during a 150 s cycle. The sensor is capable to selectively measure one of every 3-5 vehicles (depending on the conveyor speed) representing the overall color and style mix of the paint shop. For each measured vehicle the AutoPelt sensor produces a vector of 90-100 controlled variables per layer.

Figure 3. AutoPelt contact sensor scans the film thickness distribution of each coating layer in up to 100 locations.

VII. PAINT PROCESS CONTROL

From a system perspective, the process of painting even a single coating layer on a vehicle body is extremely complex – the number of possible control inputs (bell/gun parameters) is in the range of hundreds, while the number of outputs is close to a hundred (the number of outputs is determined by the sensor resolution, i.e. vehicle body locations where the film thickness is measured). This complex multivariable system structure is under considerable disturbances (variations in environmental parameters, paint parameters, and equipment). In addition, the relationship between bell/gun adjustable parameters and the measured paint thickness on the vehicle body is not well defined. Those who have operated paint booths for a long time still have difficulty to formulate some clear and straightforward rules to relate the inputs to the outputs of the system. Because of this complexity, combined with the lack of reliable paint film measurement devices, the automotive painting process is often regarded as art rather than science.

In order to structure and formalize the problem of paint process control we can make at least three assumptions that would allow us to introduce a reasonably simplified model of the complex relationships in the paint booth.

The first assumption is that the automotive paint process can be decomposed into manageable independent subsystems of smaller size (each subsystem is associated with one unique combination of layer/color/style/paint booth, e.g. *base layer/amazon green/sedan/Enamel 1*). This is done by appropriate restructuring of the input/output process data. Figure 4 shows the graphical subsystem structuring interface that is used to determine subsystem structure. This process of structuring subsystems is done only once when the control algorithm is set-up or a new/color/style is included.

The second assumption is that the system dynamics can be ignored and each subsystem is covered by a MIMO nonlinear model (1) where u(k) is the vector of subsystem inputs (fluid flow rates and shaping air settings in each spray zone of all applicators that effect a particular subsystem) and y(k) is the vector of subsystem outputs (film thickness measurements for the particular subsystem in all measured locations). This assumption is justified because of the discrete nature of the paint process that has a sampling rate dictated by the conveyor speed rather than the actual process dynamics (in other words, changes in control actions are not applied while the vehicle is being painted).

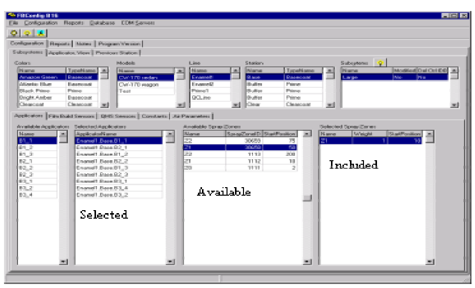

Figure 4. Graphical Subsystem Structuring Interface

Our third assumption is that selected measured locations are sufficiently representative for the paint film thickness distribution over the vehicle body.
Based on these assumptions we formulate the problem of feedback control of paint quality for any of the subsystems as a constraint nonlinear minimization problem with objective function:

$$Min_{u(k)} \, (\hat{y}(k) - y_d)^T \, (\hat{y}(k) - y_d)$$

subject to constraints:

$$\hat{y}(k) = y(k-1) + \hat{J}(k)\,(u(k) - u(k-1))$$

$$u_{min} < u < u_{max}$$

where y_d is the vector of target film thickness and u_{min}/u_{max} are lower/upper control limits.

Since the subsystems are independent by definition, one copy of the RBIC Intelligent Control Algorithm is associated with each subsystem. Therefore, in a typical paint shop with two enamel and one prime booth, two vehicle models, 12 base coat, and 4 prime coat colors, we deal with 60 subsystems each corresponding to a unique combination of applicator parameters and film thickness distribution. Each of the 60 subsystems is approximated by one linearized model (3) and is controlled by one of the 60 alternative copies of the RBIC Intelligent Control algorithm.

Jacobian model of a representative subsystem includes 90 input variables (6 vertical and 3 horizontal applicators with 5 flow zones and 2 control parameters each – fluid flow and shaping air) and 94 output variables (the film thickness of a coat layers measured in 94 specified locations. This amounts to a Jacobian matrix of a size 94 x 90. However, the actual number of the model parameters that are being learned is considerably less than 8460 due to the block diagonal structure of the Jacobian (see Figure 5).

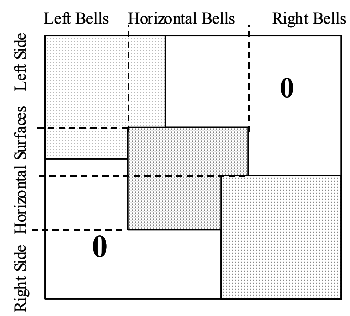

Figure 5. Block-diagonal structure of Jacobian subsystem model

The zero Jacobian parameters in Figure 5 correspond to a lack of input/output interaction (e. g. left vertical bells do not affect the film thickness of the horizontal surfaces and the right side of the vehicle) and are masked (excluded) from the learning process.

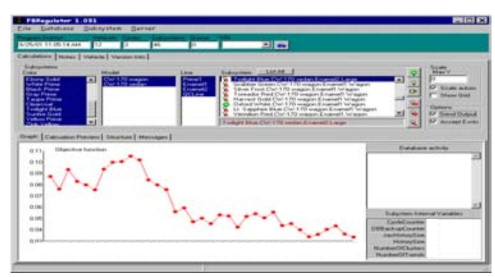

Figure 6. Graphical Feedback Control Interface

An interactive graphical interface (Figure 6) allows turning on/off a particular copy of the feedback control algorithm and tuning the parameters of the learning, control, and RBIC modules. It also enables selection one of the four alternative versions of the RBIC control algorithm. The recommended (default) version of the RBIC includes the RLS learning algorithm (6) as a model updating module and the LSQLIN constraint optimization algorithm as a control updating module. This selection was determined as the most efficient by simulation and has been used as a main control algorithm. The interface in Figure 6 also provides visual means for evaluation of overall control performance of individual subsystems, current number of rules created by the RBIC module, controller diagnostics, etc. The Graphical Feedback Control Interface (Figure 6) along with the Graphical Subsystem Structuring Interface (Figure 4) offers the capability for creating virtually independent subsystems and corresponding feedback controllers. This decomposition concept leads to a significant reduction of overall paint process complexity and transforms it into a manageable control problem.

Figure 7 shows how the RBIC Intelligent Control Algorithm is embedded in the structure of the Integrated Paint Quality Control System.

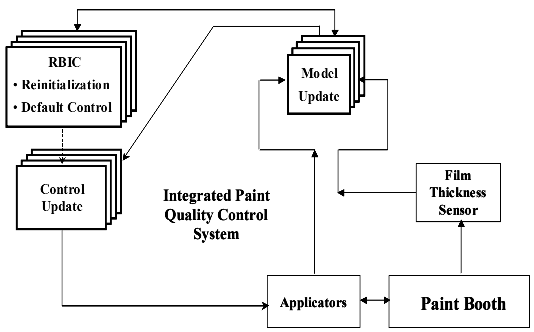

Figure 7. RBIC Intelligent Control Algorithm as a part of the Integrated Paint Quality Control System. The paint process control problem is decomposed into a set of virtual independent subsystems and a corresponding set of RBIC feedback controllers.

The RBIC Intelligent Control Algorithm automatically calculates the adjustments necessary to maintain process output (paint film thickness) on target. It eliminates shift-to-shift process variability and subjectivity and prevents production of vehicles with unsatisfactory paint quality.

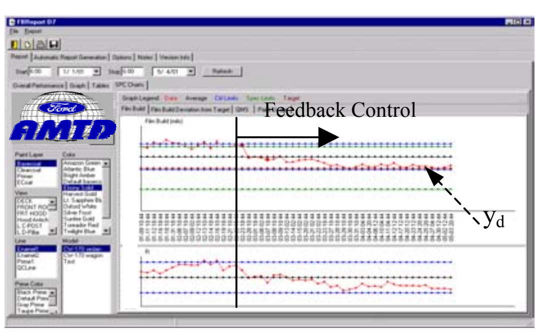

Figure 8. Effect of RBIC based feedback control on base coat film thickness. Average film thickness is maintained at the target level while the film thickness range (a measure of paint uniformity) is significantly reduced.

The Integrated Paint Quality Control System controls the film thickness of all paint layers within the specified range and has direct impact on the uniformity of the paint distribution over the vehicle body. Figure 8 shows the effect of the introduction of the feedback control based on the RBIC Intelligent Control Algorithm.

CONCLUSIONS

The RBIC Intelligent Control Algorithm was developed as a combination between conventional indirect adaptive control and RBIC - an intelligent tool supporting the adaptation process. It was inspired by the idea of introducing a set of "default" models that would reinitialize the learning algorithm with a model corresponding to the best past performance of the control algorithm under similar operating conditions. The role of the fuzzy rule base was to summarize the model information in the form of rules. We have also presented some results of the application of this control algorithm to automotive paint process control as a main component of Ford's Integrated Paint Quality Control System.

ACKNOWLEDGEMENT

The author wishes to thank the members of Ford's IPQC team Frank Migda, Gary Farquhar, Tom Larsson, Lee Feldkamp, Danil Prokhorov, and Lixing Ma for their valuable contributions to this project. The author also recognizes K-R Automation, Fanuc Robotics, JSR Ultrasonics, Behr Systems, and Perceptron for their assistance as suppliers who have helped to make this effort a success.

REFERENCES

[1] Filev, D.P., Bharitkar, S., and Tsai, M.F., "Nonlinear control of static systems with unsupervised learning of initial conditions", Proc. of NAFIPS, New York, NY, 1999

[2] Ma, L., Larsson, T., and Filev, D.P., "Intelligent control of static nonlinear MIMO systems", Proc. ACC, Chicago, IL,2000

[3] Filev, D.P., Larsson, T. and Ma, L., "Intelligent control for automotive manufacturing – rule based guided adaptation", Proc. of IEEE Int. Conference on Control & Instrumentation, Nagoya, 2000

[4] Filev, D.P., Larsson, T., and Ma, L., "Adaptive control of nonlinear MIMO systems with transport delay: conventional, rule-based or neural?", VI IEEE Conference on Fuzzy Systems, San Antonio, TX, 2000

[5] Astrom, K.J. and Wittenmark, B., "Adaptive Control", Addison-Wesley, 1989

[6] Narendra, K.S., Balakrishnan, J., and Ciliz, M.K., "Adaptation and learning using multiple models, switching, and tuning", IEEE Control Systems, 37-50, 1995

[7] Filev, D.P. and Larsson, T., "Intelligent adaptive control using multiple models", Proc. 2001 IEEE-ISIC Conference on Intelligent Control, Mexico City, Mexico, 2001

[8] Liung, L., "System Identification – Theory for User", Prentice Hall, 1987

[9] Specht, D. F., "A general regression neural network", IEEE Trans. on Neural Networks, 568-576, 1991

[10] Rosenberg, S., Moceri, M., "A 'sound' measurement for finishing", Quality On Line, http://www.qualitymag.com/articles.html, January 2000