Proceedings of the 2002 Winter Simulation Conference
E. Yücesan, C.-H. Chen, J. L. Snowdon, and J. M. Charnes, eds.

ON THE RESPONSE SURFACE METHODOLOGY AND DESIGNED EXPERIMENTS FOR COMPUTATIONALLY INTENSIVE DISTRIBUTED AEROSPACE SIMULATIONS

Paul Stewart

Department of Electronic and
Electrical Engineering
University of Sheffield.
Mappin St. Sheffield S1 3JD, UNITED KINGDOM

Peter J. Fleming

Department of Automatic Control and
Systems Engineering
University of Sheffield
Mappin St. Sheffield S1 3JD, UNITED KINGDOM

Sheena A. MacKenzie

QinetiQ Ltd.
Farnborough, UNITED KINGDOM

ABSTRACT

Distributed real-time simulation is the focus of intense development, with complex systems being represented by individual component simulations interacting as a coherent model. The real-time architecture may be composed of physically separated simulation centres. Commercial off-the-shelf (COTS) and Freeware Real-time software exists to provide data communication channels between the components, subject to adequate system bandwidth. However if the individual models are too computationally intensive to run in real time, then the performance of the real-time simulation architecture is compromised. In this paper, model representations are developed from dynamic simulation by the response surface methodology (RSM), allowing complex systems to be included in a real-time environment. A Permanent Magnet AC (PMAC) motor drive simulation with model reference control for a more electric aircraft application is examined as a candidate for inclusion in a realtime simulation environment.

1 INTRODUCTION

Extremely complex systems such as aircraft can be represented by distributed simulations, leading to the development of sophisticated architectures to administer the data transactions necessary for the components to operate as a unified whole. CORBA, a distributed architecture has acquired real-time extensions, while the High Level Architecture (HLA) has been developed in the United States by the Department of Defense to provide a real-time simulation environment. Individual research and development centres develop models which reflect their expertise, and can subsequently join a distributed environment to construct a sophisticated representation with a great deal of detail, accuracy and model flexibility at the component level. In this application, real-time flight simulators can be constructed with levels of detail focussing hierarchically down from the airframe flight characteristics, through aerothermal gas turbine engine models, to component models of fuel pumps, generators etc. Distributed components of the system are on the whole modelled in high level environments such as $Matlab$ / $Simulink$. Routines may also be coded in languages such as $Fortran$ or C. The distributed simulation architecture fulfills the task of registering the data types and protocols of these modelling environments, and providing channels for data streaming via a real-time kernel. These architectures often rely on dedicated local (LAN) or wide (WAN) area networks to provide necessary physical levels of bandwidth for the data transfer. Benchmark tests for the distributed architectures can calculate bandwidth available with any projected system. Finally, any individual component which runs slower than real-time will prevent the system from running in real-time if its operation is critical to the operation of the overall system. This last point is the most crucial, and will be considered in this paper.

Many complex systems, particularly nonlinear multivariable ones require in general relatively intense computation in order to provide an accurate dynamic simulation. This problem is exacerbated by the use of high-level simulation languages to provide the model environment. Also, highly detailed techniques such as Finite Element analysis

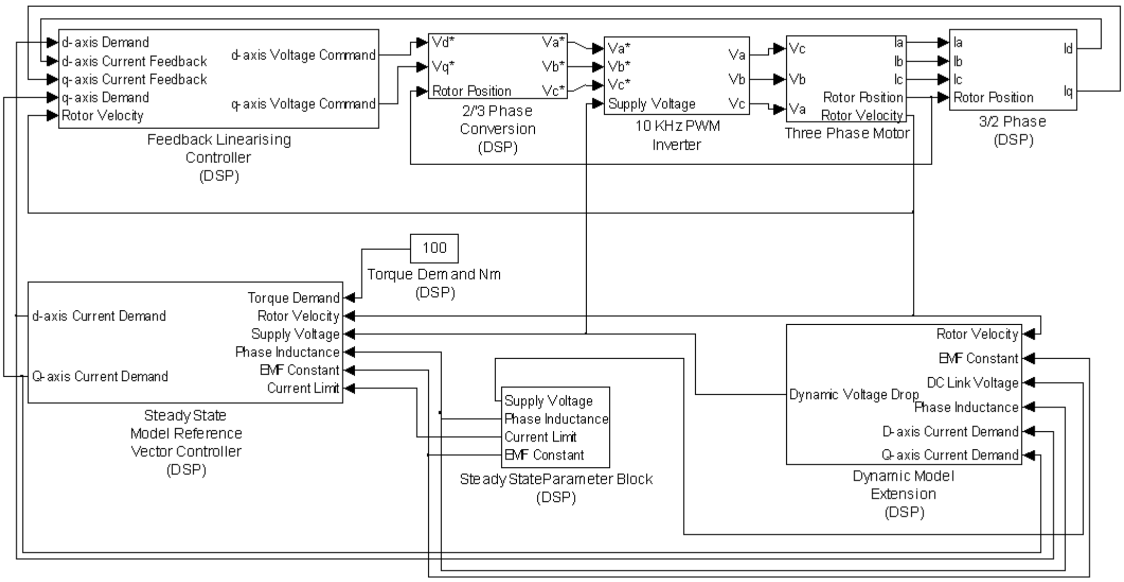

Figure 1: Simulink Motor, Power Electronics, Model Reference Controller as a Geared Rudder Actuator.

and Computational Fluid Dynamics, by definition require large amounts of time to perform incremental simulations. There exist some potential solutions to increase the speed of the model in order to bring it up to real-time capability. Firstly, the model can be further decomposed into sub-systems to run on separate machines. This solution has a penalty in terms of physical space, cost, and loading on the communications channels. Secondly, the item can be translated into highly optimised high level computer language code. This has the certain advantage of more rapid execution time, however the advantages of direct interaction with graphical development environments is lost, more specialised programming skills are required, and the code is not guaranteed to give the speed increase necessary. None of the above solve the problem of inherently intensive simulation. The model may be approximated as a lower order system. This is certainly a candidate solution, but requires a methodology to produce an accurate workable solution for highly non-linear systems. Finally, experimental or simulation data may be used to construct a minimal model of the system, in effect a technique analogous to image compression, with the aim of constructing a representation accurate to known statistical bounds, with real-time capability. This approach will be developed here, as it has the advantages of a generic approach, without the disadvantages of increased hardware costs or increased system bandwidth requirement, while retaining the advantages of graphical development environments. The solution is based around the response surface methodology (RSM) (Myers and Montgomery, 1995), and its effectiveness in application will be examined against both experimental and simulation data for a highly nonlinear Permanent Magnet AC (PMAC) motor drive model (Stewart and Kadirkamanathan, 1998, 1999). It is desired that the motor join a distributed flight simulator as the actuator motive power for an electromechanical rudder controller. In order to achieve the high velocities and torque required to achieve the operational bandwidth of the rudder, a complex highly non-linear model reference current controller is packaged in the simulation model (Figure 1) along with associated power electronics and feedback sensors (the simulation can be made more complex by the desired implementation of sensorless control).

It is imperative that wherever possible, in order to maximise the useability and applicability of the flight simulator, that its response reflects, as closely as possible, the real life dynamic response and interactions of the individual component parts. Critically, this allows the assessment and integration of new components and control systems in as "real" an environment as possible. This requirement makes real-time integration and component level complexity a critical goal. The RSM is investigated here to attain a real-time and useably accurate response for a highly complex aerospace component simulation.

Aside from its original application in chemical and process control, RSM has been used to realise low-cost computational solutions using Multi Objective Genetic Algorithms (MOGA) (Bica et.al., 1998, 2000). The RSM in this case is used to provide objective function approximations rather than evaluating these functions directly from the gas turbine engine during an optimisation procedure. In this context, response surfaces are used to model the engine performance indices and is applied to the multivariable fuzzy controller design for the Rolls-Royce Spey engine. It was found that the use of the RSM greatly reduced the computational load experienced in the controller design. The work presented in this paper uses the RSM in the context of system representation for distributed simulation. It is found to be an extremely useful and tractable tool to enable relatively slow processes to join simulations running in real-time.

2 RESPONSE SURFACE METHODOLOGY

The response surface methodology is a technique designed to optimise process control by the application of designed experiments in order to characterise a system (Myers and Montgomery, 1995). The relationship between the response variable of interest (y), and the predictor variables $(\xi_1, \xi_2, ..., \xi_k)$ may be known exactly allowing a description of the system of the form

$$y = g\left(\xi_1, \xi_2, ..., \xi_k\right) + \epsilon \tag{1}$$

where ϵ represents the model error, and includes measurement error, and other variability such as background noise. The error will be assumed to have a normal distribution with zero mean and variance σ^2. In general, the experimenter approximates the system function g with an empirical model of the form

$$y = f\left(\xi_1, \xi_2, ..., \xi_k\right) + \epsilon \tag{2}$$

where f is a first or second order polynomial. This is the empirical or response surface model. The variables are known as *natural variables* since they are expressed in physical units of measurement. In the response surface methodology (RSM), the natural variables are transformed into *coded variables* $x_1, x_2, ...x_k$ which are dimensionless, zero mean, and the same standard deviation. The response function now becomes

$$\eta = f\left(x_1, x_2, ..., x_k\right). \tag{3}$$

The successful application of RSM relies on the identification of a suitable approximation for f. This will generally be a first order model of the form

$$\eta = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k, \tag{4}$$

or a second order model of the form

$$\eta = \beta_0 + \sum_{j=1}^{k} \beta_j x_j + \sum_{j=1}^{k} \beta_{jj} x_j^2 + \sum_{i<j}\sum \beta_{ij} x_i x_j. \tag{5}$$

It may be neccessary to employ an approximating function greater than an order of two, based on the standard Taylor series expansion. The response surface methodology is intimately connected to *regression analysis*. For example when considering the first order model, the β terms comprise the unknown parameter set which can be estimated by collecting experimental system data. This data can either be sourced from physical experiments, or from previously designed dynamic computer models. The parameter set can be estimated by regression analysis based upon the experimental data. The method of least squares is typically used to estimate the regression coefficients. With $n > k$ on the response variable available, giving $y_1, y_2, ..., y_n$, each observed response will have an observation on each regressor variable, with x_{ij} denoting the ith observation of variable x_j. Assume that the error term ϵ has $E(\epsilon) = 0$ and $Var(\epsilon) = \sigma^2$ and the (ϵ_i) are uncorrelated random variables. The model can now be expressed in terms of the observations

$$\begin{aligned} y_i &= \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_k x_{ik} + \epsilon_i \\ &= \beta_0 + \sum_{j=1}^{k} \beta_j x_{ij} + \epsilon_i, \\ i &= 1, 2, ..., n. \end{aligned} \tag{6}$$

The β coefficients in equation (6) are chosen such that the sum of the squares of the errors (ϵ_i) are minimised via the least squares function

$$\begin{aligned} L &= \sum_{i=1}^{n} \epsilon_i^2 \\ &= \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{n} \beta_j x_{ij} \right)^2. \end{aligned} \tag{7}$$

The model can be more usefully expressed in matrix form as

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \tag{8}$$

where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \mathbf{X} = \begin{bmatrix} 1 & x_{11} & x_{12} & \dots & x_{1k} \\ 1 & x_{21} & x_{22} & \dots & x_{2k} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{n1} & x_{n2} & \dots & x_{nk} \end{bmatrix},$$
$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_n \end{bmatrix}, \quad \epsilon = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}. \tag{9}$$

It is now necessary to find a vector of least squares estimators **b** which minimises the expression

$$L = \sum_{i=1}^{n} \epsilon_i^2 = \epsilon^{'} \epsilon = (\mathbf{y} - \mathbf{X}\beta)^{'} (\mathbf{y} - \mathbf{X}\beta) \qquad (10)$$

and yields the least squares estimator of β which is

$$\mathbf{b} = \left(\mathbf{X}^{'}\mathbf{X}\right)^{-1} \mathbf{X}^{'}\mathbf{y} \qquad (11)$$

and finally, the fitted regression model is

$$\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}, \quad \mathbf{e} = \mathbf{y} - \hat{\mathbf{y}} \qquad (12)$$

where **e** is the vector of residual errors of the model.

3 PERMANENT MAGNET AC MOTOR AND CONTROLLER

A brief description will be presented in this section to give an overview of the complexities of a permanent magnet AC (PMAC) motor and controller, and the level of computational overhead demanded by (for example) a *Simulink* model. The controller in question is a model reference type, which allows the torque speed envelope of the motor to be greatly increased (Stewart and Kadirkamanathan, 2001). The general configuration of this class of motor is three phase AC, and a non-linear transform (*Park Transform*) is usually applied to the phase currents and voltages to allow for easier analysis and control system design (Pillay and Krishnan, 1995), to give the following description of the motor

$$V_q = r\, i_q + \omega\, L\, i_d + \omega\, \lambda + L \left(\frac{di_q}{dt}\right) \qquad (13)$$

$$V_d = r\, i_d - \omega\, L\, i_q + L \left(\frac{di_d}{dt}\right) \qquad (14)$$

where V_d, V_q are the d and q axis voltages, i_d, i_q are the d and q axis currents, r is the phase resistance, ω is the rotor velocity, L the phase inductance and λ the back EMF constant in the reference frame as volts/radian/second. The q-axis inductance is equivalent to the armature inductance, and the d-axis inductance is equivalent to the field inductance in a field wound DC machine. In the case of the surface mount PMAC motor, these quantities are equal and are denoted by L. The d and q variables are obtained from the three phase quantities via the following definition of the Park transform

$$\begin{vmatrix} V_q \\ V_d \end{vmatrix} = \frac{2}{3} \begin{vmatrix} cos(\theta) & cos(\theta - \frac{2\pi}{3}) & cos(\theta + \frac{2\pi}{3}) \\ sin(\theta) & sin(\theta - \frac{2\pi}{3}) & sin(\theta + \frac{2\pi}{3}) \end{vmatrix} \begin{vmatrix} V_a \\ V_b \\ V_c \end{vmatrix} \qquad (15)$$

where $V_{a,b,c}$ are the three phase elements. This transformation also applies to current and flux linkage quantities, and the three phase quantities may be obtained from the d and q axis variables by application of an inversion of the Park matrix in equation (15).

High performance phase current regulation is the key to high-performance motion control with the sinusoidal PMAC motor. Fast responding current regulation combined with self synchronisation via the built-in shaft position encoder make it possible to orientate the current phasor I anywhere within the d-q reference frame subject to supply current and voltage constraints. The speed of the torque response is limited only by the source voltage and stator inductance values. The baseline approach to torque control is to map the torque command T_e^* into demands for the d and q axis currents. Torque production is purely a linear function of q axis current. These current commands are then transformed into instantaneous sinusoidal currents for the individual stator phases, using the rotor angle feedback and the basic inverse vector rotation equations. PI current regulators for each of the three phases then excite the phase windings resulting in desired current amplitudes.

Examination of the motor equations (13,14) reveals cross coupling between the equations which requires correction by a feedback linearising controller to allow accurate control of the current and voltage vectors (Stewart and Kadirkamanathan, 2000). The flux weakening controller is of the model reference type, supplies current commands based upon rotor velocity and torque demand, and is based upon the standard circle diagram representation (Miller 1993, Jahns 1987). As speed and frequency increase, the current limit locus remains fixed, however the radius of the voltage limit locus decreases. The PWM control saturates when its duty cycle reaches maximum, and the available sinewave voltage from the inverter equals the phase voltage. This operating point is known as “base speed" and occurs on the circle diagram at the intersection of the q axis, current limit circle, and voltage limit circle. If the rotor velocity increases further, the radius of the voltage-limit circle decreases further, and maximum current is defined by a current vector terminating in the intersection of the two circles. As rotor velocity increases, the voltage-limit circle drags the current phasor further and further ahead of the q axis, decreasing the torque producing current, and increasing the demagnetising negative d axis current. This acceleration can increase until the point where the current vector lies entirely in the demagnetising direction, and no further torque production is possible.

The motor and controller models are coded into Simulink models, together with function blocks to describe real system objects such as the three phase PWM inverter, feedback linearising controller, rotor position encoder, Park transform blocks, and blocks to describe the voltage drop due to the current dynamics (Stewart and Kadirkamanathan, 1998). An external PID rudder position control loop supplies torque demand signals to the PMAC motor and controller simulation. However when the Simulink model of the model reference control system was run on a Pentium 3 800MHz computer with 512Mb of system memory, it was found not to be capable of real-time operation, running outside wall clock time by a factor of five at the required sampling rate of 1kHz. It was thus decided to implement a minimal description of the controller by RSM in order to achieve real time status for the motor/controller simulation.

4 RESPONSE SURFACE OF THE MODEL REFERENCE CONTROLLER BY DESIGNED FACTORIAL EXPERIMENT

The model reference controller can be converted into a response surface by viewing it as an input-output model. In this respect, the input variables are rotor velocity and torque demand, while the output variable is the angle of current vector advance. A factorial approach to experimental design was adopted (Hicks and Turner, 1999) to populate the three dimensional response space. An initial approach to deriving this population would be to combine each increment of PWM modulation depth with each increment of velocity to give the output torque surface. This would result in an output space (see Figure 2) with 180 experimental data points (all the points present in Figure 2). The operational space of the motor can however be divided into linear and nonlinear regions, by populating the map in terms of modulation depth and angle of current vector advance. This results in the linear region being defined by data points at the start and finish of the linear region for each modulation depth. The nonlinear region is defined by data points at 10^0 increments for current vector advance at each level of modulation depth. This factorial approach reduces the number of experimental data points to be collected to 90 (the black square points in Figure 2), an advantage of 50%. Modulation depths 0.3 and 0.4 have black squares across the linear region to indicate the range of experiments which have been eliminated.

Before the experimental proceedure was carried out, each data point was assigned a serial number between 0 and 90. The order of experimentation was decided via a random number generator in Matlab, according to experimental design procedures (Hicks and Turner, 1999). The characterisation study was conducted on a dynamometer rig. The method is to map the surface which describes the surface connecting torque, rotor velocity and current vector advance. The mapping is conducted for a quantization of 10^0 current vector advance, and PWM modulation depth from 0% to 100% in increments of 10%. The PWM modulation depth is given as *per unit* current on the left hand side of Figure 2. The RSM is utilised to construct a response surface which reflects the current advance profile in the flux-weakening region of the motor. The natural units ξ_1 (instantaneous torque in Nm) and ξ_2 (rotor velocity in rads/s) of the experimental dynamometer data is first transformed into the corresponding coded variables x_1 and x_2, such that

$$x_{i1} = \frac{\xi_{i1} - [max\,(\xi_{i1}) + min\,(\xi_{i1})]\,/2}{[max\,(\xi_{i1}) - min\,(\xi_{i1})]\,/2} \tag{16}$$

and

$$x_{i2} = \frac{\xi_{i2} - [max\,(\xi_{i2}) + min\,(\xi_{i2})]\,/2}{[max\,(\xi_{i2}) - min\,(\xi_{i2})]\,/2}. \tag{17}$$

The second order model to be fitted to the data is

$$\begin{aligned} y &= \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 \\ &+ \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \epsilon. \end{aligned} \tag{18}$$

Utilising equations (10-13), we obtain the coefficient matrix

$$b = \begin{bmatrix} 38.19 \\ 25.46 \\ -7.28 \\ -11.04 \\ 6.23 \\ -5.84 \end{bmatrix} \tag{19}$$

therefore the model of the flux weakening surface is

$$\begin{aligned} \hat{y} &= 38.19 + 25.46 x_1 - 7.28 x_2 - 11.04 x_1^2 \\ &+ 6.23 x_2^2 - 5.85 x_1 x_2 \end{aligned} \tag{20}$$

which may be slightly biased if the noise is coloured. The expression for the response surface model can now be implemented as a controller in *Simulink* and the preformance compared to the original *Simulink* model.

5 RESULTS

It is now possible to assess the usefulness of the second order model. By connecting the model reference controller and second order response surface model in turn to the existing PMAC motor and rudder drive model, the output can be compared to existing experimental data sets. In this case, the motor is accelerating from rest and run up to a rotor velocity of approximately 7500 rpm. This is a particularly arduous operating envelope, as the motor is being simulated with a

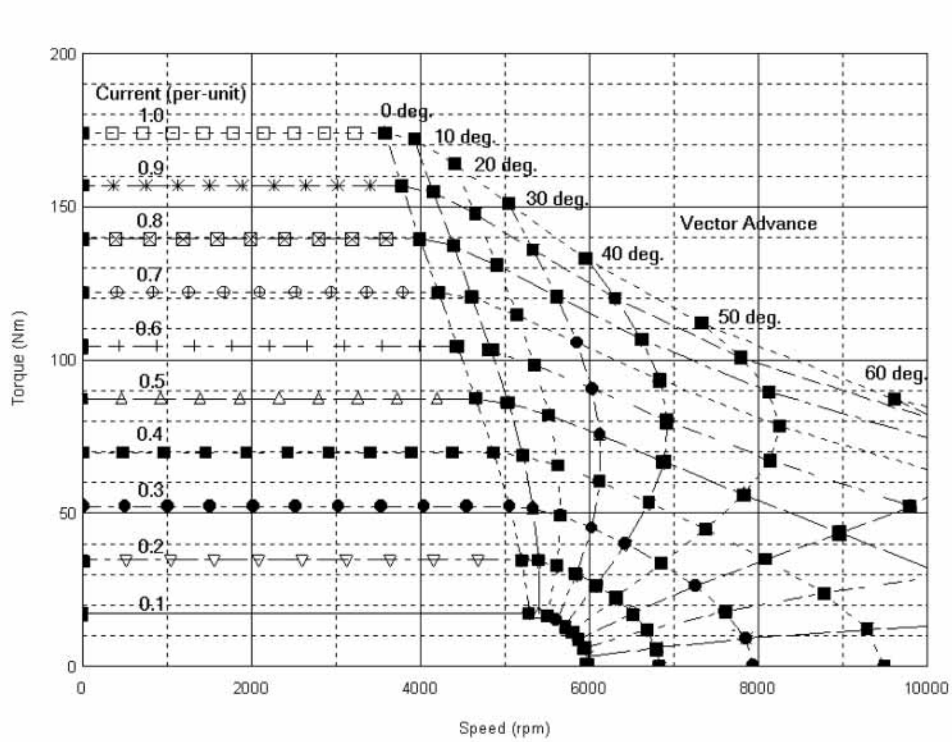

Figure 2: Torque Speed-Current-Vector Advance Experimental Data

coarse position feedback encoder rather than the more usual high-resolution encoder. This is responsible for the high level of ripple in the torque envelope. A visual inspection of the torque profiles produced by the two controllers reveals little or no difference in performance of the accelerating motor. The controller command outputs were found to have a mean difference of less than 1^0, which reflects the mean

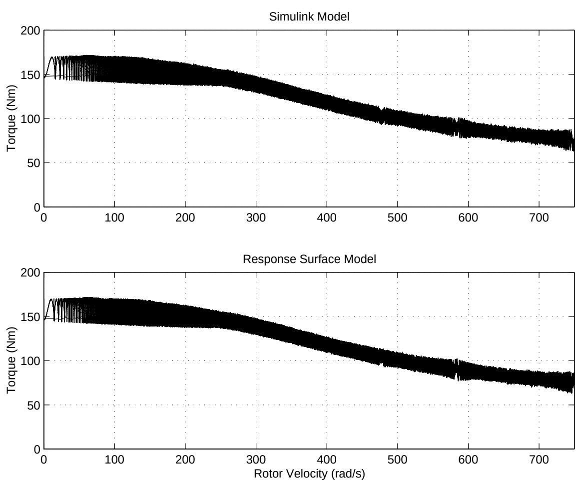

Figure 3: Comparison of Simulink and RSM Controllers in Motor Acceleration Simulation.

error in the initial controller design from experimental data, where the mean error of the second order fit was also $< 1^0$. The simulations were repeated for verification over a number of randomly chosen profiles over the entire torque/speed range. The error bounds were found to be $< 1^0$ in all simulations.

The second order controller obtained through the RSM was found to operate at more than ten times wall-clock time, compared to the original *Simulink* controller which could not attain real-time operation. The increase in performance ($>$ 50 times) was such that it was possible to run the rudder motor and controller on the same machine in real-time without any modifications to the motor-drive simulation, which is in itself a valuable improvement. The final implementation was to run the RSM designed controller on a separate machine from the motor and drive to confirm its benefits for distributed simulations. The simulation ran via a 100Mb/s LAN via standard network interface cards, and bespoke data buffers and sockets programmed in both C++ and Visual Basic. No buffer under-runs were experienced, and the controller was found to successfully stream data in real-time, operating at 1kHz.

6 CONCLUSION

A method has been presented which has been adapted from process control and optimization methodologies. The approach allows the adaptation of complex aerospace controllers and systems which are too slow to join real-time simulations to be approximated in a way such that real-time simulation becomes possible. The method has been demonstrated on a PMAC flux weakening controller for a rudder actuator, and found to provide not only the performance boost in terms of run-time, but also the accuracy required by the simulation environment. The method is shown to be a useful tool in the development of complex real-time systems.

ACKNOWLEDGMENTS

The author would like to acknowledge the Department of Trade and Industry (DTI) CARAD programme, and the Ministry of Defence's (MOD) Corporate Research Programme for funding.

REFERENCES

Bica B., G. Akat, A.J. Chipperfield, and P.J. Fleming. 1998. "Multiobjective design of a fuzzy controller for a gas turbine aero- engine", *UKACC International Conference on Control '98*, Swansea, UK, no.455, vol.2, pp.901-906.

Bica B., C. Ceconvciuc, A.J. Chipperfield, and P.J. Fleming. 2000. "Multivariable fuzzy controller design using multiple response surfaces", *UKACC International Conference on Control 2000"*, Cambridge, UK.

Hicks C.R. and K.V. Turner. 1999. "*Fundamental concepts in the design of experiments*, Oxford University Press, New York.

Jahns T.M. 1987. "Flux weakening regime operation of an interior permanent magnet synchronous motor drive", *IEEE Transactions on Industrial Applications*, vol.1A-23, no.4.

Miller T.J.E. 1993. "*Brushless permanent magnet and reluctance motor drives*, University Press, UK.

Myers R.H., and D.C. Montgomery. 1995. "*Response surface methodology: process and product optimization using designed experiments*, John Wiley and Sons Inc. USA.

Pillay P., and R. Krishnan. 1989. "Modelling, simulation and analysis of permanent magnet motor drives, part 1: the permanent magnet synchronous motor drive", *IEEE Transactions on Industrial Applications* vol.25, no.2, pp265-273.

Stewart P and V. Kadirkamanathan. 1998. "On steady state and dynamic performance of model reference control for a permanent magnet synchronous motor". *UKACC International Conference on Control 98*. University of Wales, Swansea UK. vol.455, pp.664-669.

Stewart P., and V. Kadirkamanathan. 1999. "Dynamic control of permanent magnet synchronous motors in automotive drive applications", *1999 American Control Conference*, San Diego, USA. pp.1677-1681.

Stewart P., and V. Kadirkamanathan. 2000. "Dynamic model reference control of a PMAC motor for automotive traction drives", *UKACC International Conference on Control 2000*, Cambridge UK.

Stewart P., and V. Kadirkamanathan. 2001. "Dynamic model reference PI control of flux weakened permanent magnet AC motor drives" *IFAC Journal of Control Engineering Practice.* vol.9, no.11, pp.1255-1263.

AUTHOR BIOGRAPHIES

PAUL STEWART is Research Associate with the Electrical Machines and Drives Group in the Department of Electronic and Electrical Engineering at the University of Sheffield U.K. His interests involve the application of optimisation and robust control techniques, real-time control and simulation, and advanced machine design and control. These interests have led to a wide variety of links with the automotive, aerospace and defence industries. He has over 30 publications in these areas.

PETER J. FLEMING is Professor of Industrial Systems and Control in the Department of Automatic Control and Systems Engineering at University of Sheffield and Director of the Rolls-Royce University Technology Centre for Control and Systems Engineering. His control and systems engineering interests include applications of optimisation and evolutionary computing, software for control system design and implementation, and distributed and parallel processing for real-time control and instrumentation. These interests have led to the development of close links with a variety of industries in sectors such as aerospace, power generation, pharmaceuticals, food processing and manufacturing. He has over 300 publications, including five books, in these research areas.

SHEENA A. MACKENZIE has 13 years experience in the field of control systems engineering and dynamic modelling & simulation. Since 1994 she has applied that experience to civil, military and industrial gas turbine engines in academia and industry. Her research has covered topics such as multivariable control, fuzzy logic control, the development of a virtual engine systems modelling environment and the design and development of control systems design processes for safety critical applications. She helped establish, and currently sits on, the National Advisory Committee working group for the More Electric Aircraft. She has worked for QinetiQ, and its predecessor, DERA, since 1997.