

SIMULATION & SYSTEM IDENTIFICATION OF HELICOPTER DYNAMICS USING SUPPORT VECTOR REGRESSION

Dr Sylvain Manso
Defence Science & Technology Organisation
506 Lorimer Street, Fishermens Bend, VIC 3207
Australia

ABSTRACT

This paper provides an overview of techniques developed for the application of Support Vector Regression (SVR) in the domain of simulation and system identification of helicopter dynamics. A generic high fidelity FLIGHTLAB helicopter model is used to train and validate a number of pitch response SVR models. These models are then trained using flight data from a Sikorsky Seahawk helicopter. The SVR simulation results show significant promise in the ability to represent aspects of a helicopter's dynamics at a high fidelity. To achieve this, it is important to provide the SVR kernel with knowledge of past inputs that encompass the delay characteristics of the helicopter dynamic system. In this case, the use of Nonlinear Auto Regressive eXogenous input (NARX) network architecture achieves this goal. Good performance was achieved using input data that encompassed between 300 to 500 ms worth of historic response.

1. INTRODUCTION

This paper provides an overview of techniques developed for the application of Support Vector Regression (SVR) in the domain of simulation and system identification of helicopter dynamics. SVR is a machine-learning technique that provides a 'black box' approach that may enable a more simplistic method for simulation whilst retaining acceptable levels of accuracy. SVR is the regression form of the more widely used Support Vector Machine (SVM) classification method. In this paper, the term SVM will refer to both the classification and regression methods, whereas SVR will specifically be used for the regression form.

The basis of the following work originally stemmed from the growing levels of complexity required from simulators to appropriately represent the rotary wing platforms in use by the Australian Defence Force (ADF). The ADF is currently acquiring and transitioning into service multiple helicopter platforms including the Eurocopter ARH Tiger, NHIndustries MRH-90, Sikorsky MH-60R, and Boeing CH-47F, all of which require simulation support. The Australian Defence Science & Technology Organisation (DSTO) provides some of this support. DSTO uses flight dynamic simulators to perform Human Machine Interface (HMI) studies, and to assist in accident investigation.

At present, the traditional technique for the dynamic modelling of helicopters and their systems involves the collection of flight data and aircraft specifications from which physics based theoretical equations are generated and validated. It is a time consuming process that requires the availability of a significant amount of data. The data required is often

proprietary or commercial-in-confidence, leading to a lack of availability, which can result in less than optimal simulations.

Another modelling approach involves the system identification of helicopter dynamics using frequency analysis techniques. This is a parameter estimation method in which measured aircraft responses are essentially inverted to extract a subset of the system model. This modelling approach requires flight data for aircraft response to a control input frequency sweep. Such flight data can be difficult to obtain from non flight-test aircraft due to the inherent risk of airframe structural stress when undertaking frequency response manoeuvres.

The implementation of a black box approach using machine-learning techniques may provide an option for a more simplistic method of simulation. A black box model can be defined as a machine with known or specified performance characteristics but whose constituents and means of operation are not necessarily known or specified by the user. For a given set of inputs, an expected set of outputs can be generated without explicitly knowing the relationship between input and output. A black box simulation would ideally only require flight data that is readily available to the operator of the helicopter platform.

Machine learning, using methods such as Neural Networks (NNs) and more recently SVMs, are a popular method for the implementation of black box modelling. The choice of using SVR for this investigation rather than other techniques based on NNs is due to the increased exposure and promised advantages of SVMs, including easier to train and more robust learning.

2. SUPPORT VECTOR MACHINES

SVMs arose from the area of statistical learning theory^[1] and were originally proposed by V.N.Vapnik^[2] in the early 1990s for the application of pattern classification. Since his seminal work, SVMs have been applied to a multitude of applications and undergone various transformations. Although the primary use of SVMs remains predominantly in the domain of classification, of particular interest is their use for regression in the form of SVR.

2.1 A Comparison with other Machine Learning Techniques

Since their inception, SVMs have been quite successful in solving real life problems and lifting the interest in statistical learning theory. Applications vary from face recognition^[3] and text categorisation^[4] to predicting stock market indices^[5] and modelling aerodynamic data^[6].

Scholkopf et al^[7] provided one of the original comparisons for classification between an SVM with Gaussian kernel, a Support Vector (SV) method hybrid with back-propagation, and a classical Radial Basis Function (RBF) machine. The results show that the SVM reached highest accuracy in the application of handwritten digits recognition. Another early SVR comparison was conducted by Mukherjee et al^[8]. Various approximation techniques including NNs and RBFs were applied to a chaotic time series. The SVM algorithm showed excellent performance here as well, outperforming other functions in most cases.

SVMs have performed favourably when compared to neural networks. One of the more relevant comparisons for this investigation is that of Fan et al^[6] who compare the generalisation ability of SVMs and NNs in the field of modelling aerodynamic data.

The key performance differences between SVMs and NNs relate to the minimisation principles^[9] on which they are based on. SVMs are founded on Structural Risk Minimisation (SRM) that minimises an upper bound of the generalisation error, whereas NNs are based on Empirical Risk Minimisation (ERM) that minimises the error on the training data. ERM can lead to local minima and over-fitting issues that need to be addressed by elaborate learning techniques. In contrast SRM generates a unique solution. This makes the application of SVMs in the real world a much easier prospect by removing the complexity associated with NN training for good general performance.

2.2 Support Vector Regression

A brief overview of SVR theory is presented below. A more thorough derivation is available from Vapnik's original work^[2]^[9], as well as tutorials, examples and overviews available in the references^[1]^[10]^[11]^[12]^[13]^[14].

Conceptually, SVM inputs are mapped to a higher dimensional, so-called feature space in which a decision surface lies. The support vectors themselves exist in the feature space of the SVM process and dictate the geometry of the decision surface. It is this decision surface that classifies each of the inputs in relation to the corresponding output, and it is the ability of the system to correctly classify previously unseen data, otherwise known as its ability to generalise, that dictates its usefulness. SVR differs from classification by approximating a function for the continuous output rather than that of a discrete response.

Given a set of N training points, where each example consists of an input vector, \mathbf{x}_i , and a label, y_i , such that:

$$(1) \quad \mathbf{x}_i \subseteq R$$

$$(2) \quad y_i \subseteq R$$

The object is to find a regression function $\hat{\Phi}(\mathbf{x}) = y$ that can approximate any new examples with the same underlying probability distribution $P(\mathbf{x}, y)$.

To allow for nonlinear regression functions, the training points are mapped from the current input space X to a much higher dimensional feature space Z using a nonlinear mapping φ . The regression function $\hat{\Phi}$ is defined to have at most ε deviation from the obtained targets y_i for all the training data, where the constant ε is chosen by the user. In other words, all the training points must lie within $\varepsilon > 0$ of the following linear hyperplane in feature space:

$$(3) \quad \hat{\Phi}(\mathbf{x}) = w \cdot \varphi(\mathbf{x}) + b$$

where w is the normal vector of the hyperplane, and the constant b is the bias.

To deal with noisy data, slack-variables ξ_i and ξ_i^* are introduced to penalise points outside the ε region. This corresponds to dealing with an ε -insensitive loss function defined by:

$$(4) \quad |\xi|_\varepsilon = \begin{cases} 0 & \text{if } |\xi| \leq \varepsilon \\ |\xi| - \varepsilon & \text{otherwise} \end{cases}$$

The SVR solution can then be found by solving the *primal* Quadratic Programming problem (QP):

$$(5) \quad \min_{w,b,\zeta} \left[\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N (\zeta_i + \zeta_i^*) \right]$$

subject to,

$$\begin{aligned} y_i - w \cdot \varphi(\mathbf{x}_i) - b &\leq \varepsilon + \zeta_i \\ w \cdot \varphi(\mathbf{x}_i) + b - y_i &\leq \varepsilon + \zeta_i^* \\ \zeta_i, \zeta_i^* &\geq 0 \end{aligned}$$

Conceptually, the first term achieves maximal margin for the hyperplane such that there is maximum distance between each of the points. The second term penalises the presence of any points outside the ε region. The constant C defines the trade-off between the two terms. The problem above represents a convex function with a unique minimum constrained to lie within a cube, although this solution occurs in the higher dimensional feature space due to the vector w .

Another formulation known as the *dual* QP problem is defined to constrain the solution to the input space, which is much simpler to compute. Suppose that the kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ is chosen such that the dot product in the feature space is equivalent to the kernel function in input space:

$$(6) \quad k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$$

Using Lagrangian multipliers α_i and α_i^* with the kernel trick above, the *dual* formulation is defined:

$$(7) \quad \max_{\alpha} \begin{cases} -\frac{1}{2} \sum_{i,j=1}^N (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) k(\mathbf{x}_i, \mathbf{x}_j) \\ -\varepsilon \sum_{i=1}^N (\alpha_i - \alpha_i^*) + \sum_{i=1}^N y_i (\alpha_i - \alpha_i^*) \end{cases}$$

$$\text{subject to } 0 \leq \{\alpha_i, \alpha_i^*\} \leq C, \text{ and } \sum_{i=1}^N (\alpha_i - \alpha_i^*) = 0$$

Conceptually, the optimisation problem above corresponds to finding the flattest function in the

feature space. Solving for α_i and α_i^* , the regression function for $\hat{\Phi}$ is given by:

$$(8) \quad \hat{\Phi}(\mathbf{x}) = \sum_{i=1}^N (\alpha_i - \alpha_i^*) k(\mathbf{x}, \mathbf{x}_i) + b$$

2.3 Kernel Functions

The constraint on the choice of kernel function in the SVM is to enable operations to be performed in the input space rather than the potentially high dimensional feature space. Specifically, the kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ chosen must satisfy the property such that the dot product in the feature space is equivalent to the kernel function in input space (Equation 6). This provides a way of addressing the curse of dimensionality which states that the difficulty of an estimation problem increases drastically with the dimension, Z , of the space.

Smola and Scholkopf^[1] describe the theorems and relevant corollaries used to characterise such kernels. Several well-known functions that can be used as kernels are provided in Table 1. Other possible kernel types include Splines, closed form B Splines, additive summing of kernels and Tensor products. For the results presented herein, a linear kernel is used for its computational performance and simplicity.

Table 1: List of commonly used SVM kernels.

Kernel	Function	Comments
Polynomial	$(\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$	Becomes a linear kernel when $d=1$
Radial Basis (Gaussian)	$\exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ ^2}{2\sigma^2}\right)$	Commonly referred to as the Gaussian function
Radial Basis (Exponential)	$\exp\left(-\frac{\ \mathbf{x}_i - \mathbf{x}_j\ }{2\sigma^2}\right)$	Commonly referred to as the radial basis function (RBF)
Multi Layer Perceptron	$\tanh(\rho(\mathbf{x}_i \cdot \mathbf{x}_j) + \vartheta)$	This is representative of the Neural Network equivalent
Fourier Series	$\frac{\sin(N + \frac{1}{2})(\mathbf{x}_i \cdot \mathbf{x}_j)}{\sin(\frac{1}{2}(\mathbf{x}_i \cdot \mathbf{x}_j))}$	Defined on the interval $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$

3. MODELLING OF HELICOPTER FLIGHT DYNAMICS

To provide a complete mathematical simulation of a helicopter's flight dynamics, one needs to represent the aerodynamic, structural and internal dynamic effects that once combined are influenced by the pilot controls and by external atmospheric disturbances [15]. Helicopter behaviour is dominated by the main and tail rotors, but limited by local effects that grow in influence at the limits of the flight envelope. These include, but are not limited to, blade stall, power limits, and control limits.

The method of modelling or extracting helicopter system dynamics or characteristics from flight test data is known as system identification. Machine learning techniques are a form of system identification when applied in this context.

There is little available in the literature on the use of SVMs for the system identification of a helicopter. Of most relevance is the recent work done by Bhandari et al [16] where an RBF kernel is investigated for the function estimation of a small scale helicopter. A few non-coupled models are developed to predict the longitudinal, lateral and tail rotor control inputs needed to achieve a desired flight trajectory, i.e. the inverse of a flight model. Flight data was initially post-processed through a Butterworth filter to reduce noise. Three data sets of 120 Hz resolution were constructed for training, validation and test purposes. These data sets relate control input directly to the appropriate angular rate of the aircraft. The initial SVR results look promising, although the extent of how well the model generalises is unclear.

Bhandari also developed a SVR model to predict pitch rate directly from longitudinal cyclic control, similar to the aims of this investigation. The testing and validation mean square errors are much higher than for the inverse problem above, yet the results show the correct trends. It is again unclear how well the model generalises or how the SVM was trained. It appears that input history was not implemented into their SVM model, likely resulting in the phase shifting of their results, and hence not suitable for dynamic modelling.

More progress with machine learning techniques is evident with the use of neural networks for helicopter system identification, particularly with the work of Mudigere, Kumar et al [17][18]. The predicted response of various models to control inputs have been satisfactory, though of most interest to the application of SVMs is that of the network architecture used to provide the dynamic system. The models are based on the Nonlinear Auto Regressive eXogenous input (NARX) network architecture for the identification and control of dynamical systems, first proposed by Narendra et al [19]. The NARX architecture introduces dynamics to

an otherwise static network model using Tapped-Delay-Lines (TDL) to feed past outputs and past inputs as inputs to the current model. Figure 1 depicts the architecture of a NARX network that is capable of modelling dynamics when trained using back propagation. The number of past values (TDLs) that are fed back into a NARX model is not defined and depends on an understanding of the order and degree of the system being identified.

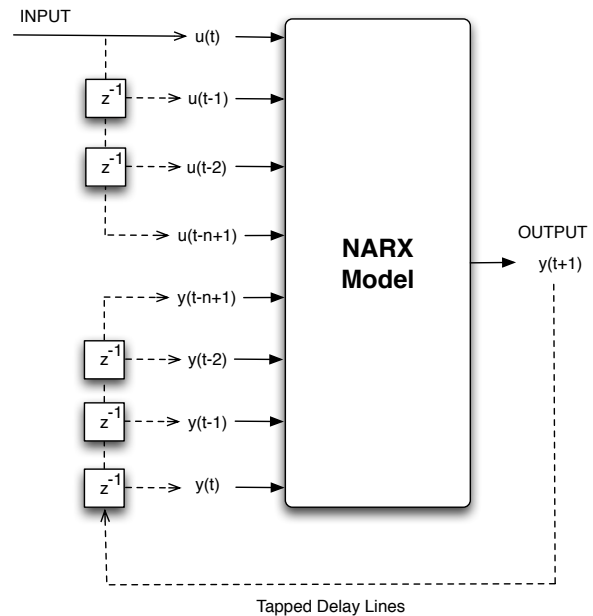


Figure 1: The NARX architecture to modelling system dynamics.

Previous work by the author [20] investigated the use of SVMs to model the longitudinal pitch dynamics of a helicopter using flight data. A simple NARX like model structure was implemented, where pitch rate was predicted based on historical pitch rate, pitch angle, and control input measurements. The model was trained using 180 Hz resolution data from a high fidelity flight dynamic model, as well as the use of real flight test data. A range of RBF and linear kernels were tested, with the results published showing good accuracy and potential for further modelling. It was stated that to achieve good results it is important to provide the machine with knowledge of past inputs that encompass the delay characteristics of the helicopter dynamic system. Also, the relationship, rather than the mechanics, between the significant variables that represent the dynamic system must be well understood.

The SVR technique proposed herein follows on from the author's previous work above and uses a pure NARX model and linear kernel to demonstrate the potential for system identification and modelling of helicopter responses.

4. SUPPORT VECTOR REGRESSION OF HELICOPTER PITCH DYNAMICS

A simulation of the pitch dynamics for a helicopter is presented to demonstrate application of the proposed SVR modelling technique. Using a NARX network and SVR with linear kernel, a FLIGHTLABⁱ helicopter model is used to provide training data. Results from 2 models (Figure 2) are presented in this paper. Each model predicts the pitch angle, θ , in response to a longitudinal control input, XB. The first model is trained using control response data at 30 knots airspeed. The second model is trained using control response data ranging from hover to 40 knots airspeed. The second model also includes airspeed as an additional training input.

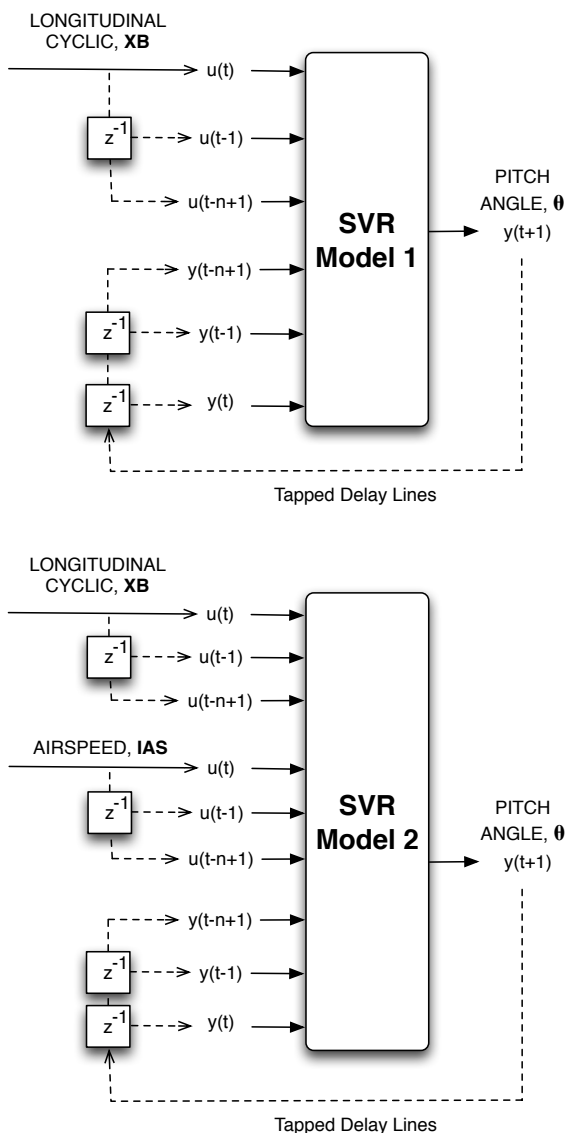


Figure 2: SVR Model Architectures implemented for simulation of helicopter pitch response.

ⁱ Developed by Advanced Rotorcraft Technology, Inc (ART), Sunnyvale, California, USA

The MATLABⁱⁱ environment is chosen to implement and develop the SVR models. This is achieved using the Spider SVM toolboxⁱⁱⁱ with LIBSVM as the primary code for the regression algorithms.

4.1 Flight Data: FLIGHTLAB Helicopter Model

FLIGHTLAB is the current helicopter-modelling environment used by DSTO. It is a commercial tool developed by Advanced Rotorcraft Technology Inc. (ART), for rotorcraft modelling and analysis. FLIGHTLAB is based on the Scope environment, which is an interpretive language that uses MATLAB-like syntax together with new language constructs for building and solving non-linear dynamic models. FLIGHTLAB provides a large range of aerospace and dynamics related components, which are used to develop flight models using object oriented design.

FLIGHTLAB uses multi-body dynamics to simulate real-time models. Generic modelling components are assigned specific values and parameters defining the aircraft. Each component is a self-contained dynamic entity that is interconnected to all other components through a child and parent structure. Solution components then take care of the kinematic and force interactions throughout the model.

A conventional medium sized twin-engine helicopter, with counter clockwise rotating rotor, was developed in the FLIGHTLAB environment. This model provides a source of noiseless data, which is highly amenable to the development of SVR modelling techniques. Table 2 provides a brief list of the major model parameters. The FLIGHTLAB model provides data at 180 Hz, which is then reduced to 10 Hz when training the SVR model.

Table 2: FLIGHTLAB Helicopter Parameters.

Rotor Parameters	Main Rotor	Tail Rotor
Radius (ft)	26.7	5.2
Chord (ft)	2.1	N/A
No. of blades	4	4
Rotor Speed (rpm)	256	1232
Rotor Twist (deg)	-12	-14
Airfoil Type	NACA 23012	N/A
Weight	20,400 lbs	
Engine Type	2 X Turboshaft	
Engine Power	2,800 SHP	
Control System	Rate based, with attitude stabilisation	
Longitudinal Cyclic Range	0 to 100% (+ve nose up)	

ⁱⁱ Produced by MathWorks, 3 Apple Hill Drive, Natick, Massachusetts, USA

ⁱⁱⁱ Developed by Weston, J., Elisseff, A., Baklr, G., and Sinz, F.: <http://www.kyb.tuebingen.mpg.de/bs/people/spider/index.html>

4.2 Method: SVR Training and Validation

The training data is scaled so that its mean and standard deviation are equal to one. Although not necessarily required, the normalisation of training data avoids certain inputs having more influence than others. This is particularly important when input variables have vastly different ranges, such as angles in radians and velocity in knots. It also allows some consistency in the choice of hyper-parameters ε and C when using different datasets, which would otherwise require more specific choices.

$$(9) \quad \mathbf{x}_{scaled} = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\sigma_x}$$

The regularisation coefficient, C , controls the trade off between training error and model complexity. A small value will increase the training errors, while a large value will lead to minimal training errors and a stronger correlation with the training data at the expense of generalisation (referred hereon as hard margin behaviour). It is noted from the literature^[21] that the value of C seems to have negligible effect when the insensitivity factor, ε , is well chosen. Values of C in this paper are varied from 0.01 to 1000.

The insensitivity parameter, ε , determines the level of training accuracy for the SVM by controlling the width of the ε -insensitive zone. If ε is larger than the range of the target values, then fewer support vectors are chosen. If ε is set to zero, hard margin behaviour is expected. Generally, the value of ε should increase when greater noise levels are present in the data. A good initial selection is to set ε to the accuracy desired. Values of ε in this paper are varied from 0.0001 to 1.

Quantitative validation is conducted by measuring the Mean Quadratic Loss (MQL) with comparison to the FLIGHTLAB output. A validation data set is then used as a method of both kernel parameter selection and performance testing.

$$(10) \quad \text{Mean Quadratic Loss} = \frac{1}{N} \sum_{i=1}^N |y_{actual,i} - y_{predicted,i}|^2$$

The training data set is chosen such that the model is taught aspects of positive and negative pitch response over a range of pitch angles and frequencies. A successive positive and negative impulse response and a pulse frequency sweep from 0 to 2Hz are used for this paper as shown in Figure 5.

Validation is performed using the training dataset as well as a specific validation dataset. The validation datasets include responses that have not been previously seen by the SVM. Three (3) validation datasets are chosen such that the generalisation

capability of the SVR Plant is tested. In this case a higher amplitude sinusoidal doublet and frequency pulse are chosen to test responses to unseen pitch dynamics for Model 1 (see Figure 6 & Figure 7). A step input response is used for Model 2 (Figure 8).

For the validation process, the initial conditions and input profile are chosen to begin the simulation. The initial conditions are used to begin the SVR prediction process where every subsequent time step builds upon the previous prediction of the SVR model. The predictions are then compared using MQL to the dynamic response of the FLIGHTLAB model that also began with the same initial conditions and input profile.

4.3 Discussion: Model 1 Pitch Response

An SVR model was developed to predict pitch angle response to a longitudinal control input at an airspeed of 30 knots. The model was trained using data at 10 Hz resolution. A linear kernel with a NARX network was developed as shown for Model 1 in Figure 2. One training dataset (Figure 5), and two validation datasets (Figure 6 & Figure 7) were used. This model required three variables to be defined. These included the SVM related insensitivity factor, ε , the regularisation coefficient, C , and the NARX related number of TDLs.

Figure 3 and Figure 5 show the effect in choice of TDL on the performance of the SVR when compared to both the original training data and the validation datasets. Good performance can be achieved provided that knowledge of past inputs is available. These inputs need to encompass the delay characteristics of the helicopter dynamic system. In this case, performance is deemed to become adequate with a TDL value of 3 or above. Because the model performs at the same rate as the training data, in this case 10 Hz, a TDL value of 3 represents the last 300ms of data.

Using a TDL value of 5, the insensitivity factor and regularisation coefficients are then varied to determine the models performance characteristics. When tested against it's training dataset, Figure 9 shows that a well chosen ε has greater influence than C . In this case, a low value of ε is guaranteed a low MQL error. The same trend is seen when tested against the unseen validation datasets (see Figure 11). The training data and validation data error surfaces have very similar shapes with a local minimum located at $\varepsilon = 0.1$ and $C = 0.1$. No hard margin behaviour is apparent, most likely due to the noise free flight data used from the FLIGHTLAB model.

Interestingly, the value of ε does not have as much influence on the number of support vectors used

when compared with the choice in regularisation coefficient, C (see Figure 13). A value of C that is greater than 0.1 requires significantly less support vectors, yet provides for similar good performance when compared to lower values of C . A lower number of support vectors would allow faster computational performance and a quicker training time.

4.4 Discussion: Model 2 Pitch Response

The second SVR model was developed to predict pitch angle response to a longitudinal control input at various airspeed values. This model was also trained using data at 10 Hz resolution. Similarly, a linear kernel with a NARX network was developed as shown for Model 2 in Figure 2. One training dataset, similar to Model 1 but for speeds from hover to 40 knots in 5-knot increments, and one validation dataset (Figure 8) were used. For this model, the validation dataset involved a sustained step input whose pitch response varied the airspeed of the helicopter. Similar to Model 1, the insensitivity factor, ϵ , the regularisation coefficient, C , and the NARX related number of TDLs needed to be defined.

Again, the importance of the NARX network in allowing a length of past inputs to capture the delay

characteristics of the helicopter response can be seen from the results in Figure 4. For this case, a TDL of 5 (500ms of data) provides best performance against the training and validation datasets. Of interest is the degradation in performance against the training data when higher values of TDL are used.

The MQL error surface is shown for a TDL value of 5 and variation in ϵ and C for testing against the training (Figure 10) and validation (Figure 12) datasets. Similar surfaces are seen for both data sets, and again the influence of ϵ on model performance is most evident, similar to Model 1. As was also seen with Model 1, the value of C has the greatest influence on the number of support vectors used (Figure 14).

In comparison to Model 1, Model 2 is inherently better able to generalise when larger pitch angles are achieved, particularly when such pitch angles involve significant variation in airspeed. This can be seen in Figure 8 where the time response of both models is shown against the third validation data set. Here, Model 2 is better able to predict the reduction in pitch angle over time, although not to the same level as the validation data.

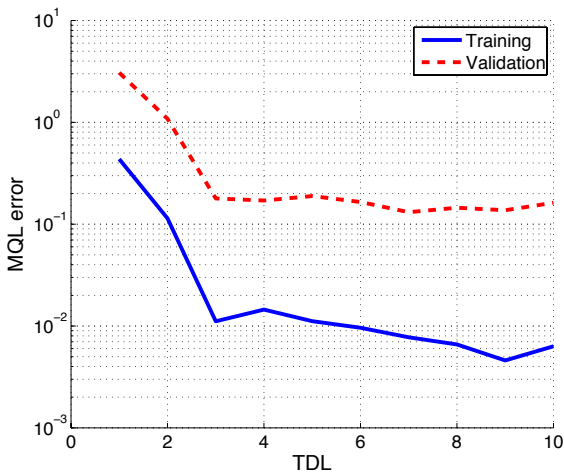


Figure 3: Model 1 MQL predictive error against the Training and Validation datasets with variation in TDL. $\epsilon = 0.01$ and $C = 0.1$. Validation dataset 1 & 2 are used

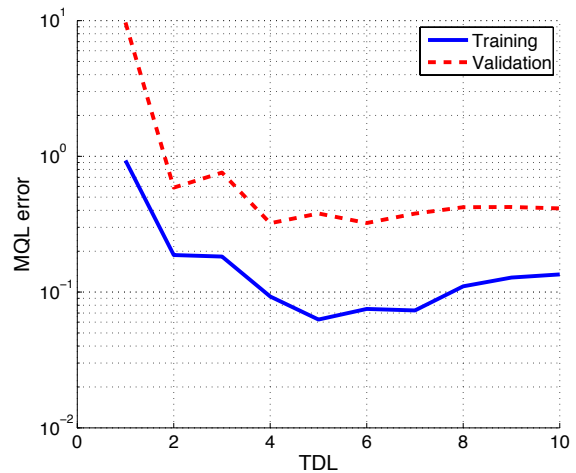


Figure 4: Model 2 MQL predictive error against the Training and Validation datasets with variation in TDL. $\epsilon = 0.01$ and $C = 0.1$. Validation Dataset 3 is used.

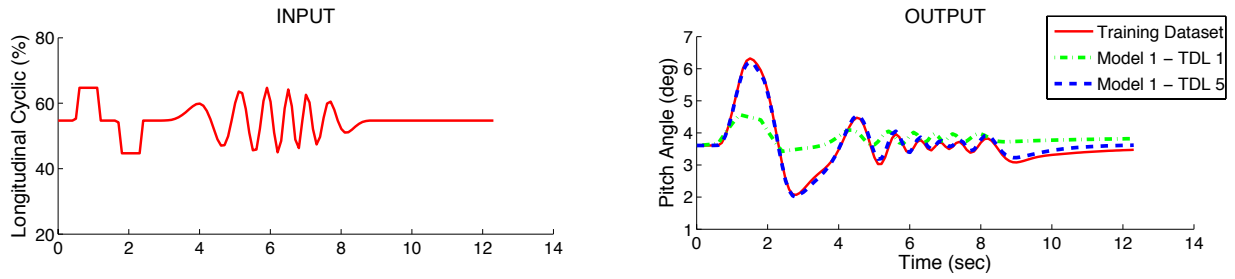


Figure 5: Training Dataset. Model 1 Performance is shown against its training dataset for TDL values of 1 and 5. $\epsilon = 0.01$ and $C = 0.1$.

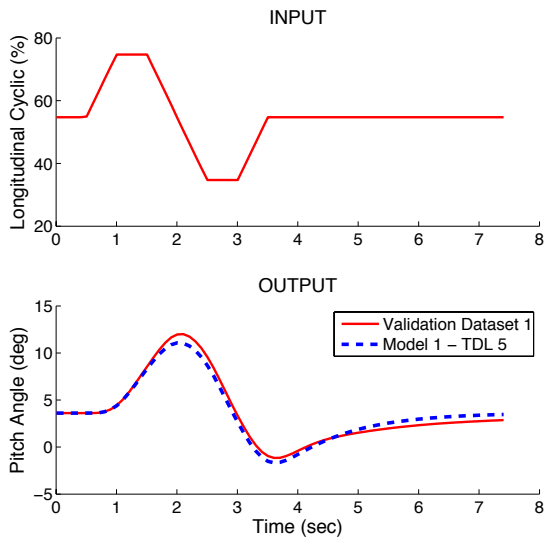


Figure 6: Validation Dataset 1. Model 1 Performance is shown against unseen data for a TDL value of 5. $\epsilon = 0.01$ and $C = 0.1$.

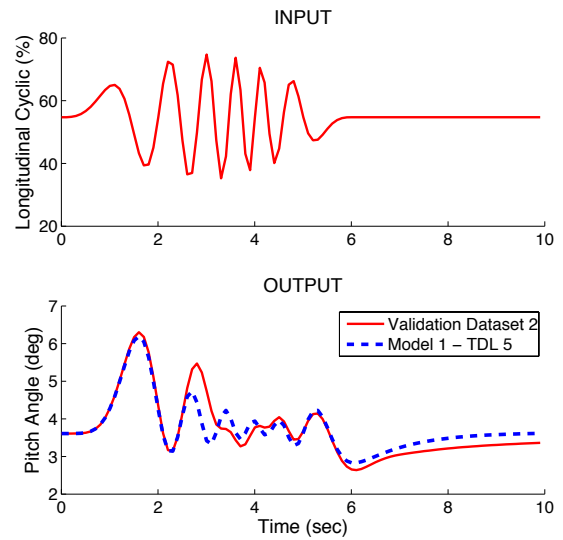


Figure 7: Validation Dataset 2. Model 1 Performance is shown against unseen data for a TDL value of 5. $\epsilon = 0.01$ and $C = 0.1$.

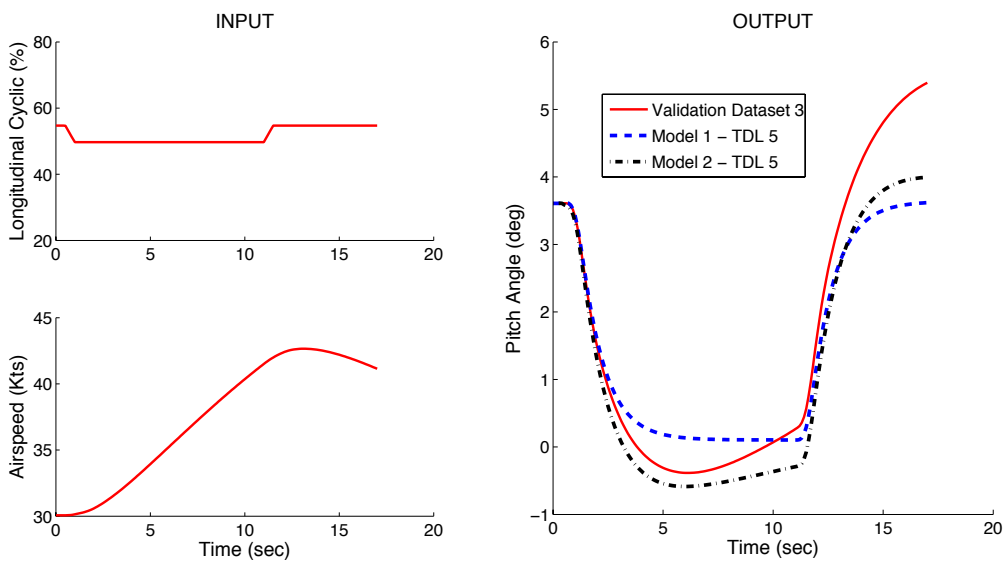


Figure 8: Validation Dataset 3. Performance of Model 2 is shown against unseen data for a TDL value of 5. For comparison, Model 1 results are also presented. $\epsilon = 0.01$ and $C = 0.1$.

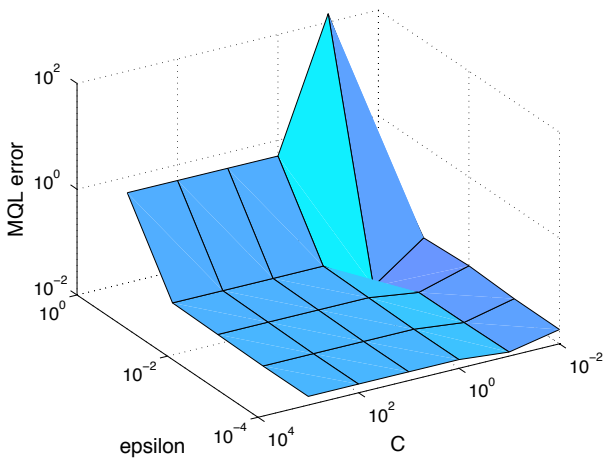


Figure 9: Model 1 MQL predictive error against the Training dataset. Variation in $\epsilon = 0.0001:1$ and $C = 0.01:1000$, TDL = 5.

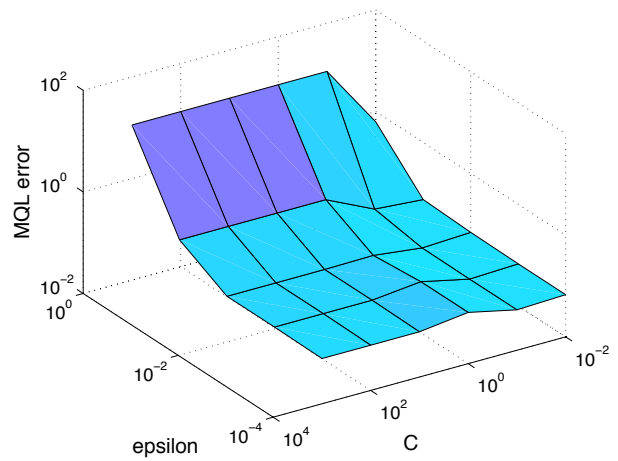


Figure 10: Model 2 MQL predictive error against the Training dataset. Variation in $\epsilon = 0.0001:1$ and $C = 0.01:1000$, TDL = 5.

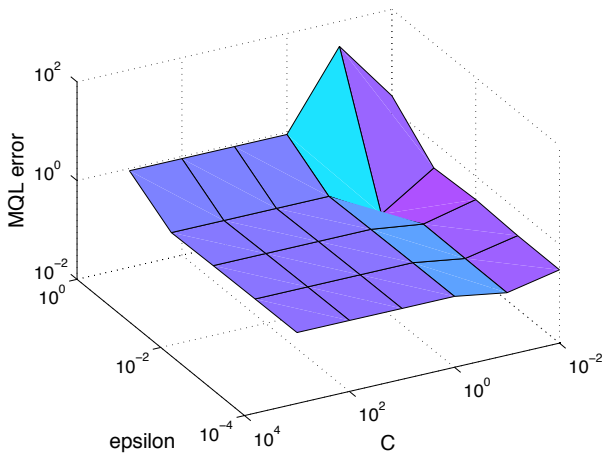


Figure 11: Model 1 MQL predictive error against the Validation Datasets 1&2. Variation in $\epsilon = 0.0001:1$ and $C = 0.01:1000$, TDL = 5.

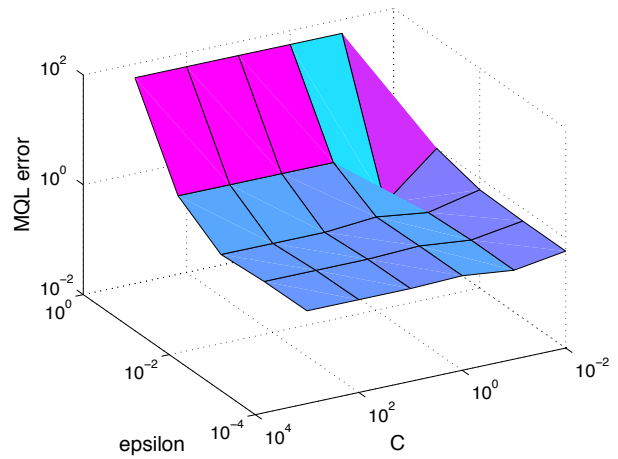


Figure 12: Model 2 MQL predictive error against Validation Dataset 3. Variation in $\epsilon = 0.0001:1$ and $C = 0.01:1000$, TDL = 5.

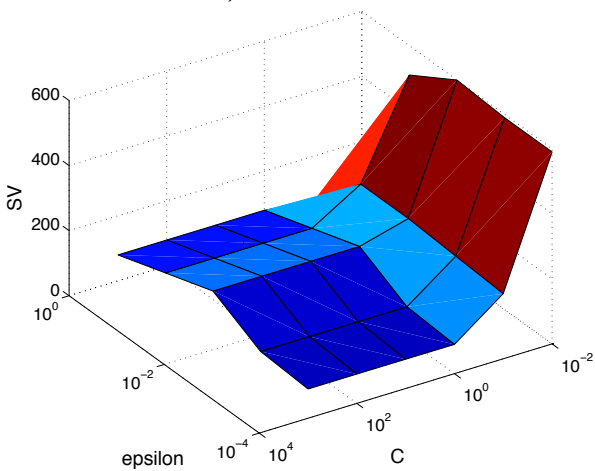


Figure 13: Model 1 Number of Support Vectors produced after training. Variation in $\epsilon = 0.0001:1$ and $C = 0.01:1000$, TDL = 5.

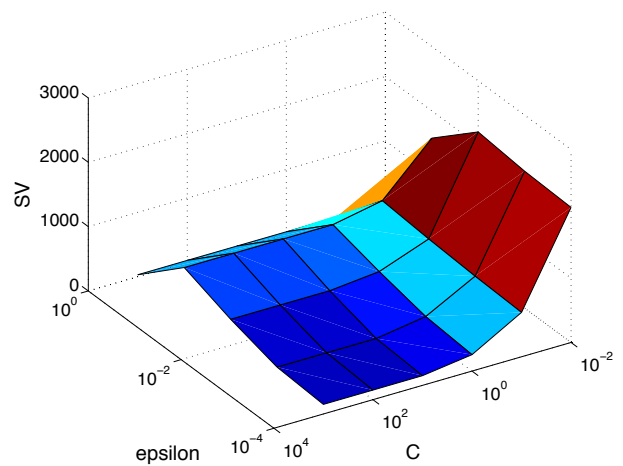


Figure 14: Model 2 Number of Support Vectors produced after training. Variation in $\epsilon = 0.0001:1$ and $C = 0.01:1000$, TDL = 5.

5. APPLICATION WITH FLIGHT TEST DATA

A small selection of flight test data recorded at 20 Hz from a Sikorsky Seahawk helicopter^{iv} is provided for training and testing of the SVR model. The training (Figure 15) and validation (Figure 16) datasets show pitch response to longitudinal cyclic input at hover conditions. Although the datasets are very small, this data provides an opportunity to train an SVR model using non-noise free data. Because only hover data was provided, the Model 1 NARX architecture was used.

Best performance against the validation dataset represents a TDL of 8, corresponding to 400ms of historic data. In this case, increasing level of TDL does not aid in performance against the unseen validation data (see Figure 17). The MQL predictive error surface against the training dataset (Figure 18) is very similar to the previous models taught with noiseless FLIGHTLAB data. In this case, a low value for ϵ and C provide the best performance against the validation data (Figure 19), even though the number of support vectors required show hard margin behaviour (Figure 20).

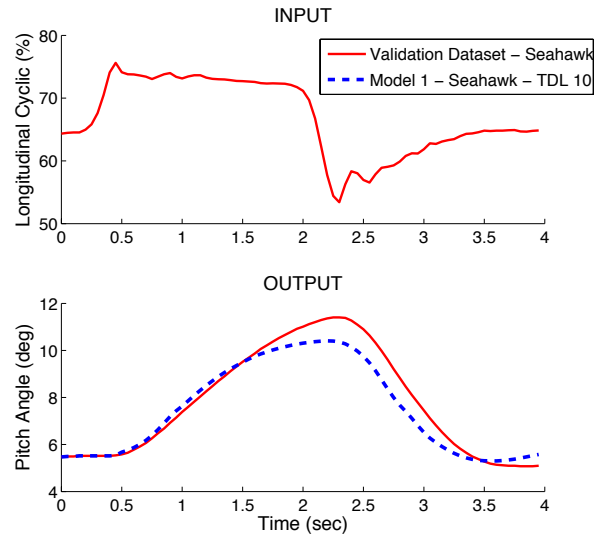


Figure 16: Validation Dataset. Model 1 Performance is shown against unseen Seahawk helicopter data for a TDL value of 10. $\epsilon = 0.01$ and $C = 0.1$.

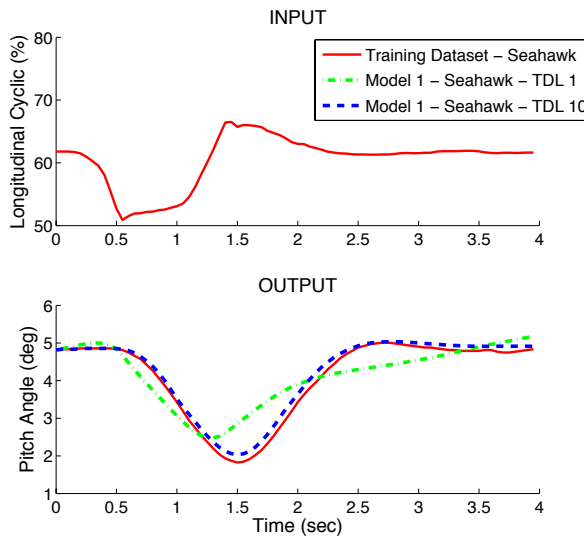


Figure 15: Training Dataset. Model 1 Performance is shown against Seahawk helicopter training data for TDL values of 1 and 10. $\epsilon = 0.01$ and $C = 0.1$.

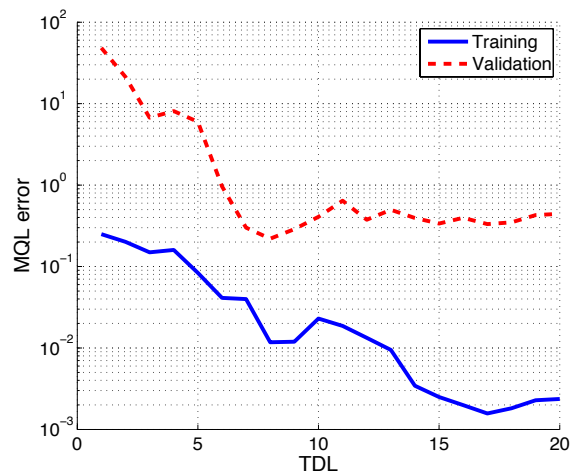


Figure 17: Model 1 MQL predictive error against Seahawk Training and Validation datasets with variation in TDL. $\epsilon = 0.01$ and $C = 0.1$.

^{iv} Recorded during the 1994 ADF Airborne Trials of the S-70B2 Helicopter. Provided courtesy of the Aircraft Maintenance and Flight Trials Unit (AMAFTU)

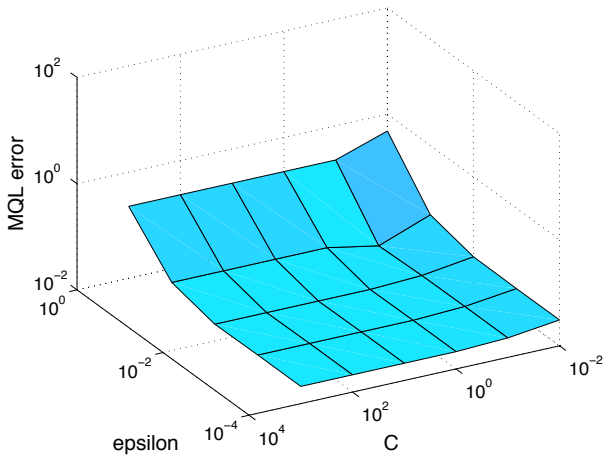


Figure 18: Model 1 MQL predictive error against the Seahawk Training dataset. Variation in $\epsilon = 0.0001:1$ and $C = 0.01:1000$, TDL = 10.

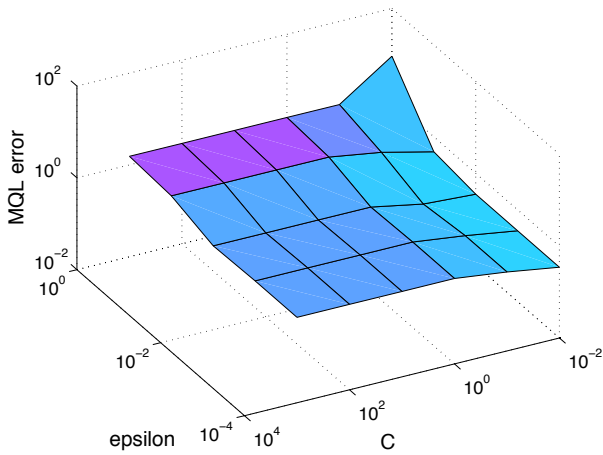


Figure 19: Model 1 MQL predictive error against the Seahawk Validation dataset. Variation in $\epsilon = 0.0001:1$ and $C = 0.01:1000$, TDL = 10.

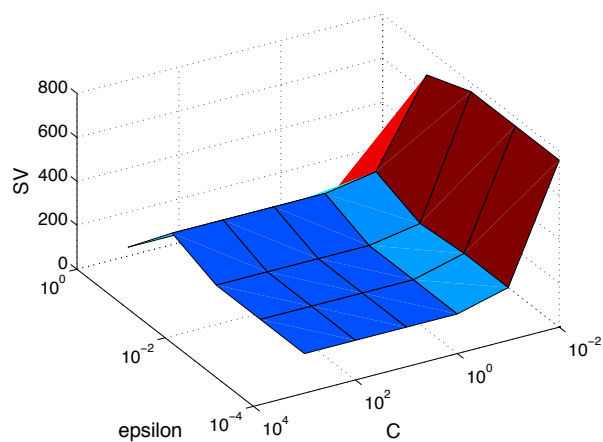


Figure 20: Model 1 Number of Support Vectors produced after training. Variation in $\epsilon = 0.0001:1$ and $C = 0.01:1000$, TDL = 10.

6. FURTHER DISCUSSION

The SVR results presented here show good generalisation capability when presented with unseen data. The choice in insensitivity factor, ϵ , has greater influence than the regularisation coefficient, C , to achieve low MQL. But in all cases, it is important to provide the plant with knowledge of past inputs that encompass the delay characteristics of the helicopter dynamic system. The use of NARX network architecture achieves this goal. Good performance requires a number of TDL that encompass between 300 to 500 ms worth of historic data.

The amount of Seahawk flight data available for training and validation was too limited to draw any major conclusions with comparison to the other SVR models. Although the other models were presented with noise free FLIGHTLAB data, the training performance was found to be similar to the Seahawk based model.

SVR exhibits one major disadvantage in comparison to traditional modelling and other machine learning techniques such as NNs. In its current form, a single SVR model is at best only a Multiple Input Single Output (MISO) system. A complete non-linear helicopter flight dynamic model will require many individually trained SVRs linked together as sub systems to provide the outputs that define a flight path.

It is likely that the most efficient method of developing a high fidelity flight model will be one that is a combination of both SVR and traditional modelling techniques. An SVR model may also be used to provide methods in which to reshape or reduce noise in flight response data. For example, a pure sinusoidal control response could be simulated from a limited flight data set, such that system identification can then be achieved using frequency analysis techniques.

The application of SVR models would lend itself well to application in the control system domain. This may include online system identification and over the horizon control, similar to many NN applications that are popular for use in Unmanned Aerial Vehicles today.

Unlike NNs, the use of SVR may lead to easier certification for use in manned aircraft or within commercial airspace. Risk mitigation would be easier due to the mathematical basis of the Structural Risk Minimisation and statistical learning principles on which SVMs are founded on.

7. CONCLUSIONS

The SVR model results show significant promise in the ability to represent aspects of a helicopter's dynamics at a high fidelity. To achieve this, it is important to provide the model with knowledge of past inputs that encompass the delay characteristics of the helicopter dynamic system. In this case, the use of NARX network architecture achieves this goal. Good performance requires a number of Tapped Delay Lines (TDL) that encompass between 300 to 500 ms worth of historic data.

None of the SVR models presented here model the effects of dynamic cross coupling, hence further work may be beneficial in this area. Further work is also recommended to investigate the SVR generalisation capability when appreciable noise is evident in the flight data stream.

8. ACKNOWLEDGEMENTS

The author wishes to thank the Aircraft Maintenance and Flight Trials Unit (AMAFTU) of the ADF for their permission to use Seahawk flight data in this paper.

9. REFERENCES

-
- [1] Smola, A. J., and Scholkopf, B., 'A tutorial on support vector regression', *Statistics and Computing*, no. 14, 2004, pp. 199-222.
- [2] Vapnik, V. N., *The nature of statistical learning theory*, Springer-Verlag New York, 1995.
- [3] Osuna, E., 'Applying SVMs to face detection', *IEEE Intelligent Systems*, vol. 13, no. 4, July/August 1998, pp. 23-26.
- [4] Dumais, S., 'Using SVMs for text categorization', *IEEE Intelligent Systems*, vol. 13, no. 4, July/August 1998, pp. 21-23.
- [5] Abraham, A., Philip, N., and Saratchandran, P., 'Modeling Chaotic Behavior of Stock Indices Using Intelligent Paradigms', *International Journal of Neural, Parallel & Scientific Computations*, vol. 11, 2003, pp. 143-160.
- [6] Fan, H., Dulikravich, G., and Han, Z., 'Aerodynamic data modeling using support vector machines', *Inverse Problems in Science and Engineering*, vol. 13, no. 3, June 2005, pp. 261-278.
- [7] Scholkopf, B., Sung, K., Burges, C., Girosi, F., Niyogi, P., Poggio, T., and Vapnik, V., 'Comparing Support Vector Machines with Gaussian Kernels to Radial Basis Function Classifiers', *IEEE Transactions on Signal Processing*, November 1997, pp. 2758-2765.
- [8] Mukherjee, S., Osuna, E., and Girosi, F., 'Nonlinear Prediction of Chaotic Time Series Using Support Vector Machines', *IEEE Neural Networks for Signal Processing*, 1997, pp. 511-520.
- [9] Vapnik, V. N., 'An Overview of Statistical Learning Theory', *IEEE Transactions on Neural Networks*, September 1999.
- [10] Gunn, S., *Support Vector Machines for Classification and Regression*, University of Southampton, 10 May 1998.
- [11] Platt, J., 'How to implement SVMs', *IEEE Intelligent Systems*, vol. 13, no. 4, July/August 1998, pp. 26-28.
- [12] Muller, K., Mika, S., Ratsch, G., Tsuda, K., and Scholkopf, B., 'An Introduction to Kernel-Based Learning Algorithms', *IEEE Transactions on Neural Networks*, vol. 12, no. 2, March 2001, pp. 181-201.
- [13] Scholkopf, B., 'SVMs - a practical consequence of learning theory', *IEEE Intelligent Systems*, vol. 13, no. 4, July/August 1998, pp. 18-21.
- [14] Anguita, D., Boni, A., and Ridella, S., 'A Digital Architecture for Support Vector Machines: Theory, Algorithm, and FPGA Implementation', *IEEE Transactions on Neural Networks*, vol. 14, no. 5, September 2003.
- [15] Padfield, G., *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modeling*, AIAA education series, 1999.
- [16] Bhandari, S., Chen, B., Colgren, R., and Chen, X., 'Application of Support Vector Machines to the Modeling and Control of a UAV Helicopter', *AIAA Modeling and Simulation Technologies Conference and Exhibit*, Hilton Head, South Carolina, 20 - 23 August 2007.
- [17] Mudigere, D., Omkar, S., and Kumar, M., 'Identification of Helicopter Dynamics Based on Flight Data Using a PSO Driven Recurrent Neural Network Model', *AHS 64th Annual Forum*, Montreal, Canada, 29 April 29 - 1 May 2008.
- [18] Kumar, M., Omkar, S., Ganguli, R., Sampath, P., and Suresh, S., 'Identification of Helicopter Dynamics using Recurrent Neural Networks and Flight Data', *AHS 59th Annual Forum*, Phoenix, Arizona, USA, 6-8 May 2003.
- [19] Narendra, K., and Parthasarathy, K., 'Identification and control of dynamical systems using neural networks', *IEEE Transactions on Neural Networks*, vol. 1, no. 1, 1990, pp. 4-27.
- [20] Manso, S., *Support Vector Regression of a High Fidelity Helicopter Flight Model*, Ph.D. Thesis, RMIT University, Australia, August 2008.
- [21] Cherkassky, V., and Ma, Y., 'Selection of Meta-parameters for Support Vector Regression', *ICANN 2002*, 2002, pp. 687-693.

COPYRIGHT STATEMENT

The author(s) confirm that they, and/or their company or organisation, hold copyright on all of the original material included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The author(s) confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ERF2014 proceedings or as individual offprints from the proceedings and for inclusion in a freely accessible web-based repository.