Effect of Measurement Noise on the Performance of a Depth and Pitch Controller using the Model Predictive Control Method

Leo V. Steenson,
Alexander B. Phillips
and Stephen R. Turnock
Fluid Structure Interactions Research Group
University of Southampton
Hampshire, UK
Email: l.v.steenson@soton.ac.uk

Maaten E. Furlong
Marine Autonomous and
Robotic Systems
National Oceanography Centre, Southampton
Hampshire, UK

Eric Rogers
Communications Signal
Processing and Control
Research Group
University of Southampton
Hampshire, UK

***Abstract*—In this paper a depth and pitch controller for a hover-capable AUV is designed and implemented in simulation. The effect on controller performance of random Gaussian noise on the feedback signals is evaluated. It has been shown that very small levels of measurement noise will result in the controller performance degrading substantially and behaving in an erratic fashion. A polynomial type filter has been proposed and integrated into the model predictive control algorithm. This modification reduces the effect of the measurement noise substantially and improves controller performance.**

I. Introduction

With increasingly complex demands on autonomous underwater vehicles (AUVs), high precision control systems are required. Typically gain-scheduled classical controllers are used on AUVs due to the controller simplicity and low computational effort ([5] and [4]) however, for precise manoeuvring, alternative control algorithms may provide better performance. For these control algorithms to transition from theory to application requires that simulations are performed that encompasses realistic mathematical models and measurement inaccuracies.

MPC has been recently gaining a significant amount of interest outside of the process industry from where it originated [3]. The reason for this is due to two main reasons; good performance of multi-input multi-output (MIMO) systems without significant tuning effort, and ease of implementation due to the standardisation of MPC tools. There exists only a handful of applications, with varying success, of the MPC algorithm to AUVs; both simulated, [1] and [7], and experimentally tested, [6] and [2]. Further work is required before MPC algorithms are ready to be applied to operational AUVs.

In this work a depth and pitch controller is derived using the model predictive control (MPC) algorithm for a hover-capable AUV. A similar controller has previously been developed for this AUV, [6]. The focus of this work is to examine the effect on system performance as the quality of feedback, to the MPC controller, degrades.

First the MPC algorithm is introduced and the derivations given. Next the depth and pitch system that is to be controlled is introduced and the simulation process described. The controller performance is then evaluated with and without measurement noise added to the state feedback. Performance is evaluated using the standard deviation of the outputs at the end of each simulation. A modification to the MPC algorithm is made and its performance evaluated against the standard algorithm.

II. Model Predictive Control

The concept of MPC is that the controller utilises a mathematical model (of the system to be controlled) so as to predict the future trajectory of the system for a given future control trajectory. Using this ability, the optimal future control trajectory can be found.

An overview of the MPC algorithm is given in Fig. 1. First the controller is initialised starting with a linear model of the system that is to be controlled, this is often a linear approximation of a non-linear system as is the case in this work. The augmented model is then built, this is used to predict the future system trajectory. Finally before starting the main control loop, the cost function that will be used to find the optimal control trajectory is defined and any other controller variables set, such as input constraints.

The main control loop starts by finding the optimal solution to the cost function. If this optimal solution violates the predefined constraints then the Hildreth programming procedure is utilised to find an optimal solution within the constraints. The first set of the control trajectory vector is then added to the previous system input set and applied to the system

In the rest of this section the derivation is given so as to build the augmented model, predict the future system trajectory, and find the optimal control trajectory. The full derivation of the algorithm can be found in [8].

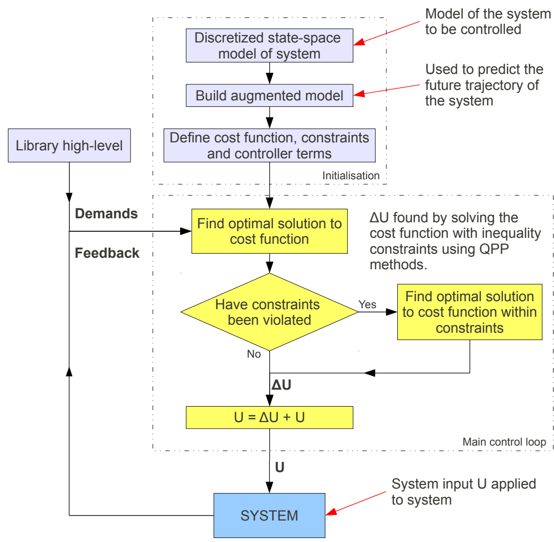

Fig. 1. Flow diagram describing the main steps within the model predictive controller.

A. Augmented Model

A linear system can be modelled with a linear state-space model with m inputs, q outputs and n_1 states:

$$\dot{x_m}(t) = A_m x_m(t) + B_m u(t), \quad (1)$$
$$y(t) = C_m x_m(t), \quad (2)$$

where u and y are the system input and output respectively and x_m is the state variable vector. It is assumed that $q \leq m$, otherwise it is unlikely that each output can be controlled without steady-state errors.

In order for the MPC algorithm to predict the future trajectory of the system, the system output can only be a function of the current state variable vector and not the input vector. This is the case with all inertia type mechanical systems; the input cannot have an instantaneous effect on the system output and therefore the D matrix, commonly found within state-space literature, is taken to equal zero.

The MPC algorithm used here embeds an integrator into the model, enabling the controller to deal with model inaccuracies. This is done by changing the system input from u to Δu by taking the difference between the current and previous state and input vectors:

$$\Delta x_m(k) = x_m(k) - x_m(k-1), \quad (3)$$
$$\Delta u(k) = u(k) - u(k-1), \quad (4)$$

Using equations (3) and (4) the state-space model becomes:

$$\Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k), \quad (5)$$

Connecting $\Delta x_m(k)$ to the output $y(k)$, a new state variable vector is created:

$$x(k) = \left[\Delta x_m(k)^T y(k)\right]^T, \quad (6)$$

Combining equations (5) and (6) results in the state-space model used within the MPC controller:

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \overbrace{\begin{bmatrix} A_m & o_m^T \\ C_m A_m & I_{q\times q} \end{bmatrix}}^{A} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \overbrace{\begin{bmatrix} B_m \\ C_m B_m \end{bmatrix}}^{B} \Delta u(k) \quad (7)$$

$$y(k) = \overbrace{\begin{bmatrix} o_m & I_{q\times q} \end{bmatrix}}^{C} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix}, \quad (8)$$

where $o_m = 0_{q\times n_1}$

B. Prediction of System Trajectory

In order to create a controller that utilises the system dynamics, first the ability to predict the future system trajectory, for a given future control trajectory, is created. The future system trajectory is calculated for a prediction horizon of N_p samples, for a future control trajectory of N_c samples. N_c must be less than or equal to N_p. The prediction will be denoted here as starting from sample number k_i where $k_i > 0$.

The future control trajectory can be denoted as:

$$\Delta u(k_i), \Delta u(k_i+1), ..., \Delta u(k_i+N_c-1), \quad (9)$$

The prediction horizon is the future state variables for the system and can be denoted as:

$$x(k_i+1|k_i), x(k_i+2|k_i), ..., x(k_i+m|k_i), ..., x(k_i+N_p|k_i), \quad (10)$$

where x represents the state variable vector at each sampling instant starting with the current plant information (measured or observed) at sample k_i. Using the augmented state space model from equations (7) and (8) the future state vectors can be calculated:

$$\begin{aligned} x(k_i+1|k_i) &= Ax(k_i) + B\Delta u(k_i) \\ &\vdots \\ x(k_i+N_p|k_i) &= A^{N_p}x(k_i) + A^{N_p-1}B\Delta u(k_i) \\ &\quad +A^{N_p-2}B\Delta u(k_i+1)... \\ &\quad +A^{N_p-N_c}B\Delta u(k_i+N_c-1). \end{aligned} \quad (11)$$

And therefore the future system outputs can be predicted as:

$$\begin{aligned} y(k_i+1|k_i) &= CAx(k_i) + CB\Delta u(k_i) \\ &\vdots \\ y(k_i+N_p|k_i) &= CA^{N_p}x(k_i) + CA^{N_p-1}B\Delta u(k_i) \\ &\quad +CA^{N_p-2}B\Delta u(k_i+1) + ... \\ &\quad +CA^{N_p-N_c}B\Delta u(k_i+N_c-1). \end{aligned} \quad (12)$$

The future output vector Y, of length N_p, can be represented as:

$$Y = [y(k_i+1|k_i), y(k_i+2|k_i), ..., y(k_i+N_p|k_i)]^T \quad (13)$$

And the future control horizon ΔU, of length N_c, can be represented as:

$$\Delta U = [\Delta u(k_i), \Delta u(k_i+1), ...\Delta u(k_i+Nc-1)]^T \tag{14}$$

Combining equations (11) to (14), the future system trajectory can be represented in a compact matrix form:

$$Y = Fx(k_i) + \phi \Delta U, \tag{15}$$

where

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}, \tag{16}$$

$$\phi = \begin{bmatrix} CB & 0 & ... & 0 \\ CAB & CB & ... & 0 \\ CA^2B & CAB & ... & 0 \\ \vdots & & & \\ CA^{N_p-1}B & CA^{N_p-2}B & ... & CA^{N_p-N_c}B \end{bmatrix}. \tag{17}$$

C. Optimization

In order to find an optimal control solution, first the set-point vector must be defined and a cost function created. The set-point vector, of length N_p, is defined as:

$$R_s^T = [1\ 1\ ...\ 1]\, r(k_i), \tag{18}$$

where $r(k_i)$ is the set-point at instant k_i. The cost function is defined as:

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U, \tag{19}$$

where the first quadratic term defines the objective of minimizing the error between the future system prediction vector, Y, and the set-point vector R_s. The second quadratic term is a function of ΔU and $\bar{R}$, where $\bar{R}$ is a diagonal matrix of the form:

$$\bar{R} = r_w I_{mN_c \times mN_c}, \tag{20}$$

and with $r_w > 0$ penalises the magnitude of ΔU. Therefore if it is desirable to minimise the error as quickly as possible then an r_w value of zero should be taken, resulting in large values within the ΔU vector. These large values may not be feasible in reality and will need to be handled using constraints.

To find the optimal ΔU vector, J must be minimized. Substituting equation (15) into equation (19), the cost function becomes:

$$\begin{aligned} J \;=\; & (R_s - Fx(k_i))^T (R_s - Fx(k_i)) \\ & -2\Delta U^T \phi^T (R_s - Fx(k_i)) \\ & +\Delta U^T (\phi^T \phi + \bar{R}) \Delta U. \end{aligned} \tag{21}$$

To minimize J, the first derivative with respect to ΔU is found:

$$\frac{\delta J}{\delta \Delta U} = -2\phi^T (R_s - Fx(k_i)) + 2(\phi^T \phi + \bar{R}) \Delta U, \tag{22}$$

and taking $\frac{\delta J}{\delta \Delta U}$ to equal 0, the optimal ΔU is found using:

$$\Delta U = (\phi^T \phi + \bar{R})^{-1} (\phi^T R_s - \phi^T F x(k_i)), \tag{23}$$

where matrix $\phi^T \phi$ has dimensions of $mN_c \times mN_c$, $\phi^T F$ has dimensions of $mN_c \times n$, and $\phi^T \bar{R}_s$ is the last q columns of $\phi^T F$.

Using equation (23) the future control trajectory can be easily calculated. When actually implementing this algorithm it is also desirable to apply constraints on the U and ΔU values. This is done using the active set method with the Hildreth programming procedure to identify the active constraints. The optimal solution is then found by treating the active inequality constraints as equality constraints. The handling of the constraints does not have a substantial influence on this work so the derivation is not given.

D. Feedback

Examining equation (23) shows that the feedback to this algorithm is in the form of the latest state-variable vector. From equation (6) it can be seen that the state-variable vector contains the system outputs y and the change of the original state-variable vector (of the original model) Δx_m. The state-variable vector used for feedback has the form:

$$x(k_i) = \begin{bmatrix} x_m(k) - x_m(k-1) \\ y(k) \end{bmatrix} \tag{24}$$

III. System Model

In this work the effect of measurement noise on a depth and pitch controller, built using the MPC algorithm, is evaluated. This system is an approximation of the low-speed hovering operational mode of the Delphin2 AUV; with the system inputs as the front and rear vertical tunnel thrusters and the system outputs of depth (z) and pitch (θ), see Fig. 2.

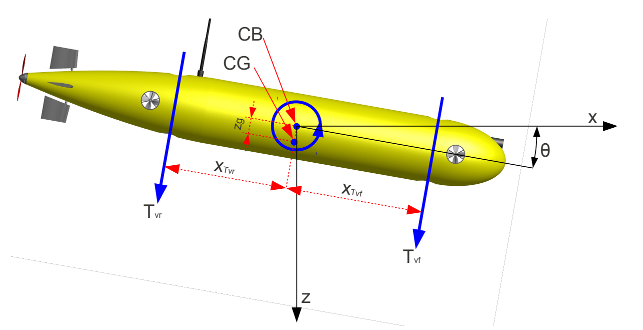

Fig. 2. Diagram illustrating the location of the front and rear vertical tunnel thrusters and the axis system.

A. Non-linear Model

In order to effectively test this controller in simulation, a non-linear model is used as the system. This non-linear system can be modelled as two coupled subsystems; depth and pitch:

1) Depth Model:

$$\dot{w_v} = \frac{1}{m_z}[T_{vf}\cos\theta + T_{vr}\cos\theta + (W-B) - \frac{1}{2}\rho V^{2/3}C_{Dw}|w_v|w_v] \tag{25}$$

$$w_v = \int_0^t \dot{w_v}\ dt,\ z = \int_0^t w_v\ dt \tag{26}$$

2) Pitch Model:

$$\dot{q_v} = -\frac{1}{I_y}[x_{Tvf}T_{vf} + x_{Tvr}T_{vr} - z_g W\sin\theta - \frac{1}{2}\rho V^{2/3}C_{Dq}|q_v|q_v] \tag{27}$$

$$q_v = \int_0^t \dot{q_v}\ dt,\ \theta = \int_0^t q_v\ dt \tag{28}$$

B. Linear Model

Both the pitch and depth models, equations (27) and (25), have non-linear trigonometric and quadratic terms that need to be linearised for use with the MPC algorithm. In this section the linearisation process will be described and a linear state-space model will be produced that estimates the AUV dynamics.

The trigonometric terms within the model include both the Sine and Cosine functions. The Cosine function is used to calculate the magnitude of thrust in the z axis. Here the Cosine function is approximated to equal 1, effectively removing the influence of pitch on thrust in the z axis. As the controller is being designed to maintain zero pitch, it is unlikely that the AUVs pitch will exceed $\pm 10^o$. Therefore the maximum the error produced by this approximation, equation (29), is within $\pm 1.5\%$. The Sine term is used to model the restoring moment within the pitch subsystem. Using small angle approximation, the Sine function is emitted from the equation leaving the angle (in radians). The maximum error produced using this approximation, equation (30), within pitch angles of $\pm 10^o$ is $\pm 0.5\%$.

$$\cos(\theta) = 1 \tag{29}$$

$$\sin(\theta) = \theta \tag{30}$$

Linearising the quadratic damping terms is more challenging. When the AUV has successfully arrived at the defined depth and pitch set-points, and is stable, the heave and pitch velocities are zero, thus the damping forces and moments are equal to zero. Linearising about zero heave and pitch velocities results in the damping coefficients also equalling zero, therefore eliminating damping from the model. Removing damping from the model would force the MPC algorithm to reduce system velocities using the system inputs (thrusters) rather than the natural dynamics, causing the controller to perform slowly and inefficiently. It is also intuitive that if the damping terms are estimated to be too high then the controller will overshoot the set-point.

The linear damping terms need to be a compromise between low and high speed operation. The linearised and non-linear damping terms are compared using the square of the normalised error:

$$(\text{Normalised error})^2 = \left[\frac{\text{non-linear term} - \text{linear term}}{\text{non-linear term}}\right]^2 \tag{31}$$

The sum of the squared normalised errors, from zero to the maximum velocities (estimated), is then minimised. The value of the linear damping coefficient that at the minimum is taken as the the coefficient for this work.

Combining equations (25) to (30) the AUV dynamics can be approximated with a time invariant state space model (assuming zero initial conditions):

$$\begin{aligned}\dot{x}(t) &= \overbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{\rho V^{2/3}k_z}{2m_z} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{-z_g W}{I_y} & \frac{\rho V^{2/3}k_q}{2I_y} \end{bmatrix}}^{A_m} \begin{bmatrix} z \\ w_v \\ \theta \\ q_v \end{bmatrix}(t) \\ &+ \overbrace{\begin{bmatrix} 0 & 0 \\ \frac{1}{m_z} & \frac{1}{m_z} \\ 0 & 0 \\ \frac{-x_{Tvf}}{I_y} & \frac{-x_{Tvr}}{I_y} \end{bmatrix}}^{B_m} \begin{bmatrix} T_{vf} \\ T_{vr} \end{bmatrix}(t) \end{aligned} \tag{32}$$

$$y(t) = \overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}^{C_m} \begin{bmatrix} z \\ w_v \\ \theta \\ q_v \end{bmatrix}(t) \tag{33}$$

where k_z and k_q are the linear damping coefficients. The values of the parameters used in this state-space model are given in Table I.

TABLE I
MODEL PARAMETERS

Parameter	Value
ρ	$1000kg/m^3$
V	$0.08m^3$
m_z	$167.5kg$
I_y	$70kg.m^2$
W	$540N$
B	$545N$
k_z	27.1
k_q	2.3
x_{Tvf}	$0.55m$
x_{Tvr}	-0.49
V	0.08
C_{Dq}	3.0
C_{Dw}	3.28

C. Simulation

In this work the controller that is evaluated will have fixed values of N_p, N_c and r_w of 80, 8 and 4 respectively. All the simulations have the same set-points for depth and pitch of $1.0m$ and 0.0^o respectively. A simulation of the controller performance is given in Fig. 3 where the non-linear model provides full state feedback without measurement noise.

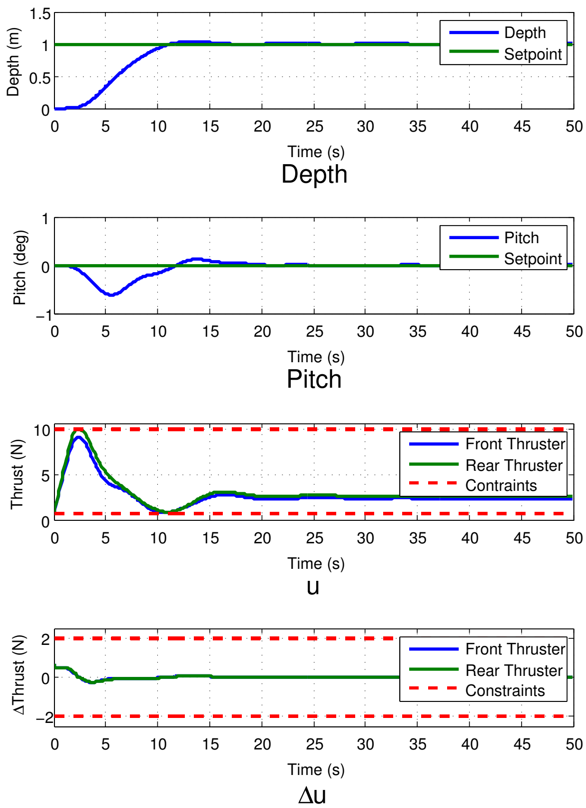

Fig. 3. Simulation results showing the performance of the depth and pitch controller without measurement noise. Note the quick and smooth convergence of both depth and pitch to the relevent set-points, and that the Δu signals are approximately zero once the set-points have been achieved.

The top two sub-plots are the depth and pitch values against time, with their corresponding set-points. As can be seen the depth converges quickly and smoothly to the set-point with minimal overshoot, while the pitch does experience some disturbance during the diving phase but also converges to the set-point of zero.

The bottom two sub-plots are of the system input u and the Δu value at each sample, note that the u values are the integral of the Δu values. As can be seen the controller dives the AUV with high values of thrust then reduces the thrust as the system approaches the set-points.

IV. Measurement Noise

By definition closed loop control systems require some form of feedback acquired by; measurement using sensors, estimation using observers, or a combination of both. A measured signal can be represented as:

$$signal = x + e_s + e_r, \tag{34}$$

where x is the true value of the variable that is being measured, e_s is systematic error and e_r is random error (measurement noise).

The systematic error is a variable offset from the x value due to inherent inaccuracies with the sensor or observer. Such errors can be caused by poor sensor calibration or model inaccuracies within an observer. Random error, also known as measurement noise, can be due to many factors, but for sensors it is often caused by electromagnetic interference. As time tends towards infinity the mean of the random error should tend towards zero. In this work, the effect of random error on the performance of the model predictive control algorithm will be investigated. The effect of systematic errors will not be investigated.

A. *Quantification*

On the Delphin2 AUV, feedback for depth and pitch is provided by a digital compass sensor with a pressure transducer. The derivatives of these values (q_v and w_v) are also required for full state feedback. For this section these derivatives will be approximated by taking the difference derivative between the most recent two samples.

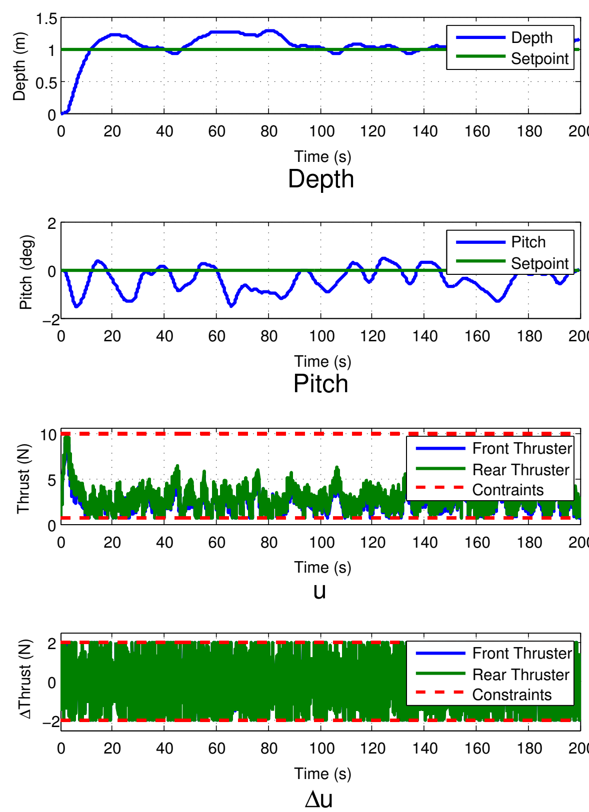

Fig. 4. Simulation results showing the performance of the depth and pitch controller with measurement noise (standard deviation of 0.001) added to the depth and pitch feedback signals. Note the random behaviour on both the depth and pitch signals, and the magnitude of the Δu values.

Measurement noise with a Gaussian distribution will be added to the depth and pitch signals from the non-linear model. The magnitude of this additional measurement noise will be quantified by its standard deviation (σ_n) and will vary from zero to 0.001 for both the depth and pitch signals.

B. Effect on Performance

The effect of measurement noise on controller performance will be quantified by calculating the standard deviations of the depth (σ_z), pitch (σ_θ) and Δu ($\sigma_{\Delta u}$) signals for the final 75% of each simulation. Five simulations are run for each σ_n value.

Fig. 4 presents simulation results, for the the same set-points as in Fig. 3, but with measurement noise (with a σ_n value of 0.001) added to the depth and pitch signals. Both the depth and pitch signals experience significant disturbances causing erratic, unacceptable performance. It is impossible to make sense of the Δu signal as it appears to be extremely noisy. The u signal follows a similar trend to that in Fig. 3 but is also noisy and erratic.

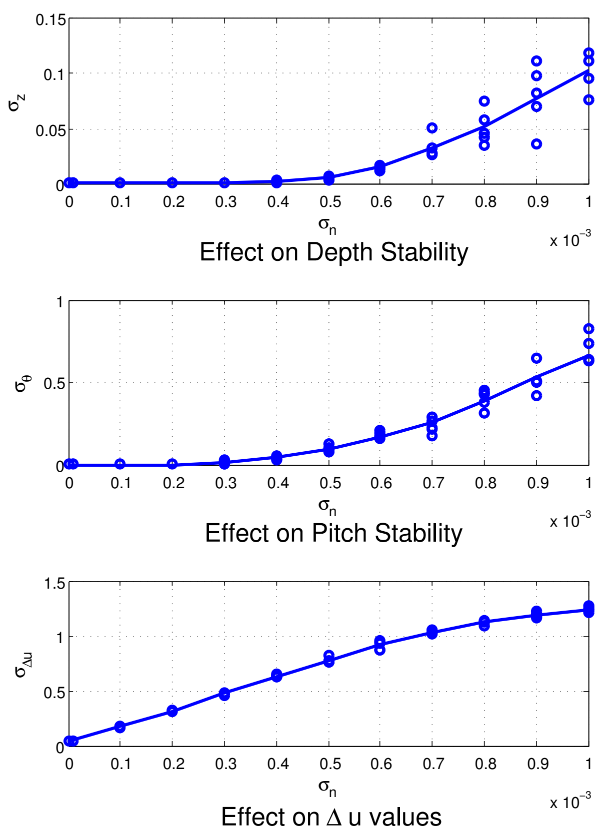

Fig. 5. The standard deviations of the depth, pitch and Δu signals against the standard deviation of the measurement noise.

Fig. 5 presents the results of σ_z, σ_θ and $\sigma_{\Delta u}$ values against the σ_n. Both the σ_z and σ_θ start to increase to unacceptable levels when σ_n is above approximately 0.4×10^{-3}. The values of $\sigma_{\Delta u}$ appears to be linearly proportional to σ_n for values from zero to 0.6×10^{-3}. Above this the constraints start to restrict the magnitude of each Δu value, without the constraints it would be expected that this linear trend would continue.

V. MPC Algorithm Improvements

Clearly this substantial degradation of controller performance, for a small magnitude of measurement noise, is unacceptable. The obvious solution is to filter the depth and pitch feedback signals. To do this without a time delay that could also be detrimental to system performance, and provide an accurate estimation of the derivatives, a polynomial type filter is proposed.

This filter stores a vector of the previous f samples and a vector of the time-stamps for each sample. A second-order polynomial in then fitted to these vectors using least-squares regression. The sample vectors take the form:

$$[s(k), s(k-1), ..., s(k-f+1)] \tag{35}$$

where $s(k)$ is the latest sample, $s(k-1)$ the previous sample etc. The time vector takes the form:

$$[0, (dt \times 1), (dt \times 2), ..., (dt \times (f-1))] \tag{36}$$

where dt is the sampling interval of the controller.

The length of the vectors for the depth and pitch samples are 8 and 5 respectively. The resulting polynomials for the depth and pitch signals are P_z and P_θ respectively. The derivatives of these polynomials are then calculated resulting in the polynomials $\dot{P}_z$ and $\dot{P}_\theta$.

One method would be to filter the data so as to reduce the magnitude of noise on each sample, and then implement the filtered signals as before. This could still provide noisy feedback to the controller due to the variations in the fitted polynomials. Instead the state-variable vector, used in the MPC cost function, is modified.

To make full use of the information obtained by fitting a polynomial to the sampled data, the polynomial will be evaluated at time equal to zero and one sample. The state-variable vector is now of the form:

$$x(k) = \begin{bmatrix} P_z(0) - P_z(dt) \\ -(\dot{P}_z(0) - \dot{P}_z(dt)) \\ P_\theta(0) - P_\theta(dt) \\ -(\dot{P}_\theta(0) - \dot{P}_\theta(dt)) \\ P_z(0) \\ P_\theta(0) \end{bmatrix} \tag{37}$$

A. Results

Using the new state-variable vector, the same simulations as in Fig. 5 are performed. Fig. 6 presents the data from these simulations using the new state-variable vector and the data from the original case. Note that in this figure the y-axis has a logarithmic scale to help visualise the variation in performance. All three plots show similar trends with significant reductions in the standard deviation of the depth, pitch and Δu signals. This implies that the system is more stable and the erratic behaviour, shown in Fig. 4, has not been eliminated but reduced to acceptable levels.

Interestingly the standard deviations on the depth and pitch signals are higher than the unfiltered case with σ_n below approximately 0.3×10^{-3}. This is most likely due to the filter

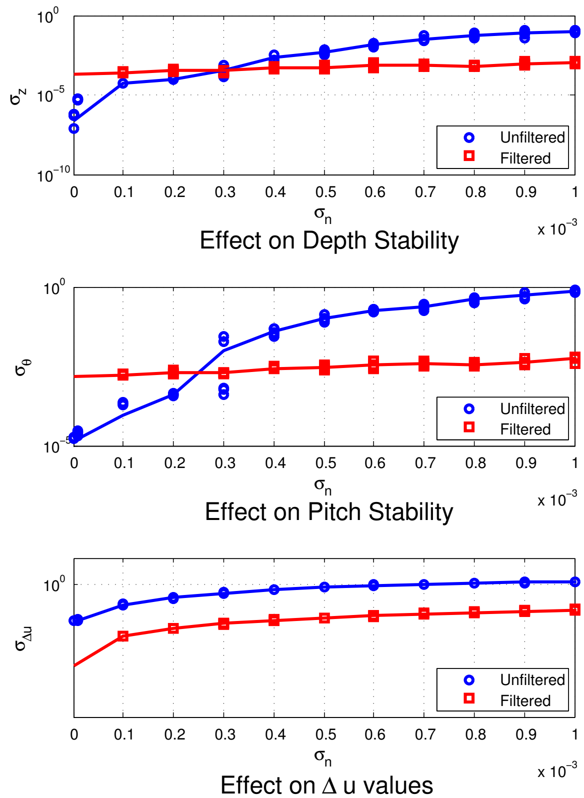

Fig. 6. The standard deviations of the depth, pitch and Δu signals against the standard deviation of the measurement noise with and without the filter.

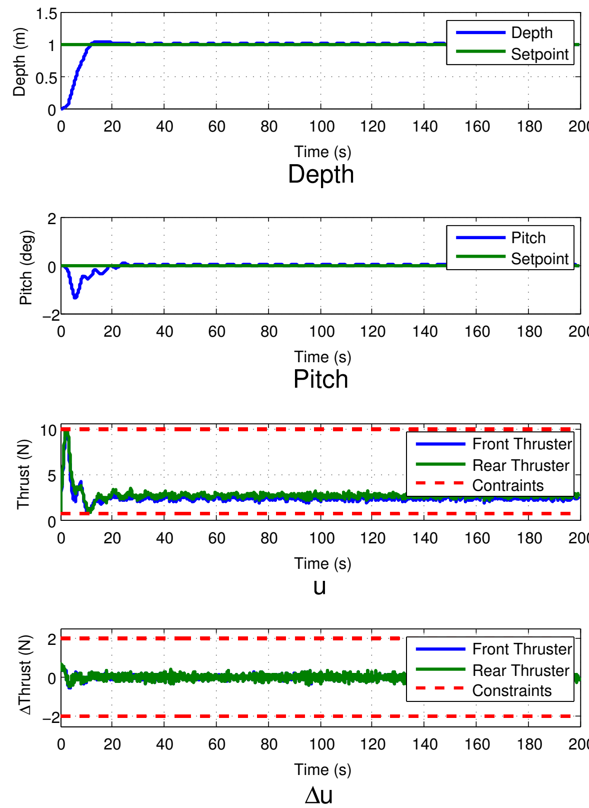

Fig. 7. Simulation results showing the performance of the depth and pitch controller with measurement noise (standard deviation of 0.001) added to both the depth and pitch feedback signals. Note in comparison with Fig. 4 the performance is much better, with reduced disturbances to the depth and pitch signals and lower Δu values.

performing poorly, an effort must be made to tune the length of the sample vectors and the order of the fitted polynomials so that they remove the influence of the measurement noise without filtering out the system dynamics.

Finally Fig. 7 presents the simulation results for a σ_n value of 0.001 on both the depth and pitch signals. In comparison to Fig. 3 the performance has been greatly improved and is not substantially worse than the simulation without measurement noise, shown in Fig. 3.

VI. Conclusions

In this work a depth and pitch controller has been developed, using the MPC algorithm, for a hover-capable AUV. It has been simulated using a non-linear model of the system and shown to provide excellent performance. However, when low levels of measurement noise are added to the feedback signals the performance quickly degrades to unacceptable levels.

The measurement noise has been shown to cause the system to experience significant disturbances to the depth and pitch signals. The magnitude of these disturbances has been quantified using the standard deviation of the depth, pitch and Δu values.

A polynomial type filter has been proposed and integrated within the MPC algorithm using a modified state-variable vector. The performance of the controller when using the filter is several orders of magnitude better than those without the filter. Clearly when using feedback with measurement noise the use of this filter will improve system performance when compared to the original algorithm. However, care must be taken when designing the filter so as not to influence the dynamics of the feedback.

Future work should include the development of a state observer that could directly estimate the state-variable vector from the augmented model using either the filtered or unfiltered depth and pitch signals. It could also be expected that the method proposed here could be combined with an observer to provide better overall system performance.

This work has shown the importance of including realistic levels of measurement noise when simulating novel control algorithms. In order for new control algorithms that are emerging from the theoretical control community to be accepted and applied to real systems requires that these algorithms go through scrupulous examination.

References

[1] Budiyono A. Model predictive control for autonomous underwater vehicle. *Indian Journal of Geo-Marine Sciences*, Vol. 40(2):pp. 191–199, 2011.

[2] Wasif Naeem Marine. Model predictive control of an autonomous underwater vehicle.

[3] S.Joe Qin and Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733 – 764, 2003.

[4] McEwen R. and Streitlien K. Modeling and control of a variable-length auv. Technical report, Monterey Bay Aquarium Research Institute, Bluefin Robotics, Corp., 2006.

[5] L.V. Steenson, A.B. Phillips, E. Rogers, M.E. Furlong, and S.R. Turnock. Preliminary results of a hover capable auv attempting transitional flight. In *Unmanned Untethered Submersible Technology (UUST)*, 2011a.

[6] L.V. Steenson, A.B. Phillips, E. Rogers, M.E. Furlong, and S.R. Turnock. Experimental verification of a depth controller using model predictive control with constraints onboard a thruster actuated auv. In *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCIV)*, 2012.

[7] Quan Truong, Liuping Wang, and P. Gawthrop. Intermnittent model predictive control of an autonomous underwater vehicle. pages 1 –6, dec. 2006.

[8] L. Wang. *Model Predictive Control System Design and Implementation Using MATLAB*. Springer, 2010.