

An Autopilot Based on a Local Control Network Design for an Unmanned Surface Vehicle

SK Sharma¹, W Naeem² and R Sutton¹

¹(School of Marine Science and Engineering, Plymouth University, Plymouth)

²(School of Electronics, Electrical Engineering and Computer Science,
Queen's University Belfast, Belfast)

(E-mail: sanjay.sharma@plymouth.ac.uk)

Over recent years, a number of marine autopilots designed using linear techniques have underperformed owing to their inability to cope with nonlinear vessel dynamics. To this end, a new design framework for the development of nonlinear autopilots is proposed. Local Control Networks (LCNs) can be used in the design of nonlinear control systems. In this paper, a LCN approach is taken in the design of a nonlinear autopilot for controlling the nonlinear yaw dynamics of an unmanned surface vehicle known as *Springer*. It is considered the approach is the first of its kind to be used in marine control systems design. Simulation results are presented and the performance of the nonlinear autopilot is compared with that of an existing *Springer* Linear Quadratic Gaussian (LQG) autopilot using standard system performance criteria. From the results it can be concluded the LCN autopilot out-performed that based on LQG techniques in terms of the selected criteria. Also it provided more energy saving control strategies and would thereby increase operational duration times for the vehicle during real-time missions.

KEY WORDS

1. Unmanned Surface Vehicle.
2. Autopilot Design.
3. Linear Quadratic Gaussian Control
4. Local Control Network.

1. INTRODUCTION. During the summer of 1588, Admiral Charles Howard and Vice Admiral Francis Drake arranged for eight fire ships to be sent against the Armada which caused the majority of the Spanish ships to break formation and leave their safe anchorage in Calais for the open sea. Soon after this incident the Battle of Gravelines ensued. On a later occasion in 1718, a Royal Navy blockade was set up by Captain Woodes Rogers in Nassau Harbour with a view to capturing the pirate Captain Charles Vane. Unfortunately for Rogers, Vane directed a blazing crewless brigantine at the blockade. As a consequence, Rogers' small squadron of ships was forced to take evasive action that allowed Vane to make good his escape to continue his life of piracy until his eventual capture in 1720. Subsequently, he was hanged in Jamaica on 29 March 1721. It can be argued that these two incidents illustrate early examples of Unmanned Surface Vehicles (USVs) being deployed in naval warfare.

Nowadays USV technology is becoming widespread in the commercial, naval and scientific sectors. Indeed they are now used for mine counter-measures (Yan et al., 2010), surveying (Majohr et al., 2000) and environmental data gathering (Caccia, 2007), to name just three.

In recent years the *Springer* USV has been designed, built and continues to be developed. *Springer* is intended to be a cost effective and environmentally friendly USV; it was designed primarily for undertaking pollutant tracking, environmental and hydrographical surveys in rivers, reservoirs, inland waterways and coastal waters, particularly where shallow waters prevail. An equally important secondary role was also envisaged for *Springer* as a test bed platform for other academic and scientific institutions involved in environmental data gathering, sensor and instrumentation technology, control systems engineering and power systems based on alternative energy sources.

For the vehicle to be capable of undertaking the kinds of mission that are contemplated, *Springer* requires robust, reliable, accurate and adaptable Navigation, Guidance and Control (NGC) systems which allow seamless switching between automatic and manual control modes. Such properties in NGC systems are necessary for the changes in the dynamic behaviour of the vehicle that may occur owing to the deployment of different payloads, mission requirements and varying environmental conditions. Of course, an important integral component in a NGC system is the autopilot (control) sub-system.

The development of practical ship autopilot systems can be traced back to the Sperry Gyroscope Company (Sperry Gyroscope Company, 2011). Sperry constructed the first automatic ship steering mechanism which he called the *automatic pilot* or *gyropilot* (Sperry, 1922). Sperry's gyropilot was known as '*Metal Mick*' because in its operation it appeared to replicate the actions of an experienced helmsman. In addition, the work of (Minorski, 1922) is also regarded as making key contributions to the development of automatic ship steering systems. Minorski's main contribution was the theoretical analysis of automatic steering and the specification of the three term or Proportional-Integral-Derivative (PID) controller for automatic ship steering. As in the case of Sperry's work, his PID controller designs were predicated on visual observation of the way the experienced helmsman would steer a ship. From those early pioneering days ship autopilots based on PID controllers or their variants became the norm for a number of years. While such autopilots performed satisfactorily within given specifications, their overall operational effectiveness was limited. In an attempt to overcome the shortcomings of these devices, more sophisticated autopilots have been and are still being proposed. One of the more modern and favoured approaches to autopilot design has been based on Linear Quadratic Gaussian (LQG) techniques (Hozhuter and Schultze, 1996; Fossen, 2000; Tran et al., 2004; Moreira and Soares, 2005; Naeem et al., 2008; Fossen, 2011). However, it should be noted that, in general, basic PID and LQG autopilots can only reliably be applied to linear ship dynamics.

The dynamic characteristics of marine vessels are invariably nonlinear and the *Springer* USV is no exception. This being so it would seem appropriate to control the vehicle with a nonlinear autopilot. Local Control Networks (LCNs) can be used in nonlinear control system design. Furthermore, LCNs are transparent, and simple to design compared to many other more complex nonlinear controllers. Thus LCNs were deemed worthy of investigation for application as autopilots. Hence the intention of this paper is to report a novel autopilot system based on a LCN design for the USV



Figure 1. The *Springer* Unmanned Surface Vehicle (USV).

and to benchmark its performance against an existing *Springer* autopilot design developed using LQG control techniques. It is believed that this is the first application of a LCN in the role of a marine autopilot and, therefore, the paper offers a new framework for marine control systems design. Details of the navigation and line-of-sight guidance sub-systems for the vehicle can be found in (Naeem et al., 2008).

With regards to the structure and content of the paper, on completion of this introductory material, Section 2 reports details of the *Springer* vehicle hardware, and of linear and nonlinear models of its yaw dynamics. Section 3 describes the two autopilot designs, and in Section 4 simulation results and a discussion are presented. Finally concluding remarks are given in Section 5.

2. THE SPRINGER UNMANNED SURFACE VEHICLE. The *Springer* USV was designed as a medium waterplane twin hull vessel which is versatile in terms of mission profile and payload. It is approximately 4 m long and 2.3 m wide with a displacement of 0.6 tonnes. Each hull is divided into three watertight compartments. The NGC system is carried in watertight Peli cases and secured in a bay area between the crossbeams. This facilitates the quick substitution of systems on shore or at sea. The batteries which are used to provide the power for the propulsion system and onboard electronics are carried within the hulls, accessed by a watertight hatch. In order to prevent any catastrophe resulting from a water leakage, leak sensors are utilized within the motor housing. If a breach is detected the onboard computer immediately issues a warning to the user and/or takes appropriate action in order to minimize damage to the onboard electronics (Naeem et al., 2008). A mast has also been installed to carry the Global Positioning System (GPS) and wireless antennas. The wireless antenna is used as a means of communication between the vessel and its user and is intended to be utilized for remote monitoring purpose, intervention in the case of erratic behaviour and to alter the mission parameters. The *Springer* is shown in Figure 1 and the arrangement inside its hulls is depicted in Figure 2. The Peli case layouts may be found in (Naeem et al., 2008).

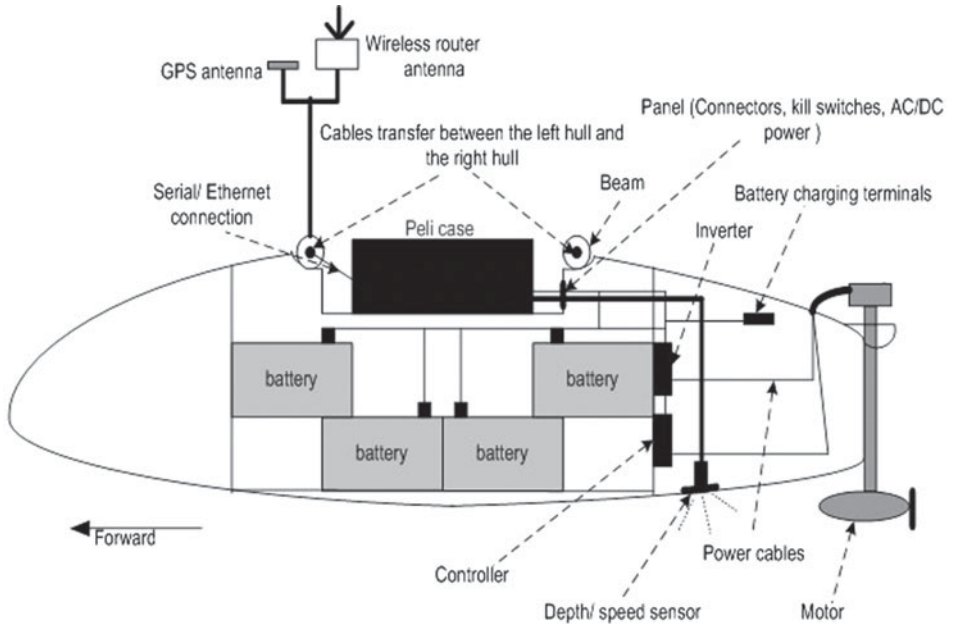


Figure 2. Side view of the *Springer* Unmanned Surface Vehicle (USV).

The *Springer* propulsion system consists of two propellers powered by a set of 24 volt 74 lbs Minn Kota Riptide transom mounted saltwater trolling motors. As will be seen in Subsection 2.1, steering of the vessel is based on differential propeller revolution rates.

2.1. *Vehicle Steerage.* The vehicle has a differential steering mechanism and thus requires two inputs to adjust its course. This was simply modelled as a two input, single output system in the form depicted in Figure 3, where n_1 and n_2 are the two propeller thrusts in Revolutions Per Minute (RPM).

Clearly, straight line manoeuvres require both the thrusters to run at the same speed; the differential thrust is zero in this case. By letting n_c and n_d represent the common mode and differential mode thruster velocities defined, then n_c and n_d are defined by Equations (1) and (2):

$$n_c = \frac{n_1 + n_2}{2} \quad (1)$$

$$n_d = \frac{n_1 - n_2}{2} \quad (2)$$

In order to maintain the velocity of the vessel, n_c must remain constant at all times. The differential mode input, however, oscillates about zero depending on the direction of the manoeuvre. While the actual steerage system operates using RPM, for reasons of convenience and ease of comparisons, results are shown in Revolutions Per Second (RPS) where appropriate.

2.2. *Linear Vehicle Yaw Dynamics.* As a linear yaw dynamic model of the vehicle is required in the design of the LQG autopilot, System Identification (SI) (Ljung, 1999) was performed on three different sets of experimental data, collected



Figure 3. Block diagram representation of a two-input Unmanned Surface Vehicle (USV).

with a sampling interval of 1 s for training, validating and testing the model. In this case, only n_d has been manipulated and therefore acts as the sole input to the system. This alters both n_1 and n_2 whereas n_c is maintained to conserve the operating regime.

The SI of the vehicle dynamics was undertaken using a proprietary MATLAB™ toolbox. Initial results showed that dynamic models based on an AutoRegressive with eXogenous (ARX) terms approach failed to achieve a good fit. Thus an iterative prediction-error minimization method using the SI toolbox was employed to estimate the parameters of an AutoRegressive Moving-Average with eXogenous (ARMAX) variable terms type fourth-order linear state space model, which has more flexibility in the handling of disturbance modelling than the ARX model. The dynamic model of the *Springer* vehicle was obtained in State Space (SS) form as shown in Equation (3) which was subsequently used to design the LQG autopilot.

$$\begin{aligned} \mathbf{x}(k + 1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) + \mathbf{K}e(k) \\ y(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}u(k) \end{aligned} \tag{3}$$

where:

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 1.0035 & -0.0012 & -0.0056 & -0.0057 \\ 0.0247 & 0.9752 & -0.1372 & -0.1241 \\ 0.0349 & -0.0103 & -0.4166 & 0.7517 \\ 0.0325 & -0.0987 & -0.0604 & 0.0073 \end{bmatrix}, & \mathbf{B} &= \begin{bmatrix} 0.0000025 \\ -0.0000654 \\ 0.0004235 \\ -0.0003025 \end{bmatrix}, \\ \mathbf{K} &= \begin{bmatrix} 0.0275 \\ -1.2455 \\ 1.8949 \\ -1.6802 \end{bmatrix}, & \mathbf{C} &= [42.3573 \quad -0.0940 \quad -0.0288 \quad 0.0354], & \mathbf{D} &= [0] \end{aligned} \tag{4}$$

Figures 4a, 4b and 4c show the performance of the SS model on training, validation and test data sets which produces mean-squared errors of 0.00057465 rad², 0.0075 rad² and 0.00042696 rad² respectively.

Although the SS model fits well (Figure 4c) on the series-parallel configuration (one-step-ahead prediction), its performance deteriorates for the parallel response (iterated prediction) and indicates the poor predictive capability of the model. Figure 5 shows the series-parallel and parallel configuration of the model and the Figure 6 illustrates the predictive capability of the SS model on a parallel model output of the test dataset producing a mean-squared error of 0.307 rad². The poor predictive capability may be because of the highly nonlinear dynamics of the *Springer* vehicle were not captured in

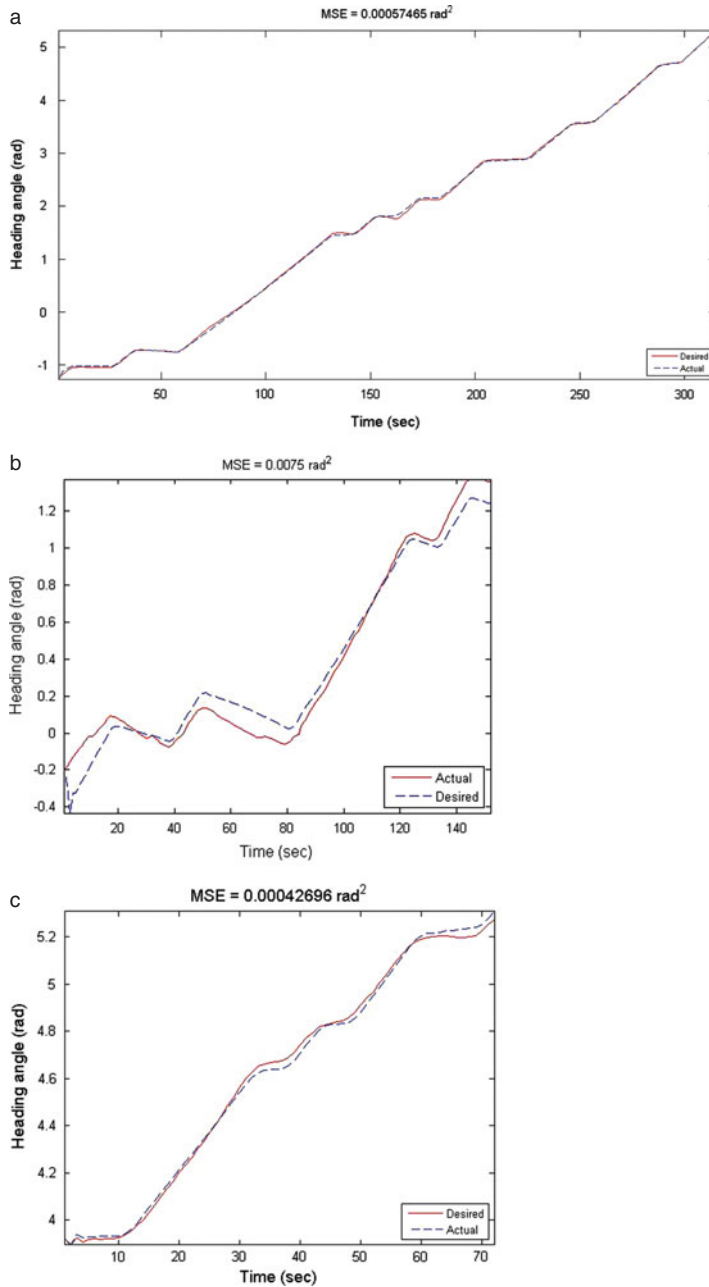


Figure 4. (a) Performance of the State Space (SS) model: training. (b) Performance of the State Space (SS) model: validation. (c) Performance of the State Space (SS) model: testing.

the linear state-space model. Nonlinear modelling of the *Springer* in a parallel configuration is thus required to obtain the true characteristics of its dynamic. Subsection 2.3 studies the nonlinear modelling of the *Springer* using neural networks and compares its predictive capability with the SS model.

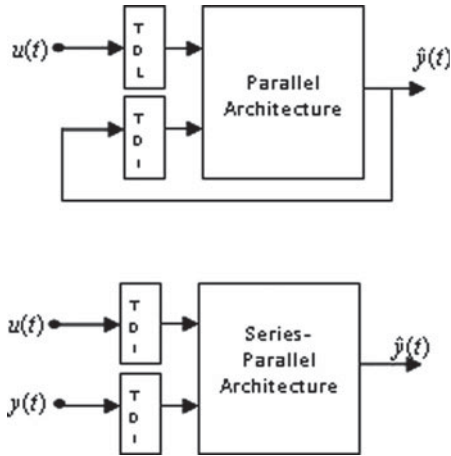


Figure 5. Parallel and series-parallel configurations of the model.

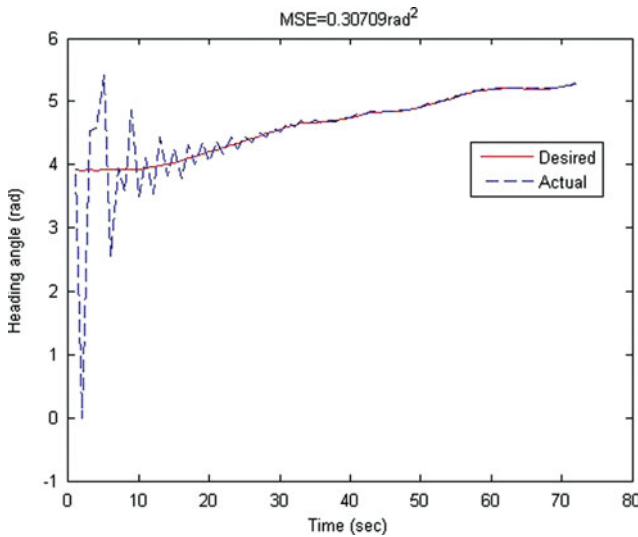


Figure 6. Predictive capability of a parallel model.

2.3. *Nonlinear Vehicle Yaw Dynamics.* As shown above in the SS model case, the linear model across an operating point does not contain the true characteristics of the highly nonlinear *Springer* dynamics, thus the requirement for a nonlinear model. A Multi-Layer Perception (MLP) type Neural Network (NN) model was developed here using the previous dataset. A Genetic Algorithm (GA) (Sivanandam and Deepa, 2008) with a population of 20 chromosomes and a crossover probability of $p_c=0.65$ and mutation probability of $p_m=0.03$ was used to obtain the best NN architecture with optimized weights and biases (Sharma et al., 2002; 2005). The GA was run for 3000 generations and after a period of trial and error, a parallel architecture network

with ten inputs was derived:

$$\hat{y}(t) = f_{NN} \{u(t-1), u(t-2), \hat{y}(t-1), \hat{y}(t-2), \hat{y}(t-3), \\ \times \hat{y}(t-4), e(t-1), e(t-2), e(t-3), e(t-4)\} \quad (5)$$

where:

$$e(t) = y(t) - \hat{y}(t)$$

and three hidden nodes were selected for modelling.

Figures 7a, 7b and 7c show the parallel model performance of the NN model on the training, validation and test data set which produced mean-squared errors of 0.0003746 rad², 0.003904 rad² and 0.0007106 rad² respectively.

The results show that NN achieved better predictive capability compared to the state-space model considered in the Subsection 2.2.

This NN model was used to replicate the nonlinear yaw dynamics of the *Springer* USV and to train a LCN autopilot to follow set point trajectories, with its performance being judged against that of a LQG autopilot.

3. AUTOPILOT DESIGNS. In this section of the paper, both autopilot designs are presented.

3.1. Linear Quadratic Gaussian (LQG) Autopilot Design. In this autopilot design, a LQG controller was selected which consists of a linear combination of a Linear Quadratic state feedback Regulator (LQR) and a Kalman filter. The LQG controller is inherently multi-variable, therefore modification to a multi-input, multi-output model is rather straightforward. To construct the autopilot, an LQR problem is solved which assumes that all states are available for feedback. However, this is not always true because either there is no available sensor to measure that state or the measurement is very noisy. A Kalman filter can be designed to estimate the unmeasured states. The LQR and Kalman filter were developed independently and then combined to form an LQG controller, a method known as the *separation principle* (Burl, 1999). A block diagram of the autopilot is depicted in Figure 8 showing the individual components of the LQG. In this figure, r is the reference input which is transformed to a corresponding reference state vector using the transformation matrix N_x . The block N_u compensates for any steady state errors present in the output of the closed loop system.

The LQG controller requires an SS model of the system in the form specified in Equation 3. The parameters A, B, C, and D are also defined in Equation 4 for the *Springer* vehicle. A unique closed form solution of the LQG control law is defined as:

$$u(k) = \mathbf{K}_{LQR}(x_r(k) - \hat{x}(k)) \quad (6)$$

The Kalman filter equations and a derivation of the N_x and N_u blocks respectively are at Appendices A and B.

3.2. Local Control Network Autopilot Design. A LCN provides a ‘divide-and-conquer’ approach to the design of a global controller for complex nonlinear systems. It consists of several linear Local Model Controllers (LMCs) spread across the operating regions. The operating space is decomposed into a number of regimes and the required global controller is then formed by interpolating between simpler LMCs that are locally valid (Johansen and Foss, 1995).

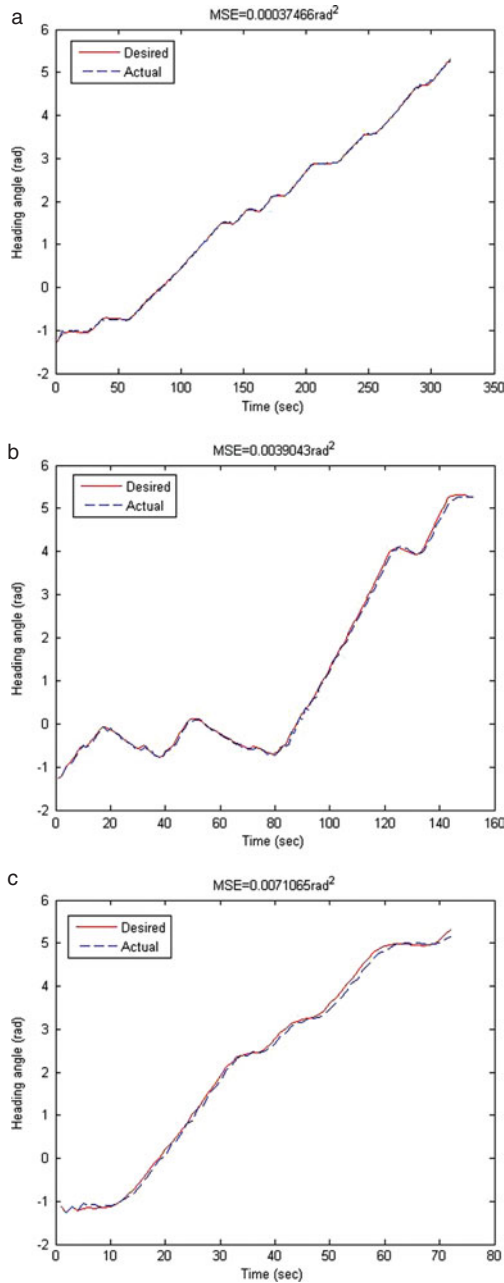


Figure 7. (a) Performance of the Neural Network (NN) model: training. (b) Performance of the Neural Network (NN) model: validation. (c) Performance of the Neural Network (NN) model: testing.

Knowledge of these local operating regimes is therefore a key requirement for building such controllers. It is known that a large class of nonlinear systems can be controlled in this way, including most batch processes and many control system

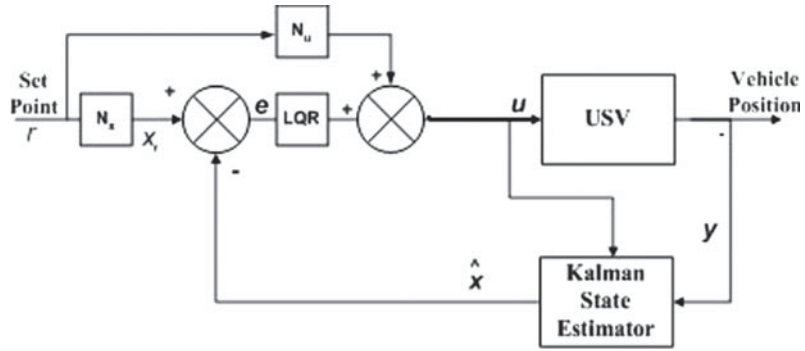


Figure 8. Block diagram representation of a Linear Quadratic Gaussian (LQG) autopilot design.

applications (Rippin, 1989). Apart from normal operation, the control system may also have to function correctly during startup and shutdown cycles and to operate during maintenance and fault conditions, all of which constitute different operating regimes. In most applications, the design of a suitable LCN is done on the basis of *a priori* plant knowledge (Brown et al., 1997; Townsend et al., 1998). The interpolation is done such that the LMC which is most valid at an operating point will be given the greatest weight, neighbouring ones will be weighted less and those for distant operating regimes will not contribute anything to the global control at that point.

Each LMC is thus associated with a weighting function that provides smooth interpolation and also indicates the relative validity of it at a given operating point. *Springer* had already been tested within its linear operating region using SS models and LQG techniques. However, until this study, no work has been undertaken to test the USV while operating in the nonlinear region of its dynamic system response. For practical operations, the identification of distinct operating regimes are important, which can only be obtained with *a priori* plant knowledge. The identification of local operating regimes and simultaneous design of a global LCN is difficult in the absence of *a priori* knowledge about the unknown plant. This paper finds such operating regimes and constructs a suitable LCN based on GAs for *Springer*.

3.2.1. *Local Control Networks.* The general discrete LCN representation is shown in Figure 9. Here, the same inputs, x , are fed to all the LMCs and the outputs are weighted according to some scheduling variable or variables, ϕ . The LCN output \hat{y} is given by the weighted sum:

$$\hat{y} = \sum_{i=1}^N \rho_i(\phi) f_i(x) \tag{7}$$

where:

- $\rho_i(\phi)$ is the validity or interpolation function associated with the i^{th} LMC, $f_i(x)$.
- N is the total number of LMCs.

The validity functions $\rho_i(\phi)$ are normalized so that the total contribution from all the LMCs is 100%. The most widely used $\rho_i(\phi)$ in the literature are normalized Gaussian

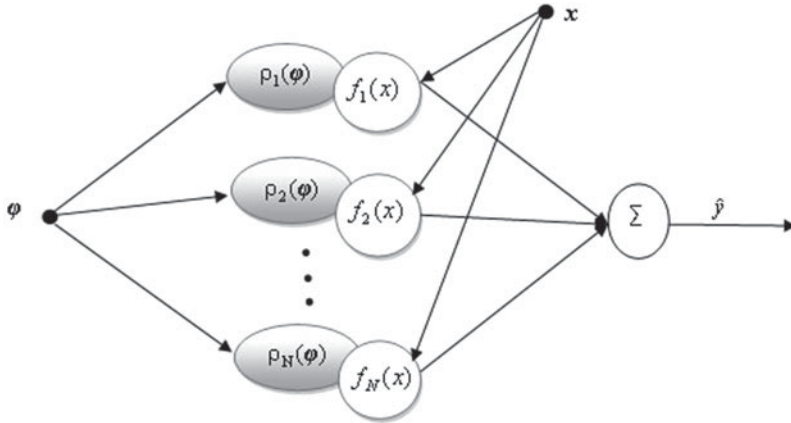


Figure 9. General architecture of a Local Control Network (LCN).

functions represented as:

$$\rho_i(\varphi) = \frac{\exp(-\|\varphi - s_i\|^2/2\sigma_i^2)}{\sum_{j=1}^N \exp(-\|\varphi - s_j\|^2/2\sigma_j^2)} \tag{8}$$

The $f_i(x)$ are linear controllers. In this paper, a discrete-type PID controller is considered for the LCN construction as it requires a smaller number of parameters compared to that of ARMAX type linear controllers; it also avoids controller initialization problems while reducing the GA search space.

A continuous-time PID control law is described by:

$$u(t) = k_p e(t) + k_d \dot{e}(t) + k_I \int e dt \tag{9}$$

where:

k_p, k_d and k_I are the proportional, differential and the integral gains.

u is the control action.

e is the error.

The equivalent PID controller in discrete form is

$$u(k) = u(k - 1) + k_p[e(k) - e(k - 1)] + k_d[e(k) - 2e(k - 1) + e(k - 2)] + T_s K_I e(k) \tag{10}$$

where:

k is the sample number.

T_s is the sampling interval.

For Equations (8) and (10) the unknown parameters in each regime are the validity function centres, s_i , standard deviations σ_i and the PID control parameters k_p, k_d and k_I . The identification of local operating regimes and simultaneous design of a global LCN is difficult in the absence of *a priori* knowledge about the unknown plant.

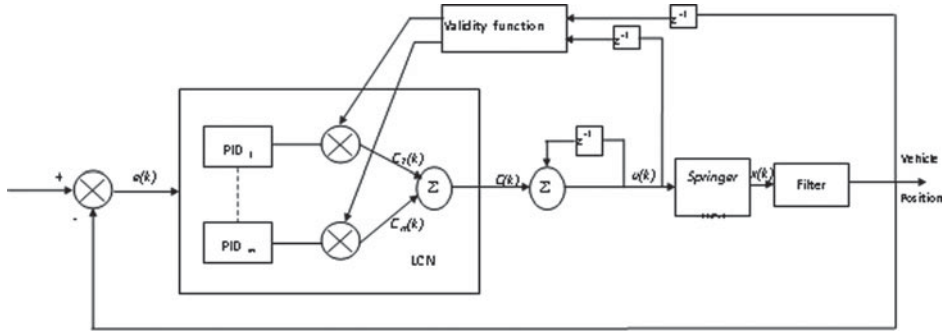


Figure 10. PID-type local control network acting on a Springer USV.

Figure 10 shows the design of a LCN involving PID-type LMCs. Here the LCN consists of m PID-type local model controllers. The output of the i^{th} PID-type LMC at sample k is $c_i(k)$ and the overall LCN output is defined as $c(k) = \sum_{i=1}^m c_i(k)$. The control action applied to the Springer USV at sample k is given by $u(k) = c(k) + u(k - 1)$. The same error $e(k) = r(k) - y(k)$, is applied to all LMCs in the network. The scheduling variable for the validity function, $\rho_i(\varphi)$, were chosen as $\varphi = [y(k - 1), u(k - 1)]$, where $y(k)$ is the heading angle output and $r(k)$ is the reference setpoint. A GA is then used to construct a LCN for the USV. The GA then simultaneously searches for the optimal number of LMCs (from a given maximum number), the parameters of these LMCs and the parameters of the validity functions and filter. Apart from reducing the tracking error and total controller effort, the fitness function of the GA is also incorporated to promote transparency by encouraging all valid LMCs to be mutually orthogonal, such that each local controller acts independently of the rest at its operating point. In this application a steady-state GA (Michalewicz, 1996) with crossover probability of $p_c = 0.65$ and mutation probability of $p_m = 0.03$ was applied to a population of 20 chromosomes.

The filter used in Figure 10 is of second order with the input/output relation described in Equation (11):

$$a_1y(k) = b_1x(k) + b_2x(k - 1) + b_3x(k - 2) - a_2y(k - 1) - a_3y(k - 2) \quad (11)$$

4. SIMULATION RESULTS AND DISCUSSION. The GA was run for 2000 generations and, to allow for the stochastic nature of genetic learning, the training process was repeated five times. The number of LMCs from the best chromosome was selected as the optimum, along with the centres and covariances of the associated validity function and the parameters for the filter. Figure 11a shows the closed-loop response from the LCN in response to a step change demand in the vehicle heading and it also reveals that a good tracking performance is achieved. Whereas Figure 11b indicates that a smooth LCN controller effort was expended in order to achieve such satisfactory results.

The GA selected two LMCs; Figures 12a and 12b illustrate the respective interpolation regions for the two LMCs. Figure 13 contains a plan view with the training data superimposed showing the operating region for Springer. It can be seen that the GA has selected the local controllers that are occupying 100% and are thus transparent somewhere in the operating regions. The shaded portion in the plan view

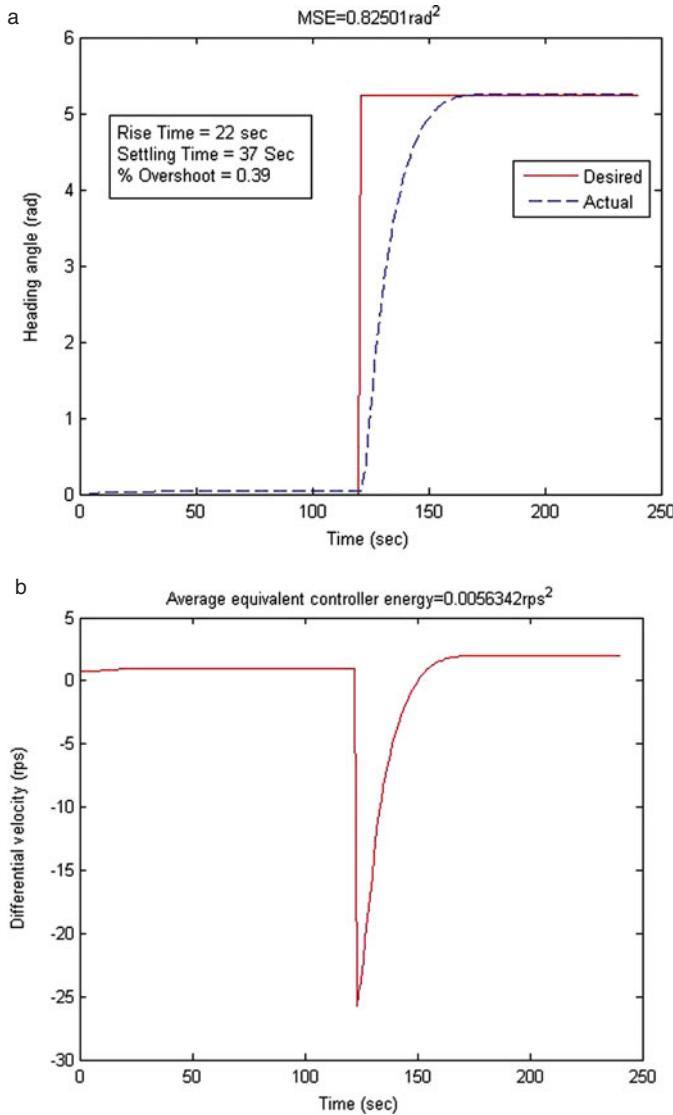


Figure 11. (a) Local Control Network (LCN) response to a step change in vehicle heading: heading response. (b) Local Control Network (LCN) response to a step change in vehicle heading: controller effort.

show the areas where each controller will be acting independently in the operating regions.

The two PID-type LMCs defining the LCN were:

$$C_1(k) = -1.9366[e(k) - e(k-1)] - 0.7826 \times [e(k) - 2e(k-1) + e(k-2)] + 0.1201e(k) \tag{12}$$

$$C_2(k) = -1.6429[e(k) - e(k-1)] + 0.2200 \times [e(k) - 2e(k-1) + e(k-2)] + 0.0429e(k) \tag{13}$$

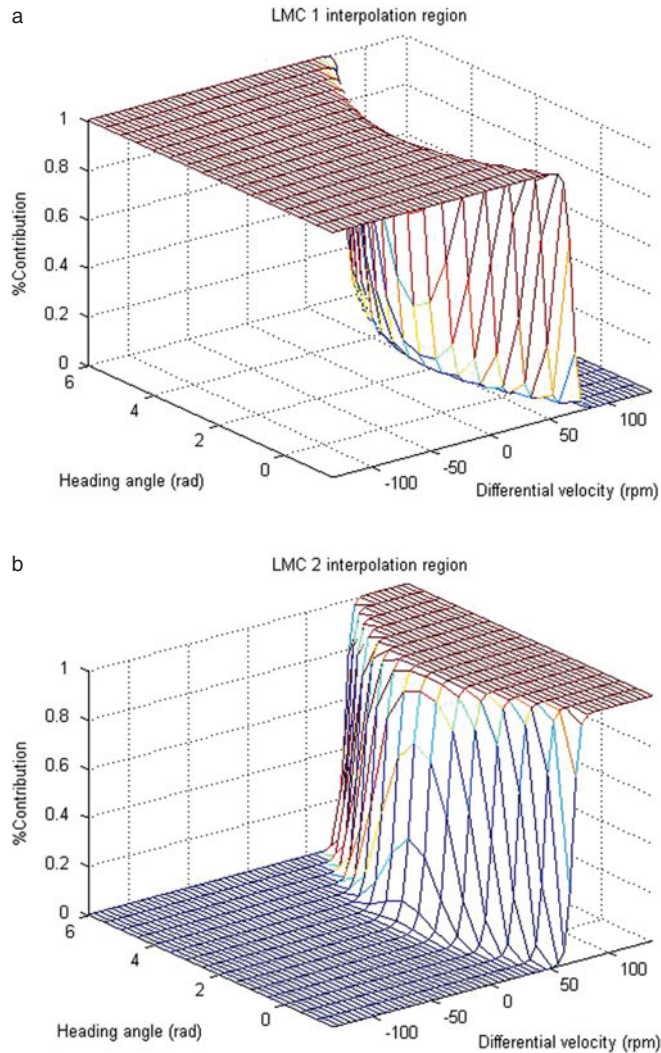


Figure 12. (a) Interpolation regions for two of the Local Model Controllers (LMCs): LMC 1 interpolation region. (b) Interpolation regions for two of the Local Model Controllers (LMCs): LMC 2 interpolation region.

where $e(k) = r(k) - y(k)$ is again the error between the reference and controller trajectory.

The centres and covariances for the Gaussian interpolation functions were ρ_1 : $(-14.0660, 2.0260)$ and $(14.8186, 0.6666)$ and ρ_2 : $(82.8059, 1.9678)$ and $(20.1629, 0.4573)$. The coefficients for the filter were: $a_1 = 1.0$, $a_2 = -0.9845$, $a_3 = 0.0576$, $b_1 = 0.0500$, $b_2 = -0.0063$ and $b_3 = 0.0318$.

The robustness of the controller was next tested. Results in (Brown et al., 1997; Townsend et al., 1998; Sharma et al., 2002) indicate that while the stability of a local controller acting on a local plant model is easily proved, the global stability of the overall closed-loop system is more problematic. One way to overcome this

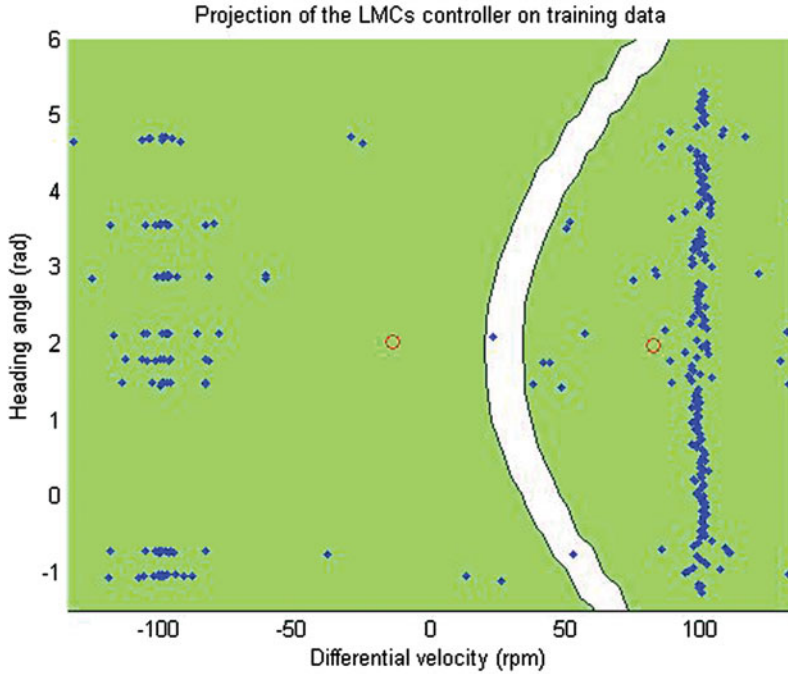


Figure 13. Plan view of validity regions with training data.

and to test the robustness of the controller is to demonstrate that it is stable throughout the operating space of a vehicle manoeuvre. To this end, a random sequence of course-changing manoeuvres was employed which covered the entire operating region as shown in Figure 14. Figure 14 also reveals that the LCN generated a stable closed-loop response and smoothly followed the reference trajectory.

The NN model of the *Springer* was used directly to design a LCN in the absence of much *a priori* knowledge. These results show the capacity of this approach to designing a LCN based autopilot for *Springer*. In addition, they clearly illustrate the autopilot’s ability to cope successfully when operating in both the linear and nonlinear realms of the vehicle’s dynamic system response.

In order to make quantitative comparisons between the two autopilot designs, the standard system performance criteria of Rise Time (T_R), Settling Time (T_S) and Percentage Overshoot ($\%M_P$) were employed (Ogata, 2002). Here T_R is taken as the time required for the system response to rise from 10% to 90% of its final value. It is used to denote the speed of response of a system. The T_S is the time required for the system response to reach and stay within a specified tolerance band of the final value which in this case is taken as 2%. T_S is the minimum time in which the transient phase of the system response is assumed to have decayed away, therefore, indicating the time at which the system may function at the new operating point. Whereas $\%M_P$ is the percentage maximum amount a system overshoots its final value and is used to signify the oscillatory nature and relative stability of the system.

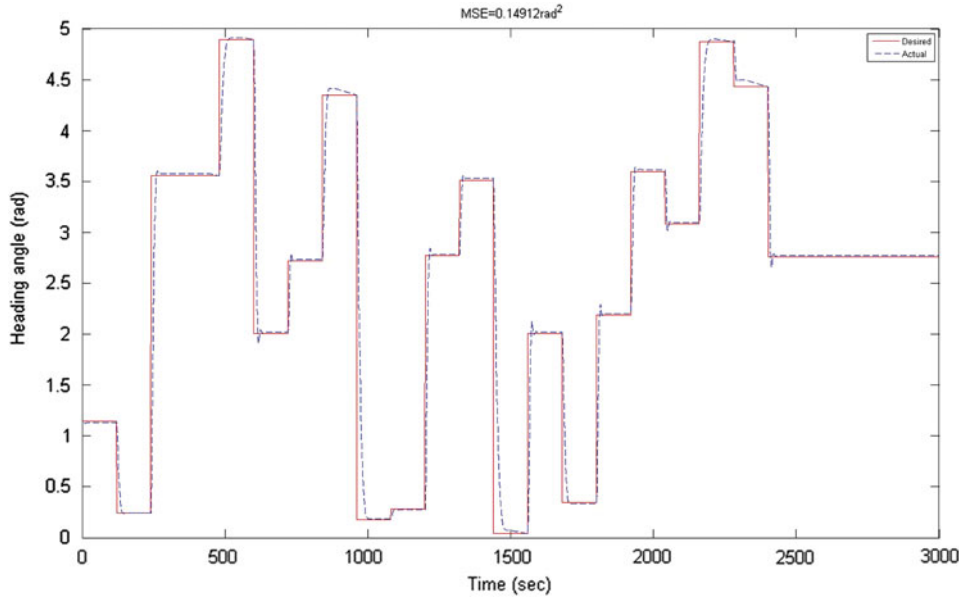


Figure 14. Vehicle response to a random sequence of course-changing demands in heading using the Local Control Network (LCN).

Additionally, as a measure of accuracy and autopilot control activity, the Mean-Square Error of the Yaw Error ($MSE(\psi_\epsilon)$) and the Average Equivalent Controller Energy ($ACE(E_u)$) were used. These may be considered in their discrete forms as:

$$MSE(\psi_\epsilon) = \frac{1}{N} \sum_{k=1}^N [y(k) - \hat{y}(k)]^2 \tag{14}$$

and:

$$ACE(E_u) = \frac{1}{N} \sum_{k=1}^N [u(k)/60]^2 \tag{15}$$

where:

- $y(k)$ is the desired output at k^{th} instant in rad.
- $\hat{y}(k)$ is the actual output at k^{th} instant in rad
- $u(k)$ is the controller effort at k^{th} instant in RPM.
- N is the total number of samples.

As was to be expected, the LQG autopilot was also subjected to the same series of course-changing demands displayed in Figures 11a and 14, the results of which can be found in Figures 15(a), and 16 respectively. The response of the LQG appears to be quite sluggish with a large T_S of over 274 s. Figure 15(b) depicts the controller output which settles to approximately zero as the vessel reaches steady state.

The step response in Figure 11 shows that the results $T_R=22$ s, $T_S=37$ s and $\%M_P=0.39\%$ were obtained using the LCN whereas the LQG produced $T_R=65$ s,

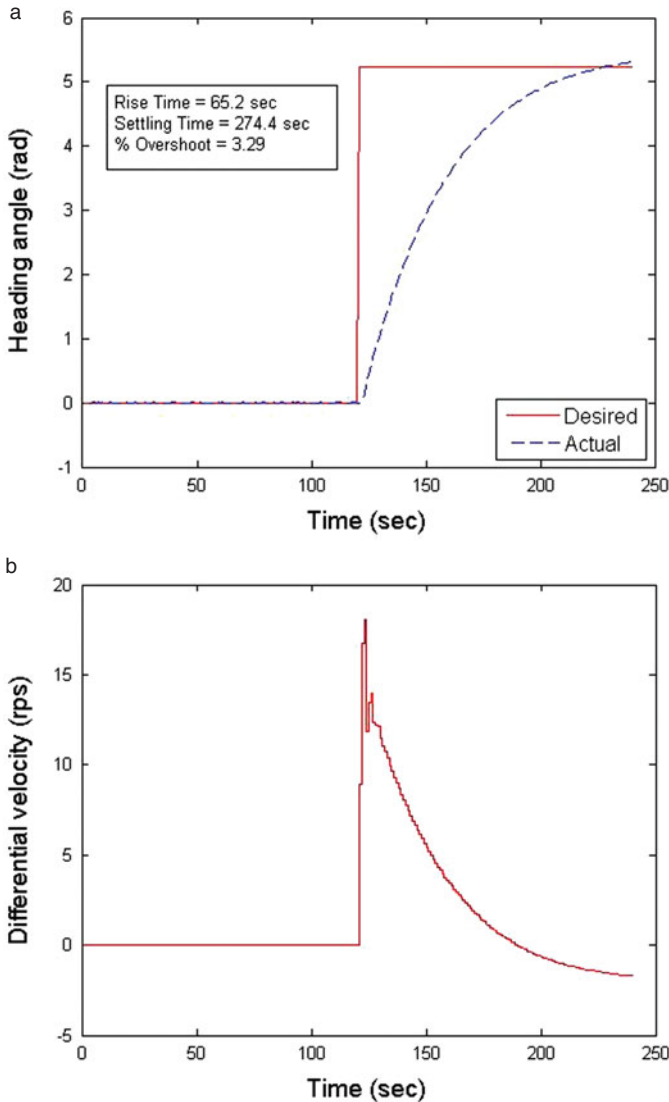


Figure 15. (a) Linear Quadratic Gaussian (LQG) autopilot response to a step change in vehicle heading: heading response. (b) Linear Quadratic Gaussian (LQG) autopilot response to a step change in vehicle heading: controller effort.

$T_S = 274$ s and $\%M_p =$ approximately 3.3% as depicted in Figure 15a. Hence the results show that faster and better responses were obtained with LCN.

Table 1 compares the LCN and LQG autopilots for the different trajectories with respect to $MSE(\psi_e)$ and $ACE(E_u)$.

As shown in Table 1, the LCN approach outperformed and produced less MSE values compared to the LQG in all cases. A major reduction in controller effort was also achieved by the LCN; this would save battery power when controlling the vehicle in real-time mission scenarios.

Table 1. Comparison between the LCN and LQG autopilots for different trajectories.

Methods used	Trajectory	$MSE(\psi_e)$ (rad ²)	$ACE(E_u)$ (rps) ²
LCN	Step	0.825	0.00563
	Random	0.1492	0.1824
LQG	Step	2.0691	14.5102
	Random	0.5092	4.2324

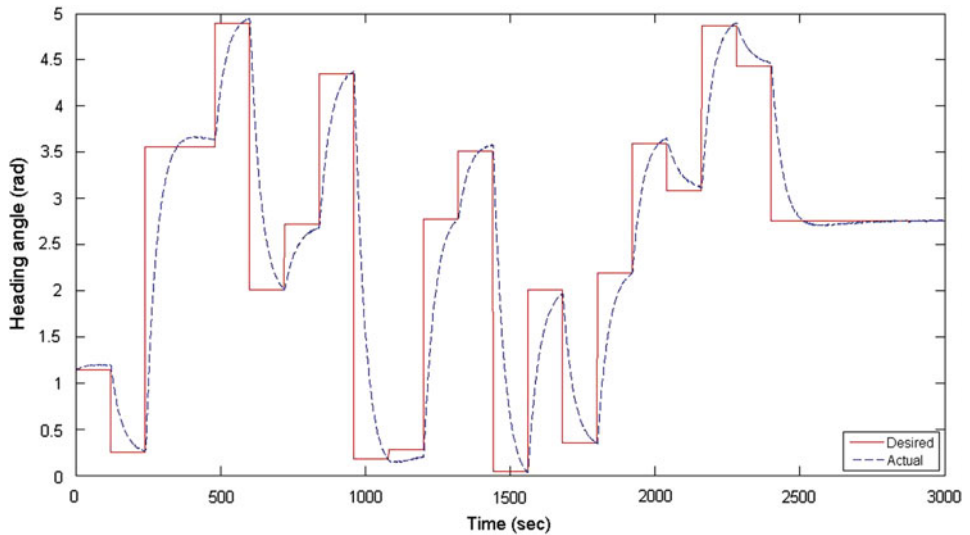


Figure 16. Vehicle response to a random sequence of course-changing demands in heading using the Linear Quadratic Gaussian (LQG) autopilot.

5. CONCLUDING REMARKS. A genetic learning approach to the construction of Local Model Controllers (LMCs) for the *Springer* Unmanned Surface Vehicle (USV) has been proposed based on Local Control Networks (LCNs). The approach optimizes an overall LMC structure and local Proportional-Integral-Derivative (PID) controller parameters. Also it facilitates the inclusion of transparency, generalization of constraints, and simplicity of design. Furthermore, instead of requiring *a priori* knowledge of the *Springer* for designing a LMC, a much easier Neural Network (NN) modelling approach from experimental data was utilized.

The results of the LCN autopilot were compared with those obtained from that developed using a LQG approach, which requires a linear model in its architecture. Even a fourth order State Space (SS) model was unable to provide better predictive capability compared to the NN model and thus the LQG approach produced an inferior autopilot which resulted in greater Mean-Square Error of the Yaw Error $MSE(\psi_e)$ values and consuming more controller energy compared to that of the LCN design.

Thus the LCN is more suited for use in the *Springer* vehicle from a practical viewpoint in terms of manoeuvring capability and also in controller energy consumption which would in reality provide the vehicle with longer mission durations.

Finally, it is considered that this work is the first to report a LCN approach to marine control systems design. As a result, a new design framework for the design of nonlinear autopilots has been presented for application in the marine sector.

APPENDIX A. A discrete-time controlled process may be described by the linear stochastic difference equations (A1):

$$\begin{aligned} \mathbf{x}(k + 1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}u(k) + w(k) \\ z(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}u(k) + v(k) \end{aligned} \tag{A1}$$

where:

- $x(k)$ is an $n \times 1$ state vector.
- \mathbf{A} is an $n \times n$ state transition matrix.
- $u(k)$ is an $l \times 1$ input vector.
- \mathbf{B} is an $n \times l$ matrix.
- $w(k)$ is an $n \times 1$ process noise vector.
- $z(k)$ is an $m \times 1$ measurement vector.
- $\mathbf{C}(k)$ is an $m \times n$ measurement matrix.
- $v(k)$ is an $m \times 1$ measurement noise vector.

Both the $w(k)$ and $v(k)$ are assumed to be uncorrelated zero mean Gaussian white noise sequences with covariances are given by

$$E[w(k)w^T(i)] = \begin{cases} \mathbf{W}(k), & i = k \\ 0, & i \neq k \end{cases}$$

$$E[v(k)v^T(i)] = \begin{cases} \mathbf{V}(k), & i = k \\ 0, & i \neq k \end{cases}$$

$$E[w(k)v^T(i)] = 0, \text{ for all } k \text{ and } i$$

The Kalman filter equations may be written down into time update and measured update Equations (A2) to (A6), where:

- $\hat{\mathbf{x}}$ is an estimate of the system state vector x .
- K is the Kalman gain
- \mathbf{P} is the covariance matrix of the state estimation error.

Time update equations:

$$\hat{\mathbf{x}}(k + 1) = \mathbf{A}\hat{\mathbf{x}}(k) + \mathbf{B}u(k) \tag{A2}$$

$$\mathbf{P}^-(k + 1) = \mathbf{A}\mathbf{P}(k)\mathbf{A}^T + \mathbf{W}(k) \tag{A3}$$

Measurement update equations

$$\mathbf{K}(k) = \mathbf{P}^-(k)\mathbf{C}^T[\mathbf{C}\mathbf{P}^-(k)\mathbf{C}^T + \mathbf{V}(k)]^{-1} \tag{A4}$$

$$\hat{\mathbf{x}}(k) = \hat{\mathbf{x}}^-(k) + \mathbf{K}(k)[z(k) - \mathbf{C}\hat{\mathbf{x}}^-(k)] \tag{A5}$$

$$\mathbf{P}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{C}]\mathbf{P}^-(k) \tag{A6}$$

The measurement update equations incorporate a new observation into the *a priori* estimate from the time update equations to obtain an improved *a posteriori* estimate.

APPENDIX B. A derivation has been carried out herein to evaluate the contents of blocks N_x and N_u in Figure 5 for reference input tracking in an LQR control strategy (Franklin et al., 1998).

Let N_x denotes the forward block which transforms the reference input r to a reference state x_r that is an equilibrium one for that r . Mathematically, this can be stated in Equations (B1) and (B2) as:

$$N_x r = x_r \quad (\text{B1})$$

$$u = -K_{LQR}(x - x_r) \quad (\text{B2})$$

The final or steady state value of the states can be written in Equation (B3) as:

$$x(\infty) = x_{ss} = x_r \Rightarrow N_x r = x_r = x_{ss} \quad (\text{B3})$$

To compensate for any steady state output error in case of type 0 systems, a steady state control term is needed that is proportional to the reference input, in Equation (B4) as:

$$u_{ss} = N_u r \quad (\text{B4})$$

Also:

$$C x_{ss} = y = r \Rightarrow C N_x r = r \Rightarrow C N_x = I \quad (\text{B5})$$

Since the system is at steady state, i.e., $x(k+1) = x(k) = x_{ss}$ and $u = u_{ss}$, therefore:

$$x_{ss} = A x_{ss} + B u_{ss} \Rightarrow (A - I)x_{ss} + B u_{ss} = 0 \quad (\text{B6})$$

Substituting Equations (B4) and (B5) in the above equations:

$$(A - I)N_x r + B N_u r = 0 \Rightarrow (A - I)N_x + B N_u = 0 \quad (\text{B7})$$

Finally writing Equations (B5) and (B7) in matrix form in Equation (B8):

$$\begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix} \begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (\text{B8})$$

and solving for N_x and N_u yields the desired result in Equation (B9):

$$\begin{bmatrix} N_x \\ N_u \end{bmatrix} = \begin{bmatrix} A - I & B \\ C & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \quad (\text{B9})$$

REFERENCES

- Brown, M. D., Lightbody, G. and Irwin, G. W. (1997). Non-linear Internal Model Control using Local Model Networks. *IEE Proceedings on Control Theory and Applications*, **144**, 505–514.
- Burl, J. B. (1999). *Linear Optimal Control, H_2 and H_∞ Methods*. Addison-Wesley Longman Inc.
- Caccia, M., Bibuli, M., Bono, R., Bruzzone, Ga., Bruzzone, Gi. and Spirandelli, E. (2007). Unmanned Surface Vehicle for Coastal and Protected Waters Applications: The Charlie Project. *Marine Technology Society Journal*, **41**, 62–71.
- Fossen, T. I. (2000). A Survey of Nonlinear Ship Control: From Theory to Practice. *Proceedings of the 5th IFAC Conference on Manoeuvring and Control of Marine Craft*, Aalborg, Denmark.
- Fossen, T. I. (2011). *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley and Sons.
- Franklin, G.F., Powell, J. D. and Workman, M. (1998). *Digital Control of Dynamic Systems*. 3rd Edition, Addison-Wesley Longman Inc.
- Holzhtuter, T. and Schultze, R. (1996). Operating Experience with a High Precision Track Controller for Commercial Ships. *Control Engineering Practice*, Vol 4, 343–350.

- Johansen, T. A. and Foss, B. A. (1995). Semi-Empirical Modelling of Nonlinear Dynamic Systems Through Identification of Operating Regimes and Local Models. *Neural Network Engineering in Dynamic Systems*, 105–126.
- Ljung, L. (1999). *System Identification: Theory for the User*. 2nd Edition. Prentice Hall Inc., New Jersey.
- Majohr, J., Buch, T. and Korte, C. (2000). Navigation and Automatic Control of the Measuring Dolphin (Messin™). *Proceedings of 5th IFAC Conference on Manoeuvring and Control of Marine Craft*, Aalborg, Denmark.
- Michalewicz, Z. (1996). *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd Edition, Springer-Verlag, New York.
- Minoriski, N. (1922). Directional Stability of Automatic Steered Bodies. *Journal of the American Society of Naval Engineer*, **34**(2), 280–309.
- Moreira, L. and Soares, G. C. (2005). Design of Robust Steering Autopilot for Ships. *Proceedings of the 12th International Congress of the International Maritime Association of the Mediterranean*, Lisbon, Portugal.
- Naeem, W., Xu, T., Sutton, R. and Tiano, A. (2008). The Design of a Navigation, Guidance, and Control System for an Unmanned Surface Vehicle for Environmental Monitoring. *Proceedings of the Institute of Mechanical Engineers Part M: Journal of Engineering for the Maritime Environment*, **222 M2**, 67–80.
- Ogata, K. (2002). *Modern Control Engineering*. 4th Edition, Prentice-Hall Inc.
- Rippin, D. W. T. (1989). Control of Batch Processes. *Proceedings of the 3rd IFAC DYCORS+ '89 Symposium*, Maastricht, Netherlands.
- Sharma, S. K., McLoone, S. and Irwin, G. W. (2005). Genetic Algorithms for Local Controller Network Construction. *Proceedings of the IEE, Control Theory & Application*, **152**, 587–597.
- Sharma, S. K., McLoone, S. and Irwin, G. W. (2002). Genetic Algorithms for Local Model and Local Controller Network Design. *Proceedings of the American Control Conference*, Anchorage, USA.
- Sivanandam, S. N. and Deepa, S. N. (2008). *Introduction to Genetic Algorithms*. Springer-Verlag, Berlin.
- Sperry, E. A. (1922). Automatic Steering. *Trans Society of Naval Architects and Marine Engineers*, 53–61.
- Sperry Gyroscope Company (2011). *A Short History of Sperry Marine*. www.sperrymarine.northropgrumman.com/Company-Information/Corporate-History/Sperry-History/ accessed 27 September 2011.
- Townsend, S., Lightbody, G., Brown, M. D. and Irwin, G. W. (1998). Nonlinear Dynamic Matrix Control using Local Model Networks. *Trans Inst MC*, **20**, 47–56.
- Tran, V. T., Nguyen, H. T., Hoang, T. X., Nguyen, T. M. H., Cu, X.-T. and Nguyen, V. P. (2004). An Optimal Autopilot for Ships using a Regressive Exogenous Model. *Proceedings of the International Symposium on Communications and Information Technologies*, Sapporo, Japan.
- Yan, R. J., Pang, S., Sun, H. B. and Pang, Y. J. (2010). Development and Missions of Unmanned Surface Vehicle. *Journal of Marine Science and Application*, **9**, 451–457.