Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment

http://pim.sagepub.com/

Published by:

⑤SAGE

http://www.sagepublications.com

On behalf of:

Institution of MECHANICAL ENGINEERS

Institution of Mechanical Engineers

Additional services and information for *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment* can be found at:

Email Alerts: http://pim.sagepub.com/cgi/alerts

Subscriptions: http://pim.sagepub.com/subscriptions

Reprints: http://www.sagepub.com/journalsReprints.nav

Permissions: http://www.sagepub.com/journalsPermissions.nav

Citations: http://pim.sagepub.com/content/228/2/146.refs.html

>> Version of Record - May 14, 2014

OnlineFirst Version of Record - Nov 5, 2013

What is This?

Non-linear control algorithms for an unmanned surface vehicle

Sanjay K Sharma, Robert Sutton, Amit Motwani and Andy Annamalai

Abstract

Although intrinsically marine craft are known to exhibit non-linear dynamic characteristics, modern marine autopilot system designs continue to be developed based on both linear and non-linear control approaches. This article evaluates two novel non-linear autopilot designs based on non-linear local control network and non-linear model predictive control approaches to establish their effectiveness in terms of control activity expenditure, power consumption and mission duration length under similar operating conditions. From practical point of view, autopilot with less energy consumption would in reality provide the battery-powered vehicle with longer mission duration. The autopilot systems are used to control the non-linear yaw dynamics of an unmanned surface vehicle named *Springer*. The yaw dynamics of the vehicle being modelled using a multi-layer perceptron-type neural network. Simulation results showed that the autopilot based on local control network method performed better for *Springer*. Furthermore, on the whole, the local control network methodology can be regarded as a plausible paradigm for marine control system design.

Introduction

From the review by Motwani,[1] it is clear that unmanned surface vehicles (USVs) are now being used in an array of different application areas in the commercial, naval and scientific sectors. Indeed, they are currently being used for mine counter-measures,[2] surveying[3] and environmental data gathering,[4] to name but a few. In order to meet the ongoing challenges of these sectors USV technology continues to be developed particularly in the field of navigation, guidance and control systems. For such vehicles to be capable of undertaking the kinds of mission that are now being contemplated, they require robust, reliable, accurate and adaptable autopilot systems which allow seamless switching between automatic and manual control modes. Such properties in marine control systems being necessary for the changes in the dynamic behaviour of the vehicles that may occur owing to the deployment of different payloads, mission requirements and varying environmental conditions. Thus, in order to meet the testing demands being imposed by these sectors, autopilots have been designed based on, for example, fuzzy,[5] gain scheduling,[6] H infinity,[7] linear quadratic Gaussian,[8] sliding mode[9] and neural network (NN)[10]

techniques that have met with varying degrees of success.

Since management and monitoring of the environment is a major issue worldwide, an USV named *Springer*, depicted in Figure 1, has been specifically designed and developed to be a cost-effective and environmentally friendly USV primarily for undertaking pollutant tracking, and environmental and hydrographical surveys in rivers, reservoirs, inland waterways and coastal waters, particularly where shallow waters prevail.

The dynamic characteristics of marine vessels are invariably non-linear and the *Springer* USV is no exception. This is further confirmed in Sharma and Sutton[11] by showing a non-linear model predictive controller

Marine and Industrial Dynamic Analysis Group, Centre for Advanced Engineering Systems and Interactions, School of Marine Science and Engineering, Plymouth University, Plymouth, UK

Corresponding author:
Sanjay K Sharma, Marine and Industrial Dynamic Analysis Group, Centre for Advanced Engineering Systems and Interactions, School of Marine Science and Engineering, Plymouth University, Drake Circus, Plymouth PL4 8AA, UK.
Email: sanjay.sharma@plymouth.ac.uk

outperforming a linear proportional–integral–derivative (PID) controller. Thus, this article reports the application of two novel non-linear autopilot designs for the vehicle. Local control network (LCN) and non-linear model predictive control (NMPC) schemes being used in their designs. Details of the navigation and line-of-sight guidance subsystems for the vehicle can be found in Naeem et al.[12]

Yaw dynamics of the *Springer* vehicle

Full details of the *Springer's* hardware can be found in Naeem et al.[12] The *Springer* USV having been designed as a medium waterplane twin hull vessel which is versatile in terms of mission profile and payload. It is

Figure 1. The *Springer* unmanned surface vehicle.

approximately 4 m long and 2.3 m wide with a displacement of 0.6 tonnes.

A multi-layer perceptron (MLP)-type NN model of the *Springer* yaw dynamics was developed using a dataset recorded during full-scale trials. A genetic algorithm (GA)[13] was used to obtain the unknown parameters of the MLPNN model which had a population of 20 chromosomes, a crossover probability of $p_c = 0.65$ and mutation probability of $p_m = 0.03$. The GA was run till maximum of 10,000 generations or mean square error (MSE) of less than $MSE \leqslant 0.00001$ was achieved on normalised training dataset. A parallel architecture network model was then tested on validation and test data to check its predictive capability. The GA selected the MLPNN with four hidden nodes and represented in generic form as

$$\hat{y}(t) = f_{NN} \left\{ \begin{array}{l} u(t), u(t-1), u(t-2), u(t-3), \hat{y}(t-1), \hat{y}(t-2), \hat{y}(t-3), \hat{y}(t-4), \\ e(t-1), e(t-2), e(t-3), e(t-4) \end{array} \right\} \quad (1)$$
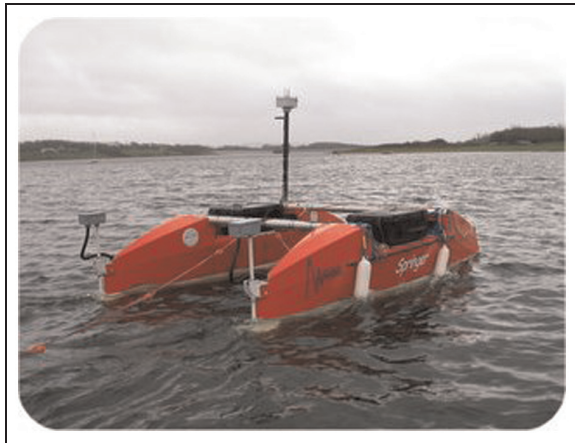
where $e(t) = y(t) - \hat{y}(t)$. In the case of the MLP, inputs are multiplied by the weights between the input and hidden layer and then between the hidden and output layer to produce the final output. Here $\tan h$ was used as an activation function in the hidden layer and linear in the output.

Figure 2(a) and (b) illustrates the performances of the MLPNN on validation and test dataset which produced mean-squared errors of 0.00012567 and 0.00018626 rad[2], respectively. Modelling of the yaw dynamics of the *Springer* vehicle is detailed in Sharma and Sutton.[14]

Thereupon this NN model was used to replicate the non-linear yaw dynamics of the *Springer* USV and to train a LCN autopilot to follow set point trajectories and also used in the architecture of the NMPC algorithm.
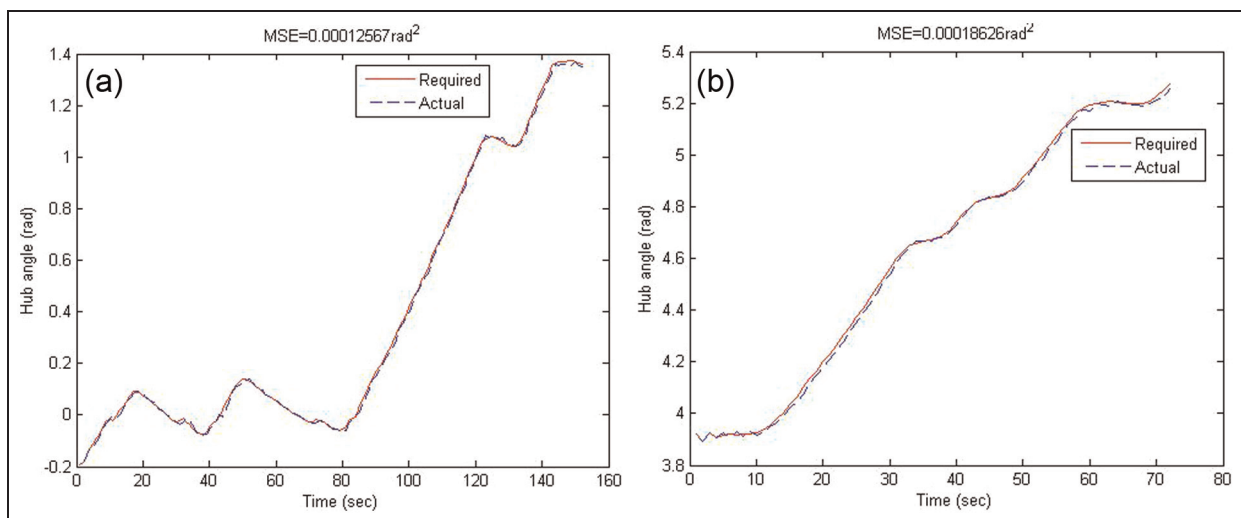

Figure 2. Predictive capability of the MLPNN model on (a) validation and (b) test data.
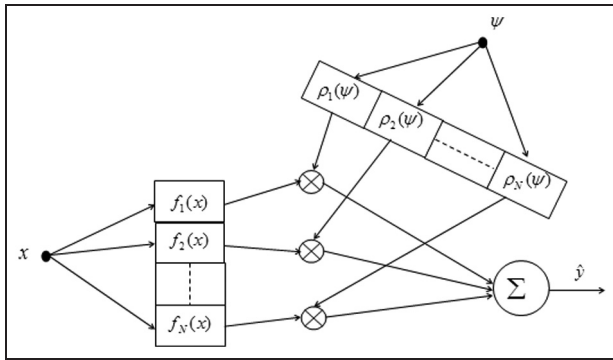MSE: mean square error.

Figure 3. General architecture of a LCN.
PID: proportional–integral–derivative.

Autopilot designs

LCN autopilot design

LCN for complex non-linear systems is designed by divide-and-conquer approach by interpolating several linear local model controllers (LMCs) spread across the operating regions.[15] A priori information of the local operating regimes is therefore needed to build a global LCN. A priori plant knowledge is used to design the LCN to control most of the non-linear systems.[16–19] The global controller acting at a point receives maximum contribution from the LMC valid around that operating point and less from the neighbouring ones and none from the distant ones. The relative validity of each LMC at operating point is decided by the associated weighting function. No work to date has been undertaken to test the USV while it operates in the non-linear range of its dynamic response. This article identifies the local operating regions along with the parameters of a global LCN based on GAs for *Springer*.

LCN. Figure 3 shows the general discrete LCN architecture. The same inputs, x, are fed to all the LMCs, and the outputs are weighted according to some weighting or scheduling variables, ψ. The output from the LCN $\hat{y}$ is provided by the weighted sum of the output obtained from each LMC

$$\hat{y} = \sum_{i=1}^{N} \rho_i(\psi) f_i(x) \tag{2}$$

where $\rho_i(\psi)$ is the validity or interpolation function associated with the ith LMC, $f_i(x)$ and N is the total number of LMCs.

The total contribution from all the LMCs is made 100% by normalising the validity functions $\rho_i(\psi)$, and the widely used $\rho_i(\psi)$ in the literature are normalised Gaussian functions which are given as

$$\rho_i(\psi) = \frac{\exp\left(-\|\psi - s_i\|^2 / 2\sigma_i^2\right)}{\sum_{j=1}^{N} \exp\left(-\|\psi - s_j\|^2 / 2\sigma_i^2\right)} \tag{3}$$

where s_i is the centre and σ_i is the standard deviation associated with the validity function of ith LMC. Herein, discrete-type PID controllers are considered for the $f_i(x)$ as linear controllers for the LCN construction.

A continuous-time PID control law is defined by

$$u(t) = k_p e(t) + k_d \dot{e}(t) + k_I \int e \, dt \tag{4}$$

where k_p, k_d and k_I are the proportional, differential and the integral gains, u is the control action and e is the error. The PID controller in discrete form is equivalently represented as

$$\begin{aligned} u(k) = u(k-1) &+ k_p[e(k) - e(k-1)] \\ &+ k_d[e(k) - 2e(k-1) + e(k-2)] + T_s k_I e(k) \end{aligned} \tag{5}$$

where k is the sample number and T_s is the sampling interval and selected 1 here in GA optimisation.

The unknown parameters for GA in each regime in equation (3) are the validity function centres s_i and the standard deviations σ_i. The unknown parameters in equation (5) are the PID control parameters k_p, k_d and k_I.

Figure 4 shows the design of a LCN with m PID-type LMCs. The output of the ith 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 $u(k) = c(k) + u(k-1)$. All LMCs in the network receive the same error $e(k) = r(k) - y(k)$, as input. The weighting or scheduling variable for the validity function, $\rho_i(\psi)$, was chosen as $\psi = [y(k-1), u(k-1)]$, where $y(k)$ is the heading angle filtered output and $r(k)$ is the reference set point. The filter was used to smooth the signal in the feedback loop. A GA was then used to construct a LCN for the USV. The optimal number of LMCs (from a given maximum number), the parameters of these LMCs and the parameters of the validity functions and filter are selected simultaneously by GA. It also incorporates as constraints to make sure that all valid LMCs to be mutually orthogonal and act independently at its operating point. The fitness function of the GA reduces the tracking error and total controller effort and is given by $ACE/(1 + MSE)$, where ACE is the average equivalent controller energy and MSE is the mean square error of the yaw error as defined in equations (10) and (11), respectively.

The filter used in Figure 4 is of second order with the input/output relation described in equation (6)

$$\begin{aligned} a_1 y(k) = b_1 x(k) &+ b_2 x(k-1) + b_3 x(k-2) \\ &- a_2 y(k-1) - a_3 y(k-2) \end{aligned} \tag{6}$$

NMPC autopilot design

The concepts and techniques of model predictive control (MPC) have been developing for over three decades
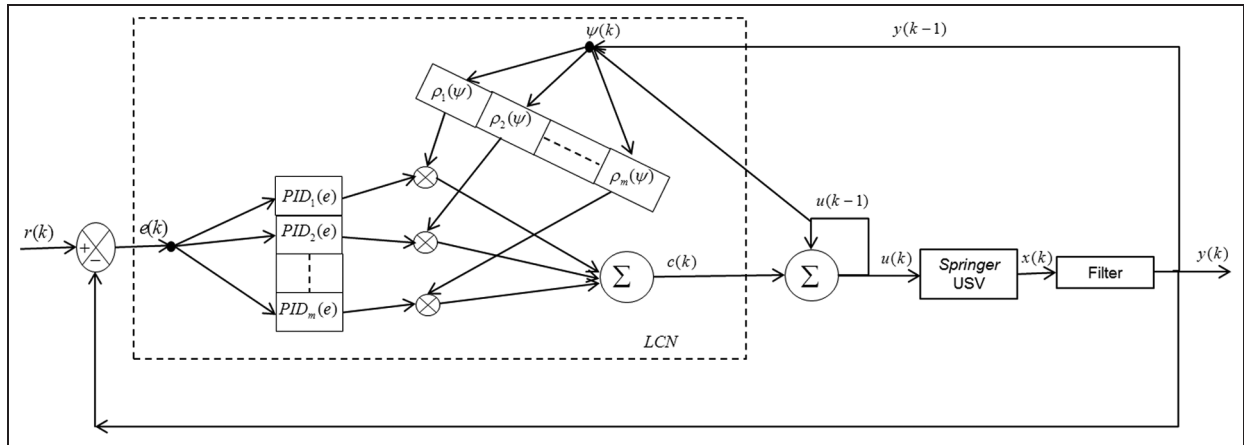
Figure 4. Proportional–integral–derivative-type local control network acting on the *Springer* unmanned surface vehicle.

and are shown to be popular in many sectors such as the process and automotive industries, and in academia as illustrated in the text of Maciejowski,[20] Rawlings and Mayne,[21] Wang[22] and Allgower et al.[23] In addition, the marine control system design fraternity have also embraced this approach since it offers the advantage of being capable of enforcing various types of constraints on the plant process as exemplified by Perez,[24] Oh and Sun,[25] Liu et al.,[26] Li and Sun[27] and Naeem et al.[28]

Although MPC is well established and has provided solutions to a number of problems, in its linear approach, however, it does have its limitations particularly when dealing with non-linear plant. Many systems are, however, inherently non-linear and operate under tight performance conditions with many satisfying constraints. These demands require systems to operate over a wide range of operating conditions and linear models are often not sufficient to describe the system dynamics adequately and hence non-linear models must be used. This inadequacy of linear models is one of the motivations for the increasing interest in NMPC. The closed-loop dynamics of the linear MPC are non-linear due to the presence of constraints and will not provide an optimal solution. NMPC works on the basis of non-linear models and takes accounts of non-quadratic cost-functional and general non-linear constraints. Excellent introductions to such techniques can be found in Tatjewski and Lawrynczuk,[29] and Grune and Pannek.[30]

Principle of NMPC. At the heart of MPC are the model of the system and the concept of open-loop optimal feedback. The model is used to generate a prediction of future behaviour of the system. At each time step, past measurements and inputs are used to estimate the current state of the system. An optimisation problem is solved to determine an optimal open-loop policy from the present (estimated) state. Only the first input move is applied to the plant. At the subsequent time step, the system state is re-estimated using new measurements.

The optimisation problem is resolved and the first input move to the plant is calculated again. Figure 5 presents the working scheme of a MPC. Here the controller predicts the dynamic behaviour of the system in the future over a prediction horizon N_p and determines the input over a control horizon ($N_c \leqslant N_p$) based on the measurements obtained at time t such that an open-loop performance objective J is minimised.

The block diagram in Figure 6 illustrates the NMPC process used in this article. The NMPC consists of the MLPNN model and the GA optimisation block. The u_m variable is the tentative control signal, y_r is the desired response and y_m is the filtered network model response. The GA optimisation block determines the values of u_m that minimise J, and then the optimal u_p is input to the plant.

The objective function J mathematically describes the control goal. In general, good tracking of the reference trajectory is required with low control energy consumption. The predictions are used by a numerical optimisation program to determine the control signal that minimises the following performance criterion over the specified horizon

$$
\begin{aligned}
J = &\sum_{i=N_0}^{N_p} [y_m(t+i) - y_r(t+i)]^2 \\
&+ \lambda \sum_{j=1}^{N_c} [u_m(t+j-1) - u_m(t+j-2)]^2
\end{aligned}
\tag{7}
$$

where N_0, N_p and N_c define the horizons over which the tracking error and the control increments are evaluated. The λ value determines the contribution that the sum of the squares of the control increments has on the performance index.

The key characteristics of NMPC are as follows:

- Non-linear models can directly be used in NMPC for prediction.
- Constraints can be easily incorporated.
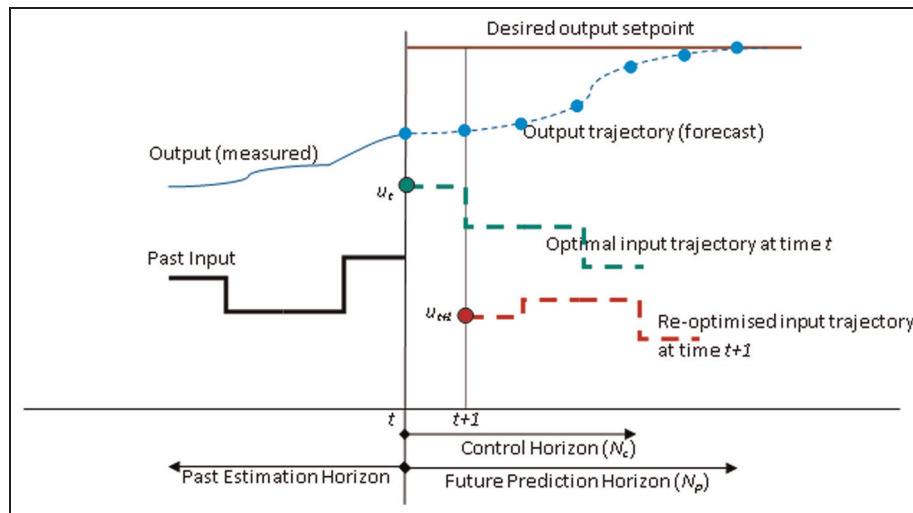- Applied on-line.

Figure 5. A conceptual picture of MPC. Only ($u_t|t$) is injected into the plant at time t. At time $t + 1$, a new optimal trajectory is recomputed.
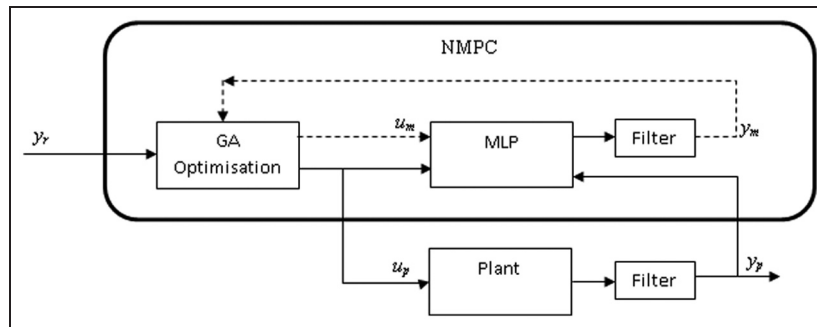

Figure 6. The NMPC process.
NMPC: non-linear model predictive control; MLP: multi-layer perceptron.

The repeated on-line solution from NMPC is in general complex and computationally expensive compare to a linear MPC where the on-line solution of the optimal control problem can be efficiently obtained by a quadratic program. This limits the successful practical application of NMPC. Thus, NMPC has been applied almost only to slow systems. For fast systems where the sampling time is considerably small, the existing NMPC algorithms cannot be used. Therefore, solving such a non-linear optimisation problem efficiently and quickly has attracted significant research interest in recent years.[31–34]

The conventional iterative optimisation method requiring initial values based on gradient descent such as sequential quadratic programming (SQP) has been applied to NMPC.[35] These techniques can succumb to local minima and can lead to infeasible solution. GA on the other hand is a global stochastic search technique that applies the concept of biological evolution to find an optimal solution in a search space and has proved to be efficient in solving complicated non-linear optimisation problems compared to the more conventional optimisation techniques such as SQP. Furthermore, the search range of the input variable constraints can easily

be incorporated in the search space of a GA during optimisation, which makes it easier to handle the input constraint problem than other descent-based methods. However, the computational burden in the case of a GA is much heavier and increases exponentially with the increase of the horizon length of the NMPC making it difficult to implement in the conventional form; thus, only a few applications of GAs to non-linear MPC[36,37] are available in the literatures. In this article, a modified NMPC algorithm based on a GA is proposed for the design of an autopilot for the *Springer* USV. In place of seeking the exact global solution for NMPC at every sampling time, suboptimal control sequences satisfying the constraints are implemented. The GA decreases the cost function within the sampling interval and the best chromosome represents the optimal control sequence at that time and so on. This requires less computational demands without deteriorating much to the control performance.[38]

GA optimised NMPC algorithm

Here a GA is used to obtain a sequence of optimal control signals. More specifically, a steady-state GA with
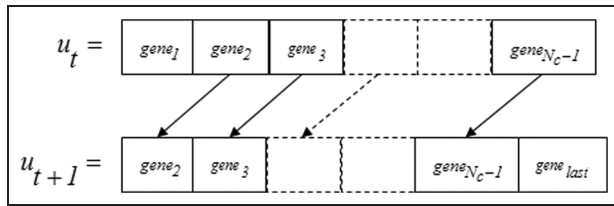
Figure 7. Chromosomes representing optimal control trajectories at time t and $t + 1$.

floating point encoding and special genetic operators including initialisation, mutation, crossover and termination was used. The fitness function of the GA is derived from the objective function of the NMPC. Mutation and crossover operators are designed with built-in constraints in order not to violate the constraints of the control inputs. A convergence measure is introduced as a termination condition. The operation of the GA used here is explained as follows.

Encoding. Every individual chromosome $\{o_i; i = 1, \ldots, Npop\}$ in the population of the GA determines a control trajectory: $\{o_i = [u_i(t), u_i(t + 1), \ldots, u_i(t + N_c - 1)]\}$.

An individual chromosome o_i is described by a set of N_c floating point numbers which are selected within the admissible control interval $[u_{min}, u_{max}]$ and with absolute difference $\{\Delta u_i(t + j); j = 1, \ldots, N_c - 1\}$ not exceeding a prescribed value Δu_{max}. Here u_{min} and u_{max} are constraints limiting the range of the control signal, whereas $\Delta u_{\max}$ limits the gradient of the control signal.

Initialisation. A suitable initialisation procedure at every sampling interval is required in order to obtain a better solution from the GA optimisation. Here the best solution of the last optimisation cycle with shift in the sequence of control signals as shown in Figure 7 is used to initialise the half of the chromosomes in a population and the rest of the chromosomes are randomly initialised with control sequence within the admissible constrained as defined in the encoding. Figure 7 shows the previous optimised control trajectory u_t as a chromosome with genes: $\{gene_1, gene_2, \ldots, gene_{N_c-1}\}$ representing the optimal control sequence at time t. The optimal chromosome at $t + 1$ is created by shifting the control sequence as shown in this figure. The last gene at $t + 1$ is randomly added in u_{t+1} satisfying limiting constraints. Now the remaining half of the chromosomes in the population are created by randomly adding a floating number within a range of $\pm \Delta u_{max}$ to each gene of u_{t+1}. The value of a gene is then adjusted to restrict it within the admissible control interval $[u_{min}, u_{max}]$.

Initialisation of the chromosomes within the close vicinity of the best solution of the previous optimisation cycle facilitates the optimisation procedures by the exploitation of previously accumulated knowledge. This strategy guarantees the quality of the current

population and the stability of the NMPC algorithm, whereas the rest of the population with randomly generated chromosomes add to the genetic diversity and are responsible for exploring the global search space in the solution.

Mutation. The mutation introduces new genetic variations into the population. The selected genes on the basis of mutation probability p_m are randomly replaced within admissible constraints of the control signals $[u_{min} \leqslant u \leqslant u_{max}]$ and $|\Delta u| \leqslant |\Delta u_{max}|$.

Crossover. The crossover is used to exchange genetic information between the two chromosomes of the population. In this article, an arithmetic crossover between the two selected parents on the basis of crossover probability p_c is used to produce two offspring. This procedure maintains the control signals within the admissible constraints.

Termination conditions. This determines when the GA optimisation loop should be stopped and first control input from the best chromosome is applied to the plant. Judicious selection of the termination criteria of the GA is the key factor in reducing the computation burden in the design of the suboptimal NMPC algorithm. Here the GA was run till 90% of the sampling interval is either elapsed or evolution converges whichever is earlier. This insures that at every sampling interval, a feasible control signal is always available for the vehicle.

Fitness value and selection. The fitness value of each chromosome is defined as $1/(J + 1)$ and the best chromosomes from the current parent and children are selected for the next generation and rest are discarded to keep the number of chromosomes in a population constant.

Simulation results

In this application, a steady-state GA with crossover probability of $p_c = 0.65$ and mutation probability of $p_m = 0.03$ was applied to a population of 20 chromosomes. GA minimises the cost function defined in equation (7) in both LCN and NMPC on infinite and prediction horizon, respectively. The range of controller input and gradient of the controller input selected from the actuator limit were $\{u_{min}, u_{max}\} = \{-132 \text{ r/min}, 132 \text{ r/min}\}$ and $|\Delta u_{max}| \leqslant 20 \text{ r/min}$, respectively. The weighting parameter λ for performance objective was kept low to provide better convergence and selected 0.01 as a value after trial and error.

LCN simulation results

For the LCN, the GA was run for 2000 generations, and to allow for the stochastic nature of the genetic learning, the training process was repeated 5 times. The
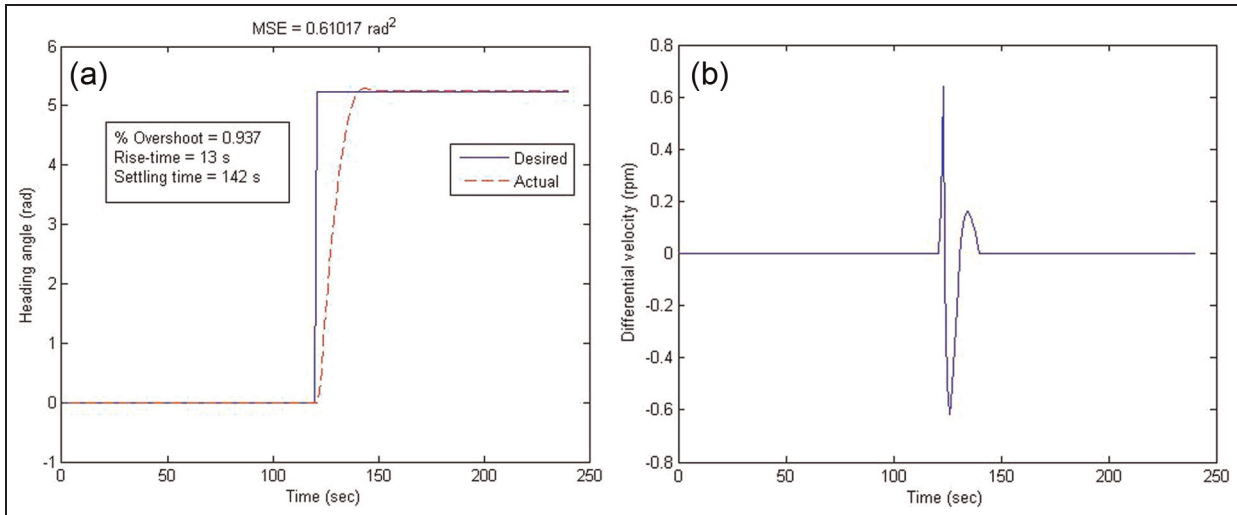
Figure 8. Local control network responses to a step change in vehicle heading: (a) step response and (b) actuator response. MSE: mean square error.

GA selected the optimum number of LMCs, along with parameters of the centres and covariances of the associated function and for the filter. Figure 8(a) shows the closed-loop response and Figure 8(b) the actuator response from the LCN in response to a step change demand in the vehicle heading and reveals that a good tracking performance is achieved.

The two PID-type LMCs defining the LCN were

$$C_1(k) = 12.3294[e(k) - e(k-1)]$$
$$+ 17.1494[e(k) - 2e(k-1) + e(k-2)] + 5.9548e(k)$$
$$(8)$$

$$C_2(k) = 12.2401[e(k) - e(k-1)]$$
$$+ 18.0646[e(k) - 2e(k-1) + e(k-2)] + 6.0891e(k)$$
$$(9)$$

where $e(k) = r(k) - y(k)$ is again the error between the reference and the controller trajectory. The centres and covariances for the Gaussian interpolation functions were ρ_1: $(-17.3445, 122.2610)$ and $(4.7751, 2.6754)$ and ρ_2: $(-108.3460, 80.9760)$ and $(3.9882, 2.8218)$.

The coefficients for the filter being $a_1 = 1.0$, $a_2 = -0.8249$, $a_3 = -0.11$, $b_1 = 0.025$, $b_2 = 0.0233$ and $b_3 = 0.0182$. The global stability of the overall closed-loop system of the LCN is difficult to prove.[39] One way to demonstrate the stability is to test a vehicle manoeuvre throughout the operating space by employing some random sequence of course-changing manoeuvres. The random sequence as shown in Figure 9 reveals that a stable and smooth closed-loop response was followed by a vehicle using LCN.

The NN model of the *Springer* was used directly to design the LCN in the absence of a priori knowledge. These results indicate the importance of this approach to design a LCN-based autopilot for *Springer*. In addition, they clearly illustrate the autopilot's ability to cope
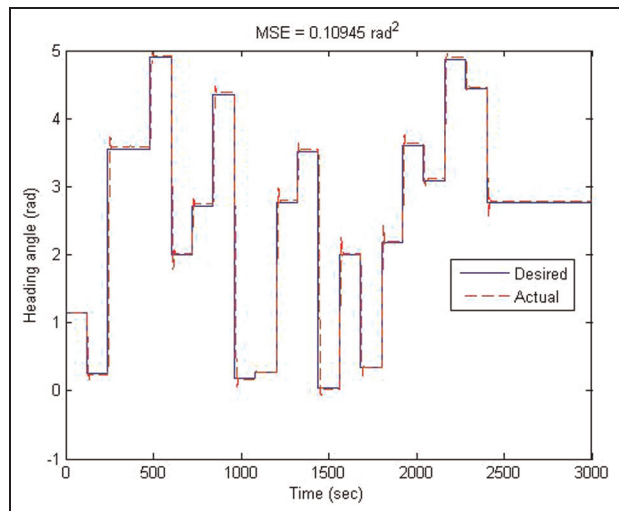

Figure 9. Vehicle response to a random sequence of course-changing demands in heading using the local control network.

successfully when operating in both the linear and nonlinear realms of the vehicle's dynamic system response.

NMPC simulation results

In the case of the NMPC, several predictive horizons were changed heuristically to investigate the suitable control strategy in terms of root mean square (RMS) prediction error. It was observed that too short a prediction horizon ($N_p = 2$) provided an intensively oscillatory prediction with a large RMS prediction error. The NMPC with a horizon $N_p = 6$ resulted in near-optimal control, and by increasing the prediction horizon above $N_p = 6$ added only a slight improvement in RMS error but the computational time increased considerably more. A prediction horizon $N_p = 6$ and control horizon $N_c = 3$ were found appropriate for this
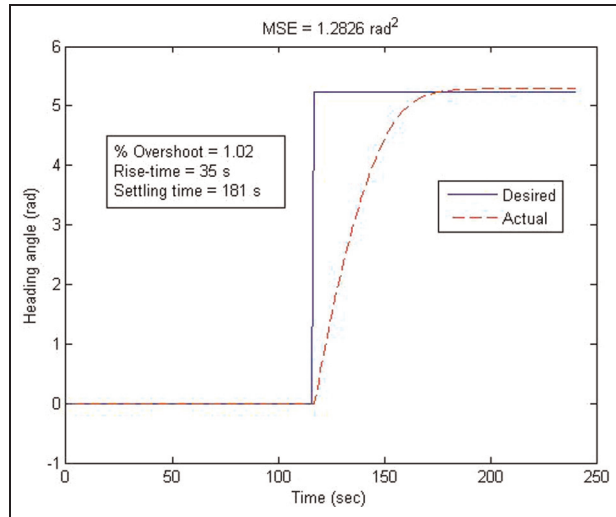
Figure 10. MPC response to a step change in vehicle heading.
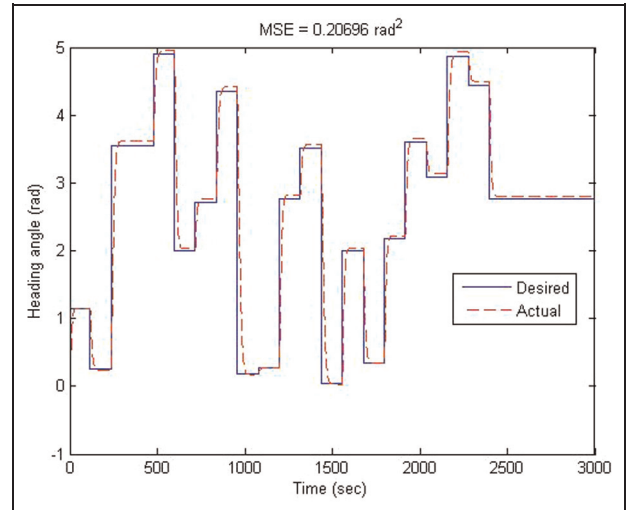MSE: mean square error.

Figure 11. Vehicle response to a random sequence of course-changing demands in heading using the NMPC.
MSE: mean square error.

application and were selected for the simulation study. The simulations were run using Visual C ++ on a desktop PC with Intel Core 2 CPU, 1.86 GHz processor and 1.97 GB of RAM. The maximum time to run a GA generation was 0.90 s, so that the next control action was readily available before the sampling time of 1.0 s. The step response results for the NMPC autopilot is shown in Figure 10 and for the multistep in Figure 11.

In order to make quantitative comparisons between the three autopilot designs, the standard system performance criteria of rise time (T_R), settling time (T_S) and percentage overshoot ($\%M_P$) were employed.[40] Here the rise time 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. While the settling time 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%. Settling time 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 the percentage overshoot 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.

Additionally, as a measure of accuracy and autopilot control activity, the *MSE* of the yaw error and the *ACE* were used as performance indices. These may be considered in their discrete forms as

$$MSE = \frac{1}{N_s} \sum_{k=1}^{N_s} [x(k) - r(k)]^2 \qquad (10)$$

and

$$ACE = \frac{1}{N_s} \sum_{k=1}^{N_s} [u(k)]^2 \qquad (11)$$

Table 1. Comparisons between the LCN and NMPC for rise time (T_R), settling time (T_S) and percentage overshoot ($\%M_P$).

Method used	T_R (s)	T_S (s)	$\%M_P$
LCN	13	142	0.902
NMPC	35	181	1.02

LCN: local control network; NMPC: non-linear model predictive control.

Table 2. Comparisons between the LCN and NMPC autopilots for different trajectories.

Method used	Trajectory	MSE (rad²)	ACE (r/min)²
LCN	Step	0.61017	0.00755
	Random	0.10945	0.012812
NMPC	Step	1.2826	0.42094
	Random	0.20696	0.14224

LCN: local control network; NMPC: non-linear model predictive control; *MSE*: mean square error; *ACE*: average equivalent controller energy.

where $r(k)$ is the desired output at kth instant (in rad); $x(k)$ is the actual output at kth instant (in rad); $u(k)$ is the controller effort at kth instant (in r/min); N_s is the total number of samples.

From the heading step responses in Figures 8 and 10, Table 1 was compiled. Table 2 compares between the LCN and NMPC autopilots for different trajectories with respect to *MSE* and *ACE*.

From the results presented, the rise time (T_R) of the NMPC autopilot is 169% and settling time (T_S) is approximately 27% more than the LCN. This can be accounted as the NMPC required computing controller actions within sampling interval which may not be optimal.

In terms of accuracy $MSE(\psi_\varepsilon)$, the LCN autopilot is better than those generated by the NMPC scheme. In addition, as shown in Table 2, the LCN methodology expends the least amount of controller effort $ACE(E_u)$. Such an achievement may be considered significant as *Springer* is battery powered. Thus, in reality, the LCN autopilot would save more battery power when controlling the vehicle in real-time missions and thereby have a longer operational range compared to the NMPC.

Discussion and concluding remarks

In the majority of the cases presented, the results demonstrate that the non-linear LCN autopilot performed better overall than its counterpart designed with NMPC control architecture. The main advantage of non-linear LCN approach is in its capability to deliver lower levels of control activity for a given task. Thus, the LCN autopilot is more economical with its power consumption than NMPC thereby endowing *Springer* with longer mission durations. From a control systems engineering standpoint, the LCN approach has several properties that make it attractive from a controller design perspective. Some of these commendable properties include the incorporation of transparency, generalisation of constraints and simplicity of design. Even so, it is surprising that the LCN approach has had very limited exposure in the field of marine control systems design. Indeed, this particular non-linear methodology offers an appealing alternative in the design of marine autopilots systems.

Marine control system designers still employ both linear and non-linear approaches in the development of marine autopilots. Thus, the work reported in this article has sought to discover the better non-linear approach to autopilot design for USVs. Therefore, to address this quandary, two autopilot designs based on non-linear LCN and NMPC structures were assessed and contrasted with each other using standard system performance criteria and indices. Given the superior performance of LCN non-linear autopilot, it is baffling as to why LCN techniques have received very little attention in marine control system design. It is considered that these techniques offer a new design framework for the development of non-linear autopilots for application in USVs and in the general marine sector.

Finally and more specifically, it is concluded that of the two control schemes scrutinised, the LCN autopilot is the more appropriate for *Springer* from a practical viewpoint in terms of controller energy consumption which would in reality provide the vehicle with longer mission durations. This being so, the intention in the near future is to undertake in full-scale real-time trials with the LCN autopilot in the control loop.

Declaration of conflicting interests

The authors declare that there is no conflict of interest.

References

1. Motwani A. *Survey of uninhabited surface vehicles.* MIDAS Technical report. MIDAS.SMSE.2012.TR.001, April 2012. Plymouth: Plymouth University.
2. Yan RJ, Pang S, Sun HB, et al. Development and missions of unmanned surface vehicle. *J Mar Sci Appl* 2010; 9: 451–457.
3. Majohr J, Buch T and Korte C. Navigation and automatic control of the measuring dolphin (MESSIN^{TM}). In: *5th IFAC conference on manoeuvring and control of marine craft*, Aalborg, Denmark, 23–25 August 2000, pp.405–410.
4. Caccia M, Bibuli M, Bono R, et al. Unmanned surface vehicle for coastal and protected waters applications: the Charlie project. *Mar Technol Soc J* 2007; 41: 62–71.
5. Park S, Kim J, Lee W, et al. A study on the fuzzy controller for an unmanned surface vessel designed for sea probes. In: *Proceedings of international conference on control, automation and systems*, Kintex, Korea, 2–5 June 2005, pp.1–4.
6. Alves J, Oliveira P, Oliveira R, et al. Vehicle and mission control of the DELFIM autonomous surface craft. In: *Proceedings of 14th Mediterranean conference on control automation*, Ancona, Italy, 28–30 June 2006, pp.1–6. IEEE.
7. Elkaim GH and Kelbley R. Measurement based H infinity controller synthesis for an autonomous surface vehicle. In: *Proceedings of 19th international technical meeting of the satellite division of the institute of navigation*, Fort Worth, TX, 26–29 September 2006, pp.1973–1982.
8. Naeem W, Sutton R and Chudley J. Soft computing design of a linear quadratic Gaussian controller for an unmanned surface vehicle. In: *Proceedings of 14th Mediterranean conference on control automation*, Ancona, Italy, 28–30 June 2006, pp.1–6. IEEE.
9. Ashrafiuon H, Muske KR, McNinch LC, et al. Sliding-mode tracking control of surface vessels. *IEEE T Ind Electron* 2008; 55: 4004–4012.
10. Qiaomei S, Guang R, Jin Y, et al. Autopilot design for unmanned surface vehicle tracking control. In: *Proceedings of 3rd international conference on measuring technology and mechatronics automation*, Shanghai, China, 6–7 January 2011, pp.610–613. IEEE.
11. Sharma S and Sutton R. An optimised nonlinear model predictive control based autopilot for an uninhabited surface vehicle. In: *The 2013 IFAC intelligent autonomous vehicles symposium*, Gold Coast, QLD, Australia, 26–28 June 2013, pp.73–78.
12. Naeem W, Xu T, Sutton R, et al. The design of a navigation, guidance, and control system for an unmanned surface vehicle for environmental monitoring. *Proc IMechE, Part M: J Engineering for the Maritime Environment* 2008; 222: 67–80.
13. Sivanandam SN and Deepa SN. *Introduction to genetic algorithms.* Berlin: Springer-Verlag, 2008.
14. Sharma S and Sutton R. Modelling the yaw dynamics of an uninhabited surface vehicle for navigation and control

systems design. *Proc IMarEST, Part A: J Mar Eng Technol* 2012; 11: 9–20.

15. Townsend S, Lightbody G, Brown MD, et al. Nonlinear dynamic matrix control using local model networks. *T I Meas Control* 1998; 20: 47–56.

16. Rippin DWT. Control of batch processes. In: *Proceedings of the 3rd IFAC DYCORD + '89 symposium*, Maastricht, The Netherlands, 1989, pp.115–125.

17. Johansen TA and Foss BA. Semi-empirical modeling of nonlinear dynamic systems through identification of operating regimes and local models. In K.J. Hunt et al.(eds.), *Neural Network Engineering in Dynamic Control Systems,* Springer-verlag London limited, 1995; pp 105–126.

18. Brown M, Lightbody G and Irwin GW. Non-linear internal model control using local model networks. *IEE P: Contr Theor Ap* 1997; 144: 505–514.

19. Sharma SK, McLoone S and Irwin GW. Genetic algorithms for local controller network construction. *IEE P: Contr Theor Ap* 2005; 152: 587–597.

20. Maciejowski J. *Predictive control with constraints.* London: Prentice Hall, Inc., 2002.

21. Rawlings JB and Mayne DQ. *Model predictive control: theory and design.* Madison, WI: Nob Hill Publishing, 2009.

22. Wang L. *Model predictive control system design and implementation using MATLAB.* Berlin: Springer-Verlag, 2009.

23. Allgower F, Glielmo L, Guardiola C, et al. *Automotive model predictive control.* Berlin: Springer-Verlag, 2010.

24. Perez T. *Ship motion: course keeping and roll stabilisation using rudder and fins.* London: Springer-Verlag, 2005.

25. Oh S-R and Sun J. Path following of underactuated marine surface vessels using line-of-sight based model predictive control. *Ocean Eng* 2010; 37: 289–295.

26. Liu J, Allen R and Yi H. Ship motion stabilizing control using a combination of model predictive control and an adaptive input disturbance predictor. *Proc IMechE, Part I: J Systems and Control Engineering* 2011; 225: 591–602.

27. Li Z and Sun J. Disturbance compensating model predictive control with application to ship heading control. *IEEE T Contr Syst T* 2012; 20: 257–265.

28. Naeem W, Sutton R and Xu T. An integrated multi-sensor data fusion algorithm and autopilot implementation in an uninhabited surface craft. *Ocean Eng* 2012; 39: 43–52.

29. Tatjewski P and Lawrynczuk M. Soft computing in model – based predictive control. *Int J Appl Math Comp* 2006; 16: 7–26.

30. Grune L and Pannek J. *Nonlinear model predictive control: theory and applications.* London: Springer-Verlag, 2011.

31. Martinse F, Biegler LT and Foss BA. A new optimization algorithm with application to nonlinear MPC. *J Process Contr* 2004; 14: 853–865.

32. Moritz D, Findeisen R, Allgower F, et al. Stability of nonlinear model predictive control in the presence of errors due to numerical online optimization. In: *Proceedings of the 42nd IEEE conference decision and control*, Hawaii, 9–12 December 2003, pp.1419–1424. IEEE.

33. Findeisen R, Diehl M, Disli-Uslu I, et al. Computation and performance assessment of nonlinear model predictive control. In: *Proceedings of the 41st IEEE conference on decision and control*, Las Vegas, NV, 10–13 December 2002, pp.4613–4618. IEEE.

34. Diehl M, Findeisen R and Schwarzkopf S. An efficient algorithm for nonlinear predictive control of large-scale systems. *Automatic Technol* 2002; 50: 557–567.

35. Tenny MJ, Wright SJ and Rawlings JB. Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming. *Comput Optim Appl* 2004; 28: 87–121.

36. Michalska H and Mayne DQ. Robust receding horizon control of constrained nonlinear systems. *IEEE T Automat Contr* 1993; 38: 1623–1633.

37. Chen W and Wu G. Nonlinear modeling of two-tank system and its nonlinear model predictive control. In: *Proceedings of the 24th Chinese control conference*, Guangzhou, China, 2005, pp.396–401.

38. Chen W, Li X and Chen M. Suboptimal nonlinear model predictive control based on genetic algorithm. In: *Proceedings of the third international symposium on intelligent information technology application workshops*, Nanchang, China, 21–22 November 2009, pp.119–124. IEEE.

39. Sharma SK, McLoone S and Irwin GW. Genetic algorithms for local model and local controller network design. In: *Proceedings of IEEE American control conference*, Alaska, 2002, paper ID ACC02-IEEE1255, pp.1693–1698. IEEE.

40. Ogata K. *Modern control engineering*. 4th ed. Prentice-Hall, Inc., 2002.