Control Engineering Practice 17 (2009) 158–169

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

System identification-based control of an unmanned autonomous wind-propelled catamaran

Gabriel Hugh Elkaim *

Computer Engineering Department, UC Santa Cruz, 1156 High Street, SOE3, Santa Cruz, CA 95064-1077, USA

ARTICLE INFO

Article history:
Received 20 August 2006
Accepted 29 May 2008
Available online 15 July 2008

Keywords:
System identification
Autonomous sailing
Wing sail
Catamaran

ABSTRACT

An autonomous catamaran, based on a modified Prindle-19 day-sailing catamaran and fitted with several sensors and actuators was built to test the viability of GPS-based system identification for precision control. Using an electric trolling motor for propulsion, and lead ballast to match all-up weight, several system identification passes were performed to excite system modes and model the dynamic response. The identification process used the Observer Kalman IDentification (OKID) method for identifying a linear time invariant plant model and associated pseudo-Kalman filter. System identification input was generated using a human pilot driving the catamaran on roughly straight line passes. A fourth order discrete time model was generated from the data, and showed excellent prediction results. Using these models, linear quadratic Gaussian (LQG) controllers were designed and tested with the electric trolling motor. These controllers demonstrated excellent line-tracking performance, with error standard deviations of less than 0.15 m. The wing-sail propulsion system was fitted, and these same controllers re-tested with the wing providing all propulsive thrust. Line-following performance and disturbance rejection were excellent, with the cross-track error standard deviations of approximately 0.30 m, in spite of wind speed variations of over 50% of nominal value.

1. Introduction

The Atlantis project set out to design an autonomous marine surface vessel. The Atlantis, pictured in Fig. 1, based on a modified Prindle-19 catamaran, uses a rigid wind sail for propulsion, along with GPS for localization and several other sensors and actuators. Functionally, the Atlantis is the marine equivalent of an unmanned aerial vehicle, and would serve similar purposes. The Atlantis has been able to demonstrate an advance in control precision of a wind-propelled marine vehicle from typical commercial autopilot accuracy of ~100 m to an accuracy of better than 1 m. This quantitative improvement enables new applications, including unmanned station-keeping for navigation or communication purposes, autonomous "dock-to-dock" capabilities, emergency return unmanned functions, and many others still to be developed.

The main goal of the Atlantis project is precise control. That is, to minimize the deviation from a desired path in the presence of disturbances such as wind, current, and waves. Precise control is achieved through the development of a predictive mathematical model. Control is predicated on having an accurate plant model which maps input to output. The controller must be able to predict the trajectory of the plant, and compensate accordingly. Unmodelled dynamics of the catamaran, combined with wind–water–wing interaction make precise control difficult. Note that even in straightforward propulsion systems, disturbances and non-linearities can be difficult to model (Smogeli, Sørensen, & Minsaas, 2008). System identification techniques are used to extract the required plant model from experimental data.

The wind-propulsion system is a rigid wing-sail mounted vertically on bearings to allow free rotation in azimuth about a stub-mast. Aerodynamic torque about the stub-mast is trimmed using a "flying tail" mounted on booms joined to the wing. This arrangement allows the wing sail to automatically attain the optimum angle to the wind, and weather vane into gusts without inducing large heeling moments. For a given set of tail and flap angles, the wing sail will fly at a constant angle of attack, α, to the relative wind (for a complete treatment of the wing-sail aerodynamics, see Elkaim & Boyce, 2007). Modern airfoil design allows for an increased lift–drag (L/D) ratio over a conventional sail, thus providing increased thrust while reducing the overturning moment.

The system architecture is based on distributed sensing and actuation, with a high-speed digital serial bus (CAN bus) connecting the various modules together. Sensors are sampled at 100 Hz, and a central guidance navigation and control (GNC) computer performs the estimation and control tasks at 5 Hz.

* Tel.: +18314593054; fax: +18314594829.
E-mail address: elkaim@soe.ucsc.edu
URL: http://www.soe.ucsc.edu/~elkaim

0967-0661/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.conengprac.2008.05.014

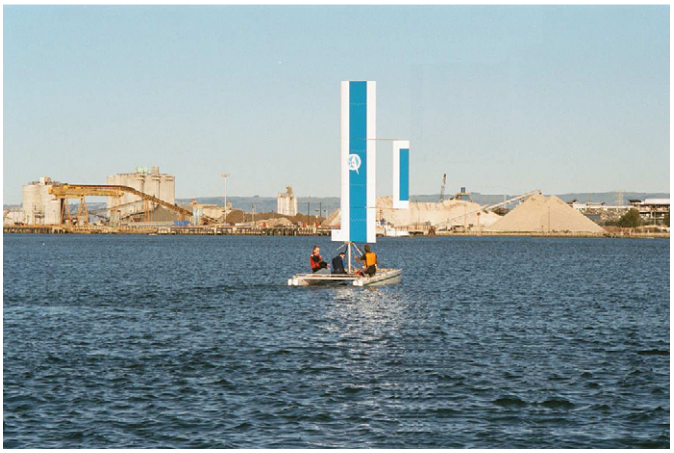

Fig. 1. The Atlantis, based on a Prindle-19 Catamaran, with a self-trimming wing sail.

This bandwidth has been demonstrated to be capable of precise control of the catamaran. The distributed architecture is both more robust and less expensive than systems that employ a high-speed, and often analog, star-configuration topology with centralized sensor interpretation and actuation. For system architecture details, see Elkaim (2001, 2006).

2. Observer/Kalman IDentification theory

There are a great many ways to approach system identification (generating a mathematical model from input/output data). This generated model is then used to predict the system response for arbitrary inputs and as a basis for control design. The system identification algorithm uses input/output data from a rich stream to produce a differential or difference equation of the system response. The models generated by the system identification techniques are most often used in feedback control systems and aid in the design of estimators and controllers.

An important advantage of the experimental approach is that does not require the tedious and often imperfect modelling of the physical system. This is especially true if questionable assumptions are made to reduce model complexity. This can also be a disadvantage as, depending on the scheme, any intuition into system behaviour is lost. Without this engineering intuition, it is difficult to view the internal system description and understand which physical parameters are being modelled. As such, any additional knowledge about physical parameter validity cannot be utilized.

Several mathematical methods have been developed for system identification. Ljung (1999) provides an excellent introduction to the subject, including the various methodologies that have been developed. Broadly, the methods split into either frequency- or time-domain-based, and tend to be grouped into single input single output (SISO) or multiple input multiple output (MIMO) solutions. Neural networks have also been used for identification purposes (van de Ven, Johansen, Sørensen, Flanagan, & Toal, 2007). Recently, a large body of state-space techniques have been aggregated into the subspace methods (Overschee & d'Moor, 1996). These techniques in general perform well, though one of their deficiencies is that the subspace methods often produce a biased state estimate. Rather than minimize the error of the output data, they minimize the model error. The minimization of the output error is non-linear, whereas the modelling error can be reduced to a simple quadratic minimization.

Another time domain technique is the Observer Kalman filter IDentification (OKID) algorithm developed at NASA Langley to model large flexible space structures. The original algorithm was developed and extended by Dr. Juang and his students to include residual whitening and several advances in the model realization algorithms (Juang, Cooper, & Wright, 1988; Phan, Horta, Juang, & Longman, 1995). The OKID algorithm minimizes the error in the observer, which will converge to the true Kalman filter for the data set used if the true world process is corrupted by zero-mean white noise.

The OKID method is utilized in this work, using only input/output data to construct a discrete-time state-space realization of the plant. The OKID identification method has several pertinent advantages for this application. It has previously been applied for identification of autonomous underwater vehicles (AUVs), see Tiano, Sutton, Lozowicki, and Naeem (2007). First, it assumes that the system in question is a discrete linear time-invariant (LTI) state-space system. Second, it requires only input and output data to formulate the model (no *a priori* knowledge of the plant is needed). Third, the OKID method produces a pseudo-Kalman state estimator, which is very useful for control applications. Lastly, the modal balanced realization of the system model means that any residual truncation errors will necessarily be small. Thus, even in the case of model order error, the results of that error will be minimized.

2.1. OKID theory

The OKID formulation begins with the standard state-space difference equation for a discrete time, LTI system:

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k$$

$$\boldsymbol{y}_k = C\boldsymbol{x}_k + D\boldsymbol{u}_k \tag{1}$$

where k is the index time variable, $\boldsymbol{x}$ is the state vector and has the dimension of $[n \times 1]$, $\boldsymbol{y}$ is the output vector and has the dimension of $[m \times 1]$, $\boldsymbol{u}$ is the input vector and has the dimension of $[r \times 1]$. The $[n \times n]$ matrix, A, is referred to as the state transition matrix, the $[n \times r]$ matrix, B, is called the input matrix, the $[m \times n]$ matrix, C, is the output matrix, and $[m \times r]$ matrix, D, is the direct feed through (or pass through) matrix. The triplet, $[A, B, C]$, is not unique, but can be transformed through any similarity transformation (i.e., the outputs are unique, but the internal states are not). This can be seen with a simple substitution:

$$\boldsymbol{x} \triangleq T\boldsymbol{z} \tag{2}$$

where $\boldsymbol{z}$ is an alternate state vector related to $\boldsymbol{x}$ through an invertible transformation matrix, T. When substituted into Eq. (1), this results in the same set of equations written as

$$\boldsymbol{z}_{k+1} = T^{-1}AT\boldsymbol{z}_k + T^{-1}B\boldsymbol{u}_k$$

$$\boldsymbol{y}_k = CT\boldsymbol{z}_k + D\boldsymbol{u}_k \tag{3}$$

Thus, the triplet $[A, B, C]$ through a similarity transformation becomes $[T^{-1}AT, T^{-1}B, CT]$. If the system is initially at rest, and a unit pulse is applied to all inputs at time zero, i.e.,

$$\boldsymbol{x}_0 = \lfloor 0 \rfloor$$

$$\boldsymbol{u}_0 = \lfloor \mathbf{1} \rfloor$$

$$\boldsymbol{u}_k = \lfloor 0 \rfloor, \quad \forall k \neq 0 \tag{4}$$

The output of the system in Eq. (1) to this unit pulse is

$$y_0 = D$$

$$y_1 = CB$$

$$y_2 = CAB$$

$$y_3 = CA^2B$$

$$\vdots$$

$$y_k = CA^{k-1}B \tag{5}$$

which are defined as the Markov parameters of the system, and are invariant under a similarity transformation. For arbitrary input, the response of the system from rest is

$$\boldsymbol{x}_k = \sum_{i=1}^{k} A^{i-1}B\boldsymbol{u}_{k-i}$$

$$\boldsymbol{y}_k = \sum_{i=1}^{k} CA^{i-1}B\boldsymbol{u}_{k-i} + D\boldsymbol{u}_k \tag{6}$$

The response reverts to the Markov parameters of the system when driven by a unit pulse from rest. These Markov parameters are assembled into the generalized Hankel matrix. Properties of the Hankel matrix are used to determine the system model order. The generalized Hankel matrix is

$$H(k-1) = \begin{bmatrix} y_k & y_{k+1} & \cdots & y_{k+\beta-1} \\ y_{k+1} & y_{k+2} & \cdots & y_{k+\beta} \\ \vdots & \vdots & \ddots & \vdots \\ y_{k+\alpha-1} & y_{k+\alpha} & \cdots & y_{k+\alpha+\beta-2} \end{bmatrix} \tag{7}$$

The Hankel matrix can be decomposed into the observability matrix, a state transition matrix, and the controllability matrix; thus the Hankel matrix (in a noise-free case) will always have rank n, where n is the system order:

$$H(k-1) = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{\alpha-1} \end{bmatrix} A^{k-1}[B \;\; AB \;\; A^2B \;\; \cdots \;\; A^{\beta-1} \;\; B] \tag{8}$$

This decomposition is at the centre of the OKID process. The system order can be determined from the rank of the generalized Hankel matrix of the Markov parameters. With real data, however, noise will corrupt the rank deficiency of the Hankel matrix and the Hankel matrix will always be full rank. Thus, the Hankel matrix is truncated using a singular value decomposition (SVD) at an order that sufficiently describes the system. In practice, the singular values of the Hankel matrix are plotted with a sudden drop that indicating the model order. This sudden decrease or "cliff" is the hallmark of the transition between real and noise modes of the system, and is often missing from real data.

This truncated Hankel matrix is then used to reconstruct the triplet $[A, B, C]$ in a balanced realization that ensures that the controllability and observability Grammians are equal. This is referred to as the Eigensystem Realization Algorithm (ERA); a modified version of this algorithm that includes data correlation is used to identify the Atlantis (Juang, 1994).

Given the Markov parameters, it is possible to determine the order of the system and generate a balanced model that is adequate for control. In real systems, however, the system pulse response cannot be obtained by simply perturbing the system with a unit pulse input. A pulse with enough power to excite all modes would likely saturate the actuators or respond in a non-linear fashion. The pulse response of the system is instead reconstructed from a continuous stream of rich system input and output data. The input/output data do not produce enough equations to solve for all of the Markov parameters. If the system were asymptotically stable, such that $A^k = 0$ for some k, then the number of unknowns would be reduced.

Taking Eq. (1) and adding zero, in the form of $(+G\boldsymbol{y}_k - G\boldsymbol{y}_k)$:

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k + G\boldsymbol{y}_k - G\boldsymbol{y}_k$$

$$\boldsymbol{y}_k = C\boldsymbol{x}_k + D\boldsymbol{u}_k \tag{9}$$

By substituting for y_k, the equations are rewritten as

$$\boldsymbol{x}_{k+1} = [A + GC]\boldsymbol{x}_k + [B + GD]\boldsymbol{u}_k - G\boldsymbol{y}_k \tag{10}$$

$$\boldsymbol{x}_{k+1} = \bar{A}\boldsymbol{x}_k + \bar{B}\boldsymbol{v}_k$$

$$\boldsymbol{y}_k = C\boldsymbol{x}_k + \bar{D}\boldsymbol{v}_k \tag{11}$$

where

$$\bar{A} = [A + GC] \tag{12}$$

$$\bar{B} = [B + GD \quad -G] \tag{13}$$

$$\bar{D} = [D \;\; 0] \tag{14}$$

and

$$\boldsymbol{v}_k = \begin{bmatrix} \boldsymbol{u}_k \\ \boldsymbol{y}_k \end{bmatrix} \tag{15}$$

Thus the output at step k can be written as

$$\boldsymbol{y}_k = D\boldsymbol{u}_k + C\bar{B}\boldsymbol{v}_{k-1} + \cdots + C\bar{A}^{p-1}\bar{B}\boldsymbol{v}_{k-p} + C\bar{A}^{p}\bar{B}\boldsymbol{x}_{k-p} \tag{16}$$

Aggregating the terms into matrix form, Eq. (16) can be rewritten as

$$\bar{y} = \bar{Y}\bar{V} + C\bar{A}^{p}X + \varepsilon \tag{17}$$

where $\bar{y}$ is the collection of output data, $\bar{Y}$ is the matrix of observer Markov parameters, $\bar{V}$ is the stack of input and output data, X is the state, and ε is the mismodelling noise:

$$\bar{y} = [y_{p+1} \;\; y_{p+2} \;\; \cdots y_l] \tag{18}$$

$$\bar{Y} = [D \;\; C\bar{B} \;\; C\bar{A}\bar{B} \;\; \cdots \;\; C\bar{A}^{p-1}\bar{B}] \tag{19}$$

$$\bar{V} = \begin{bmatrix} u_{p+1} & u_{p+2} & \cdots & u_l \\ v_p & v_{p+1} & \cdots & v_{l-1} \\ v_{p-1} & v_p & \cdots & v_{l-2} \\ \vdots & \vdots & \ddots & \vdots \\ v_1 & v_2 & \cdots & v_{l-p} \end{bmatrix} \tag{20}$$

In a noise-free case, the observer state transition matrix, $\bar{A}$, can be made deadbeat stable and $\bar{A}^p$ will be zero for time-steps greater than the system order, n. In the presence of noise, $\bar{A}$ corresponds to a pseudo-Kalman filter. In either case, it is asymptotically stable, and there will be a value of p such that the quantity $\bar{A}^p$ is negligible. p is chosen such that $C\bar{A}^pX$ is negligible, and the Markov parameters are estimated from Eq. (17) using least squares:

$$\bar{Y} = \bar{y}\bar{V}^{\mathrm{T}}[\bar{V}\bar{V}^{\mathrm{T}}]^{-1} \tag{21}$$

which minimizes the error term, $\varepsilon^{\mathrm{T}}\varepsilon$ (the difference between the output with $\bar{A}^p$ neglected and the measured values). The Markov parameters are computed and separated out using the ERA/DC algorithm (Juang, 1994; Juang et al., 1988; Phan et al., 1995).

3. Kinematic model

A plant model of the Atlantis is required for simulation and control. While several good modelling techniques exist to model a powered boat travelling through water (Fossen, 1994), they remain somewhat complicated and not easily analysed; the situation is even more complicated for sailing catamarans. Previous work (Bradfield, 1970) has developed models of hull drag and the thrust equations based on several parameters of the sails, hulls, and wind effects. These efforts produce the equations of motion suitable for light wind conditions. In order to reduce the model order and obtain a model that would have sufficient fidelity for active control, a simplified model structure is explored.

In order to formulate the equations of motion, the Atlantis is assumed to be travelling in a straight line (assumed to be coincident with the x-axis) at a constant velocity, V_x. The distance along that line is X. The perpendicular distance to the line is Y, the cross-track error, and the angle that the centreline of the Atlantis makes with the x-axis is the angular error, ψ (Fig. 2).

The local path coordinate frame can always be rotated to have the x-axis aligned to the desired path of the Atlantis. The assumption of constant velocity is not appropriate as the velocity is a function of the wind speed, which, of course, cannot be controlled and is highly variable.

The first model is a simple kinematic model that assumes that the rudders cannot move sideways through water (i.e., the flow is tangent to the rudders). This places a kinematic constraint upon the vessel motion of: $\tan\delta = L\dot{\psi}/V_x$. Using a small angle linearization produces the following continuous time state-space equations:

$$\begin{bmatrix} \dot{Y} \\ \dot{\psi} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & V_x & 0 \\ 0 & 0 & \frac{V_x}{L} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} Y \\ \psi \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{22}$$

where Y is the cross-track error (m), ψ is the azimuth error (rad), and δ is the angle of the rudders with respect to the hull centreline (rad). The distance, L, is from the boat centre of mass to the centre of pressure of the rudders (m), and the input, u, is the slew rate of the rudders (rad/s).

These simplified equations of motion are insufficient to control the boat to great precision, but generate intuition for the system identification process. Eq. (22), when cast into transfer function form, becomes a triple integrator, and as such, cannot be stabilized by simple proportional control. Closer inspection of Eq. (22) reveals that the errors in azimuth and cross-track integrate not with time, but rather with distance travelled forward. If the boat is sitting still then no amount of rudder deflection will cause the azimuth to change; likewise, at high velocity only very small inputs are required to turn through a considerable angle.

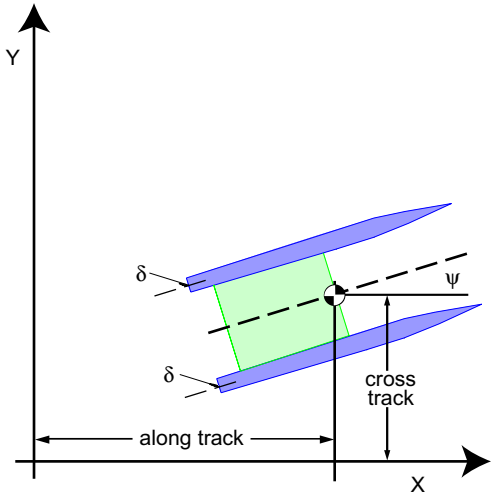

Fig. 2. The basic formulation of motion for the Atlantis on the water.

This simple kinematic model is based on prior work by the author on a GPS-guided farm tractor done at Stanford University (Elkaim, O'Connor, & Parkinson, 1997). In this work, it was shown that even with the poor model, the tractor is able to perform line-tracking tasks with very high precision. Furthermore, using an extended Kalman filter to perform parameter identification, values for the critical parameters of V_x and L can be obtained in an "on-the-fly" estimation fashion. Extensions of this work presented in Rekow (2001) show that "on-the-fly" estimation can be greatly simplified by using an extension of the least mean squares (LMS) algorithm to estimate the parameters in real-time. This technique has been validated experimentally.

Using the kinematic model, a linear quadratic regulator (LQR) controller was designed using the full state feedback of the measured state of azimuth, rudder angle, and cross-track error. Brysons rule was used to formulate the Q and R matrices, and all cross-coupling terms were assumed to be zero. That is,

$$Q = \begin{bmatrix} \frac{1}{y_{\max}^2} & 0 & 0 \\ 0 & \frac{1}{\psi_{\max}^2} & 0 \\ 0 & 0 & \frac{1}{\delta_{\max}^2} \end{bmatrix} \tag{23}$$

and

$$R = \frac{1}{u_{\max}^2} \tag{24}$$

The controller was designed such that a maximum rudder deflection rate (~25°/s) would match the maximum cross-track error of 0.1 m. That is, for this specific controller, $y_{\max} = 0.1$, $\psi_{\max} = \infty$, $\delta_{\max} = 0.44$, and $u_{\max} = 0.44$. The resulting controller designed with $L = 3.62$ m and $V_x = 2$ m/s places no penalty on heading error, but balances the rudder slew rate vs. cross-track error, while limiting the amount of rudder angle. The gain matrix, K, is

$$K = [4.36 \;\; 10.34 \;\; 3.39] \tag{25}$$

That is, the feedback control u (rad/s) is generated by multiplying $-K * \boldsymbol{x}$, where $\boldsymbol{x}$ is the state consisting of cross-track error, y (m), heading error, ψ (rad), and rudder angle, δ (rad). An entire series of these controllers in which $y_{\max}$, the maximum cross-track error, was varied from 0.01 to 1.0 m were designed and tested experimentally (Section 6 describes the control design methodology in more detail).

These controllers were tested with a trolling motor and results found to be satisfactory as long as the speed, V_x, remains below V_{design}, the design speed of the controller. As soon as $V_x > V_{design}$, the Atlantis started to hunt, eventually going unstable. When V_x was very substantially less than the design velocity, V_{design}, the response was sluggish and suffered from large (though bounded) cross-track errors. An integral state was added, but this only further destabilized the system. As the problem was not one of steady-state error, the integral control did not improve the situation.

Fig. 3 shows the root locus of the kinematic model with the full state feedback LQR controller as a function of V_x (the along-track velocity). The design velocity, V_{design}, is 1.0 m/s and the model is assumed perfect and noiseless. As the along-track velocity, V_x, is

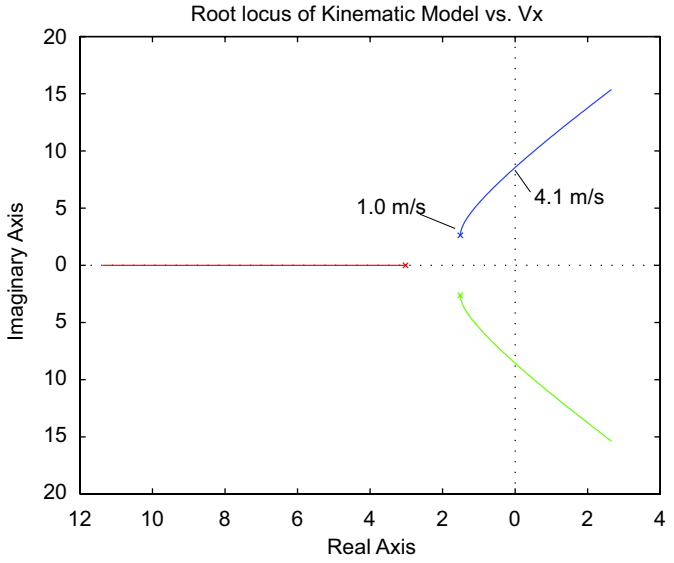

Fig. 3. Root locus plot of the closed-loop kinematic model using a fixed-velocity LQR controller as a function of the along-track velocity.

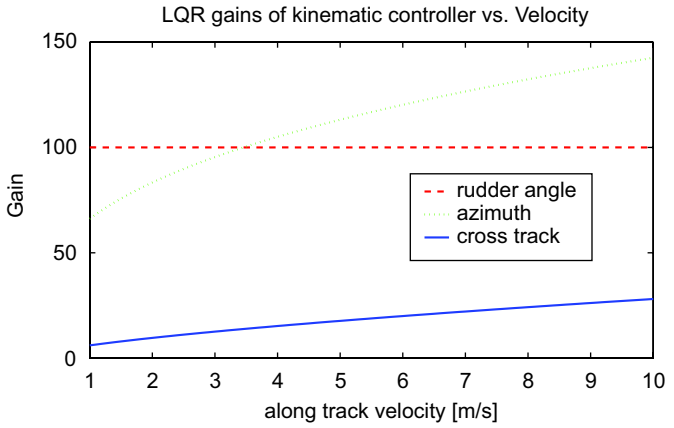

Fig. 4. LQR gains for kinematic model vs. velocity.

increased, the damping is reduced, until, at $V_x = 4.1\,\mathrm{m/s}$, the system goes unstable. This is the best case scenario—when other errors are introduced (such as mismodelling, external disturbances, sensor noise, etc.) the results are predictably worse. This is evidenced by the poor performance of the kinematic controller in all of the off design points.

4. Velocity invariance

In Section 3, it was shown that the simple kinematic model yields poor performance as soon as V_x increases above the design point, V_{design}. In order to address this shortcoming, an attempt was made to formulate a control algorithm that automatically compensates for the varying velocity. Several methods for varying the control gains as a function of velocity are possible.

Fig. 4 shows the gains for the LQR controller as a function of V_{design}. The observed gains on the rudder angle appear to be independent of velocity, but the gains on the azimuth and the cross-track error are strong functions of velocity. The gains are not simple linear or even exponential relationships to velocity, but a gain-scheduled controller could be made to interpolate the gains based on the measured velocity at the current moment.

As previously stated, the kinematic model in Eq. (22) shows that the azimuth and cross-track error do not integrate with time, but rather with distance travelled on the line. This is exactly the cause of the instability with increasing velocity present in the simple kinematic model. By introducing two new variables:

$$\tilde{Y} = \frac{Y}{V_x} \tag{26}$$

and

$$\tilde{\psi} = \frac{\psi}{V_x} \tag{27}$$

and substituting them back into Eq. (22), the equations of motion of the kinematic model can be rewritten in the following velocity invariant form

$$\begin{bmatrix} \dot{\tilde{Y}} \\ \dot{\tilde{\psi}} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & \frac{1}{L} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{Y} \\ \tilde{\psi} \\ \delta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u \tag{28}$$

The state transition matrix no longer has any terms relating to V_x. This formulation allows one single controller to be designed with the sensor input to that controller is scaled by the velocity before the control gains are applied. This has the effect of automatically reducing the sensitivity at high forward velocities and increasing the sensitivity at low velocities. In practice, a lower bound of 1.0 m/s is used on the velocity measurement, due to the presence of noise.

This velocity invariant formulation was used with great success on the GPS guided farm tractor (Elkaim et al., 1997). A similar technique was used on the Atlantis. This methodology of velocity invariance is fine for the simple kinematic model and addresses the concerns of the instability with increasing velocity. Functionally, it means that a single controller is designed, and that the inputs are scaled before injection into the controller. It does not, however, address any of the issues of mismodelling or sensor noise. In practice, with the trolling motor, the velocity invariant kinematic model performs well in calm waters but starts to oscillate badly in waves. Furthermore, the controller performance (as measured by $y_{\max}$) has to be relaxed in order for the closed-loop system to remain stable.

Of course, the kinematic model (even the velocity invariant one), completely ignores the dynamics of the vehicle and, to a certain extent, violates the laws of physics—a change in rudder angle produces an instantaneous change in the yaw rate. Other forces, such as coriolis forces, are not included. This model will get worse and worse as the velocity increases. To the control system, however, this mismodelling appears simply as another disturbance to be rejected. At low speed, this is within the capabilities of the controller, but as the velocity increases, these disturbances become larger and the control system simply cannot keep up.

Using the input pre-scaling, however, it is possible to normalize any set of input data by the velocity and thus identify the velocity invariant system. Data collected for system identification are pre-scaled by velocity (with a lower bound of 1.0 m/s) before applying the system identification techniques. This allows the best velocity invariant model to be identified by the algorithm and contributes to the robustness of the controller.

5. Atlantis system identification

In order to use the OKID method to generate the system model, high quality data are required; inputs into the system with enough

power to persistently excite the system modes along with the cleanest sensor data available. Due to the time constraints of the project, the wing was under construction when the off-line identification data were required. Without the wing, little choice was left but to use a trolling motor as the primary source of propulsion of the Atlantis, and to attempt to simulate the wing-sail propulsion system by rotating the trolling motor to point off the boat centreline, and increases the motor drive voltages. The Atlantis was ballasted with an additional 75 kg (in the form of additional batteries) equivalent to the weight of the wing. These changes were so that the collected data on the identification passes would be representative of the dynamics of wing propulsion. The Atlantis in this configuration in pictured in Fig. 5.

Fig. 5. The Atlantis on an unmanned trajectory being controlled by the identified LQG controller.

In order to gather data to perform a proper system identification of the Atlantis, a series of open-loop line-following tests were conducted in which a human driver, through the GNC computer, caused the rudders to either slew left or right at the maximum slew rate (25°/s) or to remain still in order to track a roughly straight line. This "pseudo"-random input was designed to apply the maximum power through the controls and produce a rich output that would contain information from all modes of interest.

5.1. Typical identification pass

Every attempt was made to fully exercise the dynamics of the Atlantis in order that the system identification algorithm could model the full range of expected responses. Typically, Atlantis lined up at one end of the harbour and started on a straight pass towards a marker at the other end of the harbour. During the pass, the rudders were continually slewed to port and starboard at their maximum rate. A nominally straight line was intended by the human pilot, but excursions in cross-track of several meters did occur.

Eventually, the procedure for gathering the identification data was perfected and a large number of passes were aggregated at different velocities in order to perform the final system identification. Fig. 6 shows a typical pass, with the rudder slew rate on the bottom, and the rudder angle, azimuth angle, and cross-track deviation above. This pass was over 700 m long and had azimuth excursions as high as 45° from the intended path.

5.2. OKID results

The system identification experiments resulted in several large data files from the pseudo-random input and all of the sensor

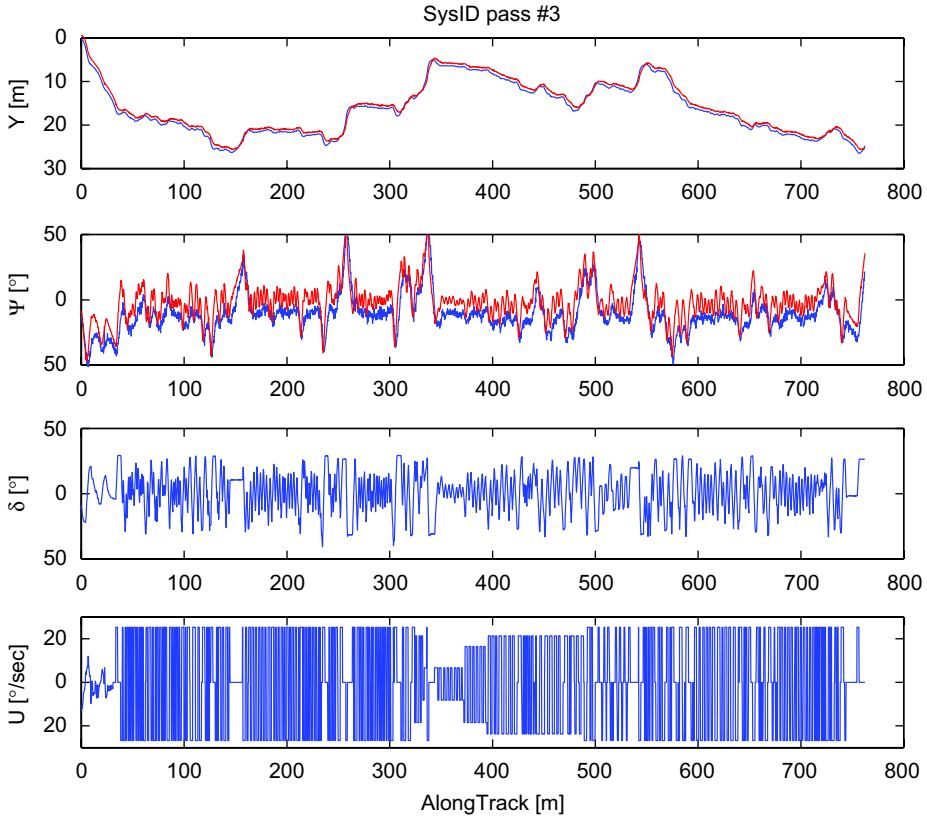

Fig. 6. Data from a typical system identification pass.

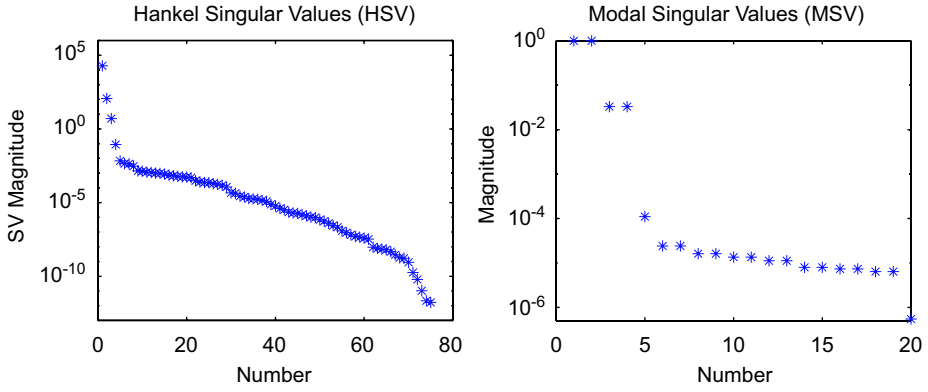

Fig. 7. Hankel singular values and modal singular values for the system identification of the Atlantis catamaran.

outputs. These files were merged together and the data normalized by the velocity as discussed in Section 4. Both the velocity-normalized and the non-normalized data were used in the identification process, but the velocity-normalized data produced much better performing controllers, and are therefore presented in this work.

The OKID method requires an upper bound on the possible system order. This is done to reduce the computation time and also to define the blocks of input and output that will be used. For the Atlantis, the maximum system order, p, was chosen to be 80. While it is conceivable that the true system order could be greater than 80 modes (indeed, most real systems have infinite modes), as an engineering approximation, 80 was deemed more than adequate.

Fig. 7 shows the Hankel singular values and the modal singular values of the OKID process on the Atlantis. On the real system, the transitions between the real modes and the noise modes are smooth. However, the first four singular values of the Hankel matrix do stand out, and this is confirmed by the modal singular values which demonstrate that even if the higher modes are used, they are virtually indistinguishable from noise.

The Atlantis input-to-output model is assumed to be a fourth order. Since the kinematic model is a third order system, some speculation is given to what the physical interpretation for the fourth mode should be. It would seem that the fourth mode is a lag term between the actuation of the rudder and the time that the Atlantis begins to rotate about her centre of mass. This remains the best guess as to the fourth state. In truth, this is difficult to decipher because the balanced realization form of the state-space model gives no physical intuition. A similarity transformation based on the pseudo-inverse of the output matrix is used to try to discern exactly what the states are in terms of the outputs. The similarity transformation results in the output matrix, C, being the identity matrix. If the number of states are equal to the number of outputs, this maps the states directly into the outputs. Unfortunately, in this case the number of states is greater than the number of outputs, and thus the information is not terribly useful.

5.3. Data reconstruction

The fourth order model performed well as shown by the matching the output not only of the cross-track error, but simultaneously of the azimuth error and of the rudder angle as well. This, of course, is because the OKID method is perfectly suited to MIMO systems as well as the traditional SISO systems. By definition, the model that it generates will match the outputs (all of them), in a least squares sense.

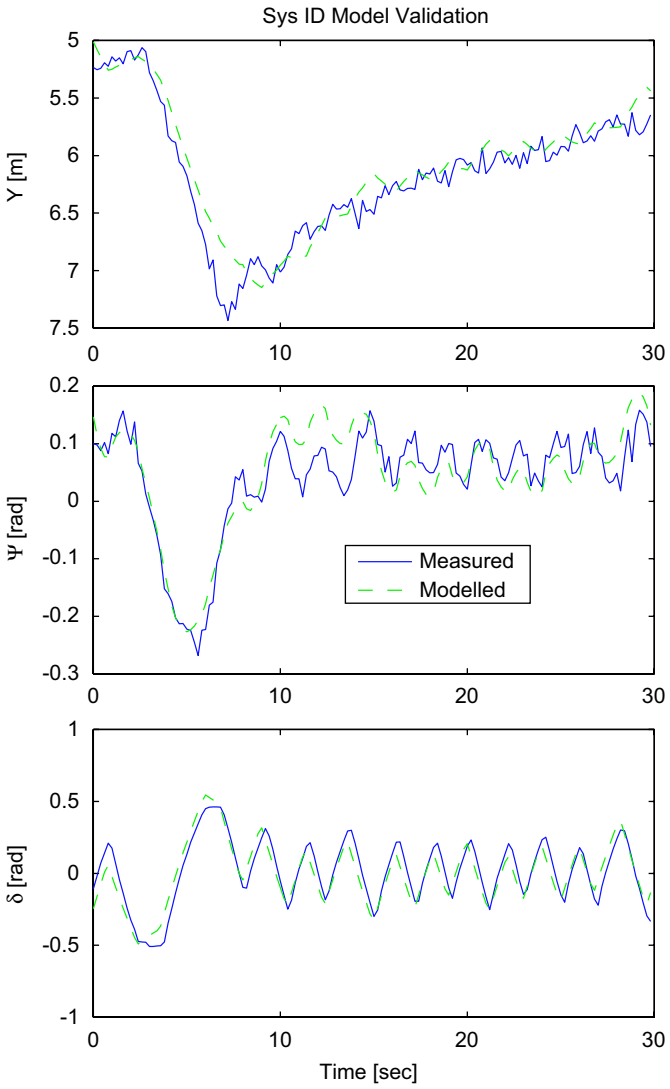

Fig. 8. The OKID data reconstruction of a system identification pass outputs from the identified model.

Fig. 8 shows that the model generated was also quite adequate at predicting Atlantis response to a known input. These particular data were from a system identification pass which had not been used in the identification process. These data were reserved to validate the OKID by tracking the output of the actual system. Perfect tracking may not, in fact, be desirable. Consider the case

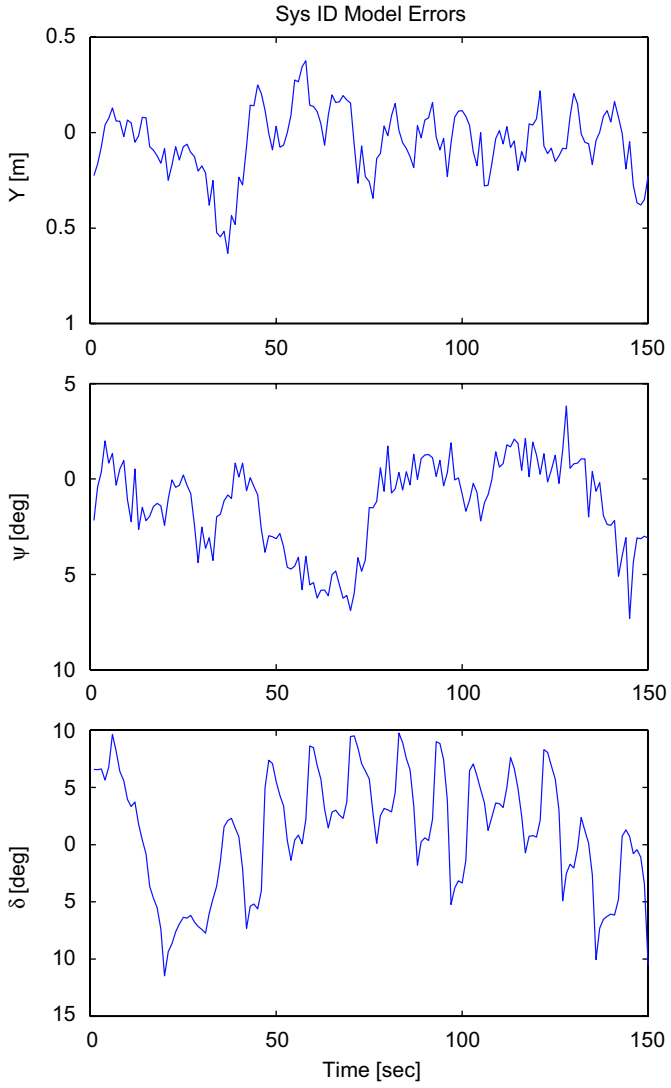

Fig. 9. OKID data reconstruction errors.

where a sensor has high frequency noise on its output—generating a model that tracks the noise perfectly is not helpful from a control standpoint. Although the high frequency sensor noise might be observable, it is certainly not controllable.

Fig. 8 shows the reconstruction of the Atlantis system identification data for approximately 30 s. The actual trace goes on much longer, but the detail is lost as the horizontal axis is stretched out. Obviously, with the addition of the observer, the agreement with the data will be much greater. This is due to the filtering sensor noise, but also because the Atlantis itself is a fairly slow moving object; the sampling rate of the control system (200 ms) is such that the motion of the catamaran during one sample interval is quite small.

Fig. 9 shows the errors from the data in Fig. 8. As can be seen, the errors are quite small. Standard deviations are ~0.1 m for the cross-track mismodelling error, 1.5° in azimuth, and 3–4° in rudder angle mismodelling error. Again, this is the open-loop model of the catamaran. These errors will become much smaller once the observer is added to the loop (since the error will then be adjusted by the data from the previous outputs).

It is interesting to note that any identified model will be the controllable and observable subset of the true mode. This is due to the rather simple fact that if it cannot be measured, then it cannot be seen by the algorithm. That is, if the sensor cannot sense (or observe) some state, then as far as the identification algorithm is concerned, it does not exist. Likewise, the same is true of controllability. If the input cannot push on a certain part or mode of the system, then the identification algorithm is also blind to that mode.

6. Atlantis controller design

The Atlantis has been identified to the extent that a controllable and observable model capable of predicting the short term behaviour of the Atlantis (while under trolling motor propulsion) was developed. In addition, the pseudo-Kalman filter that is the best state estimator/observer has also been obtained from the OKID method.

The OKID data-based observer will be used to construct a linear quadratic Gaussian (LQG) controller for the Atlantis. That is, an LQR controller is constructed using the identified model (with the assumption that there is access to the full state), and the state is then estimated from the output data using the observer generated from the OKID algorithm. There are several advantages in using the LQG control from the identified data over other methods. Firstly, the identified model is, in some least squares sense, the best mathematical representation of the system. As such, the controller design can be much more aggressive knowing that the mismodelling is minimized. Secondly, one of the main difficulties in designing Kalman filter estimators is generating appropriate process and measurement noise covariances; the OKID method already provides the Kalman gain directly. Lastly, an LQR controller can only act on the measured states, and thus forces more sensors onto the system to adequately measure all of the states.

In terms of disadvantages, the LQG controller does not have any of the guaranteed stability margins, robustness, nor simplicity that the LQR controller possesses. Thus, simulation and experimentation are required in order to develop confidence in the control implementation.

The last step in the process is to design the controller and close the loop. The control method used throughout this project is the standard LQR controller coupled with the identified state estimator that resulted from the OKID algorithm used for identification. The LQR controller is one that minimizes the cost function, J, in Eq. (29), with the terms $y_{\max}$, $\psi_{\max}$, $\delta_{\max}$, and $u_{\max}$ being design parameters. The general controller methodology is to treat this essentially as an output regulator with the cross-track error as the only output of concern (i.e., $\psi_{\max}=\infty$ and $\delta_{\max}=\infty$):

$$J=\sum_{k=0}^{\infty}\left(\boldsymbol{x}_k^{\mathrm{T}}C^{\mathrm{T}}\begin{bmatrix}\frac{1}{y_{\max}^2} & 0 & 0\\ 0 & \frac{1}{\psi_{\max}^2} & 0\\ 0 & 0 & \frac{1}{\delta_{\max}^2}\end{bmatrix}C\boldsymbol{x}_k+\boldsymbol{u}_k^{\mathrm{T}}\left[\frac{1}{u_{\max}^2}\right]\boldsymbol{u}_k\right) \tag{29}$$

The solution to the LQR minimization is a gain matrix, K, such that $\boldsymbol{u}=-K\boldsymbol{x}$. Eq. (29) uses Brysons rule to penalize the control at the same time as penalizing cross-track error, heading error, and rudder displacement. There are several excellent techniques to generate the gain matrix, K, such that J is minimized (Stengel, 1994).

6.1. Controller implementation

The implementation of identified controller and estimator is fairly straightforward. From the OKID methodology, the linear

system matrices $[A, B, C, D]$ and the estimator G are extracted. The control gain matrix, K is extracted from the LQR design (with the cost function in Eq. (29)). Abstractly, the controller is simply a mathematical function that maps (with memory), a sequence of system outputs to a set of control inputs. In this specific case, the first step is to normalize the cross-track error and heading error by the velocity (with a lower bound of 1 m/s to prevent noise amplification) as shown in Eq. (28).

Given that this implementation includes an estimator, the initial estimate of the state is set to **0** (note that there are several methods for attempting to find a better initial estimate for the state, such as taking the pseudo-inverse of the C matrix and multiplying it by the current measured outputs). The initial control, u, is also set to 0.

At any given time-step, the control is calculated from the current state estimate:

$$u = -[K]\hat{x} \tag{30}$$

and then the estimator is propagated forward one step in time given the current control and measured system outputs:

$$\hat{x}_+ = [A + GC]\hat{x}_- + [B + GD]u - [G]\tilde{y} \tag{31}$$

where $\hat{x}_+$ is the future state estimate, $\hat{x}_-$ is the previous estimate, u is the actuator command, and $\tilde{y}$ is the speed normalized sensor readings.

Note that the control portion of the propagation $[B + GD]$ could be implicity brought into the $\hat{x}_k$ term, but in this case was not due to saturation limits imposed by the actuator hardware.

Once the gains are computed, a file is generated that is uploaded to the Atlantis GNC computer. A family of controllers is designed at once and tested experimentally to determine which is the best. Fig. 10 shows the pole-zero map for the open and closed-loop kinematic model as well as the open- and closed-loop identified model (estimator poles are not shown because they are much faster than the controller).

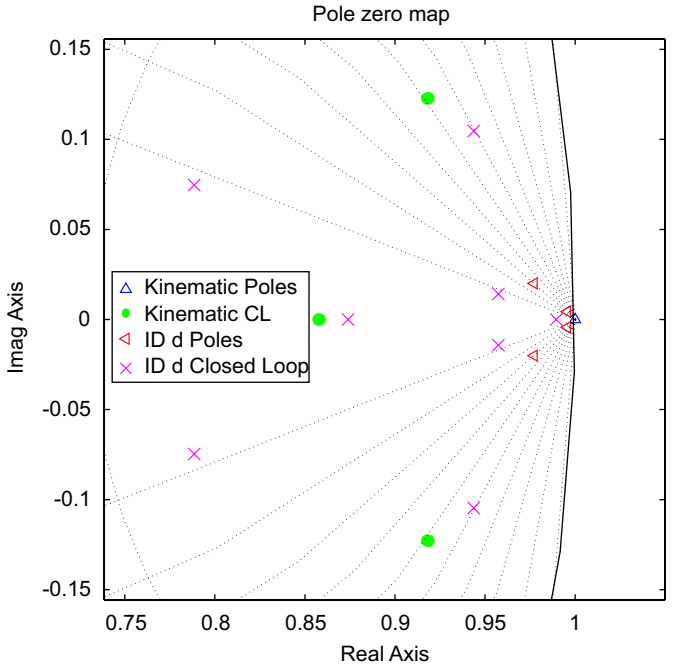

Fig. 10. Pole-zero map of the closed-loop identified system, as well as the kinematic open- and closed-loop poles.

7. Experimental results

The Atlantis controllers were validated experimentally in order to demonstrate the applicability of system identification for rapid, robust control design.

7.1. Trolling motor tests

While the wing sail was still under construction, the system identification and controller tasks had already been completed. In order to test out the controllers, a MinKota electric trolling motor was used to simulate the presence of the wing sail and wind by mounting the trolling motor at the sailboat centre of gravity (CG), and turning the trolling motor such that its direction of thrust was canted off the centreline by more than 40° (Fig. 5).

The trolling motor is at the centre of the boat, and the lead batteries provide the ballast. The boat was run unmanned, with the GNC computer providing all navigation and control functions. Of note is the fact that the anemometer is located at the front wooden cross-beam. This is only a temporary location, and moving the sensors physical location is very easy due to the CAN bus architecture employed on the Atlantis (Elkaim, 2006).

Fig. 11 shows a typical autonomous pass while under computer control. Note that the computer regulates the path to the line, but that the turn is performed open loop with a feed-forward command. To the scale pictured in Fig. 11, the recorded position data show very little cross-track error. This run includes changing currents, wind, and waves which were all injecting disturbances into the system. Fig. 12 shows a close look at the errors in the first part of the path shown in Fig. 11. The statistics show that the mean, μ, was less than 3 cm, and the standard deviation, σ, was less than 10 cm.

The azimuth shows a $-20°$ bias for most of the path length of the run pictured in Fig. 11 which is due to current. This can be verified by the velocity plot at the bottom of Fig. 11, where the top line is the hull-speed sensor, and the smooth lower line is GPS velocity. The difference in these two is current, and it can be seen in spite of the high frequency noise of the hull-speed sensor (due to the placement behind the centreboards).

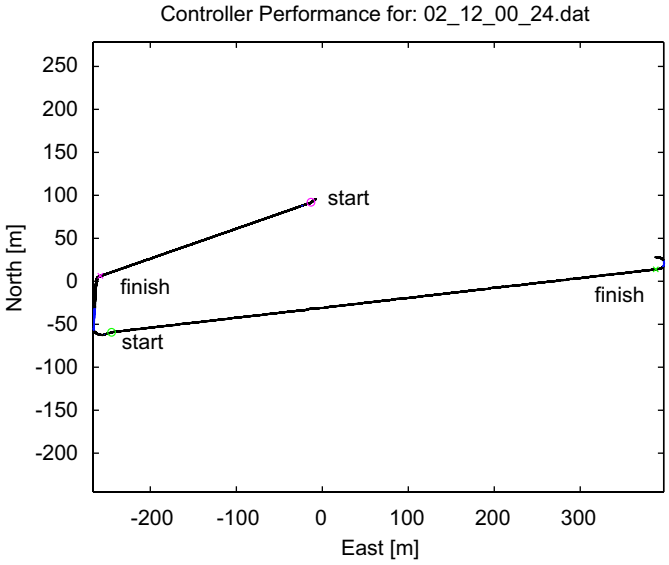

Fig. 11. The trajectory as recorded by a differential GPS receiver while under computer control.

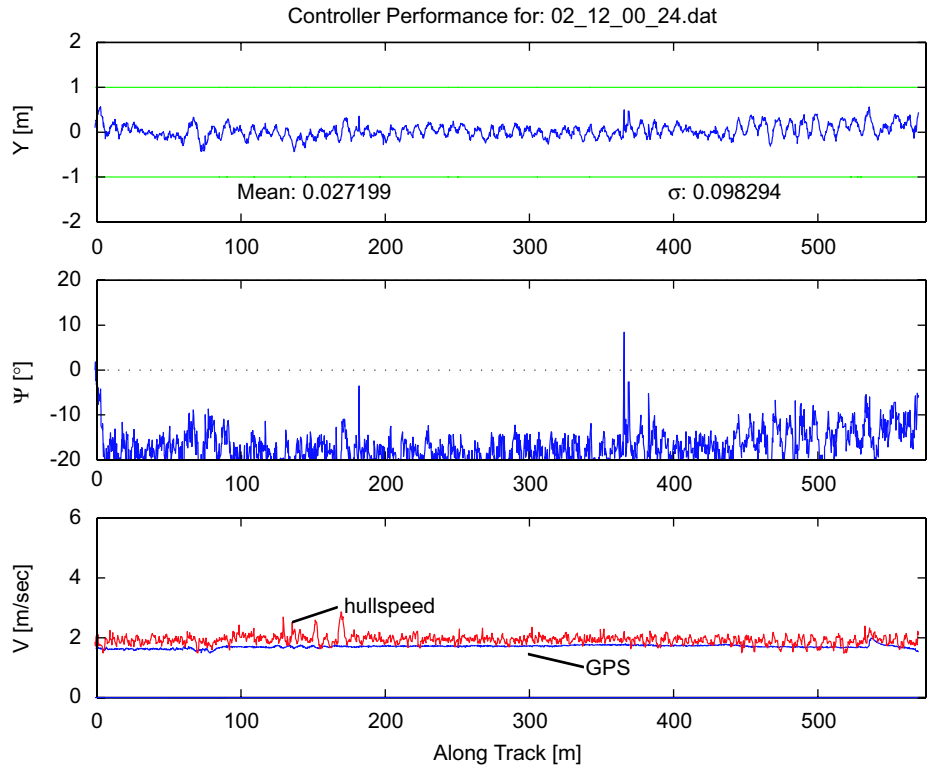

Fig. 12. Close up of the first section pictured in Fig. 11, showing very precise control while under trolling motor propulsion and autonomous control.

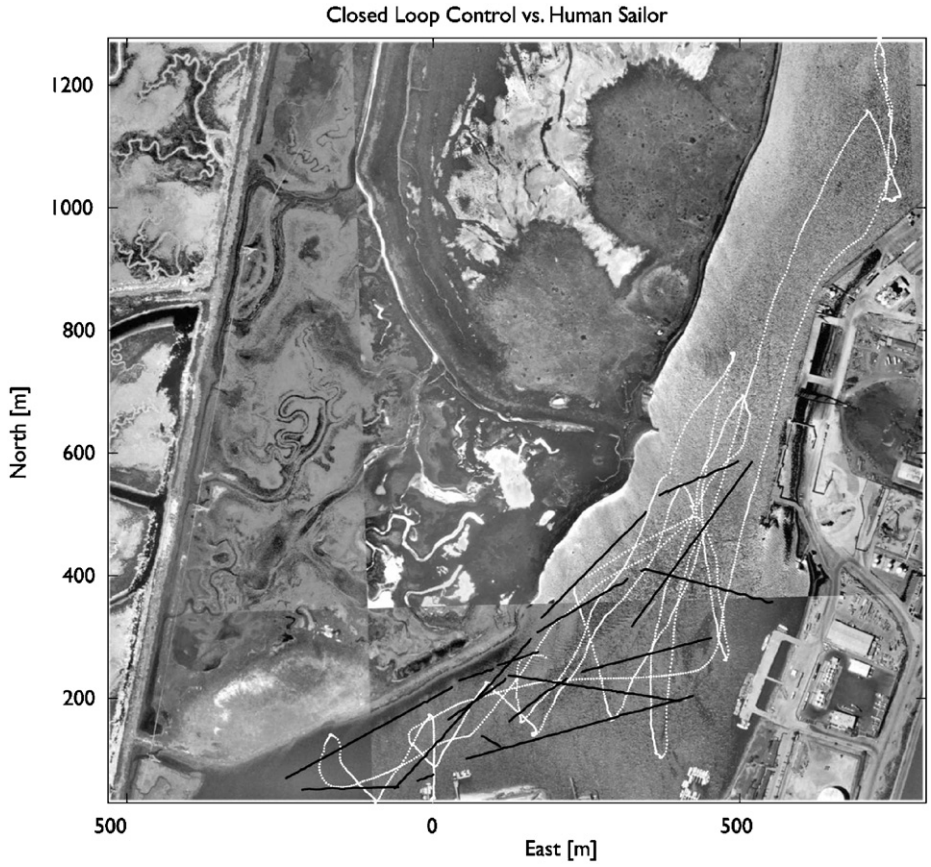

Fig. 13. Satellite photograph of harbour where the Atlantis was sailed under computer control, showing human control (white) and computer control (black).

7.2. Wing-sail tests

Further tests validated that the closed-loop controllers were robust and capable of precise line following with the increased disturbances due to the wing-sail propulsion. No modifications were made to the controllers, and the tests were run on a day with 6 m/s of wind (with gusts up to 10 m/s range).

Qualitatively, the wing-sail performed even better than anticipated. With the tail centred, there was no tendency for the Atlantis to heel what-so-ever, and the absence of aeroelastic instability (sail luffing) made the entire event quiet and pleasant. Upon turning the trailing edge of the tail in the direction of desired travel, the Atlantis smoothly accelerated to speed and quietly continued on her course. Large gusts simply caused Atlantis' wing to quickly stall and, with only a slight shudder, reposition at the new angle of attack (as evidenced by the yarn tufts on the wing surface).

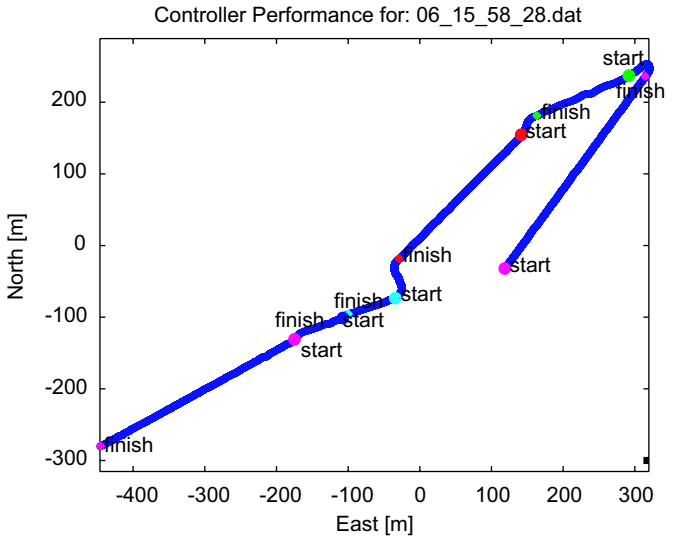

Fig. 14. Automatic control under wing-sail propulsion; control system is active between "start" and "end" labels.

Fig. 13 shows a satellite picture of the harbour where testing was performed. The white dots are from a previous year, when the Atlantis was conventionally sailed with a sloop rig, and was sailed by a human pilot. The black dots indicate the various closed-loop control passes from the recent tests. Note that the white trace has a curving, "human," look to it, whereas the black trace looks like machine control. Qualitatively, the computer control simply looks unnatural.

Fig. 14 is, once again, a closer look at an overhead view of a set of computer controlled paths. As with the trolling motor, the control system is active on the lines in between each "start" and "end" pair, while the turns were performed open loop. Fig. 15 presents a close-up of the first path of the regulated control, and looks at the cross-track error, azimuth error, and velocities. Note that the dark line in the top of the velocity graph is the wind speed, and can be seen to vary well over 50% of nominal.

The mean of the cross-track error is less than 3 cm, and the standard deviation is less than 30 cm, note that this is the sailboat technical error (STE, the sailing analog of flight technical error, that is the difference between the measured position and the reference position). Previous characterization of the coast-guard differential GPS receiver indicated that the navigation sensor error (NSE) is approximately 36 cm, thus the total system error (TSE) is less than 1 m (Elkaim, 2001).

Fig. 16 presents the aggregate of all controlled sailing runs overlaid, along with bounds indicating ±1 m. The differences in path length have to do with the location of the shore, and the desire not to run aground. Depending on the path chosen, longer or shorter distances were traversed. The control system performance always remains within the 1 m bound. As a basis for

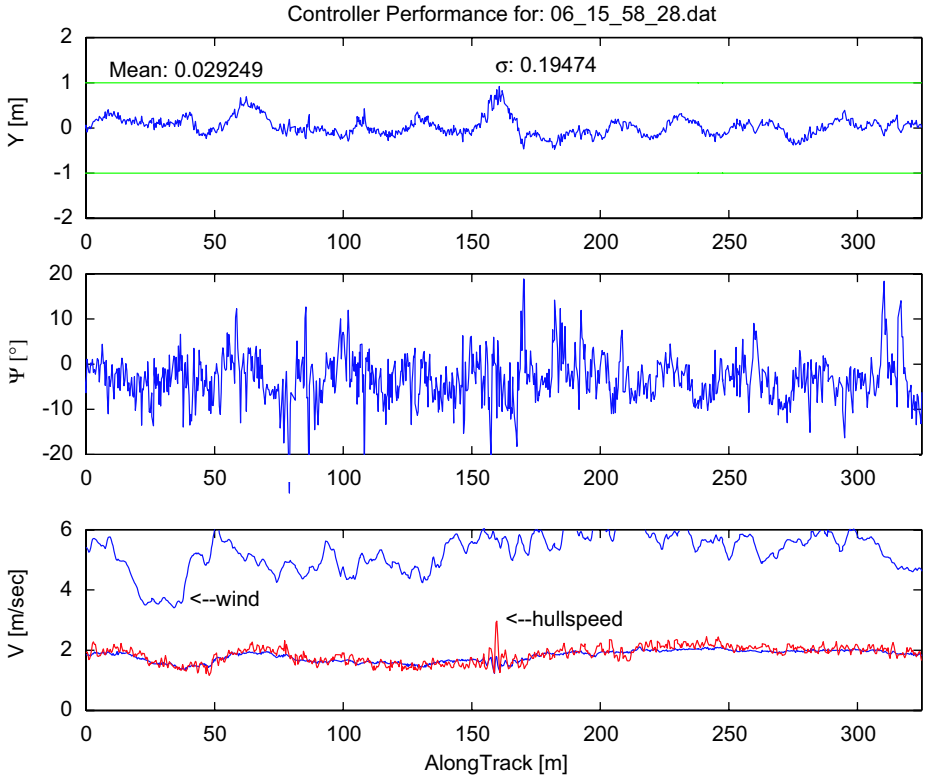

Fig. 15. Close up of one of the control segments displayed in Fig. 14, showing a mean of less than 3 cm, and a standard deviation of less than 20 cm.

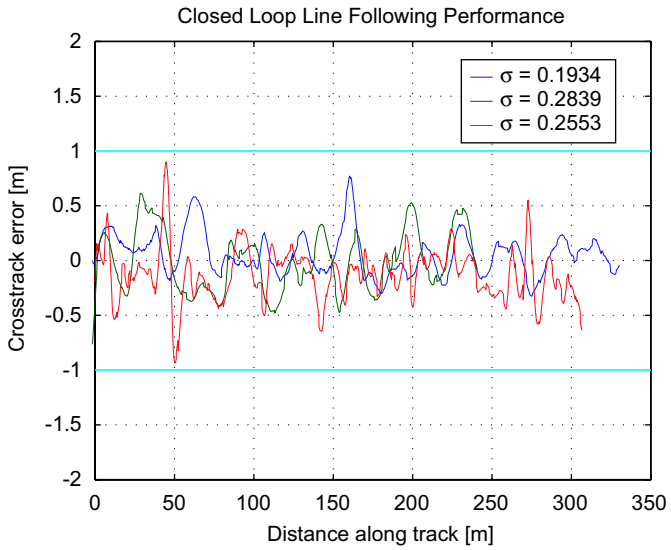

Fig. 16. Aggregate plot of computer controlled sailing passes, with lines at ±1 m bounds, overlaid on top of one another.

comparison, the specifications for the top-of-the-line AutoHelm autopilot indicate a cross-track accuracy of 0.05 nautical miles, or 92.6 m.

8. Conclusions

It has been demonstrated that with the combined advances in GPS technology, and the advent of low-cost sensors, an unmanned sailboat can be built that can navigate with unprecedented levels of accuracy. By utilizing a novel wing-sail propulsion system, the difficulties of actuating a sail have been overcome, and high authority control can be realized. With the trolling motor providing thrust, the identification-based controllers showed very good performance, with a mean of less than 3 cm, and a standard deviation in cross track error of approximately 10 cm. With the wing-sail propulsion, the identified controllers demonstrated sailboat technical error (STE) in line following less than 0.3 m was achieved, in challenging conditions. Combined with a navigation sensor error (NSE) of 0.36 m, this yields a total system error (TSE) of less than 1 m.

Within the system identification and control design issues exists a rather large caveat: the assumption that the Atlantis powered by the trolling motor is sufficiently similar to the Atlantis under wing-sail propulsion such that a control system designed for one will work well with the other.

There was, in fact, little basis for this assumption.

There simply was not enough time to wait for the wing to be finished before designing the control system and performing the open-loop system identification tasks. Nor was there enough time or funds for the project to delay while the very last bit of performance could be eked out of the system through the exploration of other control topologies. Instead, it was thought that the best strategy would be one that got the Atlantis navigating on its own as soon as possible, even without knowing if there might be more sophisticated control design techniques. In addition, the trolling configuration was deemed much less probable to capsize the catamaran or injure the crew, resulting in a lower risk for development and testing. At a minimum, the non-control engineering issues (such as sign errors in the code that performed control) could be discovered without destroying expensive and irreplaceable hardware.

But most importantly, the assumption proved valid and the controller design indeed worked quite well.

The dynamics of the wing sail are so benign that the catamaran hardly heeled. Thus the dynamics were quite well matched from the trolling motor to the sailing tasks. Interestingly enough, the trolling motor was never able to bring the Atlantis up to the speeds at which she sailed. What is remarkable about this is that the identified model/estimator/regulator combination proved sufficiently robust that it performed very well completely outside the range of parameters over which it had been designed.

Acknowledgements

The work in this paper was the result of research sponsored by University of California, Santa Cruz. The initial part of this research was conducted under a birdseed grant from the Stanford Office of Technology Licensing (OTL). The author gratefully acknowledges both the OTL and the Principal Investigator (and Thesis Advisor), Dr. Bradford Parkinson, of Stanford University for support and sharing the data.

References

Bradfield, W. (1970). Predicted and measured performance of a daysailing catamaran. *Marine Technology*, 21–37.

Elkaim, G. (2006). The Atlantis project: A GPS-guided wing-sailed autonomous catamaran. *Journal of the Institute of Navigation*, *53*(4), 237–247.

Elkaim, G. H. (2001). *System identification for precision control of a wingsailed GPS-guided catamaran*. Ph.D. thesis, Stanford University, Stanford, CA, December. URL: ⟨http://www.soe.ucsc.edu/~elkaim/Documents/GabrielElkaimThesis01.pdf⟩.

Elkaim, G. H., & Boyce, C. O. (2007). Experimental aerodynamic performance of a self-trimming wing-sail for autonomous surface vehicles. In *Proceedings of the IFAC conference on control applications in marine systems. IFAC CAMS*, Bol, Croatia, September.

Elkaim, G. H., O'Connor, M. L., & Parkinson, W. B. (1997). System identification and robust control of a GPS-guided farm tractor. In *Proceedings of the institute of navigation ION-GPS conference. ION* (pp. 1415–1426).

Fossen, T. I. (1994). *Guidance and control of ocean vehicles*. New York, NY: Wiley.

Juang, J.-N. (1994). *Applied system identification*. Englewood Cliffs, NJ: Prentice-Hall.

Juang, J.-N., Cooper, J. E., & Wright, J. R. (1988). An eigensystem realization algorithm using data correlations (ERA/DC) for modal parameter identification. *Control-Theory and Advanced Technology*, *4*(1).

Ljung, L. (1999). *System identification: Theory for the user* (2nd ed.). Englewood Cliffs, NJ: Prentice-Hall.

Overschee, P., & d'Moor, B. (1996). *Subspace identification for linear systems: Theory, implementation, applications*. Boston, MA: Kluwer Academic Publishers.

Phan, M., Horta, L. G., Juang, J.-N., & Longman, R. W. (1995). Improvement of observer/Kalman filter identification (OKID) by residual whitening. *Journal of Vibration and Acoustics*, *117*(2).

Rekow, A. K. (2001). *System identification, adaptive control and formation driving of farm tractors*. Ph.D. thesis, Stanford University, Stanford, CA.

Smogeli, Ø. N., Sørensen, A. J., & Minsaas, K. J. (2008). The concept of anti-spin thruster control. *Control Engineering Practice*, *16*(4), 465–481.

Stengel, R. F. (1994). *Optimal control and estimation* (pp. 299 – 419). New York, NY: Dover.

Tiano, A., Sutton, R., Lozowicki, A., & Naeem, W. (2007). Observer Kalman filter identification of an autonomous underwater vehicle. *Control Engineering Practice*, *15*(6).

van de Ven, P. W., Johansen, T. A., Sørensen, A. J., Flanagan, C., & Toal, D. (2007). Neural network augmented identification of underwater vehicle models. *Control Engineering Practice*, *15*(6), 715–725.