

A comparative study of control techniques for an underwater flight vehicle

R. K. LEA†, R. ALLEN†‡* and S. L. MERRY§

Unmanned, underwater vehicles have been developed considerably in recent years. Remotely operated vehicles (ROVs) are increasingly used for routine inspection and maintenance tasks but have a range that is limited by the umbilical cable. For long range operations, such as oceanographic exploration and surveying, autonomous underwater vehicles (AUVs) are emerging which have on-board power and are equipped with advanced control capabilities to carry out tasks with the minimum of human intervention. AUVs typically resemble torpedoes in that most have control surfaces and a single propulsion unit, and must move forwards to manoeuvre. Such vehicles are called flight vehicles. This paper describes techniques which are candidates for control of a flight AUV and identifies controllers used on some existing vehicles. Since underwater vehicle dynamics are nonlinear, fuzzy logic and sliding mode control were felt to have promise for autopilot application due to their potential robustness. Following development using a comprehensive simulation programme, the controllers were tested using the experimental vehicle, Subzero II, and their performance compared with that of a classical linear controller. The relative merits of the methods for practical implementation are discussed.

1. Introduction

Remotely operated vehicles (ROVs) are teleoperated robots that are used underwater, mostly for installation, inspection and repair tasks. They have advantages over human divers in that they can descend to greater depths, can stay there for greater lengths of time and require less support equipment. Thus they can reach places divers cannot, and they can be less expensive to operate. The ROV is linked to a surface ship (or other operating platform) by an umbilical cable which carries power and provides a communications link. There is, however, an immediate problem associated with this arrangement: the drag on the umbilical cable is a limiting factor on the range of the ROV i.e. how far away from the support ship it can travel.

Autonomous underwater vehicles (AUVs) do not suffer from these problems. The AUV concept is of a self-contained vehicle with enough on-board power and 'intelligence' that it can carry out tasks with minimal human intervention. Being self-contained brings a number of advantages: it has no umbilical cable to limit its range, or to become entangled in a surrounding structure and it can undertake missions that would be impractical or impossible with an ROV, such as long range collection of oceanographic data and under-ice surveying (Ferguson and Pope 1995).

Whereas ROVs tend to be box-type structures with bluff bodies, AUVs are normally much more streamlined, varying from standard torpedo shapes such as the NERC's *Autosub DTV* (Collar and McPhail 1995) to 'flatfish' types such as the NPS's *AUV II* (Healey and Lienard 1993) to low-drag vehicles such as the *NDRE-AUV* (Jalving 1994). There is an additional difference in that most ROVs can hover and manoeuvre around an operating point whereas most AUVs cannot. A typical AUV is like a torpedo or submarine in that it has a propulsion unit and control surfaces rather than a multitude of thrusters and thus must be

Accepted 3 August 1998.

* Author for correspondence.

† Institute of Sound and Vibration Research, University of Southampton, Southampton SO17 1BJ, UK.

‡ Mechanical Engineering, University of Southampton, Southampton SO17 1BJ, UK.

§ DERA Haslar.

moving forwards in order to manoeuvre. Such vehicles are known as *flight vehicles*, and are the subject of this paper.

Control of such a vehicle is a complex problem, ranging from navigation and task planning to low-level autopilot activity: it is the latter that is of interest here. This paper identifies the controllers that are used on certain existing flight vehicles. Three of these controllers were tested experimentally on the vehicle *Subzero II*; both the vehicle and the controllers are

described in detail before results for speed control are presented and discussed.

2. Control overview

Recent work concerning control techniques for underwater flight vehicles is summarized in table 1. A brief commentary follows.

Jalving (1994) has used PID control for the *NDRE-AUV* which has a low drag hull; control of the vehicle

Table 1. Summary of research into underwater flight vehicles

| Paper | Group | Year | Variables Controlled | | | | | | | | | | | | Type | | Control | Vehicle | |
|----------------------------------|-----------------|------|----------------------|-----|-----|--------|----------|--------|------------|-----|-----|-----|-----|-----|------|-----|--|----------------------|-------------------|
| | | | Global | | | | | | Body-fixed | | | | | | Sim | Exp | | | |
| | | | x | y | z | ϕ | θ | ψ | u | v | w | p | q | r | | | | | |
| Rodríguez and Dobeck | US Navy | 1989 | ✓ | ✓ | | ✓ | ✓ | | | | | | | | | ✓ | Gain scheduling | <i>LSV</i> | |
| Cristi, Papoulios and Healey | NPS | 1990 | | | ✓ | | | | | | | | | | | ✓ | Adaptive sliding | NPS <i>AUV II</i> | |
| Dougherty and Woolweaver | Martin Marietta | 1990 | | | ✓ | ✓ | ✓ | ✓ | ✓ | | | | | | | ✓ | Sliding mode | <i>MUST</i> | |
| Healey and Marco | NPS | 1992 | ✓ | ✓ | ✓ | | | ✓ | ✓ | | | | | | | ✓ | Classical, SMC | NPS <i>AUV II</i> | |
| Venugopal, Sudhakar and Pandya | FAU | 1992 | | | ✓ | | ✓ | ✓ | | | | | | | | ✓ | Neural network | <i>Ocean Voyager</i> | |
| Healey and Lienard | NPS | 1993 | ✓ | ✓ | ✓ | | | ✓ | ✓ | | | | | | | ✓ | Sliding mode | NPS <i>AUV II</i> | |
| Fryxell <i>et al.</i> | Lisbon | 1994 | ✓ | ✓ | ✓ | | | | | | | | | | | ✓ | Gain scheduled H_∞ | <i>MARIUS</i> | |
| Hills and Yoerger | US Navy | 1994 | | | ✓ | | ✓ | ✓ | | | | | | | | ✓ | Sliding mode | <i>LDUUV</i> | |
| Jalving | Norway | 1994 | | | ✓ | | ✓ | ✓ | ✓ | | | | | | | ✓ | PID | <i>NDRE-AUV</i> | |
| Smith, Rae, Anderson and Shein | FAU | 1994 | | | ✓ | | ✓ | ✓ | | | | | | | | ✓ | Fuzzy logic | Torpedo | |
| Xu and Smith | FAU | 1994 | | | ✓ | | | | | | | | | | | ✓ | Fuzzy logic | Torpedo | |
| DiBitetto | CSDL | 1995 | | | ✓ | | | | | | | | | | | ✓ | Fuzzy logic | ARPA <i>UUV</i> | |
| Liceaga-Castro and van der Molen | Strathclyde | 1995 | | | ✓ | | | | | | | | | | | ✓ | Gain scheduling | Submarine | |
| Liceaga-Castro and van der Molen | Strathclyde | 1995 | | | ✓ | | | | | | | | | | | ✓ | H_∞ | Submarine | |
| Lea | Southampton | 1997 | | | | | | ✓ | ✓ | | | | | | | ✓ | ✓ | PID | <i>Subzero II</i> |
| Lea, Allen and Merry | Southampton | 1997 | | | | | | | ✓ | | | | | | | ✓ | Gain scheduling, fuzzy logic, Sliding mode | <i>Autosub</i> | |
| Silvestre, Pascoal and Healey | Lisbon/NPS | 1997 | | | ✓ | | | | ✓ | | | | | | | ✓ | H_∞ | NPS <i>AUV II</i> | |
| Suto and Ura | Tokyo | 1997 | | | ✓ | | | | | | | | | | | ✓ | Neural network | <i>Manta-Ceresia</i> | |

x, y, z —position; ϕ —roll; θ —pitch; ψ —heading; u —surge, v —sway, w —heave, p —roll, q —pitch, r —yaw velocities.

was successful over a 4 hour sea trial. Although PID is a linear control method, successful control was to be expected as the vehicle was operating at a constant speed throughout. (Vehicle dynamics are typically non-linear with regard to speed.) In an earlier paper, Rodríguez and Dobeck (1989) described the US Navy's *Large Scale Vehicle* which used three sets of PID controllers scheduled with respect to the vehicle's speed. Although the details given of the controller and the experimental tests carried out were not exhaustive, the results presented suggested that the vehicle exhibited tight tracking of heading and depth.

Fryxell *et al.* (1994) used an H_∞ controller to control the AUV *MARIUS*. Although only brief simulation results are given, it exhibited good tracking of position and depth. However, three H_∞ controllers were used, each gain scheduled for a different vehicle speed, and no comparison was made between this, and a simpler controller that could have produced similar results. In contrast, Liceaga-Castro and van der Molen (1995b) investigated H_∞ for submarine depth control, as well as presenting a classical controller designed using frequency-domain techniques for the same control problem (Liceaga-Castro and van der Molen 1995a). Their conclusion was that the H_∞ controller achieved a similar performance to the classically designed controller.

In the area of adaptive control, Farrell and Clauberg (1993) have reported successful control of the *Sea Squirt* vehicle which used an extended Kalman filter as a parameter estimator with pole placement to design the controller. However, although the algorithms behind an adaptive controller are straightforward, the practical implementations are not—the whole of their paper is devoted to the obstacles and solutions involved in implementing the controller on the actual vehicle. Cristi *et al.* (1990) presented simulation results for the implementation of an adaptive sliding mode controller on a Swimmer Delivery Vehicle. They linearized the vehicle model around a nominal speed so that state-space methods could be used for control; the adaptive part was used to update the model as the vehicle's speed changed. However, it would appear that adaption was not necessary as, in a later paper, Healey and Lienard (1993) described the sliding mode controller for their NPS *AUV II* vehicle which used fixed-gain state-feedback (i.e. without adaption) but still produced satisfactory results over a range of operating speeds.

Fuzzy logic for depth control appears popular as both Xu and Smith (1994), and DeBitetto (1995) describe systems for control of the Florida Atlantic University's (FAU's) *Ocean Voyager* and ARPA's *UUV* respectively. However, both cases involve the use of ballast tanks rather than flight methods for depth control. FAU does, however, use fuzzy logic for flight control of their *Ocean Voyager* vehicle as reported by Smith *et al.*

(1994). The system is quite complex in that heading, pitch and depth are all controlled, with three controllers for each subsystem, each controller being designed to operate over a given speed range. In addition, the fin commands are functions of the input errors and rates and thus the controller operates as an elaborate gain-scheduling system. Good results from simulations are reported in the paper.

Venugopal *et al.* (1992) described another neural network for control of FAU's *Ocean Voyager* vehicle, which was able to learn online. Although the controller was indeed able to cope with varying vehicle dynamics, its responses were very slow—the results for the pitch controller indicate that it took around three minutes to alter the vehicle's pitch by some 20°; the yaw response was of similar speed and also suffered from low frequency oscillations.

From the table, two points become apparent: firstly, there does not appear to be a favourite type of control method. Secondly, many control methods have only been tested on a simulation—there is much less information on the results of in-water trials. In addition, few papers actually compare controllers with each other, Seube (1994) being a notable exception comparing as he does direct adaptive control, a neural network and sliding mode. In a related field Parsons *et al.* (1995) present a fuzzy logic controller for ship path tracking against an optimal controller, but researchers in the underwater field seem either to compare against a fixed linear controller, or nothing at all.

The non-adaptive controllers that are used include PID, gain-scheduled linear controllers, fuzzy logic, sliding mode and H_∞ . However, until very recently only the paper by Liceaga-Castro and van der Molen (1995b) addressed H_∞ control for an underwater vehicle. In their paper on submarine control the conclusion was that 'a comparison with designs obtained using classical methods would show that the H_∞ controllers may achieve similar performance'. Given that the H_∞ design procedure is significantly more complex than for classical controllers, its lack of a performance advantage puts it at a disadvantage. Of the other four, gain-scheduling would not be necessary if a classical controller could perform adequately. Thus, the obvious control candidates for initial investigation are classical controllers, fuzzy logic and sliding mode.

3. Modelling

Motion analysis of an underwater vehicle involves six degrees of freedom (DOF), as six independent coordinates are required to determine the position and orientation of a rigid body in three dimensions. The first three coordinates and their time derivatives represent the translational position and motion of the body while

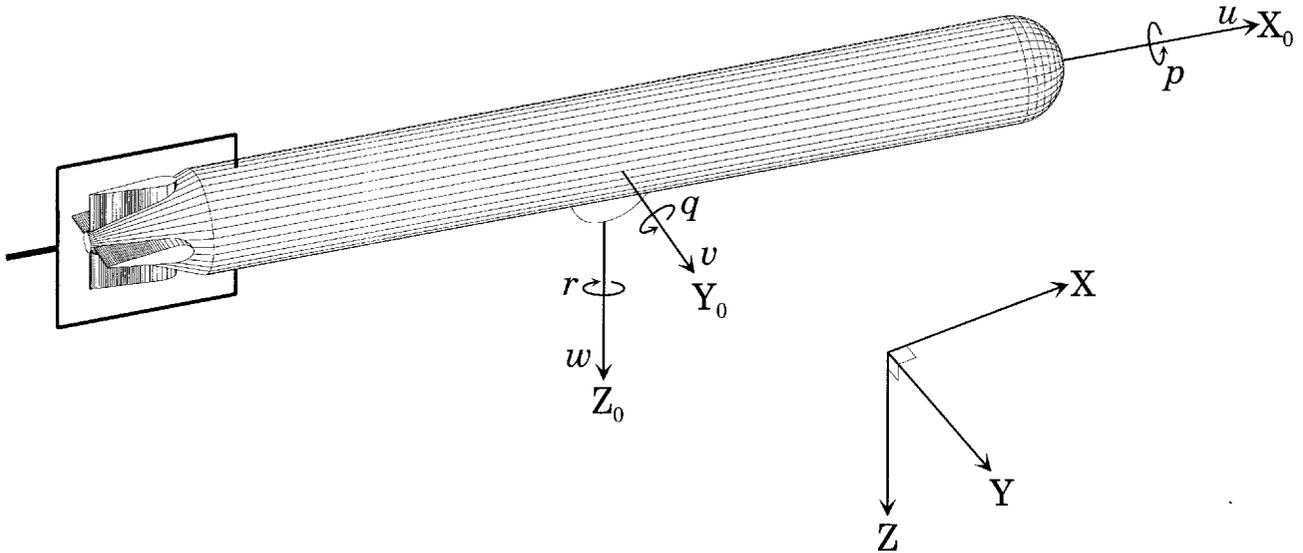


Figure 1. Coordinate systems.

the last three describe the rotational position and motion.

When considering a dynamic model of an underwater vehicle, it is convenient to use two coordinate systems: a global (earth-fixed) coordinate system or frame XYZ and a body-fixed system $X_0 Y_0 Z_0$ as shown in figure 1. In terms of vehicle position and motion, the earth-fixed system is the frame of interest whereas the equations describing the vehicle's behaviour are more easily developed in the body-fixed system.

The Euler angle transformation was used to map between the two coordinate systems. This method is routine and has been described by many authors (e.g. Fossen 1995 or Lea 1997). However, there is a problem with this representation as there exists a singularity at a pitch angle of $\theta = \pm 90^\circ$. Since this represents the vehicle pointing straight up or straight down, and whilst this is a possible manoeuvre, it is envisaged that equipment limitations will make this extremely unlikely (for instance the global pitch sensor on the *Subzero II* vehicle (section 4) has a range of $\pm 50^\circ$). By way of comparison with other flight vehicles, the specification for the *Autosub* vehicle (Collar and McPhail 1995) states that the maximum pitch angle will be $\pm 45^\circ$; the *NDRE-AUV* had an operational pitch angle of $\pm 20^\circ$ during descent and ascent (Jalving 1994).

Other transformation methods do exist; the main alternative is Euler parameters, as described by Fjellstad and Fossen (1994). As Fossen (1995) states, Euler parameters do not suffer from the singularity problem and they are also less computationally intensive than Euler angles. However, Fjellstad and Fossen (1994) show that they are of benefit only when the pitch angle is above 88.5° . They are also less intuitive, and Euler

angles have another advantage in that they can easily be decomposed into subsystems for control.

It is common practice to simplify vehicle control by decoupling the six degree of freedom system into four subsystems comprising speed, roll, horizontal plane motions, and vertical plane motions. The unmodelled coupling effects and interactions are then treated as disturbances. This arrangement has been explicitly used on the *NPS AUV II* as reported by Healey and Lienard (1993) and on the *NDRE-AUV* as reported by Jalving (1994). Smith *et al.* (1993) used separate subsystems for speed, heading, pitch and depth of the *FAU Ocean Voyager*.

With regard to speed control, this simplification produces a 1 degree of freedom subsystem with a single state variable, u , and a single input, m_d , the motor command.

4. Experimental vehicle

The Southampton University *Subzero II* vehicle was used to test experimentally the three controllers mentioned at the end of section 2. This evolved from an earlier vehicle, *Subzero*, but considerable redesign and enhancement has provided much great opportunities for controller testing (Lea *et al.* 1996).

The new vehicle is a self-powered, remotely-operated vehicle that is controlled by an IBM-compatible PC on the shore. Torpedo-shaped, it is driven by a single propeller and guided by four control surfaces. The vehicle is shown in figure 2; visible in the background is the Ocean Basin testing facility at DERA Haslar where the experimental tests were carried out.



Figure 2. The *Subzero II* vehicle.

The original vehicle specification was for a 1 m long vehicle with a maximum diameter of 10 cm. The payload carried would be a maximum of 3 kg with a target speed range of 0 to 8 knots (approximately 0 to 4 m/s) and a depth capability of 6 m. The duration of a mission was to be 15 min. The vehicle has a cylindrical hull, is made from Perspex, and has removable nose and tail sections. For ease of access, which is important in such a test-bed, the drive and control gear are mounted on a removable tray inside the centre section.

Propulsion is from a 200 W, 16 000 rpm, samarium-cobalt DC motor, powered by a 9.6 V Ni-Cad battery pack. The supply is controlled by power MOSFETs in an H-bridge chopper arrangement to allow forward and reverse action. The propeller diameter is 10 cm and has a pitch ratio of 1.0 and a blade area ratio (BAR) of 0.12. The current design cruising speed is 1.3 m/s.

The control surfaces, a rudder and two independent sternplanes, are actuated by model aircraft servo's. Roll is currently passively stable, as the heavy components such as the battery pack and the motor are mounted as low in the vehicle as possible. This has proved sufficient so far, although the roll mode may be actively controlled by means of the independent sternplanes.

Vehicle control is achieved using a host PC on the shore which communicates with the ROV over a bi-

directional link. To reduce the load on the PC, communications are handled by two Motorola 68HC11 8-bit microcontrollers (MCUs) running at 2 MHz: one on board the ROV, the other operating within the PC on a custom-built communications card. The vehicle MCU collects sensor data such as propeller speed and depth before transmitting this to the PC. The host uses this information together with pilot demands to control the vehicle. The control signals are transmitted to the ROV MCU which adjusts the pulse width modulated signals to the motor drive and the control surface actuators. Sensor limitations reduce the ROV PC communications update rate to 10 Hz although the actual data transmission rate is much higher.

The physical data link between the ROV and the PC is either a fibre-optic or wire cable. The fibre-optic link is a glass-fibre duplex cable of 62.5/125 μm size and uses standard SMA connectors at the ROV end and ST connectors at the PC end. For the trials described in this paper the wire link was used—it employs RS-422 drivers and receivers and thus requires two twisted-strand cable pairs. (For the short distances involved in these trials, which were all under 50 m, a four-core cable was used which was thinner than twisted-pair cable, having an external sheath diameter of 2.5 mm.) Although the vehicle has an umbilical cable unlike an AUV, it was

Table 2. Summary of the *Subzero II* sensor package

| Measurand | State | Sensor | Range | Resolution | LPF | HPF | Update rate | Latency† |
|-------------|-----------|-----------------------------------|-------------------------|--------------------------|-------------|--------|--------------|--------------|
| Depth | z | Pressure | 0–6 m 0–3 m | 2.3 cm 1.1 cm | — | — | — | <0.1 ms |
| Roll | ϕ | TCM2 digital compass module | $\pm 50^\circ$ | 0.3° | — | — | 10 Hz | 100 ms |
| Pitch | θ | | $\pm 50^\circ$ | 0.3° | — | — | | |
| Heading | ψ | | 0–360° | 0.1° | — | — | | |
| Speed | u | Pressure Impeller | 0–1.6 m/s >0.7 m/s | variable†† variable†† | 100 Hz — | — — | — >18 Hz‡ | <0.1 ms — |
| Surge accel | \dot{u} | Accelerometer | $\pm 4 \text{ m/s}^2$ | 3.1 cm/s^2 | 50 Hz | 0.2 Hz | — | <0.1 ms |
| Sway accel | \dot{v} | | | | | | | |
| Heave accel | \dot{w} | | | | | | | |
| Roll rate | p | Rate gyro | $\pm 90^\circ/\text{s}$ | $0.7^\circ/\text{s}$ | — | — | — | <0.1 ms |
| Pitch rate | q | | $\pm 90^\circ/\text{s}$ | $0.9^\circ/\text{s}$ | — | — | — | <0.1 ms |
| Yaw rate | r | | $\pm 73^\circ/\text{s}$ | $0.57^\circ/\text{s}$ | — | — | — | <0.1 ms |
| Prop speed | n | Optical encoder | $\pm 2750 \text{ rpm}$ | 1.2 rpm | — | — | 10.17 Hz | — |

LPF = low pass filter; HPF = high pass filter.

† Time between when the state is sampled and when valid data is received by MCU. Where no value is given, the figure reported by the sensor is an average across a sample period.

‡ Speed sensor characteristics are speed dependent.

Sensors displayed in light type were not used in tests and so were not calibrated. Any data given is nominal.

found—and reported by Lea (1997)—that this did not significantly affect speed control response. However, thruster saturation and a consequent drop in velocity was noted at the higher operating speeds.

The sensors currently installed in the ROV are three rate gyros, three accelerometers, two pressure sensors, an external speed sensor, an optical shaft encoder and a digital compass module. Together they provide data on the vehicle, as shown in table 2.

The motor command, m_d entered at the PC is a signed number representing the drive cycle on-time in MCU clock units. As the MCU runs at 2 MHz and the PWM cycle at 800 Hz, a command of 2500 corresponds to a duty cycle of 1. In practice, MCU processing overheads mean that the motor command ranges from –2100 to +2100 with 0 being no movement.

5. Classical control

Although many underwater vehicles employ advanced control methods, the successful use of classically-based controllers for underwater vehicles has been reported by Jalving (1994) on the *NDRE-AUV*. The disadvantage of such controllers is that they require that the system characteristics be linearized around a nominal operating point, and then the controller can control the process for small perturbations around that point. Performance at the operating point is usually good, as the controller has been designed for that purpose, but is unlikely to be as good elsewhere.

Transfer-function based controllers can be designed either by looking at the frequency response or by root-locus methods. The transfer-function based controllers that were used here were designed using root-locus techniques in the s -plane before being discretized into a digital controller using Tustin's mapping. The only input to such controllers was the error signal from the reference state; such controllers do not have an explicit derivative term and are thus less amenable to employing a derivative input.

6. Fuzzy logic control

Fuzzy control was conceived in an attempt to apply imprecise human thinking to control problems. The traditional application is one where no plant model is available, but a human operator can control the process satisfactorily. A rule-based system is used, with rules set up by the designer. By 'fuzzifying' crisp input data into linguistic sets it allows an automatic control strategy to be developed from a linguistic control strategy based on expert knowledge.

As an example to illustrate the fuzzy control process, take the situation of steering a small boat. If the boat was on course then the tiller would be centred; if the boat had veered off to the left then the tiller would be turned to the left; if the boat had veered to the right then the tiller would be turned to the right. This can be viewed as the basis for a fuzzy controller:

- If the *heading error* is **ZERO**, then set the *tiller* to the **MIDDLE**
- If the *heading error* is **POSITIVE**, then set the *tiller* to the **LEFT**
- If the *heading error* is **NEGATIVE**, then set the *tiller* to the **RIGHT**

where the heading error is the desired heading minus the actual heading, and thus veering to the left results in a positive value and veering to the right a negative. Each of the above statements are known as *rules* and the collection of rules as the *rule base*. The rules link inputs and outputs together: if the input is one thing, do something; if the input is another thing do something else. The style convention used here is italics for variables, whether input in the case of heading error or output in the case of tiller position. In addition, set names are given in small capitals. The sets are the categories into which the input and output variables can be partitioned. In the case of the heading error, 0° off would be **ZERO**, although 1° off might also be acceptable and thus also **ZERO**. However 10° off would definitely not be **ZERO**. An error of $+5^\circ$ is somewhere between **ZERO** and **POSITIVE** and thus will have membership of both sets. The process of determining input-set membership is called input fuzzification as it takes a ‘crisp’ input value and turns it into membership values of various fuzzy sets.

Essentially, a mapping is drawn up and applied to the input variable; see figure 3. It should be noted that the membership functions have been arranged so that the total membership in all the sets at any given value of heading error adds up to one. This is not essential, but often simplifies the control process.

A similar reverse mapping exists for the output function, namely **MIDDLE**, **LEFT** and **RIGHT**; the membership values of each output set need to be turned into a crisp output value that is used to control the position of the tiller. The mapping can again be described as for the input, or in the special case that the output sets are symmetrical and identically shaped, only the centre values need be specified. For the example here, **MIDDLE** is set to have a centre value of 0, **LEFT** to be -5° and **RIGHT** to be $+5^\circ$.

Thus, suppose the heading error is -6° . The membership of **NEGATIVE** is therefore $2/3$ and the membership of **ZERO** is $1/3$. Using the rule base described earlier, the validity of each rule is equal to the value of the appropriate input-set membership. Thus, the membership of each output is the same as that of the equivalent input, and so the final commanded tiller position is $2/3 \times 5^\circ + 1/3 \times 0^\circ = 3.33^\circ$.

Practical experience allows the designer to choose values for the set memberships and the rule base that will result in a suitable controller, without recourse to a system model. It should be noted that, although the control has been specified linguistically, the end result (here) is that of a piecewise interpolated proportional controller—the output tiller position is in some way proportional to the input heading error value. This is due to the choice of the rule base. A different rule base that could have been used is:

- If the *heading error* is **POSITIVE**, then make the *tiller position* **MORE LEFT**
- If the *heading error* is **ZERO**, then keep the *tiller position* **CONSTANT**
- If the *heading error* is **NEGATIVE**, then make the *tiller position* **MORE RIGHT**

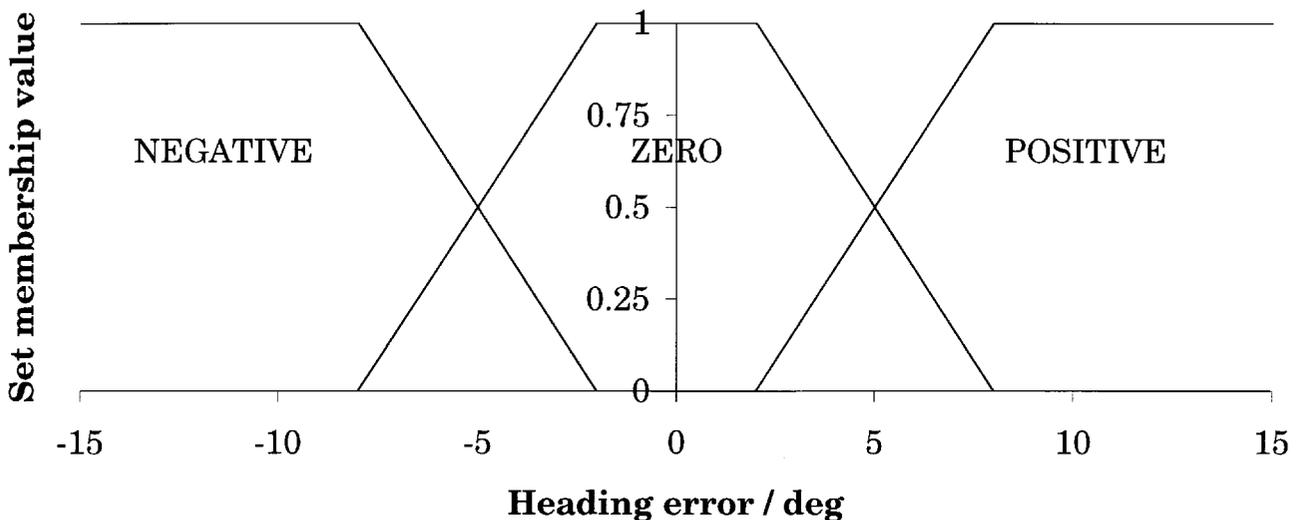


Figure 3. Fuzzy logic heading error input function.

In this case, the net effect is that of an integral controller. It can be appreciated that a fuzzy controller is unlimited in the type of control it can use; instead of simple output values, rules could have been specified such that, for example:

- If the *heading error* is ZERO, then the *tiller* should be set to minus the *heading error*.
- If the *heading error* is NEGATIVE, then the *tiller* should be set to the *heading error* squared.

Given such a range of options, it is perhaps not surprising that fuzzy controllers are not perceived as rigorous, although they are capable of performing many control actions successfully (Kosko 1994). However, fuzzy logic control is well suited to nonlinear control problems as it is fully tunable and covers as much of the state space as the designer wants in a piecewise interpolated fashion. In addition, when expressed in linguistic terms, fuzzy logic controllers do not require a mathematical system model, as long as the system can be controlled qualitatively. In such a case the fuzzy logic controller is designed by creating the input membership functions, the rule base and then the output functions or values. For the results presented here, this was done heuristically.

6.1. General fuzzy control procedure

As shown in the above example, there are three stages in the operation of a fuzzy logic controller:

- (1) First, crisp input data is fuzzified into a number of sets: the degrees of membership of the appropriate sets $\mu_{\text{SET}}(\text{variable})$ are found using the fuzzy input mapping.
- (2) Secondly, the fuzzy rule base is applied: where there is one input per rule, the validity v_i of rule number i is equal to the set membership. Where a fuzzy rule has more than one input—e.g. if x_1 is A and x_2 is B then y is C—then either the minimum membership value is taken, i.e. $v_i = \min[\mu_A(x_1), \mu_B(x_2)]$ or alternatively the validity is taken as being the product of the input values.
- (3) Thirdly, the fuzzy output sets are turned into a single crisp command value. In the case of output sets which are only specified by their centre values y_i , the output is the sum of each rule validity multiplied by the rule output divided by the sum of the rule validities (n is the number of rules):

$$y_{\text{total}} = \frac{\sum_{i=1}^{i=n} v_i y_i}{\sum_{i=1}^{i=n} v_i} \quad (1)$$

See Lee (1990) for a complete discussion of fuzzy logic controllers.

7. Sliding mode control

Sliding mode control (SMC), also known as variable structure control (VSC), uses the concept of sliding surfaces to increase control robustness. An impression of this is shown in the phase plane diagram of figure 4.

In this example, a model of the plant has been estimated as being $G(s) = 1/s$ and it has been determined that an acceptable unit step response can be achieved with unity feedback, i.e. $u = 1 - x$, where u is the actuator command and x the vehicle state. This desired state trajectory is shown by the grey line. However, over time the plant changes and eventual plant dynamics of, for example, $1/s(s + 0.5)$ result in the response shown by the dotted line.

Sliding mode control aims to keep the response to that desired, even in the presence of modelling errors or disturbances. This is achieved by attempting to track the desired response trajectory by ‘sliding’ along it. Thus, the desired trajectory is known as the *sliding surface*. The simplest method of achieving this is a control law that has one action when the response is above the sliding surface, and a different one when it is below it. The solid line in figure 4 shows the response when the simple relay control law:

$$u = 1 - x + \begin{cases} -0.25 & \text{if } \dot{x} > 1 - x \\ +0.25 & \text{if } \dot{x} < 1 - x \end{cases}$$

was used: $\dot{x} = 1 - x$ is the sliding surface and 0.25 is the sliding gain here. Figure 5 shows the time responses of the two controllers with the final plant stated above; it may be appreciated that the controller with sliding action has better performance, but at the cost of chattering in the control action. However, as will be shown later, it is possible to reduce or remove this problem.

The approach described here is based on the one by Healey and Lienard (1993) which is also presented by Fossen (1995). This is based on state feedback techniques with the addition of a nonlinear part due to the sliding action. The sliding surface is defined as

$$\sigma(\tilde{\mathbf{x}}) = \mathbf{h}^T \tilde{\mathbf{x}} \quad (2)$$

where the state error

$$\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d \quad (3)$$

Note that this is the reverse of the classical error definition—c.f. \mathbf{h} is a gain vector that will be found later. It is important that the sliding surface is defined such that $\sigma(\tilde{\mathbf{x}}) \rightarrow 0 \Rightarrow \tilde{\mathbf{x}} \rightarrow 0$ or, in other words, that as the sliding surface tends to zero then the state error also tends to zero. Now assuming that the dynamic model can

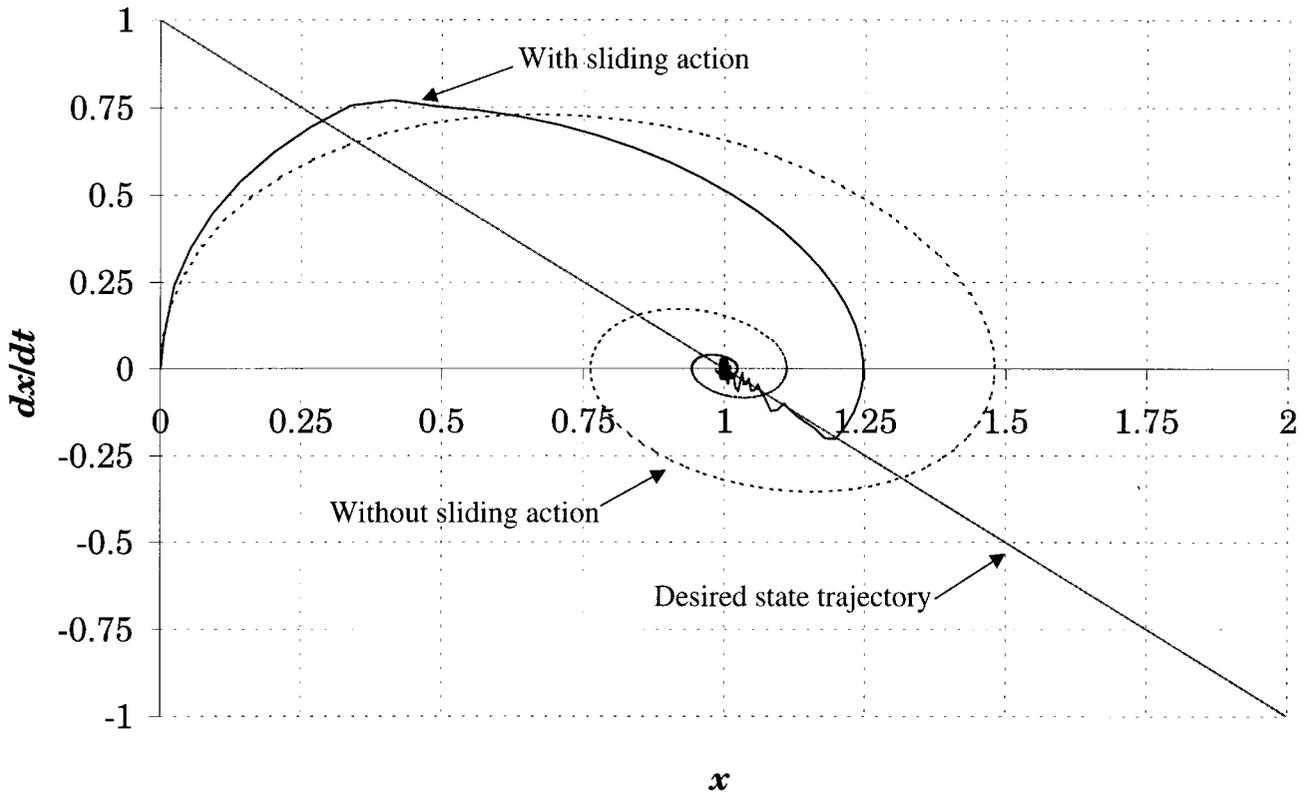
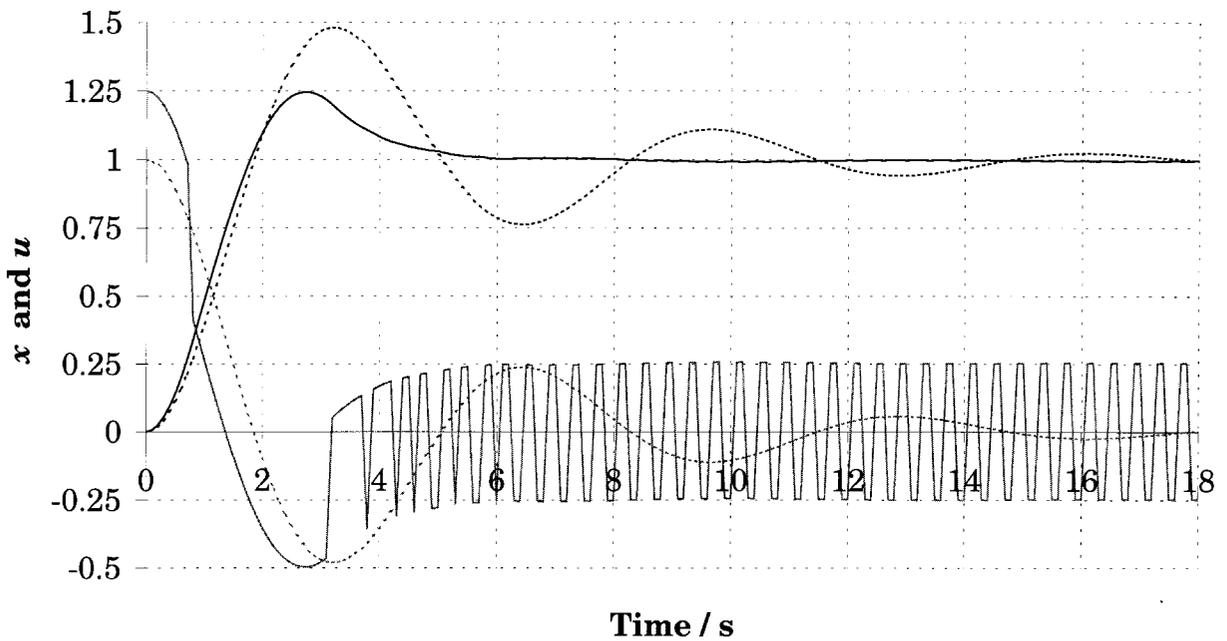


Figure 4. Phase plane plot for sliding mode example.



| | |
|---|--------------------------------------|
| Response without sliding action | — Response with sliding action |
| Control effort without sliding action | — Control effort with sliding action |

Figure 5. Time response for sliding mode example.

be written as a single input, multiple state (SIMS) equation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u + \mathbf{f}(\mathbf{x}) \quad (4)$$

where $\mathbf{f}(\mathbf{x})$ is a nonlinear function describing disturbances and unmodelled coupling effects, then the control law has two parts:

$$u = \hat{u} + \bar{u} \quad (5)$$

where \hat{u} is the nominal or computed part and \bar{u} is the nonlinear sliding part. \hat{u} uses the linear part of (4) to determine a control action that would result from the usual state feedback design process. \bar{u} ensures that the state will remain on the sliding surface despite imperfections in the model.

The sliding surface $\sigma(\tilde{\mathbf{x}})$ can be defined by pole placement, which in the case of state variables is most easily done by specifying its eigenvalues λ_i . By using state feedback techniques to define a vector of feedback gains \mathbf{k} ,

$$\hat{u} = -\mathbf{k}^T \mathbf{x} \quad (6)$$

and substituting into (4) gives the closed loop dynamics as

$$\dot{\mathbf{x}} = \mathbf{A}_c \mathbf{x} + \mathbf{b}\bar{u} + \mathbf{f}(\mathbf{x}) \quad (7)$$

where $\mathbf{A}_c = \mathbf{A} - \mathbf{b}\mathbf{k}^T$ and has eigenvalues λ_i as specified.

To determine the nonlinear part of the control it should be noted that if (4) is multiplied by \mathbf{h}^T and then $\mathbf{h}^T \dot{\mathbf{x}}_d$ is subtracted, the following is obtained:

$$\left. \begin{aligned} \mathbf{h}^T \dot{\mathbf{x}} - \mathbf{h}^T \dot{\mathbf{x}}_d &= \mathbf{h}^T \mathbf{A}_c \mathbf{x} + \mathbf{h}^T \mathbf{b}\bar{u} + \mathbf{h}^T \mathbf{f}(\mathbf{x}) - \mathbf{h}^T \dot{\mathbf{x}}_d \\ \text{therefore} \end{aligned} \right\} \quad (8)$$

$$\sigma(\tilde{\mathbf{x}}) = \mathbf{h}^T \mathbf{A}_c \mathbf{x} + \mathbf{h}^T \mathbf{b}\bar{u} + \mathbf{h}^T \mathbf{f}(\mathbf{x}) - \mathbf{h}^T \dot{\mathbf{x}}_d$$

Thus, if

$$\mathbf{h}^T \mathbf{b}\bar{u} = \mathbf{h}^T \dot{\mathbf{x}}_d - \mathbf{h}^T \hat{\mathbf{f}}(\mathbf{x}) - \eta \operatorname{sgn}(\sigma) \quad (9)$$

is chosen, where $\eta > 0$ and $\hat{\mathbf{f}}(\mathbf{x})$ is an estimate of $\mathbf{f}(\mathbf{x})$ then

$$\dot{\sigma}(\tilde{\mathbf{x}}) = \mathbf{h}^T \mathbf{A}_c \mathbf{x} - \eta \operatorname{sgn}(\sigma) + \mathbf{h}^T [\mathbf{f}(\mathbf{x}) - \hat{\mathbf{f}}(\mathbf{x})] \quad (10)$$

where $\operatorname{sgn}(x)$ is -1 if x is negative, 0 if x is 0 and $+1$ if x is positive. The system will be convergent if σ always has the opposite sign to $\dot{\sigma}$. Noting that

$$\mathbf{h}^T \mathbf{A}_c = (\mathbf{A}_c^T \mathbf{h})^T \quad (11)$$

one of the properties of eigenvectors is that an eigenvector \mathbf{m} of the matrix \mathbf{A} satisfies

$$\mathbf{A}\mathbf{m} = \lambda\mathbf{m}$$

where λ is an eigenvalue of \mathbf{A} . Thus, by choosing \mathbf{h} as the eigenvector of \mathbf{A}_c^T for $\lambda = 0$ then $\mathbf{h}^T \mathbf{A}_c \mathbf{x} = 0$ and hence (10) becomes:

$$\dot{\sigma}(\tilde{\mathbf{x}}) = -\eta \operatorname{sgn}(\sigma) + \mathbf{h}^T [\mathbf{f}(\mathbf{x}) - \hat{\mathbf{f}}(\mathbf{x})] \quad (12)$$

Therefore the system will always be convergent if

$$\eta > |\mathbf{h}| \cdot |\mathbf{f}(\mathbf{x}) - \hat{\mathbf{f}}(\mathbf{x})| \quad (13)$$

However, in practical implementations, the system will chatter as shown earlier because of the discontinuity due to the relay-action $\operatorname{sgn}(\sigma)$ term. This can be avoided (at the cost of tracking authority) by using a tanh term, as Healey and Lienard use, or alternatively the saturation function $\operatorname{sat}(x)$ which is defined as shown in figure 6 and is used here in preference to tanh for its computational efficiency. Thus, the final control law is a combination of this with (6) and (9):

$$u = -\mathbf{k}^T \mathbf{x} + (\mathbf{h}^T \mathbf{b})^{-1} \left[\mathbf{h}^T \dot{\mathbf{x}}_d - \mathbf{h}^T \hat{\mathbf{f}}(\mathbf{x}) - \eta \operatorname{sat} \left(\frac{\sigma}{\phi} \right) \right] \quad (14)$$

where ϕ is the boundary layer thickness for the sat function (not the vehicle's roll angle) and acts as a low-pass filter to remove chattering and noise effects. Both η and ϕ are tunable parameters: the greater the unmodelled disturbance, the larger the gain η must be, but at the cost of increased overshoot and ringing.

7.1. Integral action

The sliding mode controller described above does not involve any integral action as it relies on state feedback. However, there are advantages to having integral action, as will be seen later.

To achieve this where desired, an explicit integral term was added to the control law given by (14) to yield:

$$u = -\mathbf{k}^T \mathbf{x} + (\mathbf{h}^T \mathbf{b})^{-1} \left[-\frac{k_i}{T} \sum_{n=1}^{n=k} |\tilde{u}_n| + \mathbf{h}^T \dot{\mathbf{x}}_d - \mathbf{h}^T \hat{\mathbf{f}}(\mathbf{x}) - \eta \operatorname{sat} \left(\frac{\sigma}{\phi} \right) \right] \quad (15)$$

where k_i is the gain of the integral term and k is the time in discrete controller intervals.

8. Speed control

Three controllers were designed for speed control of the *Subzero II* vehicle—a transfer-function based controller, a fuzzy logic controller and a sliding mode controller. All three were tuned by their response to a step demand from 0 m/s to the nominal cruise speed of 1.3 m/s. The desired response had no overshoot with a fast settling time.

The simplification of the manoeuvring model into four subsystems means that only one degree of freedom is involved and the complex equations of motion that

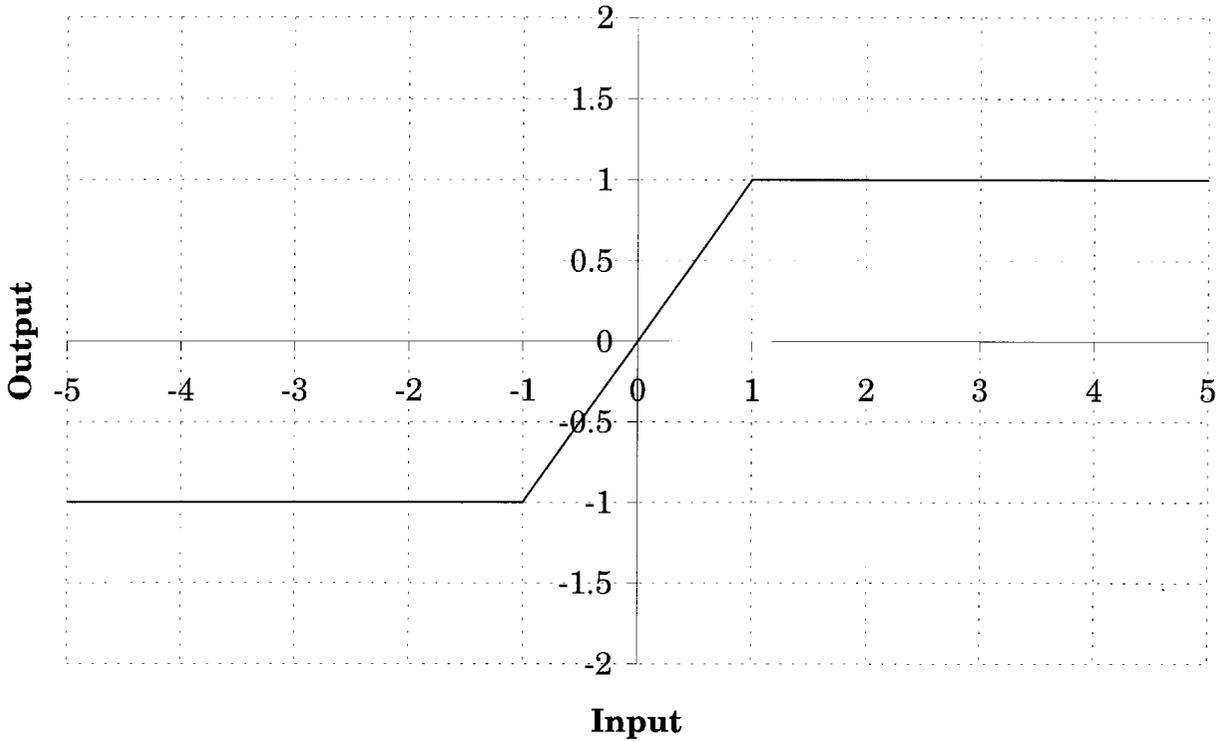


Figure 6. Saturation function $\text{sat}(x)$.

may be found in sources such as Fossen (1995) or Healey and Lienard (1993) can be simplified to:

$$(m - X_u)\dot{u} = T_{prop} - \frac{1}{2}\rho l^2 X'_{uu}u^2 \quad (16)$$

where m is the vehicle's mass, X_u is the hydrodynamic 'added mass', T_{prop} is the thrust produced by the propeller, l is the vehicle's length and X'_{uu} is the non-dimensional form of the vehicle's drag coefficient, incorporating its geometry.

Some idea of the nonlinearities present in the system may be obtained from figure 7, which illustrates the drag coefficient, as well as from figure 8, where the propeller shaft speed is shown against motor command.

8.1. Classical control

Although the thrust generated by the propeller is central to the equation given above, the actual command to the vehicle is in terms of m_d , the motor command. As a baseline, the vehicle is assumed to be operating at its nominal cruising speed of 1.3 m/s and the equation of motion (16) can therefore be linearized around that speed. This yields a transfer function of:

$$\frac{\Delta u}{\Delta m_d} = \frac{7.0 \times 10^{-4}}{s + 0.50} \quad (17)$$

This was used together with root locus techniques, which indicated that a digital controller of the form:

$$\frac{m_d(z)}{u_d(z) - u(z)} = K \frac{z - b}{z - 1} \quad (18)$$

would give a critically damped response (u_d is the demanded speed). This was tuned—again using root-locus methods—to give the controller

$$\frac{m_d(z)}{u_d(z) - u(z)} = 4000 \frac{z - 0.96}{z - 1} \quad (19)$$

which produced the response shown in figure 9 when combined with the transfer function model of the vehicle. However, as the transfer function model was generated from a cruising speed of 1.3 m/s, it is only likely to be valid around that speed and thus the actual step response from zero may not be the same.

8.2. Fuzzy logic

In the literature, there appear to be few examples of fuzzy logic used for speed control. For example, a major proponent of fuzzy control, Smith *et al.* (1993), presented work on simulated fuzzy control of heading, depth and pitch, but speed was controlled by specifying the propeller speed.

The rule-base and methodology used in this work is similar to that described by Lea *et al.* (1997) on the

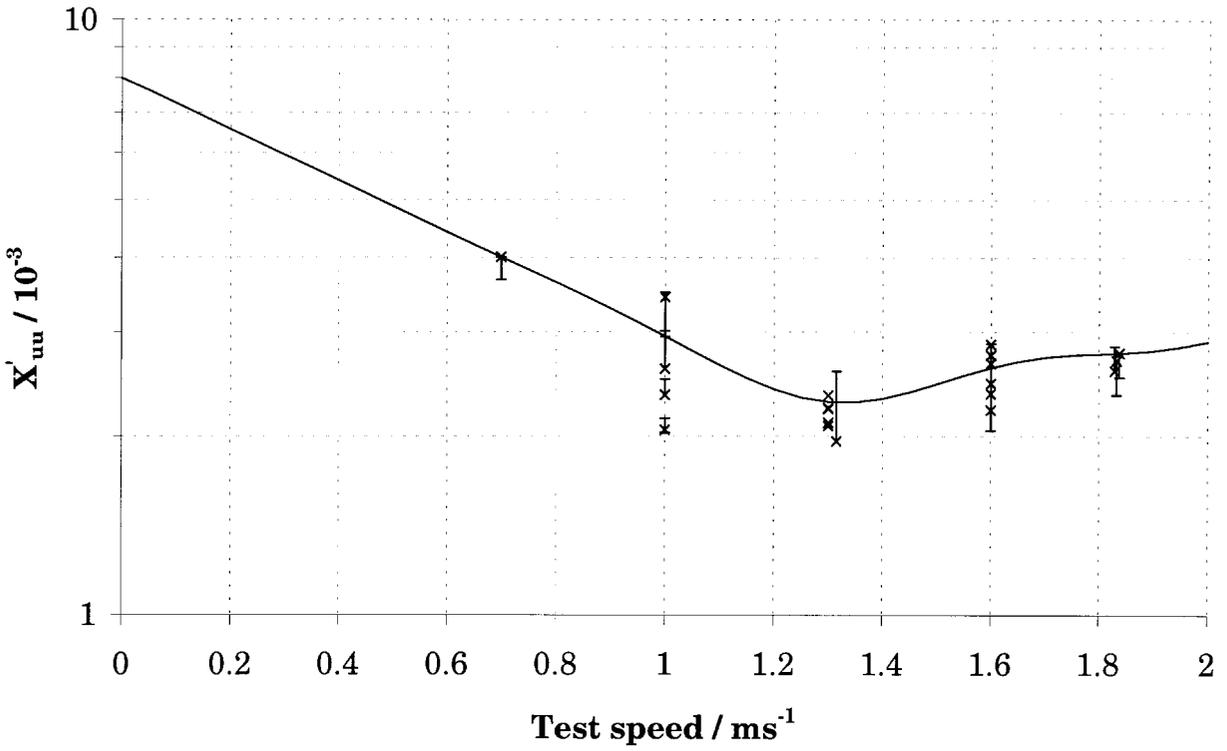


Figure 7. Drag coefficient X'_{uu}

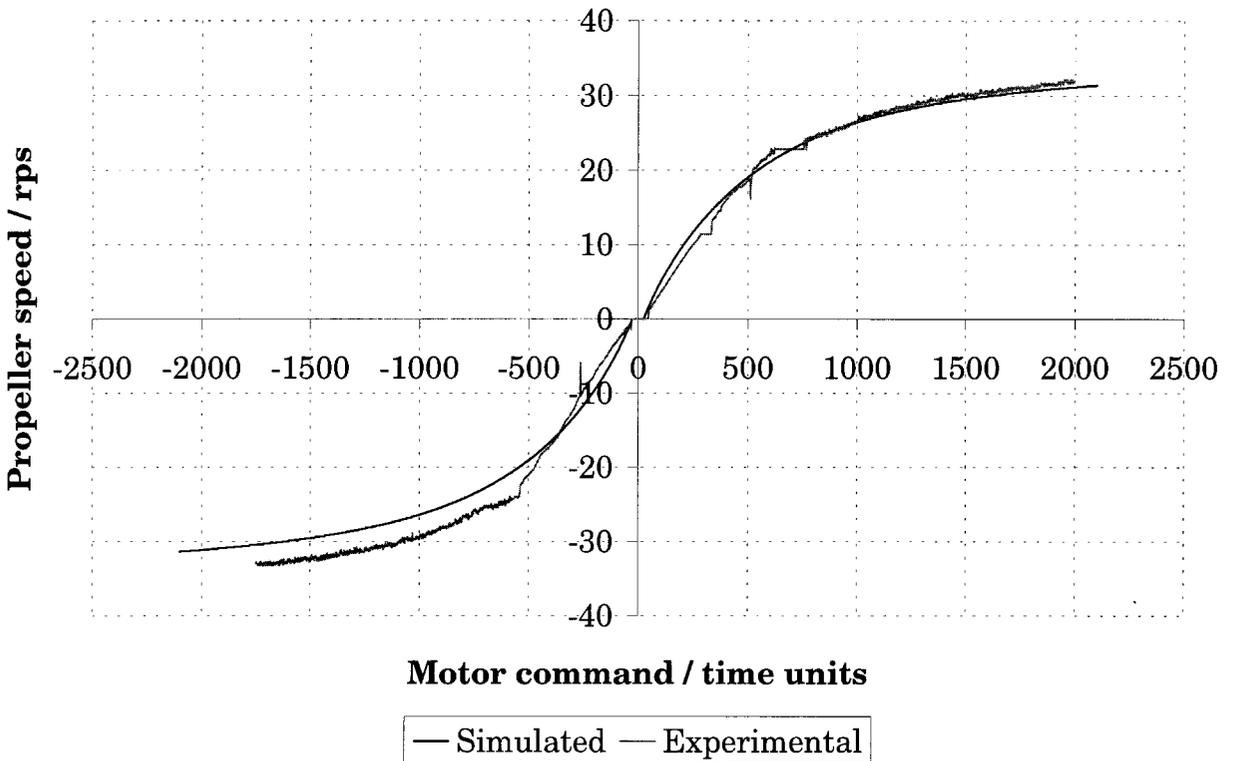


Figure 8. Propeller shaft speed in air against motor command.

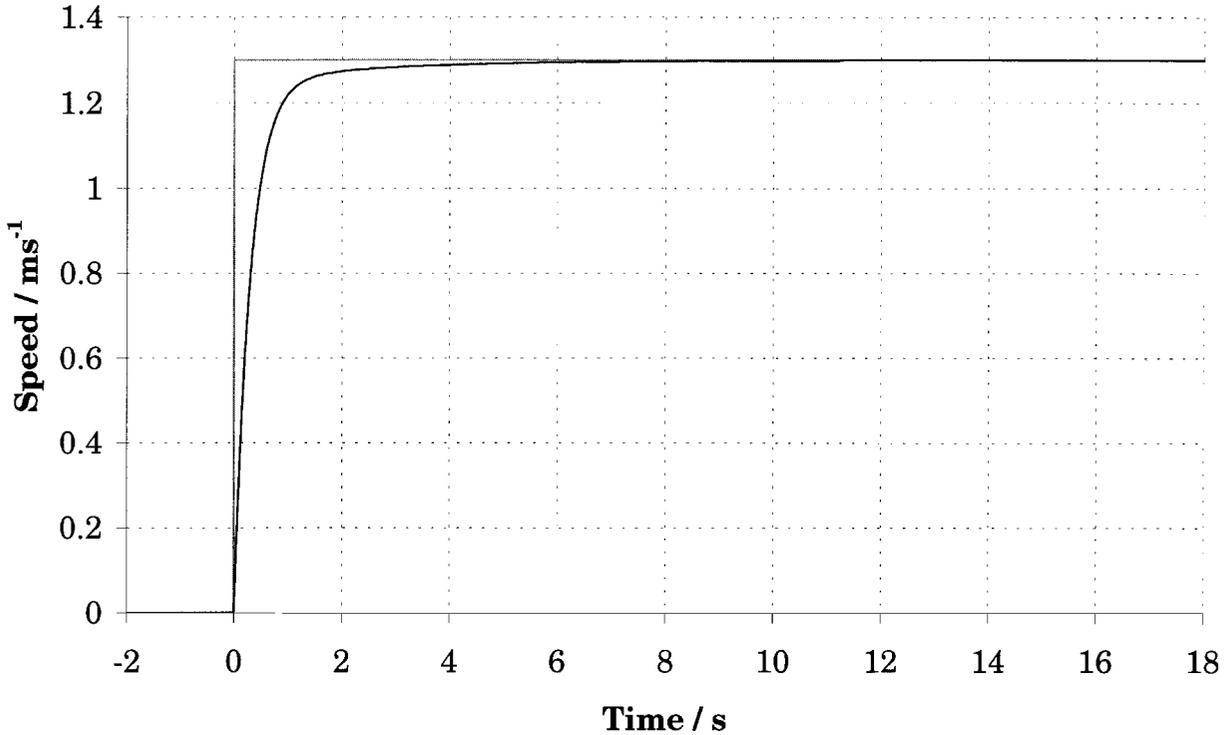


Figure 9. Transfer function-based speed response.

simulated speed control of a low-drag vehicle. In both cases, the fuzzy control is based on the premise that, whilst the propeller speed for a given vehicle speed may not be known because of disturbances such as currents, it is possible to make statements of the type:

- If *speed* is TOO SLOW then *motor command* should be INCREASED.
- If *speed* is TOO FAST then *motor command* should be DECREASED.
- If *speed* is OKAY then *motor command* should be KEPT THE SAME.

In practice, two inputs were used for speed control—speed error $u' = u_d - u$ and acceleration \dot{u} . The controller was tuned by hand using a simulation of the vehicle's dynamics. The fuzzy input sets are shown in figure 10 and the fuzzy rule base with output values is shown in table 3. The values given are the centroids for rules of the type:

- If u' is OKAY and \dot{u} is STEADY, then $m_{d_k} = m_{d_{k-1}} + 0$.

In other words, the output values are added to the current motor command to give the new motor command and thus the values in the rule base are increments rather than absolute values. The controller is, in effect, using purely integral action.

8.3. Sliding mode

The input to the sliding mode controller is the speed error $\tilde{u} = u - u_d$. As this is completely decoupled from the other state variables, the sliding surface can be selected to be the state error (Healey and Lienard 1993):

$$\left. \begin{aligned} \sigma &= \tilde{u} = u - u_d \\ \dot{\sigma} &= \eta \text{sat} \left(\frac{\sigma}{\phi} \right) \end{aligned} \right\} \quad (20)$$

Given an accurate model, the simplified equation of motion (16) can be used to generate the motor command:

$$\left. \begin{aligned} (m - X_{\dot{u}})\dot{u} &= T_{prop} - \frac{1}{2}\rho l^2 X'_{uu}u = T_{prop} - drag \end{aligned} \right\} \quad (21)$$

therefore

$$T_{prop_d} = (m - X_{\dot{u}})(\dot{u}_d - \dot{\sigma}) + drag_{predicted}$$

where the 'd' subscript represents a demanded value. The vehicle drag was predicted using the surge drag coefficient for the vehicle as well as the drag due to the control surfaces. As the actual position of the fins δr and δs was not measured, the commanded values δr_d and δs_d were used instead. This gave the following expression for drag:

$$drag_{predicted} = 0.5u^2 + 3.5u^2(\delta r_c^2 + \delta s_c^2). \quad (22)$$

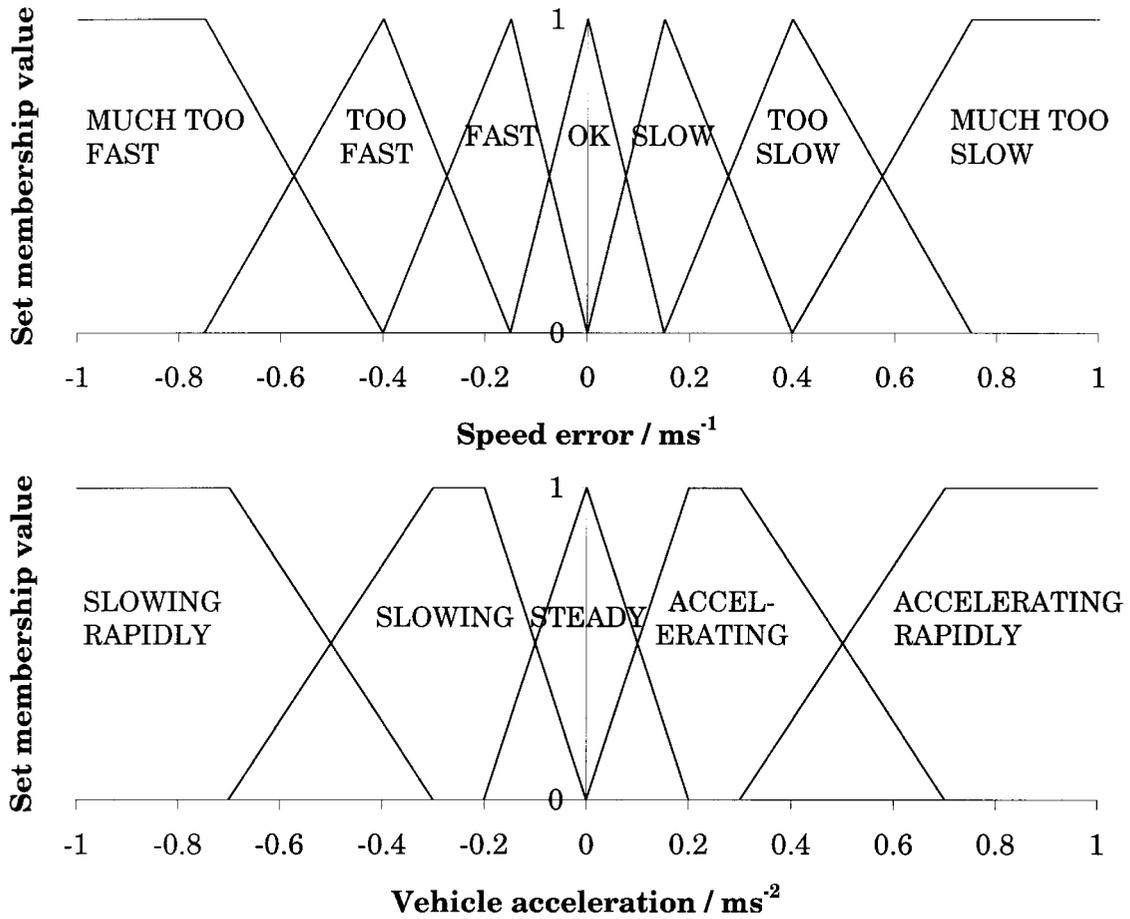


Figure 10. Fuzzy speed controller input mappings.

Note that the vehicle drag was predicted using a simpler model than that given by figure 7 as part of the data for that model came from the tests described below. The propeller thrust was found using a four-quadrant thrust model (see Lea 1997 for details) such that:

$$\left. \begin{aligned}
 T_{prop} &= C_T^* \frac{\rho}{2} [u^2 + (0.7\pi n_d D)^2] \frac{\pi}{4} D^2 \\
 \text{therefore} \quad n_d &= \frac{1}{0.7\pi D} \sqrt{\frac{8T_{prop_d}}{C_T^* \cdot \pi \rho D^2} - u^2}
 \end{aligned} \right\} \quad (23)$$

where the second expression is a rearrangement of the first. It can be seen that if u is known then the desired propeller speed n_d can be obtained if T_{prop_d} is specified and the thrust coefficient C_T^* can be found.

Thus (21) and (23) can be combined to give:

$$n_d = \frac{1}{0.7\pi D} \sqrt{\frac{8[(m - X_u)(u_d - \dot{\sigma}) + drag_{predicted}]}{C_T^* \cdot \pi \rho D^2} - u^2} \quad (24)$$

Using the model shown in figure 8 of propeller speed against motor command, the motor command needed here was found by inverting the model to get the empirical expression:

$$m_d = 0.67n_d^{2.4545} \quad (25)$$

Thus, the basic sliding mode control algorithm was obtained by using the above two expressions. The sliding part of the control law is given by $\dot{\sigma}$ and the linear part is given by u_d . To give a smooth response

Table 3. Fuzzy speed controller rule base

| | | u' | | | | | | |
|-----|--------------|------|------|------|------|------|------|-----|
| | | MTF | TF | Fast | OK | Slow | TS | MTS |
| u | SR | 0 | 200 | 300 | 400 | 450 | 500 | 800 |
| | Slowing | 0 | 0 | 25 | 40 | 50 | 200 | 600 |
| | Steady | -400 | -150 | -25 | 0 | 25 | 150 | 400 |
| | Accelerating | -600 | -200 | -50 | -40 | -30 | 0 | 200 |
| | AR | -800 | -500 | -450 | -400 | -300 | -200 | 100 |

with no overshoot, the demanded acceleration was set to be proportional to the speed error so that:

$$\dot{u}_d = -2\tilde{u}. \quad (26)$$

However, speed control requires a non-zero propeller speed to maintain a constant speed; looking closely at the equation for motor command (24), it can be seen that there is no integral term involved. If the model is correct, then this does not pose a problem as the essentially open-loop predictive part of the control law will produce the correct motor command; if the nominal part does not produce sufficient thrust, then the non-linear part of the control law will act to make up the difference and keep the vehicle responding as per the design. However, there are limits to the authority of the nonlinear part. If this is not enough, then a steady-state error will result. Whilst the gain of this term could be increased, this would result in larger overshoots and greater ringing during normal operation. (This is similar to what occurs with a classical controller involving proportional action, as here the nonlinear part is essentially adding a proportional term.)

To deal with this, an explicit integral term was added to the control law to produce

$$n_d = \frac{1}{0.7\pi D} \sqrt{\frac{-\frac{1}{4T} \sum_{n=1}^{n=k} \tilde{u}_n + 8[(m - X_u)(\dot{u}_d - \dot{\sigma}) + drag_{predicted}]}{C_T^* \cdot \pi \rho D^2}} - u^2 \quad (27)$$

where T is the sample time (0.1 s). The tuning of this was arranged to give a slow integral action so that the dynamics of the response would not be affected significantly.

9. Experimental results

The *Subzero II* vehicle was tested at the Ocean Basin at DERA Haslar in 12 sessions which were spread over five months. In total, over 180 test runs were performed. The basin is a concrete tank measuring 120 m × 60 m × 5.5 m deep and is equipped with wave-making and rotating arm systems, although these were not used during the tests.

The vehicle began each run 0.4 m underwater on a wooden 'beach' at the edge of the Ocean Basin testing pool. The autopilot was commanded to accelerate the vehicle from stationary to the commanded speed, whilst maintaining heading and depth (not shown). Thus, the three controllers were tested with regard to their step responses to demanded speeds of 1.0 m/s, 1.3 m/s (the

nominal cruising speed), 1.6 m/s and 2.0 m/s. Representative results are shown in figures 11–13.

Figure 11 shows the experimental speed response of the classical controller. The nominal cruising speed of 1.3 m/s is reached after about 8 s and is tracked satisfactorily, albeit with some noise. (The initial spikes in the response between 0 and 1 s are due to sensor noise and scaling.) Target speeds of 1.0 m/s and 1.6 m/s are also reached successfully after the same length of time, with the slower speed being tracked less noisily.

Owing to sensor limitations a demand speed of 2.0 m/s could not be reached without problems. (The impeller for measuring speed had a 'wrap around' problem and thus the speed range had to be determined via the low speed sensor. Limitations in the range of the low speed sensor meant that when the speed was close to 2.0 m/s the sensor fusion algorithm incorrectly reported a speed of around 0.6 m/s.)

The experimental response due to the fuzzy logic controller is shown in figure 12. It can be seen that the response is good at the design speed of 1.3 m/s which is obtained after some 2 s. However, as might be expected from a more optimally tuned controller, it appears more sensitive to target speed as the response to a demand of 1.0 m/s exhibits an overshoot. In addition, noise levels appear higher than with the classical controller. Although the controller could achieve a speed of 1.6 m/s successfully, a demand of 2.0 m/s was not met with the speed reaching a maximum of around 1.9 m/s before falling due to actuator saturation resulting from tether drag. Overall, it seems that the fuzzy logic controller has a faster response than the classical controller, but the tracking is worse.

Figure 13 shows the experimental response obtained using the sliding mode controller. It can be seen that there is a fast initial response to a demand of 1.3 m/s (as with the fuzzy logic controller), but the vehicle takes a significant length of time to reach the demand speed. This is doubtless due to the model underestimating the drag of the vehicle and thus not producing a large enough motor command. (The underestimated drag is most likely due to three factors: the simplified drag term, the fact that the motor mapping is based on results obtained from experiments in air and the extra drag due to the tether.) The long settling time is due to the integral action added to the control law; without it there would be a steady-state error.

To show the tracking accuracy of the controller over time, the other responses have a non-zero initial integrator value and thus reach the target speed quickly. The tracking of the 1.0 m/s demand is especially good with minimal noise. The response to the 1.6 m/s demand is less good as there appears to be a small steady-state error, as with the 2.0 m/s response.

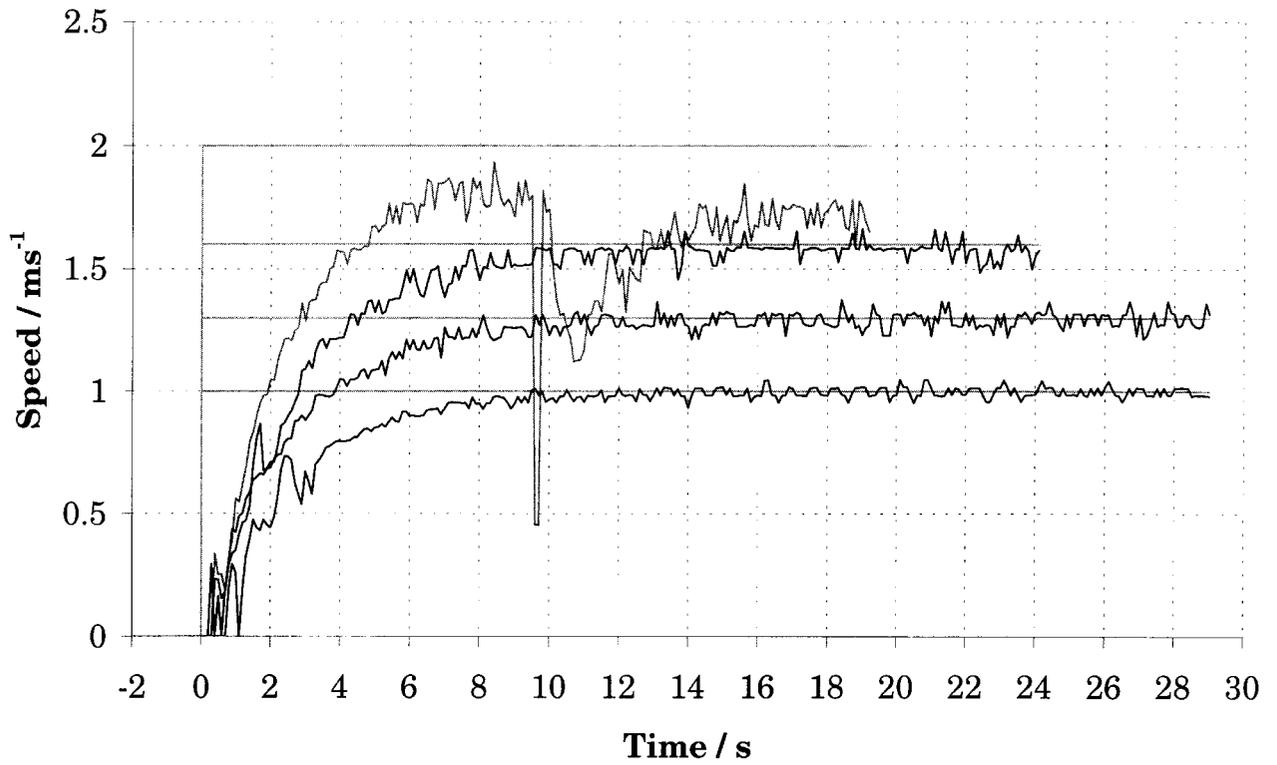


Figure 11. Experimental speed response for the classical controller.

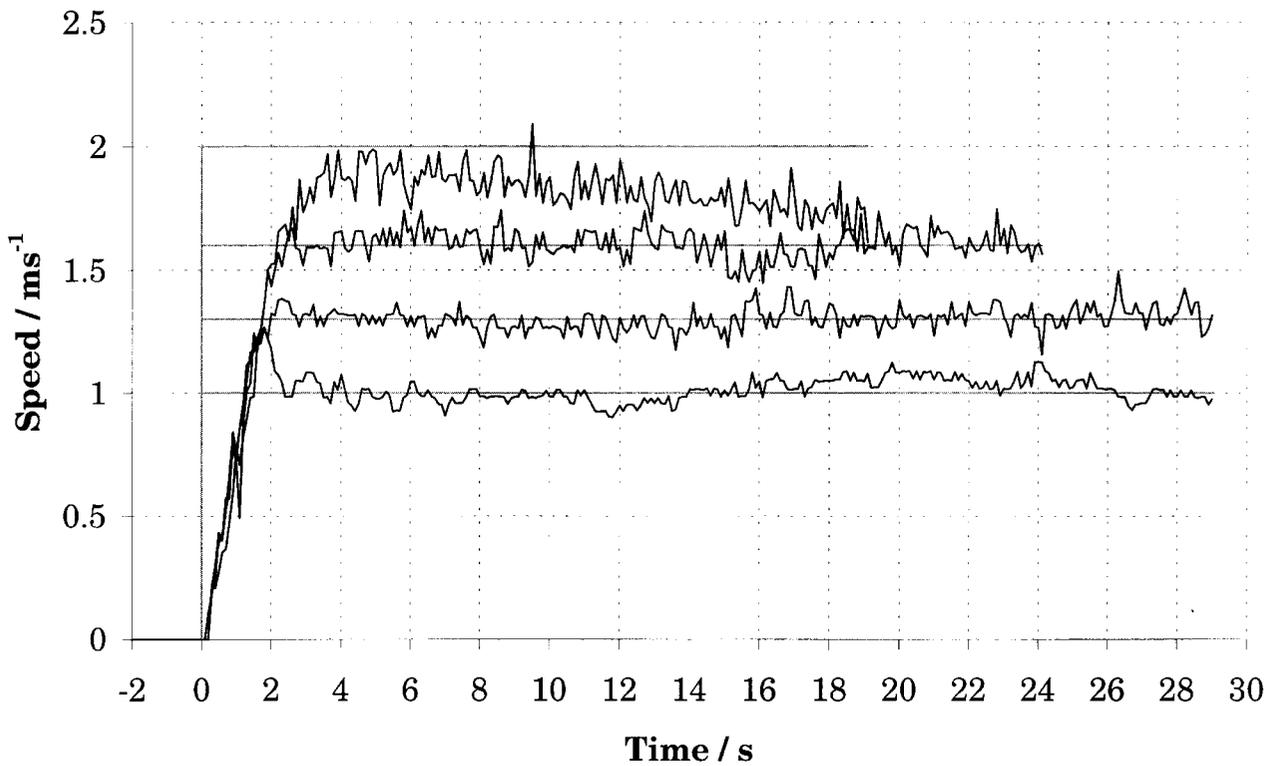


Figure 12. Experimental speed response for the fuzzy logic controller.

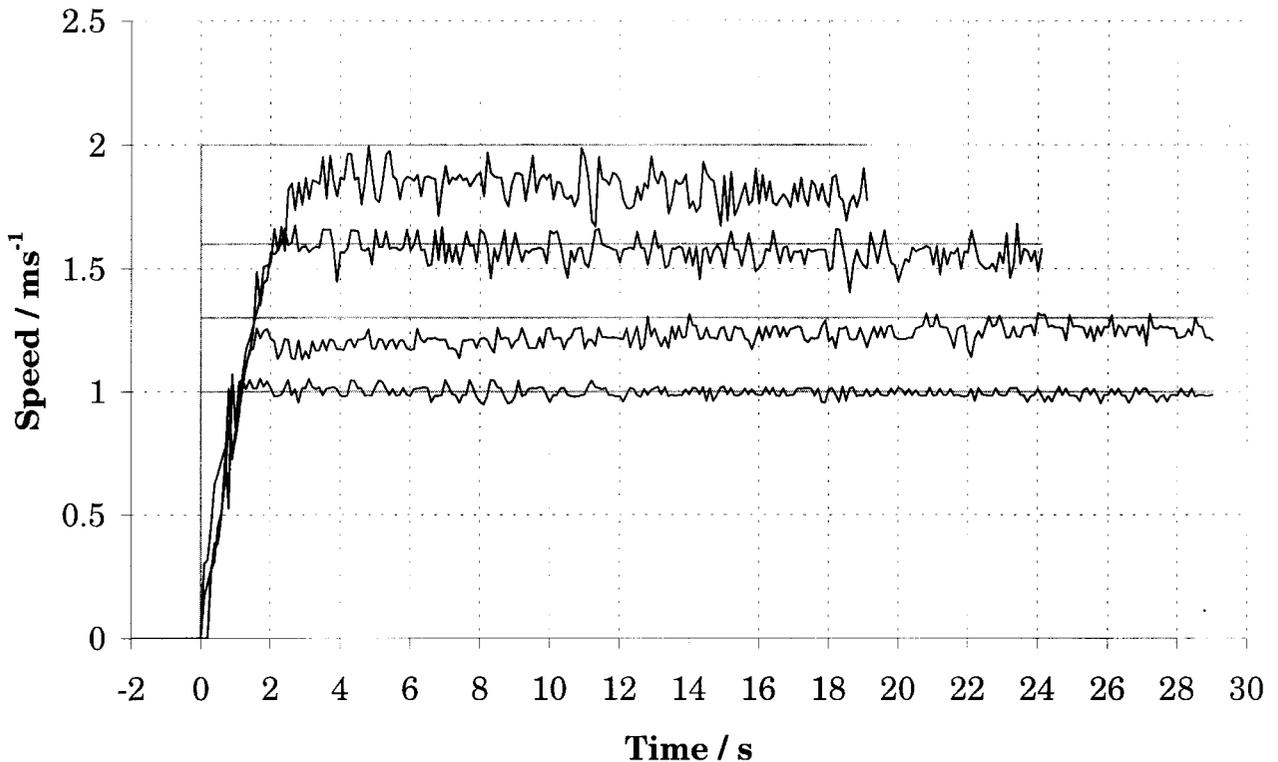


Figure 13. Experimental speed response for the sliding mode controller.

10. Conclusions

Of the three controllers, the classical controller was the most simple, but required a basic system model for its design. The fuzzy controller was generated without a model as such, but required extensive tuning via the simulation program (which could have been done on the actual vehicle if such a program was not available). The sliding mode controller was the most complex and required a complete system model.

With regard to performance, the fuzzy logic and sliding mode controllers can be compared directly as they were both tuned to give similar rise times and responses. Due to nonlinearities in the system, the classical controller could not produce such a fast rise without significant overshoots and thus had to be detuned, which resulted in a rise time of some 8 s compared to 2 s for the other two controllers. Given its less optimal nature, it is unsurprising that of the three controllers it was the most robust, i.e. the least affected by changes in target speed. Of the other two controllers, the sliding mode controller clearly has better performance, being more robust to changes in target speed as well as being generally less noisy.

Overall, there is no one controller that is clearly 'best'. If a detailed system model is available, perhaps as used in a simulation program, then the sliding mode controller may be readily generated and appears to provide

the best response. If such a model is not available, then it is up to the designer to decide whether such modelling is worthwhile.

Whilst some modelling is necessary to produce the classical controller, it is significantly less involved than that for the sliding mode controller, but has the disadvantage here of producing a slower response. If this can be tolerated, then such a controller would appear to be a reasonable solution. It might even be possible to tune the controller online as only the gain and the value of b in (18) need to be selected.

From this it would appear that the fuzzy logic controller is the least worthy of consideration. Its response appears to be less robust and more noisy than the others and the number of parameters that require tuning is significant. Its advantage that it does not require a system model may be reduced if the classical controller can also be tuned through testing. However, it should be pointed out that this does not mean that fuzzy control of speed is unsuccessful, merely that this particular implementation could be improved. For instance, it was mentioned earlier that the controller is essentially an integral one; better results may be achieved if a term is added which gives a nominal propeller speed for a given vehicle speed.

Thus, it may be concluded that no one technique appears as the most promising, but it can be appreciated

that each controller has its advantages (e.g. performance) and disadvantages (e.g. complexity) that need to be considered carefully with skill and judgement by the designer to produce a suitable solution to the desired task.

Acknowledgments

The authors are very grateful to Cable & Wireless (Marine) Ltd who kindly supported Roy Lea throughout this work. The technical skills of Keith Hallas, Jon Haughton and Colin Bielby (ISVR), and Peter Wheeler (Mechanical Engineering) are also warmly appreciated. The vehicle tests were carried out in the Ocean Basin at Haslar and the authors are very grateful to the DERA for access to this excellent facility.

References

- CRISTI, R., PAPOULIAS, F. A., and HEALEY, A. J., 1990, Adaptive sliding mode control of autonomous underwater vehicles in the dive plane. *IEEE Journal of Oceanic Engineering*, **15**, 152–160.
- COLLAR, P. G., and MCPHAIL, S. D., 1995, Autosub: an autonomous unmanned submersible for ocean data collection. *Electronics & Communications Engineering Journal*, June, 105–114.
- DI BITETTO, P. A., 1995, Fuzzy logic for depth control of unmanned undersea vehicles. *IEEE Journal of Oceanic Engineering*, **20**, 242–248.
- DOUGHERTY, F., and WOOLWEAVER, G., 1990, At-sea testing of an unmanned underwater vehicle flight control system. *Proceedings of the 1990 Symposium on Autonomous Underwater Vehicle Technology*, 65–73.
- FARRELL, J., and CLAUBERG, B., 1993, Issues in the implementation of an indirect adaptive control system. *IEEE Journal of Oceanic Engineering*, **18**, 311–318.
- FERGUSON, J., and POPE, A., 1995, 'Theseus': multipurpose Canadian AUV. *Sea Technology*, April, 19–26.
- FJELLSTAD, O.-E., and FOSSEN, T. I., 1994, Position and attitude tracking of AUV's: a quaternion feedback approach. *IEEE Journal of Oceanic Engineering*, **19**, 512–518.
- FOSSEN, T. I., 1995, *Guidance and Control of Ocean Vehicles* (New York: Wiley).
- FRYXELL, D., OLIVEIRA, P., PASCOAL, A., and SILVESTRE, C., 1994, Integrated design of navigation, guidance and control systems for unmanned underwater vehicles. *Proceedings OCEANS '94*, **III**, pp. III/105–III/110.
- HEALEY, A. J., and LIENARD, D., 1993, Multivariable sliding mode control for autonomous diving and steering of unmanned underwater vehicles. *IEEE Journal of Oceanic Engineering*, **18**, 3, 327–339.
- HILLS, S. J., and YOERGER, D. R., 1994, A nonlinear sliding mode autopilot for unmanned undersea vehicles. *Proceedings OCEANS '94*, **III**, III/93–III/98.
- JALVING, B., 1994, The NDRE-AUV flight control system. *IEEE Journal of Oceanic Engineering*, **19**, 497–501.
- KOSKO, B., 1994, *Fuzzy Thinking* (London: Flamingo).
- LEA, R. K., 1997, Control of a tethered underwater flight vehicle in the horizontal plane. *Proceedings of the 10th International Symposium on Unmanned Untethered Submersible Technology*, pp. 261–271.
- LEA, R., ALLEN, R., and MERRY, S., 1996, A low-cost remotely operated underwater vehicle. *Journal of Measurement and Control*, **29**, 201–204; 1997, Surge speed control methods for a laminar-flow autonomous underwater vehicle. *Oceanic Engineering International*, **1**, 10–15.
- LEE, C. C., 1990, Fuzzy logic in control systems: fuzzy logic controller. *IEEE Transactions on Systems, Man and Cybernetics*, **20**, 404–435.
- LICEAGA-CASTRO, E., and VAN DER MOLEN, G., 1995a, A submarine depth control system design. *International Journal of Control*, **61**, 279–308; 1995b, Submarine H_∞ depth control under wave disturbances. *IEEE Journal of Oceanic Engineering*, **3**, 338–346.
- PARSONS, M. G., CHUBB, A. C., and CAO, Y., 1995, An assessment of fuzzy logic vessel path control. *IEEE Journal of Oceanic Engineering*, **20**, 276–284.
- RODRIGUEZ, R. R., and DOBECK, G. J., 1989, Guidance and control system of the large scale vehicle. *Proceedings of the 6th International Symposium on Unmanned Untethered Submersible Technology*, 434–451.
- SEUBE, N., 1994, Review of control methods for underwater vehicle navigation in uncertain environments. *Proceedings OCEANS '94*, **III**, pp. III/99–III/104.
- SILVESTRE, C., PASCOAL, A., and HEALEY, A. J., 1997, AUV control under wave disturbances. *Proceedings of the 10th International Symposium on Unmanned Untethered Submersible Technology*, pp. 228–239.
- SMITH, S. M., RAE, G. J. S., and ANDERSON, D. T., 1993, Applications of fuzzy logic to the control of an autonomous underwater vehicle. *Proceedings of the 2nd IEEE Conference on Fuzzy Systems*, pp. 1099–1106.
- SMITH, S. M., RAE, G. J. S., ANDERSON, D. T., SHEIN, A. M., 1994, Fuzzy logic control of an autonomous underwater vehicle. *Control Engineering Practice*, **2**, 321–331.
- SUTO, T., and URA, T., 1997, Development of a small cruising-type AUV 'Manta-Ceresia' and guidance system constructed with neural network. *Proceedings of the 10th International Symposium on Unmanned Untethered Submersible Technology*, pp. 69–74.
- VENUGOPAL, K. P., SUDHAKAR, R., and PANDYA, A. S., 1992, 'On-line learning control of autonomous underwater vehicles using feed-forward neural networks.' *IEEE Journal of Oceanic Engineering*, **17**, 308–319.
- XU, M., and SMITH, S. M., 1994, Adaptive fuzzy logic depth controller for variable buoyancy system of autonomous underwater vehicles. *Proceedings of the 3rd IEEE Conference on Fuzzy Systems*, pp. 1191–1196.