TRANSNAV International Journal on Marine Navigation and Safety of Sea Transportation
Volume 5 Number 1 March 2011

Path Following Problem for a DP Ship Simulation Model

P. Zalewski
Maritime University of Szczecin, Szczecin, Poland

ABSTRACT: For the dynamic positioning (DP) operations the equations of ship's motion can be simplified to a 3 degrees-of-freedom (DOF) model. DP station-keeping is a set-point regulation problem, i.e., forcing the output of the ship to maintain a constant reference position and heading. However since offshore vessels reposition themselves at different locations, the path following problem has to be solved. The article presents various solutions of this problem including fuzzy control design which could be utilised in autonomous ship simulation as well.

1 INTRODUCTION

The Auto Track DP modes enable the vessel to follow a predefined path, described by a set of waypoints, with a high degree of accuracy [Kongsberg, 2008]. These modes cover both low-speed and high-speed operations using different control strategies. The system can automatically switch between the strategies depending on the requested speed.

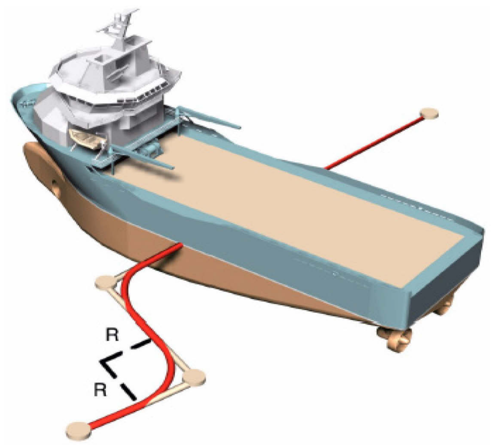

Figure 1. DP MSV (multipurpose supply vessel) in Auto Track mode. [Kongsberg, 2008]

The positions of the waypoints and the vessel's heading and speed that are to be used for each track section are specified by the operator and stored in waypoint tables. Waypoints can be inserted, modified and deleted as required. The vessel's heading is controlled by the following functions:

– Present Heading
– Set Heading
– System Selected Heading

The speed of the vessel along each section of the track can either be taken from the waypoint table or specified on-line by the operator using the Set Vessel Speed function.

Depending on the thrusters' installation and the vessel design, the maximum speed for a vessel in Auto Track low speed mode should not exceed approximately 3 knots since the effect of the lateral thrusters is reduced considerably at speeds higher than this.

The figure 1 shows the track a vessel will follow in Auto Track low speed mode according to the information contained in the table:

Table 1. Autotrack parameters

Waypoint No.	North/East Co-ordinates [m]	Speed [m/s]	Heading [°]
1	1501530 / 503600	0.3	0
2	1501570 / 503680	0.4	330
3	1501650 / 503630	0.4	300
4	1501740 / 503900	0.3	300

The operator can select between two alternative strategies for passing waypoints:

– Slowing down at each waypoint before continuing to the next (used when the vessel must remain on track, even during sharp turns)
– Passing the waypoint at a constant speed on a segment of a circle. The circle's radius can be calculated automatically according to the vessel speed, the angle of turn and the vessel's turning characteristics.

Both strategies generally require solving of the path following problem to acquire the desired dynamical behaviour of the vessel depending on the resultant thrusters' set points (geometric and dynamic problem).

DP vessels have different types of thrusters – tunnel thrusters, azimuth thrusters, main prop and rudders are the most commend used. Both RPM and pitch controlled. The pitch follow-up control is performed from the thruster process station. Set points are received cyclically from the DP controller via the dual communication network or by use of analogue signals. A PID-controller compares actual pitch against set value and controls the hydraulic pitch control valve accordingly.

It would then be convenient to parameterize all path waypoints in terms of a continuous path and constraining the vessel to this path [Skjetne, 2005]. The heading of the vessel could be taken as the direction of the tangent vector along the path, or simply as a constant heading, usually pointed against the environmental forces like waves and wind (system selected heading), or set by the DP operator according to operational demands (for instance to keep the transducer in the range of HPR transponders).

The desired dynamic behaviour along this path would be in first strategy zero speed (fixed positioning) at the waypoints, and when moving along the path from one waypoint to the next the desired path speed should be commanded online by the operator.

In the second strategy the desired dynamic behaviour along the path would be constant speed.

In the works done so far [Skjetne, 2005] the two methods were used: 1) starting with an already available tracking controller, and then converting this into a manoeuvre regulation controller; 2) starting with a parameterized path and a dynamic assignment along the path, designing the control, and tying together the geometric and dynamic objectives with the final pick of an update law. The second method seems to be more flexible and has the advantage that the path variable can be a dynamic state integrated online in the controller to satisfy the dynamic assignment.

These methods will be shortly presented in the following sections and together with the foundations of the fuzzy logic controller combining both.

2 CLASSIC TRACKING CONTROLLER IN DP SYSTEMS

The main functions to be performed in order for a dynamic positioning system to control a given vessel position (x, y) and heading (ψ) are [Cadet, 2003]:

- Estimate vessel motion
- Measure vessel response
- Determine error between prediction and measurement
- Determine corrective action to be applied
- Calculate and allocate appropriate command to thrusters to achieve desired corrective action

Figure 2 presents block diagram of a DP system. The kernel of this system is the simplified hydrodynamic vessel model. This model is a set of equations of motion that is used to predict the motion of the vessel when known forces and moment are applied. In order to separate the wave induced oscillatory part of the motion from the remaining part of the motion, the total vessel motion is modelled as the added outputs of a low-frequency model (LF-model) and a high-frequency model (HF-model). The HF-model represents oscillatory wave components in the vessel motion. The LF-model represents motions induced by wind, thrust and current in surge, sway and yaw. The low frequency portion of the model is controllable by means of thrusters. The algorithm calculates values of vessel's state vector (position, heading and motion variables) by measurements filtration and then it changes resulting force demand - thrusters allocation to meet position, heading and motion settings [Zalewski, 2010].

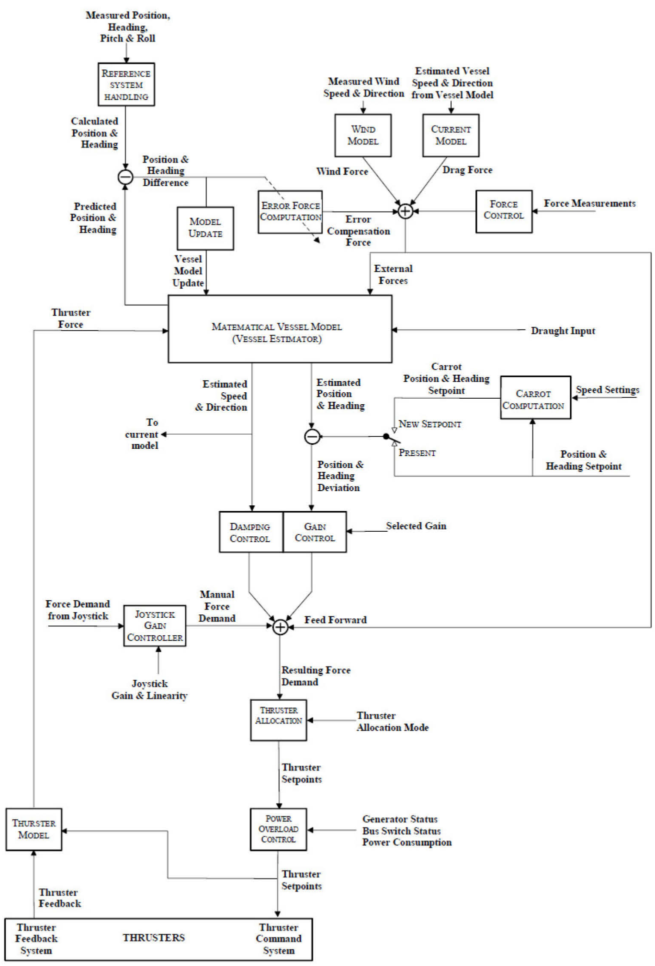

Figure 2. Block diagram of DP system. [Kongsberg, 2008]

Thrusters Allocation block is usually ship specific PID controller responsible for achieving dynamic and geometric objectives formulated by the Carrot Computation block. In a Dynamic Positioning application a Kalman filter is used to estimate the state of the vessel (for which a dynamics model has been developed) based on noisy measurements from reference systems and sensors.

2.1 *Kalman filtering*

The estimation problem solved by the Kalman filter can be expressed as follows [Cadet, 2003]: how to optimally estimate the state of a vessel with an approximate knowledge of the vessel dynamics (imperfect mathematical model) and with noisy measurements from sensors and position reference systems? What is the best state estimate one can get out of all that?

A discrete random dynamic system is described by two equations in the Kalman filter [Zalewski, 2010]:

- state equation (structural model of the process):

$$x_k = A_{k-1} \cdot x_{k-1} + w_k \tag{1}$$

- measurement equation (measurement model):

$$z_k = H_k \cdot x_k + v_k \tag{2}$$

where: x – n^{th} dimension state vector,
w – r^{th} dimension state disturbance vector,
z – m^{th} dimension measurement vector,
v – p^{th} dimension state disturbance vector (measurement noise),
A – $n \times n$ dimension transition matrix,
H – $m \times n$ dimension measurement matrix,
$r \leq n$, $p \leq m$.

Besides, it is assumed for disturbance vectors w and v that they are Gaussian noise of normal distribution, of zero mean vector and are mutually not correlated. The state equation describes the trend of the vector concerned, and the measurement model gives the functional dependence of the measurement on this vector. The solution to the equation system (1), (2), taking into consideration the limitations imposed on disturbance vectors, can be reached by the Kalman filter.

The estimation of the state vector in the filter can be presented by the equations below:

- state vector forecast:

$$\hat{x}_k^- = A_k \hat{x}_{k-1}^+ \tag{3}$$

where: $\hat{x}_k^- \in \Re^n$ – forecast or a-priori estimated value of the state vector at k-moment, $\hat{x}_k^+ \in \Re^n$ – a-posteriori estimated value of the state vector,

- covariance value of the forecast of state vector:

$$P_k^- = A_k P_{k-1}^+ A_k^T + Q_k \tag{4}$$

where Q is the matrix of the covariance of the state disturbance (of vector w), index T means matrix transposing,

- filter amplification matrix:

$$K_k = P_k^- H_k^T \left[H_k P_k^- H_k^T + R_k \right]^{-1} \tag{5}$$

where R is the covariance matrix of measurement disturbance (of vector v),

- estimate of the state vector from filtration $\hat{x}_k^+$ after making measurement z_k:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \left[z_k - H_k \hat{x}_k^- \right] \tag{6}$$

- covariance matrix of the estimated state vector:

$$P_k^+ = \left[I - K_k H_k \right] P_k^- \tag{7}$$

where I is the identity matrix.

In DP systems controlling vessel's position with fixed heading in two dimensions generally the following values are to be estimated: position coordinates (ϕ, λ), projections of the speed vector in relation to the bottom onto the meridian (N-S axis) and the parallel (E-W axis) (V_N, V_E), acceleration vector projections onto the meridian and the parallel (a_N, a_E) and the projections of acceleration vector derivatives in relation to the bottom onto the meridian and the parallel (a'_N, a'_E). In this case the state vector will have the following elements:

$$x_k = \left[\varphi, \lambda, V_N, V_E, a_N, a_E, a_N', a_E' \right]^T \tag{8}$$

The measured values will be: position coordinates of the positioning system used (ϕ_{PS}, λ_{PS}), speed components in relation to the meridian and the parallel from log or DR navigation performed by the positioning system used (position changes derivatives V_N, V_E), acceleration components in relation to the meridian and the parallel from an inertial transformer MRU (a_N, a_E). So the measurement vector will have the following elements:

$$z_k = \left[\varphi_{PS}, \lambda_{PS}, V_N, V_E, a_N, a_E \right]^T \tag{9}$$

Some assumptions like values of variance and covariance for particular measurements have to be done, for instance: DGPS system − σ_ϕ = 1.0 m; σ_λ = 1.0 m; coordinates not correlated; speed components − σ_V = 0.1 m/s; acceleration components − σ_a = 0.01 m/s^2.

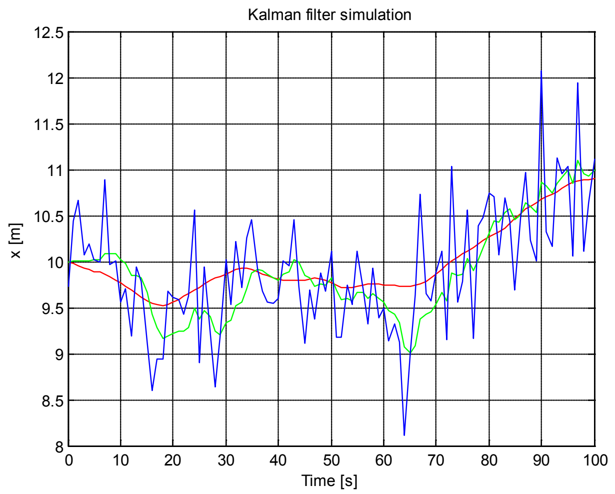

Figure 3. Kalman filter simulation of vessel x co-ordinate estimate.

In the figure 3 the results of the Kalman filter work referring to one co-ordinate are presented (simulated in Matlab®). Blue are measurements, red are unknown true values, green are estimated values.

Based on the simplified vessel model, and using the previous position estimate of the vessel, the prediction step of the Kalman filter gives us a prediction of the vessel position. Based on the forces acting on the vessel, on the vessel model and on the previous position estimate, this is where the DP system thinks the vessel is.

3 PARAMETERIZED PATH TRACKING CONTROLLER IN DP SYSTEMS

As stated in the section 1, the DP Auto Track involves two tasks called the geometric task and the dynamic task. While the main concern is to satisfy the geometric task, the dynamic task, further specified as a speed assignment, ensures that the system output follows the path with the desired speed. The main ideas were published in [Skjetne, 2005].

3.1 *Parameterization of path from waypoints – geometric assignment*

One method to generate a path of class C^r (C^r means that path function f is continuously differentiable up to order r) is to first specify a set of n+1 waypoints, and then construct a sufficiently differentiable curve that goes through the waypoints by using splines and interpolations techniques.

Designating the desired path as $p_d(\theta)$ we get n curves $p_{d,i}(\theta)$, i = 1, 2, … n, between the waypoints. Each of these is expressed as a polynomial in θ of certain order. Then the expressions for the sub paths are concatenated at the waypoints to assemble the full path. To ensure that the overall path is sufficiently differentiable at the way-points, the order of the polynomials must be sufficiently high. We will consider the path in two dimension space $\Re^2$. Let $l = \{1, 2, \ldots n\}$ be a set of indices identifying each sub path. A column vector will be stated as col. The overall desired curve is denoted $p_d(\theta) = \mathrm{col}(x_d(\theta), y_d(\theta))$, $\theta \in [0, n]$, while individual segments as $p_{d,i}(\theta) = \mathrm{col}(x_{d,i}(\theta), y_{d,i}(\theta))$, $i \in l$, and $p_i = \mathrm{col}(x_i, y_i)$, $i \in l \cup \{n+1\}$ are the waypoints.

Conveniently it is to assume that θ will reach an integer value at each waypoint, starting with $\theta = 0$ at WP 1 and $\theta = i$ at WP i. The differentiability requirement $p_d(\theta) \in C^r$, means that at the connection of two sub paths, the following conditions must hold:

$$\begin{array}{ll}
\lim\limits_{\theta \to i-1} x_{d,i-1}(\theta) = \lim\limits_{\theta \to i-1} x_{d,i}(\theta) & \lim\limits_{\theta \to i-1} y_{d,i-1}(\theta) = \lim\limits_{\theta \to i-1} y_{d,i}(\theta) \\
\lim\limits_{\theta \to i-1} x^{\theta}_{d,i-1}(\theta) = \lim\limits_{\theta \to i-1} x^{\theta}_{d,i}(\theta) & \lim\limits_{\theta \to i-1} y^{\theta}_{d,i-1}(\theta) = \lim\limits_{\theta \to i-1} y^{\theta}_{d,i}(\theta) \\
\vdots & \vdots \\
\lim\limits_{\theta \to i-1} x^{\theta^r}_{d,i-1}(\theta) = \lim\limits_{\theta \to i-1} x^{\theta^r}_{d,i}(\theta) & \lim\limits_{\theta \to i-1} y^{\theta^r}_{d,i-1}(\theta) = \lim\limits_{\theta \to i-1} y^{\theta^r}_{d,i}(\theta)
\end{array} \tag{10}$$

where: $x^{\theta^r}_{d,i} = \dfrac{\partial^r x_{d,i}}{\partial \theta^r}$, $y^{\theta^r}_{d,i} = \dfrac{\partial^r y_{d,i}}{\partial \theta^r}$ for $i \in l \setminus \{1\}$.

The task solution is to determine the coefficients $\{a_{j,i}, b_{j,i}\}$ of k order polynomials:

$$\begin{aligned}
x_{d,i}(\theta) &= a_{k,i}\theta^k + \ldots + a_{1,i}\theta + a_{0,i} \\
y_{d,i}(\theta) &= b_{k,i}\theta^k + \ldots + b_{1,i}\theta + b_{0,i}
\end{aligned} \tag{11}$$

According to equations (11) for each sub path there are $2 \cdot (k+1)$ unknowns ((k+1) unknowns a_i and (k+1) unknowns b_i) so there are $2n(k+1)$ unknown coefficients in total to be determined for the full path. Two methods are usually used for calculating these coefficients. After linearization of equations set (11) we will get $2n(k+1)$ linear equations $A\phi = c$ for the full path, which can be solved by a matrix inversion operation: $\phi = A^{-1}c$. However, as the number n sub paths increases, this soon encounters numerical problems in the inversion of A. Instead, it is possible to calculate the coefficients for each sub path independently. To ensure the desired continuity at the connection points the numerical values which are common for the neighboring sub paths must be defined.

For a k'th order polynomial $x_{d,i}(\theta)$ we have that $x^{\theta^j}_{d,i}(\theta) = 0$ for $j \geq k+1$. Hence, we can form equations from the first k derivatives of $x_{d,i}(\theta)$:

C^0: $2 \cdot 2n$ equations for $i \in l$:

$$x_{d,i}(i-1)=x_i \quad y_{d,i}(i-1)=y_i$$
$$x_{d,i}(i)=x_{i+1} \quad y_{d,i}(i)=y_{i+1} \tag{12}$$

C^1: four equations for the first and last n-waypoint:

$$x_{d,1}^{\theta}(0)=x_2-x_1 \quad x_{d,1}^{\theta}(0)=y_2-y_1$$
$$x_{d,n}^{\theta}(n)=x_{n+1}-x_n \quad y_{d,n}^{\theta}(n)=y_{n+1}-y_n \tag{13}$$

$2\cdot 2(n-1)$ equations for intermediate waypoints:

$$\left.\begin{aligned} x_{d,i}^{\theta}(i-1)=\lambda(x_{i+1}-x_{i-1}) \\ y_{d,i}^{\theta}(i-1)=\lambda(y_{i+1}-y_{i-1}) \end{aligned}\right\} \quad i=2,\ldots,n$$
$$\left.\begin{aligned} x_{d,i}^{\theta}(i)=\lambda(x_{i+2}-x_i) \\ y_{d,i}^{\theta}(i)=\lambda(y_{i+2}-y_i) \end{aligned}\right\} \quad i=1,\ldots,n-1 \tag{14}$$

where $\lambda > 0$ is a design constant. $\lambda = 0{,}5$ means that the slope at WP i is the average of pointing against WP i-1 and WP i+1; while $\lambda < 0{,}5$ means the slope is higher and $\lambda > 0{,}5$ means the slope is lower.

C^j: $2\cdot 2n$ equations for $i \in l$, if derivatives of order $j \ge 2$ are 0:

$$x_{d,i}^{\theta^j}(i-1)=0 \quad y_{d,i}^{\theta^j}(i-1)=0$$
$$x_{d,i}^{\theta^j}(i)=0 \quad y_{d,i}^{\theta^j}(i)=0 \tag{15}$$

which gives $2(j+1)\cdot 2n$ equations to solve for $(k+1)\cdot 2n$ unknowns. The path generation problem is now set up as n linear, decoupled sets of equations $A_i\phi_i = b_i$; $i \in l$; where the unknown vector ϕ_i is:

$$\phi_i = \mathrm{col}(\{a_{j,i}\}_{j=k,\ldots,0}, \{b_{j,i}\}_{j=k,\ldots,0}) \in R^{2(k+1)} \tag{16}$$

and A_i and b_i are formed correspondingly according to the above equations.

3.2 *Dynamic assignment*

The second task is to satisfy a desired dynamic behaviour along the path. This can be expressed in terms of a time assignment, speed assignment, or acceleration assignment along the path [Skjetne, 2005].

A time assignment means to be at specific points along the path at specific time instants. For a continuous parameterization $p_d(\theta)$ specific values like θ_1, θ_2; etc., must correspond to specific time instants t_1, t_2; etc., dependent through a design function $v_t(.)$ so that $\theta_1 = v_t(t_1)$ and $\theta_2 = v_t(t_2)$.

A speed assignment is to obtain a desired speed along the path. If $p_d(\theta)$ is a continuous parameterization this can be translated into a desired speed for θ. This desired speed may depend on the location along the path given by θ, or it may explicitly depend on time. A natural choice is therefore to express the desired speed for θ as a design function $v_s = (\theta, t)$.

An acceleration assignment is to obtain a desired acceleration along the path. For a continuous parameterization $p_d(\theta)$ this can be expressed by a design function $v_\alpha = (\dot{\theta}, \theta, t)$ for θ, which may depend on the speed $\dot{\theta}$ along the path in addition to θ and t.

4 FUZZY CONTROL PATH TRACKING

The design of fuzzy controller solving geometric and dynamic tasks presented in section 3 can be similar to the one described in [Zalewski, 2009].

In case of system being a DP ship, steered in Auto Track mode in accordance to the designed path, $u(n)$ will be vector of thrusters setting, $y(n)$ will be a vector containing six variables defining actual motion in 3-degrees of freedom: P_{xy} - actual position of selected reference point stored as two variables, v_x - actual longitudinal (advance) velocity, v_y - actual transverse (lateral) velocity, ω - actual angular (rotation) velocity, ψ – actual ship's heading; and $S(n)$ will be a vector containing six variables defining required motion in 3-dof: P_{xyr} - required position of reference point stored as two variables, v_{xr} - required advance velocity, v_{yr} - required lateral velocity, ω_r - required rotation velocity, ψ_r - required ship's heading At time n, $y(n)$ and $S(n)$ are used to compute the input variables of the fuzzy controller (effect of thrusters setting on motion): ΔP_{xy} - deviation between required and actual position, Δv_x - difference or deviation between required and actual longitudinal (advance) velocity, Δv_y - difference or deviation between required and actual transverse (lateral) velocity, $\Delta\omega$ - difference or deviation between required and actual angular (rotation) velocity, $\Delta\psi$ – difference or deviation between required and actual ship's heading. So generally the input variables vector can be designated by:

$$e(n) = S(n) - y(n) \tag{17}$$

Input variable scaling factors are used to conveniently manipulate the effective fuzzification on the scaled universes of discourse. The scaled factors used for $e(n)$ vector in presented research are normalization constants of the five mentioned deviations. Assuming the scaling factors for deviations as vector K_e the scaled input vector is:

$$E(n) = K_e e(n) \tag{18}$$

The scaled variables are then fuzzified by input fuzzy sets defined on the scaled universes of discourse: [0,1]. Figure 4 shows five input fuzzy sets for one of the $E(n)$ parameters that are used by the fuzzy controller implemented in Matlab®.

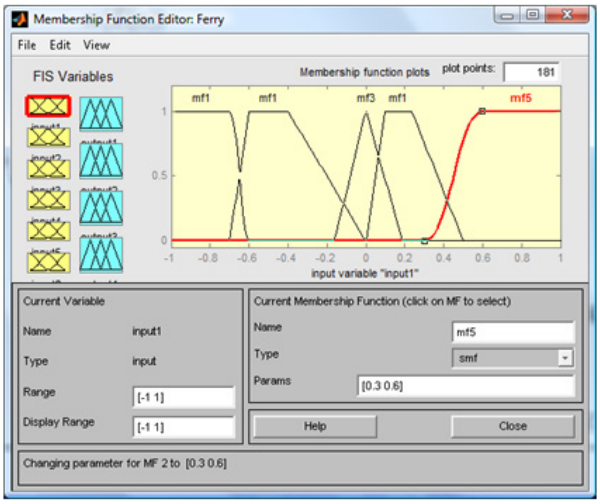

Figure 4. Membership functions of selected input parameter in DP fuzzy logic controller.

The linguistic names "Positive" and "Negative" are related directly to faster speed than required and slower speed than required respectively.

Fuzzification can be formulated mathematically replacing linguistic naming system by a numerical index system, for instance five fuzzy sets used may be represented by A_i, i = -2 (NL), -1 (NS), 0 (NZ), 1 (PS), 2 (PL).

No mathematically rigorous formulas or procedures exist to accomplish the design of input fuzzy sets – the proper determination of design parameters is strictly dependent on the experience with system behaviour, hence the expert data coming from ship manoeuvring trials is necessary similarly to designs presented in previous sections.

4.1 *Fuzzy rules*

Fuzzification results are used by fuzzy logic AND operations in the antecedent of fuzzy rules to make combined membership values for fuzzy inference. An example of a Mamdani fuzzy rule used for control of ship advance speed with main thruster is:

IF $E_2(n)$ is PL AND $E_1(n)$ is NS
THEN $u(n)$ is SAs (19)

where PL and NS are input fuzzy sets and SAs (Slow Astern) is an output fuzzy set. In essence rule (19) states that if ship's advance speed is significantly larger than the desired advance speed and the ship's position is a little off the desired track, the controller output should be the pitch setting corresponding to Slow Astern fuzzy set.

The quantity, linguistic names, and membership functions of output fuzzy sets are all design parameters determined by the controller developer. Similarly to input fuzzy sets the most popular membership functions of singleton type have been used.

The exact number of fuzzy rules is determined by the number of input fuzzy sets. For the considered system of ship control the total number of fuzzy rules will be the combination of 5 input variables and 5 fuzzy sets (if for all variables the same number of fuzzy input sets is designed): 5^5=3125; quite a large amount for only pitches setting. Actually this number of fuzzy rules can be significantly reduced by treating each input variable independently and combining the output during defuzzification. This can be achieved by utilizing coupled fuzzy controllers.

4.2 *Fuzzy inference*

The resultant membership values of input sets produced by fuzzy logic AND operation [Zadeh, 1996] or product operator can be used [Ying, 2000] are then related to the singleton output fuzzy sets by fuzzy inference. The four common inference methods produce the same inference result if the output fuzzy set is singleton.

If output fuzzy sets in rules are the same fuzzy logic OR operation can be used to combine the memberships.

4.3 *Defuzzification*

The membership values computed in fuzzy inference must be finally converted into one number by a defuzzifier. In the ongoing research the most prevalent defuzzifier in literature – centroid defuzzifier has been used [Ying, 2000]. The defuzzifier output at time n can be:

$$u(n) = \frac{m_{Z_1} \cdot u_1 + m_{Z_2} \cdot u_2 + m_{Z_3} \cdot u_3 + m_{Z_4} \cdot u_4}{m_{Z_1} + m_{Z_2} + m_{Z_3} + m_{Z_4}} \tag{14}$$

where:

u_1=-13% of pitch/throttle position (DSAs),

u_2=0% of pitch/throttle position (STOP),

u_3=-25% of pitch/throttle position (SAs),

u_4=-50% of pitch/throttle position (HAs),

$u(n)$ is the new output of the fuzzy controller at time n which will be applied to the ship system to achieve control. In comparison with conventional controllers, what is lacking is the explicit structure of the fuzzy controller behind the presented procedure. On the other hand utilizing expert knowledge for such a black box is much more straightforward and comprehensive.

5 CONCLUSIONS

For many DP operations solving of the path following problem is crucial. It can be achieved by the conventional system modelling methodology, but implementation of the fuzzy controllers seems to be promising as well especially where the system knowledge is represented mostly in an implicit and linguistic form rather than an explicit and analytical form.

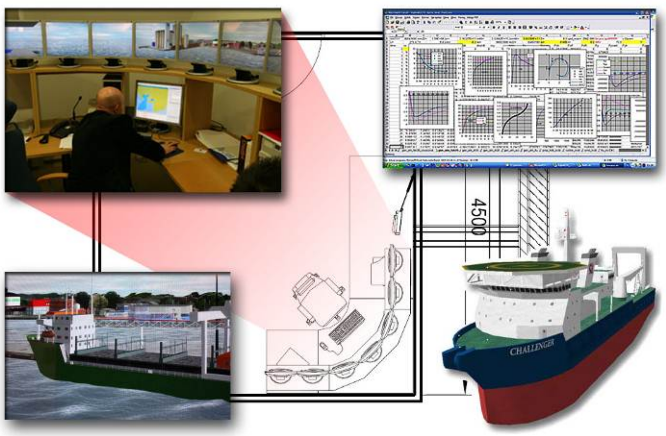

Figure 5. Instructor station in the Full Mission Ship Simulator with DP systems at Maritime University of Szczecin.

REFERENCES

Cadet O., "Introduction to Kalman Filter and its Use in Dynamic Positioning Systems", Dynamic Positioning Conference, Marine Technology Society, September 16-17, 2003.

Kongsberg Maritime AS, "Kongsberg K-Pos DP (OS) Dynamic Positioning System Operator Manual", release 7.1, doc. No. 322281/A, Horten, October 2008

Skjetne R., "The Maneuvering Problem", NTNU, PhD-thesis 2005:1, Trondheim, 2005.

Ying H., "Fuzzy Control and Modelling - Analytical Foundations and Applications", IEEE Press, New York, 2000.

Zadeh L. A., "The evolution of systems analysis and control: a personal perspective", IEEE Control Systems Magazine, 16, 95-98, 1996.

Zalewski P., "Fuzzy Fast Time Simulation Model of Ship's Manoeuvring", Taylor & Francis - Balkema in the TransNav'2009 Proceedings. Gdynia, June 17-19, 2009.

Zalewski P., "Models of DP Systems in Full Mission Ship Simulator", Scientific Journals of Maritime University of Szczecin 20(92), Szczecin, 2010.