ISSN 2229-5518

A Robust H∞ Speed Tracking Controller for Underwater Vehicles via Particle Swarm Optimization

Mohammad Pourmahmood Aghababa, Mohammd Esmaeel Akbari

Abstract— This paper presents an H_∞ controller designing method for robust speed tracking of underwater vehicles, using Particle Swarm Optimization (PSO). Nonlinearity mapping of the underwater vehicles model to a nominal linear model, by employing a linear controller for a nonlinear model, is one of the main contributions of this paper. For reaching the linear H_∞ controller, the nonlinear models linearized around an operating point. Both nonlinear and linearized models are discussed. A brief explanation of H_∞ synthesis is given. Also frequency dependent weighting functions are used for penalizing tracking errors, setpoint commands and measured outputs noises using PSO. Obtained controller is reduced order to achieve a lower order controller. After simulating the reduced order H_∞ controller it is embedded into the nonlinear model. By nonlinear simulations, robustness and efficient performance of the H_∞ controller is shown. Control efforts of actuators revealed no saturation, therefore it is feasible to implement.

Index Terms— H_∞ controller, underwater vehicles, particle swarm optimization, robustness.

1 Introduction

In the past two decades, underwater vehicles have become an intense area of oceanic researches because of their emerging applications, such as scientific inspection of deep sea, exploitation of underwater resources, long range survey, oceanographic mapping, underwater pipelines tracking and so on. Developing a control system that can achieve the aforementioned goal is challenging for a variety of reasons such as: the nonlinearity of the dynamics, the multivariable character of the vehicle with coupling among different channels, the consistent amount of uncertainty due to the lack of precise knowledge of hydrodynamic drag coefficients and evaluation of external disturbance due to environmental interaction.

Several control techniques have been proposed in literature to deal with uncertainty. Sliding mode controller for trajectory control of underwater vehicles, neglecting the cross coupling terms, is proposed in [2]. Multivariable sliding mode control for diving, steering and speed control of underwater vehicles with decoupled design is used in [3].

An H_∞ autopilot for subzero II that had two subcontrollers, the longitudinal controller for the forward speed and depth, and the lateral controller for the heading angle is presented in [4]. A reduced order H_∞ control that had three SISO decoupled controllers for the forward speed, heading angle and depth control was applied to subzero III in [5]. A time delay control law for robust trajectory control of underwater vehicles is proposed in [6].

In this paper, designing of an H_∞ controller for robust speed tracking is the major purpose. Position control cannot be performed without suitable speed tracking. Here, both linear and angular speeds are considered to be controlled. Using Particle Swarm Optimization (PSO), weighting functions, that capture the disturbance characteristics and performance requirements are selected to take advantage of H_∞ design algorithm.

For designing the speed controller, the nonlinear model is linearzed around an operating point. Afterwards, parameters changing mapped to the linear model as uncertainties. Weighted noises are also added to measured outputs. Then weighting functions for setpoint commands and tracking errors are obtained. It is assumed that all states can be measured by sensors, so that the state estimator is not necessary. After designing the H_∞ controller and order reduction, it is embedded to full nonlinear model. Tracking robustness and efficiency of the proposed controller is shown by nonlinear simulation. The required control efforts of thrusters are possible to realize.

The proposed method has the following characteristics: a) the problem of speed tracking is considered as a new work, b) designed controller is MIMO (Multi Input-Multi Output) without neglecting the cross coupling terms, c) the H_∞ controller is designed by using PSO, and d) the linear controller robustness is shown when it is embedded to the nonlinear model.

- *Mohammad Pourmahmood Aghababa, Department of Electrical Engineering, Ahar Branch, Islamic Azad University, Ahar,Iran; m-Pourmahmood@iau-Ahar.ac.ir*
- *Mohammd Esmaeel AkbariDepartment of Electrical Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran; m-Akbari@iau-Ahar.ac.ir*

ISSN 2229-5518

The rest of this paper is preceded as follows. Section 2 presents the nonlinear and linearized motion equations of underwater vehicles. In Section 3, a general scheme of H_∞ control synthesis is firstly explained. Then, the main procedure of PSO method is given. In Section 4, an H_∞ controller is designed for speed tracking aim of the underwater vehicles and numerical simulations are performed. Finally, some conclusions are given in section 5.

2 Motion Equations and Dynamics of Underwater Vehicles

2.1 The Nonlinear Model of Underwater Vehicles

Throughout the marine robotics literature a vehicle's six degrees of freedom dynamic equations are expressed as [1]:

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau \quad (1)$$

$$\dot{\eta} = J(\eta)v \quad J(\eta) = diag\{J_1(\eta), J_2(\eta)\} \quad (2)$$

where s(.)=sin(.), c(.)=cos(.), t(.)=tan(.), η is the position and orientation of the vehicle in the Earth fixed frame, $\in R^{6\times1}$, v is linear and angular velocity of the vehicle in the body fixed frame, $\in R^{6\times1}$, M is the inertia matrix including added mass, $\in R^{6\times6}$, C(v) is a matrix consisting Coriolis and centripetal terms, $\in R^{6\times6}$, D(v) is a matrix consisting damping or drag terms, $\in R^{6\times6}$, $g(\eta)$ is the vector of restoring forces and moments due to gravity and buoyancy, $\in R^{6\times1}$, and τ is the vector of forces and moments of propulsion, $\in R^{6\times1}$.The matrix $J(\eta)$ converts velocity in a body fixed frame, v, to velocity in an earth fixed frame, $\dot{\eta}$, as shown in Fig. 1. In fact $J_1(\eta)$ and $J_2(\eta)$ convert linear and angular velocities in a body fixed frame, v, to velocities in an earth fixed frame, $\dot{\eta}$, respectively. A detailed derivation of these nonlinear equations of motion can be found in [1]. Below a small summary of the modeled phenomena is given.

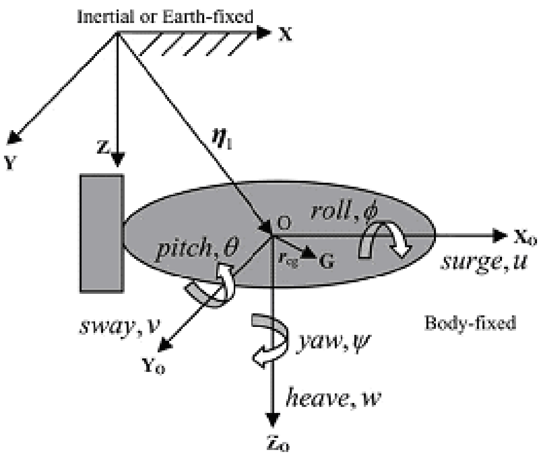

Fig. 1. Inertial and body coordinate frames

1) Mass and Inertia, M: In matrix M, two inertial components are accounted for [1],

$$M=M_{RB}+M_A,\ M=M^T,\ M>0 \quad (3)$$

The rigid body inertial matrix, M_{RB}, represents the mass and inertia terms due to the mass and other physical characteristics of the craft. However in a dense medium such as water, a considerable contribution to the mass originates from the medium. This so called added mass is accounted for by the matrix M_A.

2) Coriolis and Centripetal forces, C(v): For matrix C(x), a similar discourse can be held. Both the coriolis and centripetal forces are forces that are proportional to mass and inertia. Hence, the matrix consists of two matrices:

$$C(v) = C_{RB}(v) + C_A(v) \quad C_{RB} = -C_{RB}^T \quad (4)$$

where C_{RB} represents forces and moments due to the mass and physical characteristics of the craft, $C_A(x)$ incorporates the terms originating from the added mass.

3) Damping terms, D(v): In the damping matrix, D(x), four terms are combined [1]:

$$D(x)=D_p+ D_s(x) + D_w + D_M(x) \quad (5)$$

where D_p is the potential damping, $D_s(x)$ is linear and quadratic skin friction, D_w is wave drift damping and $D_M(x)$ is damping due to vortex shedding.

Potential damping is introduced due to forces on the body when the latter is forced to oscillate. Skin friction effects can be shown to constitute both a linear and a quadratic term. Wave drift damping only plays a major role at the surface where it can be interpreted as added resistance due to incoming waves. Damping due to vortex shedding is a result of the non-conservative nature of a moving system in water with respect to energy. The viscous damping force due to this phenomenon is a function of the relative velocity of the craft, its physical characteristics and the density and viscosity of the water.

4) Gravitation and Buoyancy, $g(\eta)$: This term models the restoring forces which result from gravitation and buoyancy.

5) Thruster model, τ: Usually, propellers are used as propulsion devices for underwater vehicles. The load torque Q from the propeller, and the thrust force T, are then usually written as [1]:

$$Q = \rho D^5 K_Q(J_0)|n|n, \quad T = \rho D^4 K_T(J_0)|n|n \quad (6)$$

where n is rotational velocity of the thruster, ρ is the mass density of water, D is the diameter of the propeller, K_Q and K_T are the torque and the thrust coefficients of the propeller, and J_0 is the advance ratio.

In this paper, the thrusters are assumed to be driven by DC motors. DC motors are usually controlled by velocity feedback. It is assumed that six propellers are erected in six freedom degrees. Therefore, n_i will be the physical input related to thruster number i. It can be also shown that an algebraic relation exists between the thrust of propeller i and the physical input. Therefore, the thrust will be chosen as input in the model $u_i = T_i$.

2.2 The Linearzed Model for Underwater Vehicles

The nonlinear speed system of the underwater vehicles can be described in state space form by defining a six dimensional state vector x=(u, v, w, p, q, r) as follows.

$$\dot{x} = f(x) + Bu, \quad u = \tau \quad (7)$$

$$f(x) = M^{-1}(-C(x) - D(x) - g(\eta)), \quad B = -M^{-1} \quad (8)$$

For a linear controller design, it is necessary to extract the linearzed model from the nonlinear model around a representative operating point. In this paper, the nominal value of rotational speed of the propellers is considered 100 rpm. Using this assumption, the operating point is

ISSN 2229-5518

obtained:

x_0=(1, 1, 1, 1, 1, 1) (9)

The linearized model is:

$$\Delta\dot{x} = A\Delta x + B\Delta u,\ \ \Delta y = C\Delta x$$
$$x = [u, v, w, p, q, r]^T,\ u = [\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6]$$
$$y = [u, v, w, p, q, r]^T \quad (10)$$

where A and B are 6× 6 matrices and C is a 6×1 vector, τ_i i=1, 2, …, 6 are the propeller forces, explained in the previous section. [u, v, w] and [p, q, r] are the linear and angular speeds of the underwater vehicle in a body fixed coordinate system, respectively.

The step response of linearized model is shown in Fig. 2. As seen in this figure, the step response is not tracked and system modes are not decoupled.

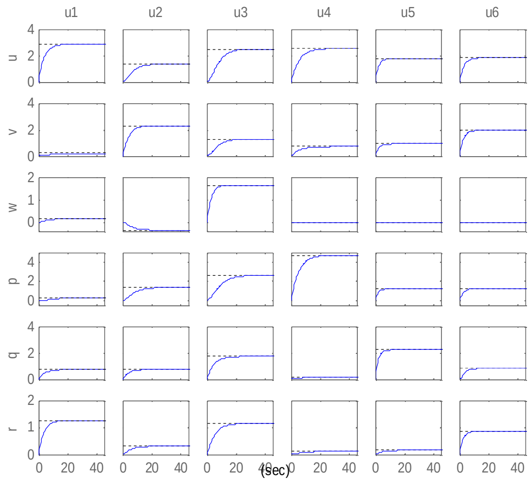

Fig. 2. Step response of open loop linear model

3 METHODOLOGIES

3.1 H∞ Synthesis Approach

Figure 3 shows a tracking problem, with disturbance rejection, measurement noise, and control input signal limitations. K is a controller to be designed, G is the system, as nonlinearity uncertainties modeled, to be controlled and W_{noise}, W_{cmd} and W_{perf} are weighting functions for sensor noises, setpoint commands and tracking errors, respectively. A reasonable design objective would be to design K to keep tracking errors and control input signal small for all reasonable reference commands, sensor noises, and external force disturbances.

Hence, a natural performance objective is the closed loop gain from exogenous influences (reference commands, sensor noise, and external force disturbances) to regulated variables (tracking errors and control input signal). Specifically, let T denote the closed loop mapping from the outside influences to the regulated variables. Good performance is associated with T being small. The mathematical objective of H∞ control is to make the closed loop MIMO transfer function T_{ed} to satisfy $||T_{ed}||_\infty<1$. The weighting functions are used to scale the input/output transfer functions such that when $||T_{ed}||_\infty < 1$, the relationship between e and d is suitable.

Without lack of generality, a mathematical overview of H∞ synthesis is as follows. Figure 4 shows a standard feedback system, where w is the input vector of exogenous signals, e is the output vector of errors to be reduced, y is the vector of measurements that are available for feedback and u is the vector of external force signals. Let T_{ew} denote the closed loop transfer matrix from w to e. The H∞ synthesis problem is to find, among all controllers that yield a stable closed loop system, a controller K that minimizes the infinity norm $||T_{ed}||_\infty$. Throughout this paper we assume that all states are available for measurement, that is, y equals the internal state of the generalized plant P.

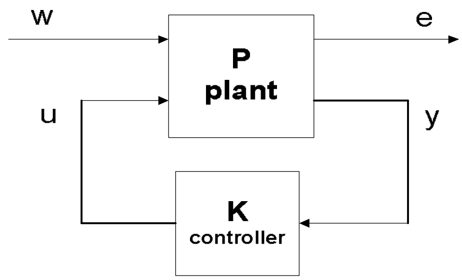

Fig. 4. Standard feedback configuration

Suppose that a state space realization for P can be written as

$$\dot{x} = C_1 x + B_1 w + B_2 u \quad (11)$$
$$e = C_1 x + D_{12} u,\ y = x \quad (12)$$

and assume that (A, B_2) is stabilizable, D_{12} has independent columns and the system with input u and output e has no zeros on the imaginary axis.

Theorem. Suppose $\gamma > 0$ is a given positive number. Let $H(\gamma) = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$ denote the Hamiltonian matrix with entries

$H_{11}=A-B_2(D_{12}^TD_{12})^{-1}D_{12}^TC_1$

$H_{12}=\gamma^{-2} B_1B_1^T-B_2(D_{12}^TD_{12})^{-1}B_2^T$

$H_{21}=-C_1^T(I- D_{12} (D_{12}^TD_{12})^{-1} D_{12}^T)C_1$

$H_{22}=-A^T+C_1^TD_{12} (D_{12}^TD_{12})^{-1} B_2^T$ (13)

Then, there exists a stabilizing controller K such that $||T_{ed}||_\infty < \gamma$ if and only if: i) $H(\gamma)$ has no eigenvalues on the imaginary axis and there exist a basis for the spectral subspace X-$H(\gamma)$ of $H(\gamma)$ of the form [X_1^T, X_2^T] where X_1 and X_2 are square matrices of appropriate di-

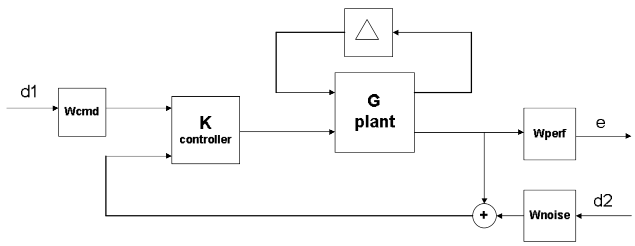

Fig. 3. Generalized Performance Block Diagram

mensions and X_1 is invertible. ii) $X(\gamma)=X_2X_1^{-1}$ is positive

ISSN 2229-5518

semi definite. In this case, one such controller is K(s) = F, where

$$F=-(D_{12}^TD_{12})^{-1}[D_{12}^TC_1+B_2^TX(\gamma)] \quad (14)$$

Existence and computation of $X(\gamma)$ is a standard matrix algebra problem that can be solved using a standard technique for solving Riccati equations based on the real Schur decomposition [9].

3.2 Particle Swarm Optimization

A particle swarm optimizer is a population based stochastic optimization algorithm modeled after the simulation of the social behavior of bird flocks. PSO is similar to genetic algorithm (GA) in the sense that both approaches are population-based and each individual has a fitness function. Furthermore, the adjustments of the individuals in PSO are relatively similar to the arithmetic crossover operator used in GA. However, PSO is influenced by the simulation of social behavior rather than the survival of the fittest. Another major difference is that, in PSO each individual benefits from its history whereas no such mechanism exists in GA. In a PSO system, a swarm of individuals (called particles) fly through the search space. Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself (i.e. its own experience) and the position of the best particle in its neighborhood. When the neighborhood of a particle is the entire swarm, the best position in the neighborhood is referred to as the global best particle and the resulting algorithm is referred to as a gbest PSO. When smaller neighborhoods are used, the algorithm is generally referred to as a lbest PSO. The performance of each particle (i.e. how much close the particle is to the global optimum) is measured using a fitness function that varies depending on the optimization problem.

The global optimizing model proposed by Shi and Eberhart [7] is as follows:

$$v_{i+1} = w \times v_i + RAND \times c_1 \times (P_{best} - x_i) + rand \times c_2 \times (G_{best} - x_i) \quad (15)$$

$$x_{i+1} = x_i + v_{i+1} \quad (16)$$

where v_i is the velocity of particle i, x_i is the particle position, w is the inertial weight. c_1 and c_2 are the positive constant parameters, Rand and rand are the random functions in the range [0,1], Pbest is the best position of the ith particle and Gbest is the best position among all particles in the swarm.

The inertia weight term, w, serves as a memory of previous velocities. The inertia weight controls the impact of the previous velocity: a large inertia weight favors exploration, while a small inertia weight favors exploitation [7]. As such, global search starts with a large weight and then decreases with time to favor local search over global search [7].

It is noted that the second term in equation (15) represents cognition, or the private thinking of the particle when comparing its current position to its own best. The third term in equation (15), on the other hand, represents the social collaboration among the particles, which compares a particle's current position to that of the best particle [8]. Also, to control the change of particles' velocities, upper and lower bounds for velocity change is limited to a user-specified value of Vmax. Once the new position of a particle is calculated using equation (16), the particle, then, flies towards it [7]. As such, the main parameters used in the PSO technique are: the population size (number of birds); number of generation cycles; the maximum change of a particle velocity Vmax and w.

Generally, the basic PSO procedure works as follows: the process is initialized with a group of random particles (solutions). The ith particle is represented by its position as a point in search space. Throughout the process, each particle moves about the cost surface with a velocity. Then the particles update their velocities and positions based on the best solutions. This process continues until stop condition(s) is satisfied (e.g. a sufficiently good solution has been found or the maximum number of iterations has been reached).

4 H$_\infty$ CONTROLLER DESIGNING PROCEDURE

4.1 Weight Selection and Building Model Uncertainty

To take advantage of H$_\infty$ design algorithm, we formulate the design as a closed loop gain minimization problem. So we select weighting functions that capture the disturbance characteristics and performance requirements to help normalize the corresponding frequency dependent gain constraints.

W_{cmd} is included in H$_\infty$ control problems that require tracking of a reference command. W_{cmd} shapes the normalized reference command signals into the reference signals that we expect to occur. It describes the magnitude and the frequency dependence of the reference commands generated by the normalized reference signal. Reference commands for underwater vehicles linear and angular speeds are usually flat. This means that underwater vehicle speed does not change frequently and has no high oscillations. Therefore, W_{cmd}, is selected equal to

$$W_{cmd} = \frac{A}{s+B} \quad (17)$$

where A and B are two constants that are determined using PSO.

W_{perf} weights the difference between the response of the closed loop system and the ideal model. Often we might want accurate matching of the ideal model at low frequencies and require less accurate matching at higher frequencies, in which case W_{perf} is flat at low frequencies, rolls off at first or second order, and flattens out at a small, nonzero value at high frequencies. Therefore, the error weights penalize setpoint tracking errors on u, v, w, p, q and r. Hence, W_{perf} is considered as follows, for all of them.

$$W_{perf} = \frac{C}{s+D} + E \tag{18}$$

as A and B; C, D and E are three constants that are found using PSO.

W_{noise} represents frequency domain models of sensor noise. Each sensor measurement feedback to the controller has some noise, which is often higher in one frequency range than another. The weighting function for the sensors would be small at low frequencies, gradually increase in magnitude as a first order or second order system, and level out at high frequencies. Then a high pass filter is selected for weighting functions of measured states.

$$W_{noise} = \frac{Fs}{s+G} \tag{19}$$

where F and G constants are found by PSO.

To complete the uncertainty model, changing of the underwater vehicle speeds due to vehicle parameters changing, that can be produced by hydrodynamic drag coefficients and propellers rotational speeds and external disturbance, should be considered in controller designing procedure. In this paper it performed by evaluating the underwater vehicle nonlinear behavior, when the mentioned vehicle parameters are changed reasonably, and mapping it to the linear model. Therefore, we will build an uncertainty model that matches our estimate of uncertainty in the physical system as closely as possible. Because the amount of the model uncertainty or variability typically depends on frequency, our uncertainty model involves frequency-dependent weighting functions to normalize modeling errors across frequency. The following frequency dependent weighting function for both linear and angular speeds is chosen.

$$W_{uncertainty} = \frac{Hs+I}{s+J} \tag{20}$$

where H, I and J constants are computed by PSO.

4.2 H$_\infty$ Controller Design and Simulation results

Now that all plant components, as illustrated in Figure 3, are described and nonlinearity uncertainties and the weighting functions are constructed. We can design a desired H$_\infty$ controller. By using *sysic* function of MATLAB Robust Control Toolbox, the weighted uncertain model is built. Nonlinearity uncertainties are modeled by using *ultidyn* function.

The weighting functions unknown parameters are computed using PSO. Therefore, minimizing a cost function, determining the vector P=[A, B, C, D, E, F, G, H, I, J] is the main purpose. For doing this, a performance index as a cost function- that should be minimized- must be selected. The performance criterion is defined based on some typical desired output specifications in the time domain such as overshoot M_p, rise time T_r, settling time T_s, and steady-state error E_{ss}. Therefore, in this paper, a time domain performance criterion defined by

$$\min_{K_{Q,R}} F(P) = \sum_{i=1}^{6}\sum_{j=1}^{6}\Big[\left(1-e^{-\alpha}\right)\times\left(T_{sij}+T_{rij}\right) + e^{-\alpha}\left(M_{pij}+E_{ssij}\right)\Big] \tag{21}$$

is used for evaluating the H$_\infty$ controller performance.
where M_{pij} is the maximum overshoot, T_{sij} is the settling time, T_{rij} is the rise time and E_{sij} is the integral absolute error of step response (i, j=1, 2, …, 6). Note that desired steady state of diagonal modes of the system (i.e. i=j) is 1 while for non-diagonal modes (i.e. i≠j) it is desired to be 0.

$\alpha \in [0,4]$ is the weighting factor. The optimum selection of α depends on the designer's requirement and the characteristics of the plant under control. We can set α to be smaller than 0.7 to reduce the overshoot and steady-state error. On the other hand, we can set α to be larger than 0.7 to reduce the rise time and settling time. If α is set to 0.7, then all performance criteria (i.e. overshoot, rise time, settling time, and steady-state error) will have the same worth.

The minimization process is performed using PSO algorithm. Step response of the plant is used to compute four performance criteria M_p, E_{ss}, T_r and T_s in the time domain. At first, the lower and upper bounds of the parameters are specified. Then a population of particles and a velocity vector are initialized, randomly in the specified range. Each particle represents a solution (i.e. weighting functions parameters P) that its performance criterion should be evaluated. This work is performed by computing M_p, E_{ss}, T_r, and T_s using the step response of the plant, iteratively. Then, by using the four computed parameters, the performance criterion is evaluated for each particle according. Then using equations (15) and (16) the next likely better particles (solutions) are determined. This process is repeated until a stopping condition is satisfied. In this stage, the particle corresponding to G_{best} is the optimal vector P. The optimal P is obtained as P=[0.15, 1.23, 98.47, 0.95, 0.11, 0.2, 1.51, 5.73, 1.29, 10.33].
After constructing the weights and the weighted plant, we have recast the control problem as a closed loop gain minimization. A gain minimizing controller for the uncertain plant can be computed by using *hinfsyn* function. By using this function, the desired H$_\infty$ controller (K in Figure 3) is obtained. The obtained controller has 12 inputs (plant noisy outputs and weighted setpoint commands), 6 outputs for control forces of the plant, and 18 states, with nominal performance $\gamma_{\min} = 0.65$. For model order reduction, *modred* and *balreal* commands are used. Small Henkel singular values indicate that the associated states are weakly coupled. With discarding these negligible Henkel singular values, the controller order is reduced to 11. Figure 5 shows the reduced order H$_\infty$ controller behavior (step response), when it is engaged with the linear plant. As illustrated in this figure, the H$_\infty$ controller can control the vehicle to track the desired speeds, efficiently.

To assess the behavior of the designed H$_\infty$ controller, it is embedded to the full nonlinear model of the underwa-

ISSN 2229-5518

ter vehicle as described in section 2.1 to form a closed loop system. Simulations are implemented in MATLAB Simulink. By the step response, the speed tracking quality is examined. Figure 6 shows the robust behavior of the designed controller against the nonlinearity of the nonlinear model. As shown in this figure, when a step is simultaneously commanded to the actuators, the proposed H_∞

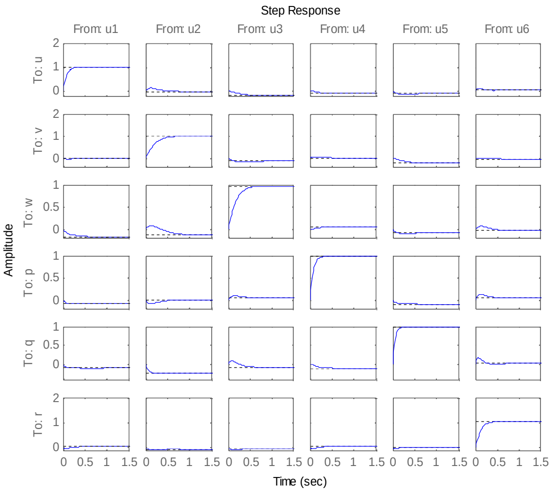

Fig. 5. Step response of linear model with reduced order H_∞ controller

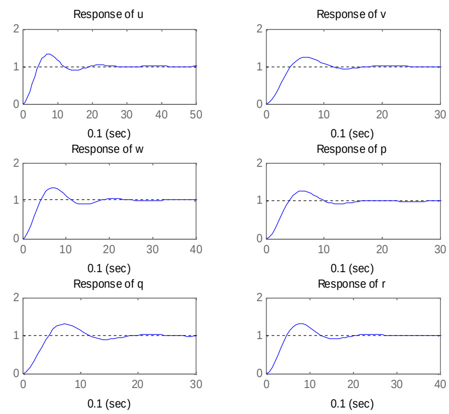

Fig. 6. Step response of the reduced order H_∞ controller integrated in the nonlinear model

controller can follow the signal with small errors. Furthermore, steady state and amplitude errors are desirably

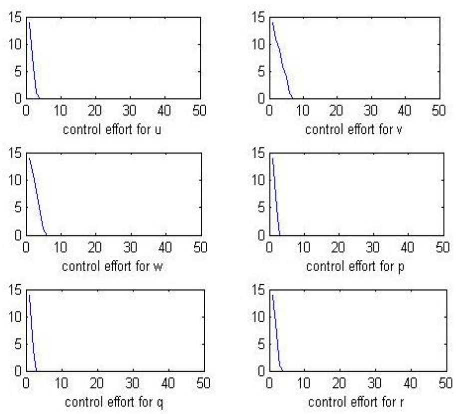

Fig. 7. Control efforts of the H_∞ controller embedded into the linear plant, (time unit is 0.1s)

small. This means that the designed controller can behave robustly against to the nonlinearities.

Figures 7 and 8 show the control efforts of the H_∞ controller with the linearized and the nonlinear models, respectively. As illustrated in the figures, the control effort of actuators reveals no saturation and so it is feasible to implement.

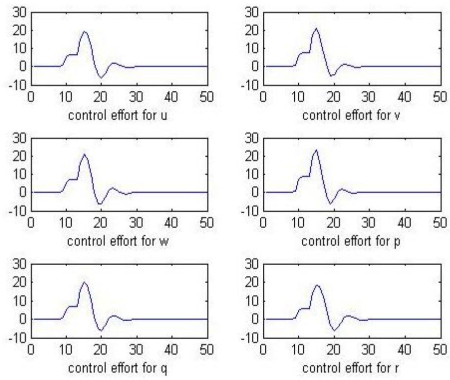

Fig. 8. Control efforts of the H_∞ controller embedded into the nonlinear plant, (time unit is 0.1s)

5 Conclusions

A robust H_∞ controller for underwater vehicles speed tracking is introduced, in this paper. Nonlinearity of nonlinear model is mapped onto the nominal linear model as uncertainties. Using frequency dependent weighting functions that are determined by PSO, tracking errors and noise errors are eliminated, robustly. The designed controller order is reduced. Using nonlinear simulations, robust behavior of the proposed controller is shown. The actuator control efforts were at the suitable rang for implementation. The future work can focus on control of underwater vehicles using nonlinear methods hybrid with intelligent techniques.

References

[1] Fossen T. I., *Guidance and Control of Ocean Vehicles*, John Wiley & Sons Ltd, 1994.

[2] D.R. Yoerger, and J. E. Slotine, "Robust Trajectory Control of Underwater Vehicles", IEEE Journal of Ocean Engineering 10 (4), 1985, pp. 462 -470.

[3] A.J. Healey and D. Lienard, "Multivariable Sliding Mode Control for Autonomous Diving And Steering of Unmanned Underwater Vehicles", IEEE Journal of Ocean Engineering 18 (3), 1993, pp. 327–339.

[4] Z. Feng, and R. Allen, "H_∞ Autopilot Design for an Autonomous Underwater Vehicle", Proceedings of the 2002 IEEE CCA / CACSD, 2002, pp. 350 -354.

[5] Z. Feng, and R. Allen, "Reduced Order H_∞ Control of an Autonomous Underwater Vehicle", Journal of Control Engineering Practice, 2004, pp. 1511-1520.

[6] R. P. Kumar, A. Dasgupta and C.S. Kumar, "Robust trajectory control of underwater vehicles using time delay control law", Ocean Engineering 34, 2007, pp. 842–849.

[7] Y. Shi, and A. R. Eberhart, "Modified Particle Swarm Optimizer", Proceedings of the IEEE conference on evolutionary computation. Piscataway, NJ: IEEE Press, 1998, pp. 69–73

ISSN 2229-5518

[8] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," In Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, vol.4, 1995, pp.1942-1948.

[9] A. Laub, "A Schur method for solving algebraic Riccati equations", IEEE Transactions on Automatic Control, AC-24(6), 1979, pp. 913-921.