International Journal of Computer Applications (0975 – 8887)
Volume 19– No.8, April 2011

Design of a PSO-based LQ Controller for Speed Tracking of Underwater Vehicles

Mohammd Pourmahmood Aghababa
Department of Electrical Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran

Mohammd Esmaeel Akbari
Department of Electrical Engineering, Ahar Branch, Islamic Azad University, Ahar, Iran

ABSTRACT
This paper solves a linear Quadratic (LQ) optimal control problem for underwater vehicles to track desired speeds. To obtain the linear controller, nonlinear model of the underwater vehicles is linearized around an operating point. After explaining both nonlinear and linearized models a brief description of LQ controller designing procedure is reviewed. It is assumed that all system states are available to measure. Weighting matrices (Q and R) of the linear quadratic performance index are tuned to a desired step response acquired, by using Particle Swarm Optimization (PSO) that minimizes a performance criterion. It is supposed that Q and R matrices are diagonal. After designing the LQ controller for the linear nominal model it is embedded to the nonlinear model. Using nonlinear simulations the speed tracking efficiency of designed LQ controller is shown. Control efforts of actuators reveal no saturation, therefore they are feasible to implement.

Keywords: Underwater Vehicles, Linear Quadratic, Speed Tracker, PSO.

1. INTRODUCTION
Nearly 70% of the earth is covered with oceans. They hold plenty of natural resources like oil, gas, minerals and fish, etc. Underwater vehicles help us to exploit these resources. They have became 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.

Automatic control of the underwater vehicles presents several difficulties due to the nonlinear behavior of a vehicle subjected to hydrodynamic forces and moments, 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.

Todate optimal control theory has been extensively used to solve various control engineering problems. Linear quadratic (LQ) optimal control procedure is based on minimization of a linear quadratic performance index representing the control objective. Unlike pole placement method, where the designer must know the exact pole locations, optimal control places the poles at some arbitrary points on the left hand of s-plane so that the resulting system is optimal in some sense. A linear quadratic state feedback regulator (LQR) problem is solved which assumes that all states are available for feedback.

Optimal state estimation (Kalman filtering) can be used to realize the autopilot in the case when not all states are measured. For instance, the LQG/LTR (Linear Quadratic Gaussian/ Loop Transfer Recovery) design methodology has been applied to underwater vehicles that two simple linear SISO (Single Input-Single Output) examples are used to demonstrate the proposed procedures [2]. LQG/LTR methodology is also applied to linear models of underwater vehicles [3]. A discrete LQR methodology is used to control a simple linear SISO underwater vehicle for a cable tracking problem in [4].

The ability of an underwater vehicle to maintain a desired and long endurance is a prerequisite for a successful operation. To maintain full trajectory control for such vehicles, it is necessary that desired speeds are followed. Position control can not be performed without appropriate speed tracking. Here, for an underwater vehicle with six degrees of freedom, both linear and angular speeds are considered to control. It is assumed that all states can be measured by suitable sensors, so that the state estimator is not required. The nonlinear model of the vehicle is linearzed around an operating point. Afterwards, an effective LQ controller is designed that its weighting matrices are found by Particle Swarm Optimization (PSO) which minimizes a time domain cost function. Using obtained controller, nonlinear simulations are assessed to test the performance of the controller over the nonlinear model. The necessary thrusters control efforts are feasible 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 LQ controller is designed with aid of PSO, and d) the linear controller efficiency is shown when it is set into the nonlinear model.

This paper is organized as follows. In Section 2 dynamics and motion equations of underwater vehicles are discussed. A linearized model for underwater vehicles is presented in Section 3. In Section 4, a brief review of particle swarm optimization is given. In Section 5, linear quadratic optimal control synthesis is explained. Section 6 deals with designing of the LQ controller for underwater vehicles speed tracking purpose. And some computer simulations are presented. Finally, in Section 7, this paper is ended with conclusions and some future works.

2. 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, $\dot{\eta}$,

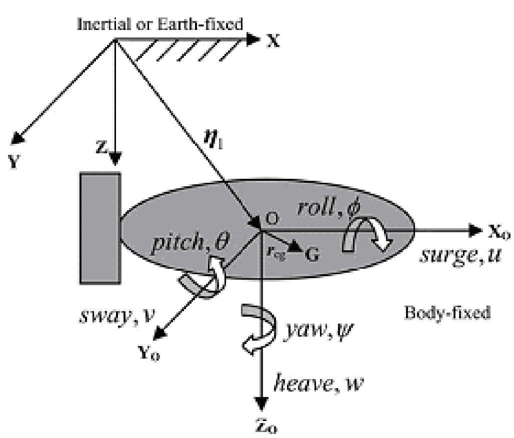

Fig 1: Inertial and body coordinate frames

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.

1) Mass and Inertia: 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: 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: 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.

Similar to added mass, 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. Low frequency friction will induce a linear term while high frequency effects will add a quadratic term.

Like potential damping, 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: The last term on the left-hand side of equation (1), $g(\eta)$, models the restoring forces which result from gravitation and buoyancy. Whereas the gravity force is a vector working along a line through the craft center of gravity, the buoyancy term is a force working along a line through the craft center of buoyancy. In general, those two points do not coincide and the restoring forces will introduce both forces and moments respectively along and about the three body axes.

5) Thruster model: 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]:

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

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.

$$J_0=V_a/nD \quad (7)$$

In the rest of this paper, the thrusters are assumed to be driven by DC motors. DC motors are usually controlled by velocity feedback, which means that the rotational velocity of the thruster tries to follow a reference velocity n_d. Therefore, n_i will be the physical input related to thruster number i. It can be also shown that an algebraic relation, although complicated, exists between the thrust of propeller i and the physical input. Therefore, the thrust will be chosen as input in the model $u_i = T_i$; This means that the model will be linear in the inputs. Here, it is assumed that six propellers are erected in six freedom degrees. So, the linear and angular speeds are controlled by these propellers. In fact, with changing in propellers rotational speeds, we can control the underwater vehicle speeds.

3. 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.

$$\dot{x} = f(x) + Bu, \; u = \tau \tag{8}$$

$$f(x) = M^{-1}(-C(x) - D(x) - g(\eta)), \; B = -M^{-1} \tag{9}$$

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 obtained:

$$x_0=(1, 1, 1, 1, 1, 1) \tag{10}$$

The linearized model is:

$$\Delta\dot{x} = A\Delta x + B\Delta u, \; \Delta y = C\Delta x$$

$$x = [u, v, w, p, q, r]^T, \; u = [\tau_1, \tau_2, \tau_3, \tau_4, \tau_5, \tau_6]$$

$$y = [\mathrm{u, v, w, p, q, r}]^T$$

(11)

where A and B are 6× 6 matrices and C is a 6×1 vector and τ_i i=1, 2, …, 6 are the propeller forces, [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 Figure 2. As it is seen in this figure, the step response is not tracked and system modes are not decoupled.

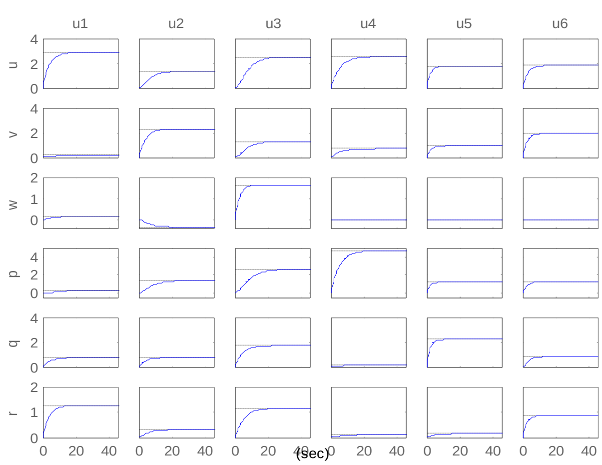

Fig 2: Step response of open loop linear model

4. 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 [6] is as follows:

$$\mathrm{v_{i+1} = w \times v_i + RAND \times c_1 \times (P_{best} - x_i) + rand \times c_2 \times (G_{best} - x_i)} \tag{12}$$

$$\mathrm{x_{i+1} = x_i + v_{i+1}} \tag{13}$$

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 i^{th} 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 [6]. As such, global search starts with a large weight and then decreases with time to favor local search over global search [6].

It is noted that the second term in equation (12) represents cognition, or the private thinking of the particle when comparing its current position to its own best. The third term in equation (12), on the other hand, represents the social collaboration among the particles, which compares a particle's current position to that of the best particle [7]. 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 (13), the particle, then, flies towards it [6]. 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 i[th] 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).

5. LINEAR QUADRATIC (LQ) TRACKER

In this section we will briefly review some results from Athans and Falb [5] on linear quadratic (LQ) optimal control theory. Consider a linear controllable system (underwater vehicle linearized model is controllable) with state x(t) $\in R^n$, input u(t) $\in R^\gamma$ and output y(t) $\in R^m$. The system performance output y is given by:

$$\dot{x} = Ax + Bu, \;\; y = Cx + Du \tag{14}$$

x(t) is assumed to be measured by sensors. Our control objective is to design an optimal controller to track a desired speed $y_d(t)$. For this purpose, we will define an error vector:

$$\hat{y} = y - y_d = C(x - x_d) \tag{15}$$

where $x_d(t)$ is the desired state. It can be shown that the optimal control law utilizing feedback from x(t) and feedforward of $y_d(t)$ can be obtained by minimization of a quadratic performance index:

$$\min J = \frac{1}{2}\int_0^T (\hat{y}^T Q \hat{y} + u^T R u)dt \tag{16}$$

where R>0 and Q$\geq$0 are the weighting matrices. Using some mathematical operations it can be showed that the optimal input is obtained as follows [5]

$$u = -R^{-1}B^T(Sx + h_1 + h_2) \tag{17}$$

where h_1 and h_2 are obtained from equations (18) and (19). S must be solved from Riccati equation (20), as follows:

$$\dot{h}_1 + (A - BR^{-1}B^T S)^T h_1 - Qx_d = 0 \tag{18}$$

$$\dot{h}_2 + (A - BR^{-1}B^T S)^T h_2 = 0 \tag{19}$$

$$\dot{S} + SA + A^T S - SBR^{-1}B^T S + \tilde{Q} = 0 \tag{20}$$

where $\tilde{Q} = C^T QC \geq 0$ and boundary conditions are simply yield:

$$h_1(T)=0,\; h_2(T)=0,\; S(T)=0 \tag{21}$$

Hence, the differential equations for S, h_1 and h_2 can be solved for all $t \in [0, T]$ by integration.

For $T \to \infty$ the solution of Riccati equation (20) will tend to the constant matrix S_∞ satisfying:

$$S_\infty A + A^T S_\infty - S_\infty BR^{-1}B^T S_\infty + \tilde{Q} = 0 \tag{22}$$

This solution is interpreted as the steady-state solution of Riccati equation (18). Furthermore, x_d is assumed to be constant. This assumption can be correct, because in many applications. The speed of underwater vehicles has slowly-varying states. For $T \to \infty$, an optimal input can be obtained as follows.

$$U=K_1x+K_2y_d \tag{23}$$

where

$$K_1=-R^{-1}B^TS_\infty,\; K_2=-R^{-1}B^T(A+BK_1)^{-T}C^TQ \tag{24}$$

The solution is shown in Figure 3.

6. DESIGN OF AN LQ CONTROLLER FOR UNDERWATER VEHICLES SPEED TRACKING

LQ method is rooted in optimal control theory and in spite of systematic design procedure, shows some useful properties of efficiency and good performance. The objective is to design an optimal control law to provide speed tracking of a linearized model of underwater vehicles using the propeller forces, as the inputs.

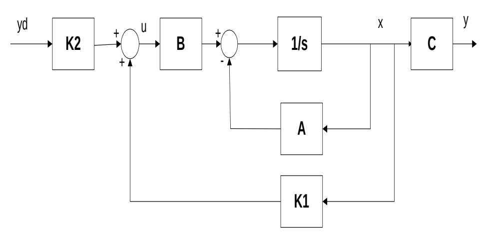

Fig 3: Linear Quadratic optimal control block diagram

As shown in Figure 2, open loop step response is not suitable, because there is coupling between channels and the step command is not followed. To achieve a suitable step response with no overshoot, no steady state error and less settling time, an LQ problem is solved for the linearized model. First the weighting matrices, Q and R, should be selected. As first attempt, Q and R matrices are considered to be represented by identical matrices. Using *lqr* function in MATLAB 7 software, K_1 and K_2 matrices are computed. The step response of the closed loop system is plotted in Figure 4. As illustrated in this figure, step response has not appropriate characteristics: there is coupling between channels and the step command is not tracked, accurately.

In this paper, it is assumed that Q and R 6*6 to be diagonal matrices. Therefore, minimizing a cost function, determining the vector $K_{Q,R}$=[Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, R_1, R_2, R_3, R_4, R_5, R_6], where Q_i and R_i (i=1, 2, …,6) are diagonal elements of Q and R, is the main purpose. The performance criterion or cost function is

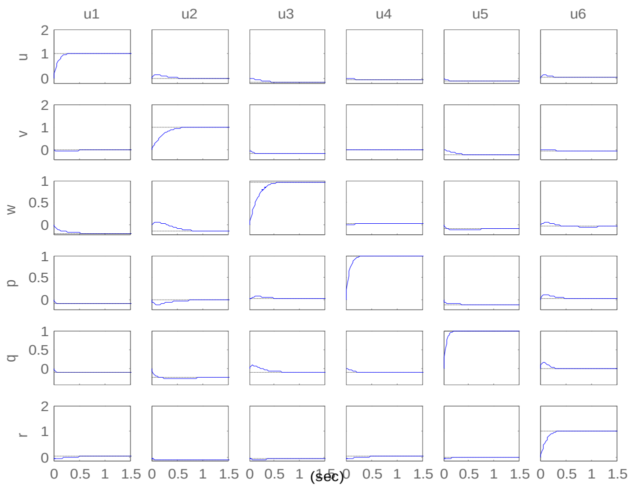

Fig 4: Step response of final system with R=Q=I

defined based on some typical desired output specifications in the time domain such as overshoot, rise time, settling time, and steady-state error.

Therefore, in this paper, a time domain performance criterion defined by

$$\min_{K_{Q,R}} F(K_{Q,R}) = \sum_{i=1}^{6}\sum_{j=1}^{6}\Big[\left(1-e^{-\alpha}\right)\times\left(T_{sij}+T_{rij}\right) + e^{-\alpha}\left(M_{pij}+E_{ssij}\right)\Big] \qquad (25)$$

is used for evaluating the LQ 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. For this purpose, step response of the plant is used to compute four performance criteria overshoot (M_p), steady-state error (E_{ss}), rise time (T_r) and setting time (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. Q and R matrices diagonal elements $K_{Q,R}$) 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 (23) and (24) 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 $K_{Q,R}$.

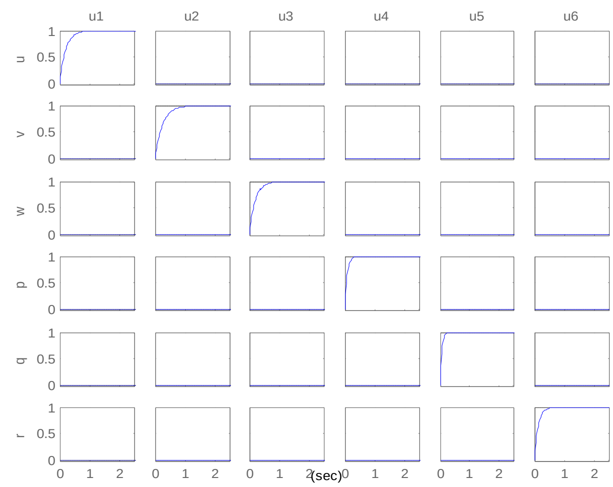

Fig 5: Step response of final system with Q and R found by PSO

Figure 5 shows the step response of the final closed loop system. Comparing figures 4 and 5, we can realize the relation between the cost function gains and overcoming the coupling effects. This means that an appropriate selection for Q and R is led to the efficient decoupled response. It can be concluded that the final step response is suitable and it has good characteristics such as less settling time, no overshoot and correct final value.

Now, to assess the behavior of the designed LQ controller, it is embedded into the full nonlinear model of the underwater vehicle to form a closed loop system. Using the step command, the speed tracking quality is evaluated. Figure 6 shows the step response of the designed controller, when it is engaged into the nonlinear model. As shown in this figure, the proposed small errors. In particular, the steady state error is very small. This means that the designed linear controller can behave efficiently versus the model nonlinearity.

Figures 7 and 8 show the control efforts of the LQ controller embedded into the linearized and the nonlinear models, respectively. As illustrated in these figures, the control effort of actuators reveals no saturation and so is feasible to implement.

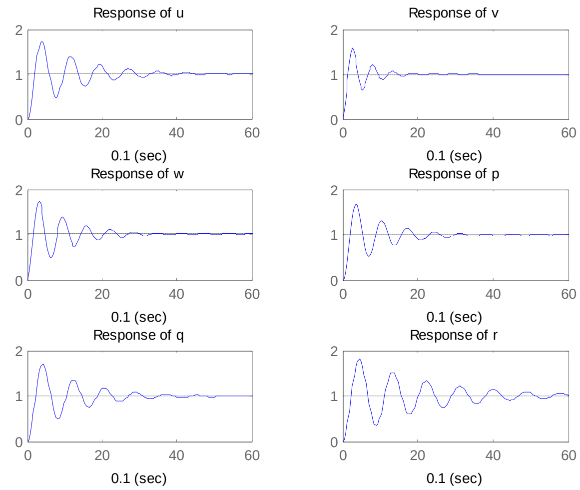

Fig 6: Step response of LQ with nonlinear model

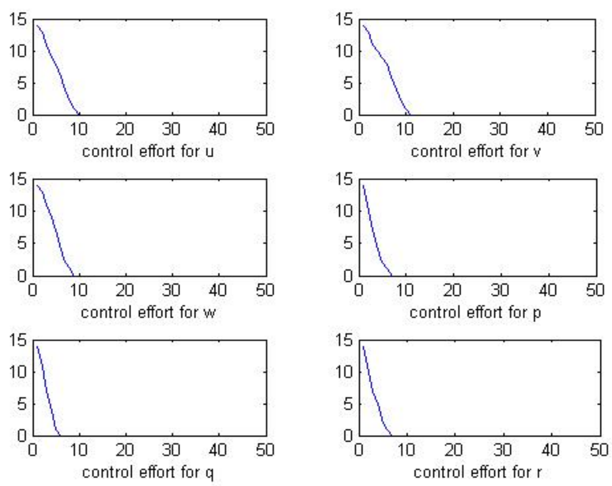

Fig 7: Control efforts of the LQ controller embedded into the linear plant, (time unit is 0.1s)

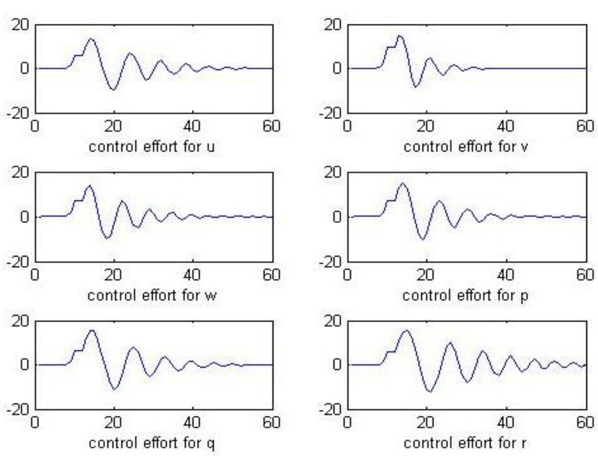

Fig 8: Control efforts of the LQ controller embedded into the nonlinear plant, (time unit is 0.1s)

7. CONCLUSIONS

An LQ controller for underwater vehicles speed tracking was introduced in this paper. Using particle swarm optimization the suitable weighting matrices were obtained. A performance index function was defined to evaluate each of the candidate Q and R matrices. The introduced controller was evaluated with linear and nonlinear models. Linear quadratic with minimizing an index performance can use of the states feedback to desired speed tracking. Simulation results showed the efficiency and effectiveness of the controller performance. The proposed controller can follow the step commands, well. The actuator control efforts were at the appropriate and acceptable rang for implementation. The future work can focus on control of underwater vehicles using nonlinear techniques hybrid with intelligent manner.

8. REFERENCES

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

[2] Triantafyllou, M. S., and Grosenbaugh, M. A. 1991. Robust Control for Underwater Vehicle Systems with Time Delays, IEEE Journal of Oceanic Engineering, vol. 16, no. 1, 146-151.

[3] Juul, D. L., McDermott, M., Nelson, E. L., Barnett, D. M. and Williams, G. N. 1994. Submersible Control Using the Linear Quadratic Gaussian with Loop Transfer Recovery Method, In Proceedings of the Symposium on Autonomous Underwater Technology, Cambridge, MA, USA, 417-425.

[4] Neam, W., Sutton, R., and Ahmad, S. M. 2003. LQG/LTR Control of an Autonomous Underwater Vehicle Using a Hybrid Guidance Law, Marine and Industrial Dynamic Analysis Group, the University of Plymouth, IFAC, 2003.

[5] Athans, M., and Falb, P. L., 1966. *Optimal Control*, McGraw-Hill Book Company, New York.

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

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

[8] Healey, A. J., and Lienard, D. 1993. Multivariable Sliding Model Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles, IEEE Journal of Oceanic Engineering, vol. 18, no. 3**,** 327-339.