Journal of Advances in Computer Research
Quarterly ISSN: 2008-6148
Sari Branch, Islamic Azad University, Sari, I.R.Iran
(Vol. 5, No. 2, May 2014), Pages: 69-88
www.jacr.iausari.ac.ir

IPSO-SQP Algorithm for Solving Time Optimal Bang-Bang Control Problems and Its Application on Autonomous Underwater Vehicle

Tahereh Taleshian✉, Abolfazl Ranjbar Noei, Reza Ghaderi

Electrical Engineering Dept., Babol University of Technology, Babol, Iran

t_taleshian@yahoo.com; a.ranjbar@nit.ac.ir; r_ghaderi@nit.ac.ir

Received: 2014/01/02; Accepted: 2014/01/21

Abstract

In this paper, an integration of Improve Particle Swarm Optimization (IPSO) in combination with Successive Quadratic programming (SQP) so called IPSO-SQP algorithm is proposed to solve time optimal bang-bang control problems. The procedure is found not sensitive to the initial guess of the solution. Due to random selection in the first stage of the search process, the chance of converging to the global optimum is significantly increased, without sticking in a local optimum. The combined technique gains both advantages of its original algorithms. The IPSO directly minimizes the cost function without the need for gradient-based techniques. The performance of the outcome will be increased when the SQP immediately undertakes the optimization task. This is shown via applying those on some other nonlinear systems. Consequently, the proposed algorithm is successfully applied on a time optimal bang-bang control of an autonomous underwater vehicle. A pitch-programming task is also investigated for the autonomous underwater vehicle by designing an optimal PID controller.

Keywords: *Autonomous Underwater Vehicle, IPSO-SQP Algorithm, Optimal PID Controller, Pitch Programming, Time Optimal Bang-Bang Control*

1. Introduction

One of the most common type of control input is piecewise (constant) function which consist of a sequence of (fixed) inputs that is applied to systems with appropriate switching times. In many mathematical models of the mechanical systems, the control input is of bang-bang type. The bang-bang solution may also be encountered in some optimal control problems. A special situation arises when the Hamiltonian is linear in terms of the control input and the response is also nonsingular [1]. Using the bang-bang Controller changes the problem to finding the switching times [2], [3], [4], [5]. Mohler in [5] and [6] presented a bang-bang control algorithm called switching time variation method (STVM) that requires information of the number of switching and the switching times as initial guesses. It generates a sequence of switching functions whilst computes the gradient of the cost function with respect to the switching times. Using this gradient information, the switching times are corrected at any iteration. During the correction process, a careful selection of the step size is crucial because the Hessian of the cost function is not estimated. In [7], the Switched Time Optimization (STO) algorithm is used for time optimal control of a two-linked manipulator. The STO algorithm needs a good initial guess for the switching times to converge to a global minimum. In [8] primarily the switching time computation (STC) method is proposed to assess the time of the switching. The work was followed to minimize the final time in a time optimal bang-bang control problems. In [9] a general algorithm for Time Optimal Switching control (TOS algorithm) is proposed of nonlinear systems using a single control input. Primarily the STC method is used to find a feasible switching control then the TOS algorithm uses this information as an initial guess to solve the time optimal bang-bang control problem. In [10] a method is proposed to use a mathematical programming formulation to solve the bang-bang constrained optimal control problems. This method not only gives what STC and TOS algorithms can give together, but also assesses sufficient conditions for a local minimum. Unfortunately, this algorithm needs a good initial guess otherwise; the algorithm may converge to a local minimum. Generally, gradient-based methods have the possibility of getting trapped at local optimum depending on the initial guess of the solution. In order to achieve a good result, these methods require very good initial guesses of the solution. Besides, as the complexity of the system increases, the specification of a suitable initial guess can become troublesome [11]. Thus global optimal control methods such as genetic algorithm (GA), particle swarm optimization (PSO), differential evolution (DE) and *etc.* can be used to find the global optimum or a sufficiently close approximation. In the heuristic algorithms, the cost function's gradient is not required. They are not sensitive to initial guess of the solution and they usually do not get stuck into a local optimum. Based on these advantages, they have been successfully applied in many optimal control problems [11], [12], [13]. In this paper, the IPSO-SQP algorithm is used to overcome the shortages of traditional methods in time optimal bang-bang control problems. First, an improved PSO (IPSO) algorithm is proposed to enhance global search ability and convergence speed of PSO. Second, to achieve faster convergence speed around global optimum and higher convergence accuracy, the IPSO is combined with successive quadratic programming (SQP) algorithm.

In recent years, AUVs have become an intense area of oceanic research because of their emerging applications, such as deep sea inspections, underwater pipelines tracking, fish tracking and different application in military industry, *etc* [14], [15]. These vehicles are controlled autonomously. The required energy, which is carried on board provides wider scope of operation in comparison with the other type of underwater vehicles such as ROVs. Despite of the complexity in the structure and the difficulty to control and due to capability of the AUVs, these are still of the researcher's interest. The task to be controlled here is the trajectory of an AUV from the depth lower than 50 meters towards the surface. It is necessary that the AUV has to come back to the nest where is located on the sea surface in minimum time. The work will be followed when a PID controller with optimal coefficients gain is tuned for the pitch-programming task. This paper is organized as follows:

In section2, a time optimal bang-bang control problem is addressed and the proposed IPSO-SQP algorithm for solving time optimal bang-bang control problem is introduced. The application of this algorithm in a time optimal bang-bang control of three nonlinear systems (Van Der Pol, Rayleigh and F8 aircraft) is presented in section3. The advantage of the proposed method is discussed via a comparison study with some other similar methods. A simulation study shows the significance of the proposed IPSO-SQP algorithm for solving time optimal bang-bang control problem. In Section4 to assess the performance of the proposed method, it is used on a time optimal bang-bang control of an AUV. Artificial tuning of a PID controller designation for pitch programming is performed in section 5. Finally, the work will be closed by a conclusion in section6.

1. IPSO-SQP ALGORITHM FOR SOLVING TIME OPTIMAL BANG-BANG CONTROL PROBLEMS

In this paper, a configuration of IPSO and SQP algorithms –so called IPSO-SQP, is used in a time optimal bang-bang control problem. In which there is no need to a have good initial guess of the solution. The IPSO is gained to solve nonlinear optimal control problems [11]. This has shown to have rapid convergence to a near optimum solution. In fact, the search process becomes very slow around the global optimum. On the other hand, the SQP algorithm is weak to escape from the local optimum. Since the convergence speed and the accuracy to reach to the global optimum are more significant, it is meaningful to combine two describe techniques. A combination process of these two methods is as follows:

First, the IPSO algorithm is used to find a near optimum solution. Thereafter the search process immediately switches to the SQP algorithm to gain the higher converge rate to achieve global optimum. More details are presented in the following sections.

2.1 IPSO algorithm

Particle swarm optimization (PSO), as an effective heuristic optimization technique, is based on simulating of the movement and flocking of birds. Eberhart and Kenedy first improved this algorithm in 1995 [16]. The PSO uses the concept of social mutual effect to solve an optimization problem. In the PSO, particles move in the search space to find best solution. Each particle is considered as a point in an N-dimensional space. The flight is updated according to the past experience of the particle and also other birds

In the search space, each particle continues the flight according to the best solution that has been achieved so far by this particle personally. This value is called the personal best (pbest). The other trajectory where the PSO follows is the best value that has been achieved so far by each particle in the vicinity of that particle. This value is called the global (or local) best (gbest or lbest). The main concept of the PSO is involved with the acceleration of each particle towards the pbest and the gbest (lbest) using an inertia weight. In the beginning of the search process initial population is randomly created in the given search space. Each particle has its own velocity vector, which is updated at any iteration. The updating velocity equation is as follows:

$$v_i^{k+1} = \omega v_i^k + c_1 r_1 \left(pbest_i^k - x_i^k \right) + c_2 r_2 \left(gbest^k - x_i^k \right) \qquad (1)$$

where x_i^k is the position of the i^{th} particle in k^{th} iteration, ω is the inertia weight. Coefficients c_1 and c_2 are the acceleration multipliers, r_i is a random uniformly distributed number in the range $[0,1]$. When the velocity is evaluated from (1) the position of every particle is updated as follows:

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

The update laws are repeated until a stopping criterion in the algorithm is met. Preventing the PSO algorithm to stick in a local minimum, a weighting factor [11] is proposed in Eq. (3) which is updated as follows:

$$w_i^k = \frac{1}{\left(1+\exp(-\alpha F(pbest_i^k))\right)} \quad (3)$$

considering $\alpha = 1/F(gbest)$ where $F(pbest_i^k)$ and $F(gbest)$ are the fitness value of the personal and global best respectively. Therefore, the algorithm is ultimately called IPSO.

2.2 SQP algorithm

SQP is an iterative analytical nonlinear programming method. This technique begins from an initial point to find a solution using the gradient information. This optimization method is faster than other population based search algorithms. Although the SQP method is highly dependent on the initial estimate of the solution [17], [18], this has successfully applied in some optimal control problems [19], [20].

The SQP method is based on an iterative formulation together with the solution of some other quadratic programming sub problems. The optimization problem in SQP method is considered as follows:

$$\begin{cases} \text{minimize: } J(x) \\ \text{subjected to: } \psi_i(x) \le 0, \quad i = 1, 2, \ldots, l \end{cases} \quad (4)$$

where $J(x)$ is the cost function and $\psi_i(x)$ stands for the constraint. In this regard the Lagrangian function $L(x, \lambda)$ is constructed in terms of the Lagrangian multiplier λ_i, the cost function together with the constraint which is as follows:

$$L(x, \lambda) = J(x) + \sum_{i=1}^{m} \lambda_i \psi_i(x) \quad (5)$$

In fact, the SQP consists of three main parts:

1- Update the Hessian of the Lagrangian function according to:

$$H_{k+1} = H_k + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k^T H_k}{s_k^T H_k s_k} \quad (6)$$

2- Solve the quadratic programming sub-problem:

$$\min \ \frac{1}{2} d_k^T H_k d_k + \nabla f(x_k)^T d_k \quad (7)$$

$$\nabla \psi_i(x_k)^T d_k + \psi_i(x_k) = 0 \quad i = 1, \ldots, m_e$$

$$\nabla \psi_i(x_k)^T d_k + \psi_i(x_k) \ge 0 \quad i = m_e, \ldots, m$$

3- Apply a linear search to find a solution for the next iteration:

$$X_{k+1} = X_k + \alpha d_k \quad (8)$$

The algorithm is repeated until the stopping criterion (maximum iteration or convergence criterion) is met. The step length parameter α_k is determined via a linear search procedure.

2.3 IPSO-SQP Algorithm for Solving Nonlinear Optimal Control Problems

In the following a brief review of the IPSO-SQP algorithm is expressed. For more details, one may refer to [11]. In this method, primarily a group of particles is randomly initiated. The IPSO is executed to find a global kind best position. Then the routine is switched to the SQP algorithm to search around the found global best. This is written here:

Step 1: Initialize the position and velocities of particles, using uniformly distribution random number.

Step 2: Evaluate the fitness value for each particle.

Step 3: If the maximum iteration is arrived, go to step7, else, go to step4.

Step 4: The global best is stored. If the change between the current global best fitness value and its previous one is smaller than a predefined value, go to step7 else continue.

Step 5: The velocities and position of all particles are updated according to Eq.(1) and (2).

Step 6: The inertia weight for each particle is updated according to Eq.(3) and go to step2.

Step 7: Switch to the SQP algorithm to search around the global best, which is found by IPSO. In this case, the best solution obtained by IPSO is considered as an initial guess for the SQP algorithm.

2.4 Problem Statement

The task is to guide a system from a given initial state to a target in a minimum time, using a bang-bang control. Consider the following nonlinear dynamic:

$$\dot{x} = f\left(x(t), u(t)\right) \tag{9}$$

where $x(t) \in \mathbb{R}^n$ is the state vector, $u:\left[0, t_f\right] \to U \subset \mathbb{R}$ is a finite time control input and $f: \mathbb{R}^n \times U \to \mathbb{R}$ is a vector field. The goal is to steer the system to a desired target by using some piecewise constant control, namely a finite sequence of constant input $\{u_1, \ldots, u_{N+1}\}$ where N is the number of switching. In other words, arcs are generated by the given constant inputs. These arcs are to be concatenated in a prescribed order such that to reach the target state. A bang-bang control input is defined as follows:

$$u(t) = u_i \quad \text{in the } i^{th} \text{ arc} \tag{10}$$

where i^{th} arc is the segment of the trajectory $x(t)$ $t \in \left(t_{i-1}, t_i\right)$, $i = 1, \ldots, N$ and t_i is the switching time. A concatenation of these arcs from x_0 to a target state x_T is schematically shown in Figure1.

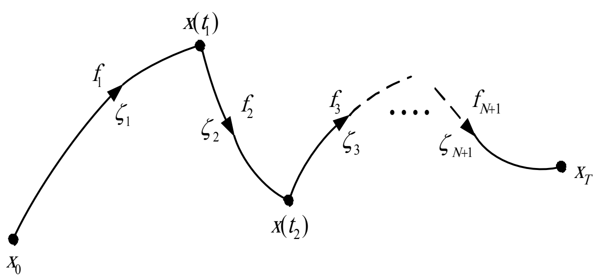

Figure1. A concatenation of arcs from x_0 *to* x_T

The time duration of each arc is defined as follows:

$$\zeta_i = t_i - t_{i-1}, \quad i = 0, 1, \ldots, N+1 \tag{11}$$

where ζ_i is the time required to travel along with the i^{th} arc. The vector of $\zeta \in R^{N+1}$ is defined by:

$$\zeta = [\zeta_1, \ldots, \zeta_{N+1}] \qquad (12)$$

Given the number of switching and the magnitude of the control input for each arc, the problem is reduced to final time (summation of the arc times) optimization subjected to equality constraint $x(t_f) = x_T$. By using the penalty method [21], the constraint is added to the cost function as a penalty function. This changes the constrained problem to an unconstrained one. Thus, the time optimal bang-bang control problem is reduced to:

$$P\left\{ \text{minimize } \sum_{i=1}^{N+1} \zeta_i + \alpha_i \sum_{j=1}^{n} \left(x_j(t_f) - x_{T_j} \right)^2 \right. \qquad (13)$$

where α_i, $i = 1, 2, \ldots, N+1$ is the weighting factors. $x_j(t_f)$ and x_{T_j}, $j = 1, 2, \ldots, n$ are the actual and desired value of the j^{th} state variable respectively. Now, the problem is to find:

- The number of switching N,
- The value of the control input in each arc
- and the time duration of each arc ζ_i, $i = 1, 2, \ldots, N$

such that the cost function in (13) is minimized. This is shown to be performed through using IPSO-SQP method which is described in the following.

2.5 IPSO-SQP Algorithm In Time Optimal Bang-Bang Control Problems

In this method by inspiring the concept in [7], the number of switching and the magnitude of the control input for each arc are determined. The switching times will be computed gaining the IPSO-SQP algorithm to take the system to the target from a given initial state in a minimum time. The discrepancy from the desired trajectory is added as a penalty function to the cost function (Eq.(13)). The particles in IPSO-SQP algorithm are treated as arc times. Thus, the dimension of each particle is $N+1$, where N is the number of switching. For example if the initial switching number is guessed $N = 3$ with $u(0) = u_{\max}$, then the arc times and the control input will be of the form $\zeta = \{\zeta_1, \zeta_2, \zeta_3, \zeta_4\}$ and $u(t) = \{u_{\max}, -u_{\max}, u_{\max}, -u_{\max}\}$ respectively. Each dimension of a particle is bounded. Since time is a nonnegative quantity, the lower bound of each dimension should be considered zero. However, the value of zero yields numerical errors, thus preventing such errors the lower bound is considered 10^{-6} which is very close to zero. Determining the upper bound value depends on designer. If a designer is familiar with the problem, he or she can choose a value for the upper bound, which is very close to the real solution. Otherwise, it is better to choose the upper bound large enough to annihilate the guessing error through running the algorithm. It must be noticed that assigning a very large value to the upper bound may lead the algorithm to converge through more number of iterations. After determining the bounds, the velocity and the position of a group of particles are randomly initialized. The IPSO algorithm is executed to search for the global best position. The SQP algorithm is then used to search around the found global best. In the following, the procedure of the IPSO-SQP algorithm for solving the time optimal bang-bang control is presented.

Step 1: Guess the number of switching *N*.

Step 2: Set the initial value of the control input $u(0) = +u_{\max}$.

Step 3: Find a possible solution with N times switching using the IPSO-SQP method and set $i=1$.

Step 4: Find a time optimal solution for $N+i$ and $N-i$ switching using the IPSO-SQP method.

Step 5: In the case of no improvement on t_f, keep the obtained solution in step3 as a possible optimal solution and let the algorithm continue. Otherwise, set $i=i+1$ and go to step4.

Step 6: if $u(0)=-u_{\max}$ assign a label S_2 to the solution and continue; else, the label of the solution will be assigned as S_1. Set $u(0)=-u_{\max}$ and go to step3.

Step 7: Among the sets of the solution S_1 and S_2, select the answer with the minimum time t_f and regard it as the desired solution and stop.

To use this procedure, primarily the value of the switching number is guessed. Then for an initial control input ($u(0)=+u_{\max}$), the algorithm is run to find the optimal arc times. If for a special switching number N, the final time is improved then the algorithm searches for the best solution for $N+1$ and $N-1$ switching. Again, the solution is checked to find a possible improvement. Detecting any improvement, the algorithm is executed again for $N+2$ and $N-2$ switching. The procedure continues until improvement in final time is not detected. In this case, the best result that has been achieved so far is stored and the algorithm is run for the other initial control input ($u(0)=-u_{\max}$) and the same initial guess of the switching number (N). Similarly, steps of the algorithm are repeated for the new guess. Consequently, the best results achieved for each initial control input are compared and the one, which yields less final time, is considered as the solution of time optimal bang-bang control problem.

Through using the proposed IPSO-SQP algorithm for solving time optimal bang-bang control problems a good initial guess for starting the algorithm is prevented. In contrast to STO method, there is no need to use any additional method to find a suitable start point. Moreover, the IPSO-SQP algorithm has a simple code and it is very easy to deal with also, it can be applied for wider range of problems. The hybrid configuration makes it a powerful algorithm, which rarely get stuck in the local optima. In the following, to verify the performance of the proposed algorithm it is used in time optimal bang-bang control of some nonlinear systems.

2. Application Of IPSO-SQP Algorithm In Time Optimal Bang-Bang Control Of Some Nonlinear System

In this section, the IPSO-SQP algorithm is applied on a time optimal bang-bang control of the Van Der Pol equation, Rayleigh system and an F8 aircraft model. The results are compared with those of obtained in [9] and [10].

3.1 Van der Pol Equation

A controlled Van Der Pol equation with the effort u is expressed as follows:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 - \left(x_1^2 - 1\right) x_2 + u \end{aligned} \tag{14}$$

where u is assumed to be of the bang-bang, namely $u \in \{-1,1\}$. The goal is to steer the states from the initial point $x_0 = [1,1]$ to the target point $x_T = [0,0]$ in minimum time. The

STO method was priory used in time optimal bang-bang control of Van Der Pol equation in [9]. In the following the results obtained using STO method is presented then the proposed IPSO-SQP method is also used for time optimal bang-bang control of Van Der Pol equation.

❖ ***STO Method***

As it is mentioned in the introduction of this paper, The STO algorithm is very sensitive to the initial guess of the solution thus, the STC method is used primarily to assess an appropriate start point. The STC method set $u(0)=1$ and found the results as follows [9]:

$\zeta=[3.92540,\ 0.43500,\ 3.28560]$, $t_f=7.64600$

The distance from the origin was also 0.00037. Using the outcome of the STC technique as an initial guess of the STO, the following result is achieved [9]:

$\zeta=[0.7230,\ 2.37220]$, $t_f=3.09520$

These results provide a fewer distance to the final states from the origin of order 10^{-4}. In fact, the length of third arc was found zero which decreases the time significantly. However, this algorithm [9] is found sensitive to the initial guess of the solution. A misappropriate choose of the initial guess may lead the algorithm to converge to a local minimum. Thus, it needs an additional algorithm (STC method) to assess an appropriate start point.

❖ ***IPSO-SQP Algorithm***

The IPSO-SQP is implemented on Van Der Pol equations. To provide a chance for the IPSO algorithm to converge, the size of the population of swarm is assumed 30 whilst both c_1 and c_2 are set to 2.1. The number of switching is initially assumed $N=4$. Hence, the number of arcs has to be one more than the number of the switching. Accordingly, the dimension of the optimization problem *i.e.* the number of arcs is equal to five. In order to show that the algorithm is not sensitive to the initial guess, the range of the arc times in each dimension is assumed $\zeta_i\in[10^{-6},10]$, and initially the particles are randomly distributed in this range. The lower bound is chosen as 10^{-6} not to be zero due to numerical consideration. The IPSO search algorithm is switched to the SQP method, when the change in the cost function value is achieved less than 0.0001 after 10 iterations.

The algorithm is run for the initial guess of $N=4$ while the initial control input is assumed $u(0)=1$. In step4, the switching number $N=3$ and $N=5$ is also tried for possible better solution. The improvement in final time is detected for $N=3$ thus the value of i is increased and the algorithm goes back to step4 to find the best solution for $N=2$ and $N=6$. This time the value of final time does not improve thus the algorithm goes to step6 to repeat the same procedure for $\{u(0)=-1,\ N=4$. In step3, the algorithm is run for $\{u(0)=-1,\ N=4$. Then in step4, the algorithm is performed for $\{u(0)=-1,\ N=3,\ N=5$. Final time is improved for $N=3$ hence, the value of i is increased and the algorithm goes to step4 to search the solution for $\{u(0)=-1,\ N=2,\ N=6$. Again, the final time is improved and as a result, the value of i is increased. In step4, the best solution is search for $\{u(0)=-1,\ N=1,\ N=7$. Improvement in final time for $\{u(0)=-1,\ N=1$ causes the algorithm to try $\{u(0)=-1,\ N=8$ for possible better solution. However, for this value of switching number, the final time does not improve and finally the algorithm goes to step7 for comparing the results obtained for $u(0)=1$ and $u(0)=-1$. Consequently, the algorithm provides $N=1$ as a result, which produces $\zeta=[0.7230,\ 2.3717]$ and $t_f=3.0947$ while the control input is $u(t)=\{-1,1\}$. In fact, the best result is as follows:

$$\begin{cases} N=1, \\ u(t)=\{-1,1\} \\ \zeta=\{0.7230, 2.3717\} \\ t_f=3.947 \end{cases}$$

The accuracy of reaching the origin is accessed by 10^{-4}. The state trajectories can be seen in Figure2.

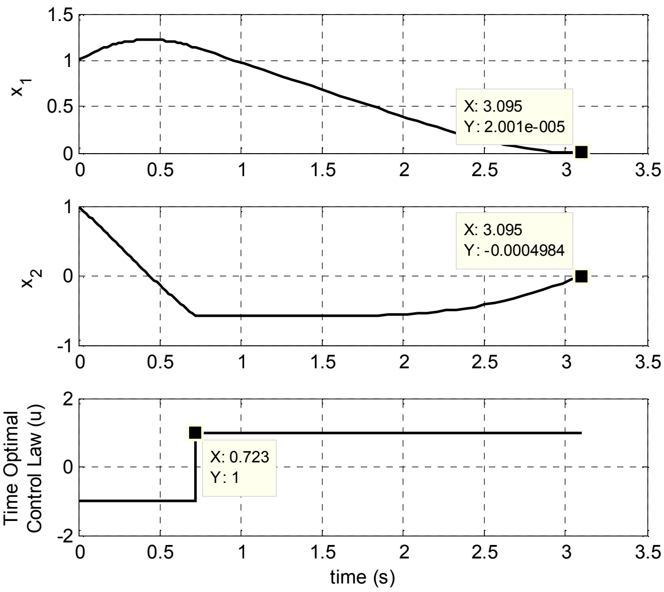

Figure2. ***x_1 and x_2 state trajectory and time optimal bang-bang control input***

It can be seen from these figure that reaching the target is possible for one switching of the control input from its minimum value $u(t)=-1$ to its maximum value $u(t)=1$ at $t_1=0.7230$ seconds. The final value of the state variables are very close to zero and this confirms the performance of the proposed method for solving time optimal bang-bang control problems. Meanwhile, using this algorithm, there is no need to a good initial guess of the solution, which is a troublesome task in the gradient based-methods such as STVM [5], [6] STC [8], STO [9] or mathematical programming [10]. Applying an additional initial algorithm is also prevented for preparing the start point through using the proposed IPSO-SQP algorithm. The quality of the work will be verified when the Rayleigh problem is also solved.

3.2 Rayleigh System

Consider the following dynamic:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -x_1 + x_2\left(1.4 - 0.14x_2^2\right) + 4u, \qquad |u(t)| \le 1 \end{aligned} \tag{15}$$

The objective is to minimize the following cost function, starting from $x_0 = [-5,-5]$ to the target $x_T = [0,0]$:

$$J(u(t)) = ct_f + \int_0^{t_f} \left(u^2 + x_1^2\right) dt \tag{16}$$

where c is a positive constant coefficient. A time optimal control that has been reported for this system is of the bang-bang type [9]. Thus, the control input is set to $u \in \{-1,1\}$. Substitution the input in (16) modifies the cost function to:

$$J(u(t)) = (1+c)\, t_f + \int_0^{t_f} x_1^2\, dt \tag{17}$$

The coefficient c is set to $1/16$ [9].

❖ *Mathematical Programming Method*

In [9] using a mathematical programming method time optimal bang-bang control problem of the Rayleigh system is solved. The initial guess of the control input and the arc times are assumed as $u(0)=1$ and $\zeta = [1.5,\ 2,\ 1,\ 0.5]$ respectively [9]. Results of applying the method in terms of the arc time intervals and the final reaching time are respectively reported as:

$\zeta = [1.47614,\ 1.76069,\ 1.76069,\ 0]$ and $t_f = 3.773841$.

It can be seen that how the initial point is close to the real solution also the time duration of the last arc is found zero. It means that two switching performs reaching the target. This algorithm is very sensitive to the initial guess of the solution, and needs the designer to be very familiar with the problem. It also needs many derivative information of the cost function. These may cause troubles when the complexity of the system is increased or when the designer is not very familiar with the problem to find a good initial guess.

❖ *IPSO-SQP Algorithm*

The IPSO-SQP is implemented for a swarm size of 30. Both c_1 and c_2 are set to 2.1. An initial guess for the number of switching is assumed $N=4$. In order to show that the algorithm is not sensitive to the initial guess, the range of the arc times in each dimension is assumed $\zeta_i \in [10^{-6}, 10]$, and initially the particles are randomly distributed in this range. The PSO search process algorithm switches to the SQP method, when the change in the cost function value is found less than 0.0001 after 10 iterations.

In step3, the algorithm is run for $\{u(0)=1,\ N=4$. In step4, $N=3$ and $N=5$ are also tried. The final time improves for $N=3$ and as a result, the value of i is increased. Then, the algorithm goes to step4 to search the solution for $\{u(0)=1,\ N=2,\ N=6$. Again, an improvement in final time for $N=2$ causes the algorithm to increase the value of i and look for possible better solution for $\{u(0)=1,\ N=1,\ N=7$. However, the value of the final time does not improve for $N=1$ and $N=7$ and the best results, which is obtained for $u(0)=1$ is stored and the algorithm goes to step6. Similar procedure is performed for $u(0)=-1$. Finally, in step7 by comparing the results obtained for both initial value of the control input, the best result is achieved as follows:

$$\begin{cases} N=2 \\ u(t)=\{1,-1,1\} \\ \zeta=\{1.2772, 1.9910, 0.4208\} \\ t_f=3.6890 \end{cases}$$

The accuracy of the results is about 10^{-3}. It can be seen that not only the final time obtained by this method is lower than what was achieved in [9] but also the algorithm is not sensitive to initial guess of the solution. The state trajectories can be seen in Figure3.

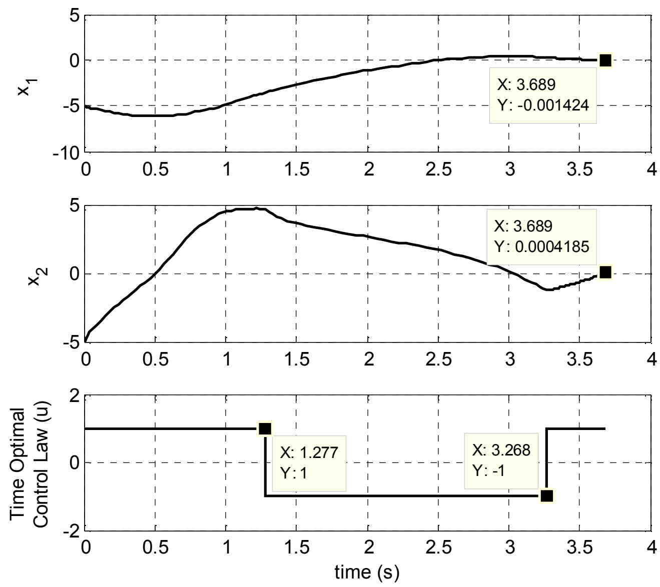

Figure3. ***x_1 and x_2 state trajectory and time optimal bang-bang control input***

It is obvious from Figure3 that the control input is switched two times at $t_1=1.2772$ and $t_2=3.2682$. This control input steers the states from the initial point from $x_0=[-5,-5]$ to the target $x_T=[0,0]$ in 3.6890 seconds. The final value of the states is very close to zero.

3.3 F8 Aircraft Model

The F8 aircraft model which is used in this paper is also used in many engineering applications [10], [22]. The model is as follows:

$$\begin{aligned} \dot{x}_1 &= -0.877x_1 + x_3 - 0.088x_1x_3 + 0.47x_1^2 - 0.019x_2^2 - x_1^2x_3 + 3.846x_1^3 - 0.215u + 0.28x_1^2u + 0.47x_1u^2 + 0. \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= -4.208x_1 + 0.396x_3 - 0.47x_1^2 - 3.56x_1^3 - 20.967u + 6.265x_1^2u + 46x_1u^2 + 61.4u^3 \end{aligned} \quad (18)$$

where x_1 is the angle of attack in radians, x_2 the pitch angle, x_3 the pitch rate in *rad/s*, and u is the tail deflection angle as a control input. The aim is to steer the aircraft from the initial state $x_0 = \frac{\pi}{180}[26.7, 0, 0]^T$ to the target $x_0 = [0, 0, 0]^T$ with appropriate switching of the control input $u = \pm 3^\circ$ in minimum time.

❖ *Mathematical Programming Method*

Time optimal bang-bang control of the F8 aircraft model is solved in [10] using a mathematical programming method. This algorithm is very sensitive to the initial guess of the solution. In [10] the method is executed for different start points and as results different results are achieved. For example for the initial guess of $\zeta = [0.5,\ 1,\ 0.5,\ 1,\ 0.5,\ 0.5]$ with $u(0) = +3^\circ$ the following result is obtained for the optimal arc times and final time, which is a local minimum:

$\zeta = [0.102917,\ 1.927923,\ 0.166868,\ 2.743384, 0.329923,\ 0.471162]$ $\quad t_f = 5.742177$

and for the initial guess of $\zeta = [1,\ 1,\ 1,\ 1,\ 1,\ 1]$ with $u(0) = +3^\circ$ the result is achieved as follows which is a global minimum:

$\zeta = [1.1327648,\ 0.3474915,\ 1.6088814,\ 0.2223491, 0,\ 0.4700298]$ $\quad t_f = 3.781517$

It can be seen that finding the global optimal solution by this method is highly dependent to the initial guess.

❖ *IPSO-SQP Algorithm*

The IPSO-SQP algorithm is used for solving time optimal bang-bang control problem of the F8 aircraft. First, the algorithm is initialized. The initial value of the switching number is guessed $N = 4$. The parameters c_1 and c_2 are set to 2.1. The swarm size is chosen $s = 30$. The lower and upper bound of each dimension of particle is determined $\zeta_i \in [10^{-6}, 10]$ hence, the particles are initially distributed randomly in this range. The search process of IPSO algorithm is switched to SQP method, when the change in cost function value is smaller than 0.0001for 10 iterations.

In step3 of the proposed IPSO-SQP algorithm, $\{u(0) = +3^\circ, N = 4$ is executed. In step4, $\{u(0) = +3^\circ, N = 3, N = 5$ is also tried. The final time is improved for $\{u(0) = +3^\circ, N = 3$. Thus, the value of *i* is increased and the algorithm goes to step4 to search for possible better solution with $\{u(0) = +3^\circ, N = 2, N = 6$. However, the final time does not improve for these value of switching number and the algorithm goes to step6 to repeat similar procedure for $\{u(0) = -3^\circ$. After executing the algorithm for different switching number, in step7, the best results are compared and the global optimal solution is achieved as follows:

$$\begin{cases} N = 3 \\ u(t) = \{3^\circ, -3^\circ, 3^\circ, -3^\circ\} \\ \zeta = [1.1348, 0.3464, 1.6083, 0.6905] \\ t_f = 3.78 \end{cases}$$

The state trajectories for the best result are depicted in the following figure.

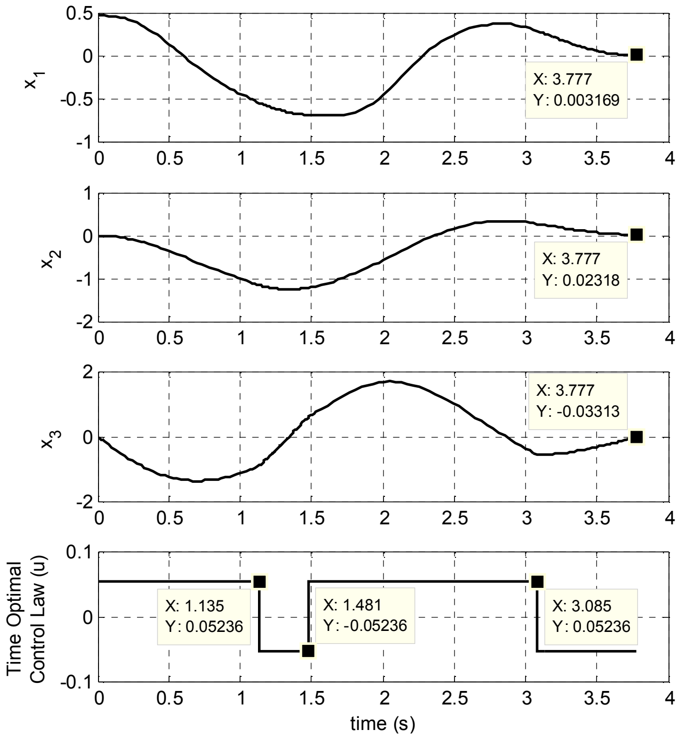

Figure4. ***x_1, x_2 and x_3 state trajectory and time optimal bang-bang control input***

It can be seen from these figures that reaching the target from the initial point is possible for 3 times switching of the control input at times: $t_1 = 1.1348$, $t_2 = 1.4812$ and $t_3 = 3.0895$.

The final values of the states are shown in the figure and are very close to zero. It indicates that the algorithm is highly capable for solving time optimal bang-bang control problems.

Besides, a practical time optimal control problem will be investigated here to verify the quality of the work. The problem is concerned with an autonomous underwater vehicle where is searching the seabed. The supply energy is maintained with carried on board batteries. The problem is expressed in the next section.

3. Time Optimal Bang-Bang Control of An Autonomous Underwater Vehicle

The AUV is used to search the seabed in depth of lower than 50 meters. It is necessary that the AUV to come back to the nest where is located on the sea surface in minimum time. Besides the AUV has to have an angle of attack of about 10 degree while comes out of the sea to successfully locate in the nest. The IPSO-SQP method is used to control the vehicle. The AUVs general equation of motion is presented in Appendix A. However, the equations in xz plane are derived in the next section.

4.1 Equations of Motion In xz Plane

The F8 aircraf From Appendix A it can be easily seen that equations of motion of the AUVs are nonlinear, coupled with six degree of freedom. Fortunately, they can be shown in terms of six first order equations. On the other hand, the derivative of each variable is a nonlinear function with respect of other variables. In this paper the motion is assumed to take place only in the pure depth-plane. Thus u, w, q, θ, x, z are considered whereas other variables are immediately neglected. In this case, the equation of motion is expressed as follows:

$$
\begin{aligned}
\dot{u} &= a_1 \sin\theta + a_2 \cos\theta + a_3 u|u|^{0.86} + a_4 qu + a_5 qw + a_6 uw + a_7 \delta_e u^2 + a_8 q|q| + a_9 w|w| + a_{10} q^2 + a_{11}\text{Thrust} \\
\dot{w} &= b_1 \sin\theta + b_2 \cos\theta + b_3 u|u|^{0.86} + b_4 qu + b_5 qw + b_6 uw + b_7 \delta_e u^2 + b_8 q|q| + b_9 w|w| + b_{10} q^2 + b_{11}\text{Thrust} \\
\dot{q} &= c_1 \sin\theta + c_2 \cos\theta + c_3 u|u|^{0.86} + c_4 qu + c_5 qw + c_6 uw + c_7 \delta_e u^2 + c_8 q|q| + c_9 w|w| + c_{10} q^2 + c_{11}\text{Thrust} \\
\dot{\theta} &= q \\
\dot{x} &= u\cos\theta + w\sin\theta \\
\dot{z} &= -u\sin\theta + w\cos\theta \\
\delta_e &= \pm\delta_{e\max}
\end{aligned} \tag{19}
$$

where the value of parameters are presented at the table B.I in appendix B.

4.2 Goals And Assumptions

The goal is to steer the AUV from the depth of 50 meters towards the nest where is located on the surface, of course in a least time as possible. Due to practical restriction, the AUV is forced to have a pitch angle of about 10 degrees when it reaches the surface. Meanwhile batteries of one per unit supply the thruster force. It is also assumed that the external disturbance of environmental forces and moments like waves and currents are negligible. The control input is considered as bang-bang controller. Consequently, the IPSO-SQP method is preferred to be used for a time optimal bang-bang control purpose.

4.3 Simulation

The IPSO-SQP is implemented for a swarm size of 30. Both c_1 and c_2 are set to 2.1. Initial guesses of the number of switching and the control input are assumed to be $N = 4$ and $u(0) = -7\pi/180(rad)$ respectively. In order to show that the algorithm is not sensitive to the initial guess, the range of the arc times in each dimension is assumed $\zeta_i \in [10^{-6}, 30]$, and initially the particles are randomly distributed in this range. The PSO search process algorithm is switched to the SQP method when the change in the fitness value becomes less than 0.0001 after 10 iterations. The algorithm is accordingly results $\zeta = [13.8698,\ 8]$ and $t_f = 21.8698$ for an optimum number of the arcs and the final time. This verifies that the reaching time to the target is achieved with two arcs:

$$
\begin{cases}
N = 1 \\
u(t) = \{-7\pi/180, -7\pi/180\}(rad) \\
\zeta = \{13.8698, 8\} \\
t_f = 21.8698
\end{cases}
$$

The results are depicted in Figure 5 to 11.

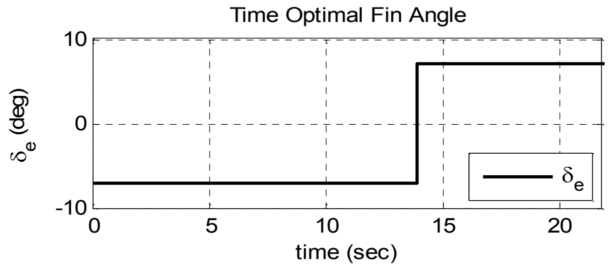

Figure5. The fin angle in the time optimal bang-bang control

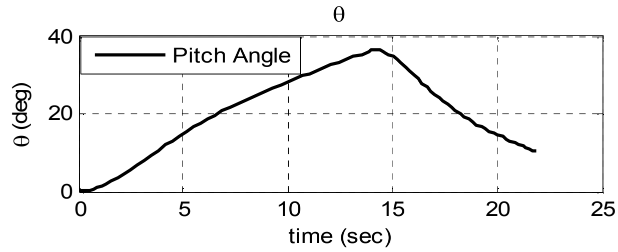

Figure6. The pitch angle

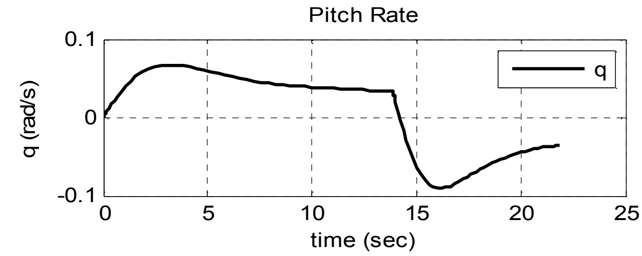

Figure7. The pitch rate (the angular velocity around the z axis)

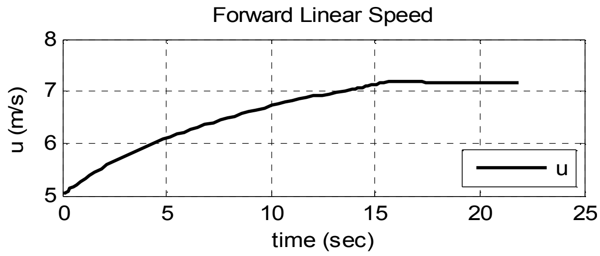

Figure8. The forward linear speed along with the longitu

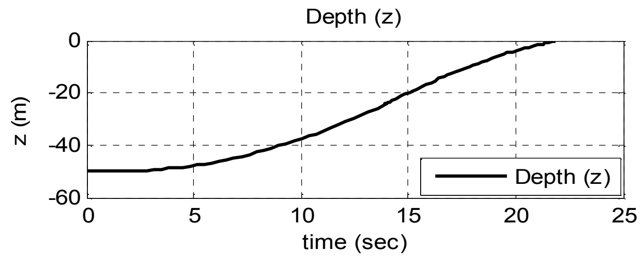

Figure9. The Vehicle depth towards the surface

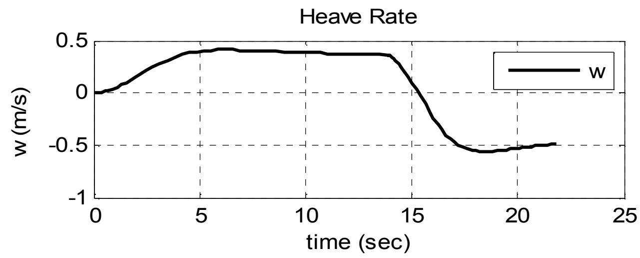

Figure10. The linear velocity along with the z axis

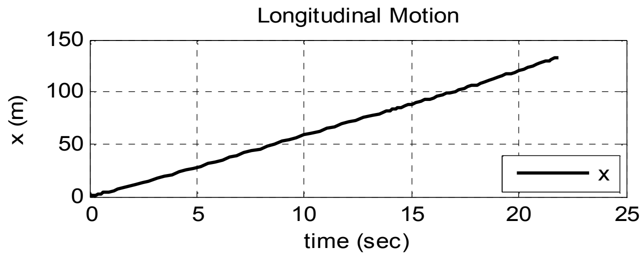

Figure11. The Vehicle longitudinal motion

From Figure5 it can be seen that the control input (the fin angle) with just one switching forces the AUV towards the surface whilst keeping the pitch angle (Figure6) 10 degrees when it reaches the surface. The pitch rate is also depicted in Figure7. The forward linear speed, the Vehicle depth, the heave rate and the longitudinal motion along with the x axis are respectively shown in Figs. 8, 9, 10 and 11, verifying that the proposed technique does the duty well enough.

4. Pitch Programming Task

An auxiliary aim of controlling the AUV is to measure and control the pitch angle via an optimum based control input δ_e. It should be noted that the required trajectory in the tracking problem is yielded from another optimum job through the IPSO-SQP algorithm. A PID controller in the closed loop system (Figure12) is designed to achieve the goal. The PID gains (K_P, K_I, K_D) are tuned by using the IPSO-SQP algorithm described in subsection2.3 to minimize the tracking error.

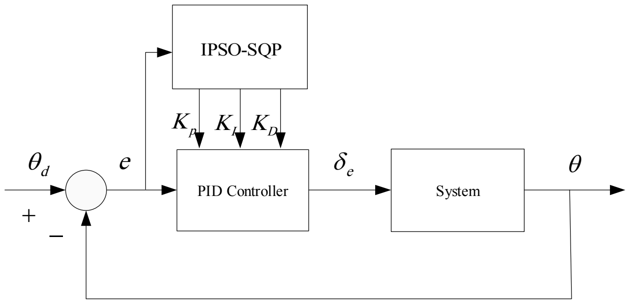

Figure12. The closed loop control system in the pitch programming task

In the IPSO-SQP algorithm described in subsection2.3, coefficients K_P, K_I and K_D are considered as particles. In this regard, the cost function, which is required to be minimized, is defined as follows:

$$J = \int_0^{t_f} e(t)^2\, dt \qquad (20)$$

Defining the error by:

$$e(t) = \theta_d(t) - \theta(t) \qquad (21)$$

The desired trajectory $\theta_d(t)$ is achieved from the time optimum bang-bang control problem in the last section. Furthermore $\theta(t)$ is the actual pitch angle of the vehicle.

The IPSO-SQP optimization technique is implemented for a swarm size of 30. Both c_1 and c_2 are set to 2.1. The range of the each dimension is assumed $K_p,\ K_I,\ K_D \in [-1000, 0]$, and initially the particles are randomly distributed in this range. Choosing the negative value of the bound is due to the difference in sign conventions between the stern plane angle (δ_e) and vehicle pitch angle. Positive stern plane angle will generate a negative moment about the y-axis, forcing the vehicle to pitch down (negative pitch rate). The PSO search process algorithm is switched to the SQP method when the change in the fitness value becomes less than 0.0001 after 10 iterations. The algorithm achieves the gains by:

$K_P = -600,\ K_I = 0$ and $K_D = -285.2014$.

In the following, outcome of the simulation of applying the PID controller are illustrated in Figure 13 to 15.

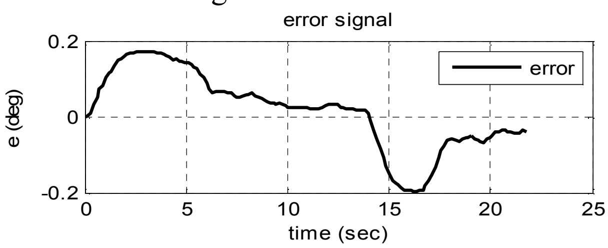

Figure13. The pitch tracking error signal

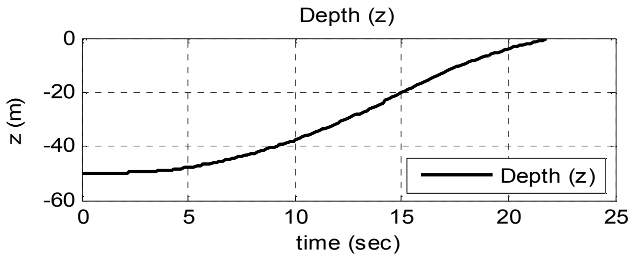

Figure14. The vehicle depth

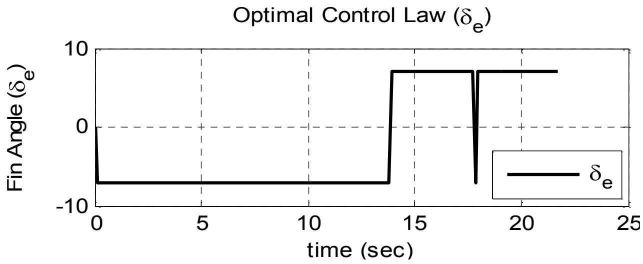

Figure15. The Control input vs. the fin angle (δ_c)

Apart from the smooth fluctuation in the graph (Figure13), the magnitude of the tracking error is seen negligible. It confirms that the controller is capable to track the desired pitch angle. The quality of the achieved depth trajectory is also seen in Figure14.

5. Conclusion

In this paper, the IPSO-SQP algorithm is used for the first time in a time optimal bang-bang control problem. The algorithm is shown to have a high capability of finding the global minimum. It is also shown that there is no need to have good initial guess. The primary stage of the search procedure uses the IPSO method to minimize the defined cost function. This confirmed the advantage of the proposed algorithm, *i.e.* directly minimizing the cost function without using a gradient based technique. Then the algorithm switches to the SQP method to find the global solution more rapidly and precisely. The algorithm is followed by the IPSO-SQP method in time optimal bang-bang control of three distinct nonlinear systems and an autonomous underwater vehicle. The final time, which is achieved by this method, was shown less than what was achieved by previous gradient based methods. The significance of the proposed technique is also verified when it is used to tune the coefficients of the PID controller *i.e.* K_P, K_I and K_D in a minimization of the tracking error. Ultimately, the tuned PID controller is successfully used in the pitch programming of AUV. Simulation results showed that the performance of the IPSO-SQP algorithm in both time optimal bang-bang control and pitch programming are satisfactory.

Dynamic motion of an AUV (Fig. A.1) can be described in both reference frames of attached to the body and inertia (the earth) one. A coordinate transformation states the variables on each. It must be noted that the effect of the motion of the earth in the fixed reference frame is neglected with respect to the vehicle motion.

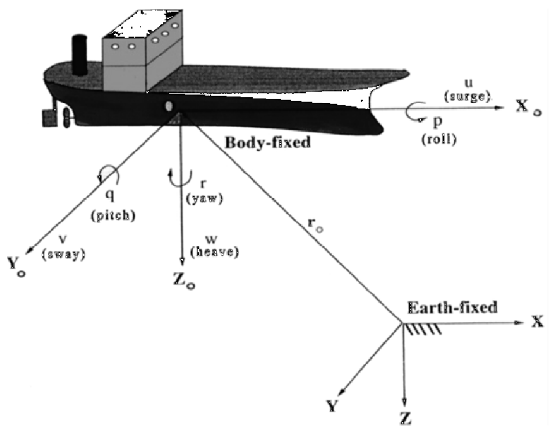

Figure A.1. The body and the inertia fixed reference frames

AUV is usually expresses by six degree of freedom equations of motion. Theses equations can be described in either of two prescribed reference frames. One may state the equations in terms of the body fixed coordinates as follows:

$$M\dot{v}+C(v)v+D(v)v+g(\eta)=\tau \qquad \text{(A.1)}$$

where:

$$M=M_{RB}+M_A,\quad C(v)=C_{RB}(v)+C_A(v),\quad D(v)=D_p(v)+D_v(v) \qquad \text{(A.2)}$$

and

$$\tau=\tau_C+\text{Thrust} \qquad \text{(A.3)}$$

M_{RB} and $C_{RB}(v)$ are the rigid body mass and coriolis matrices respectively. $M_A$and $C_A(v)$ are added mass and coriolis matrices whilst $D(v)$ is the drag force. Vector $g(\eta)$ consists of the gravity, the buoyancy forces and the moments which act on their center of the relevant forces. Furthermore τ, as vector of containing the control forces and moments, states the control surfaces (τ_C), the thruster (Thrust) and the environment forces. However the environment forces and moments, including those caused by the sea currents and waves are neglected. Meanwhile the linear and angular velocity vector v in the body fixed reference frame together with the position and the Euler angel vector η in the inertia reference frame are described by:

$$v=[u\ v\ w\ p\ q\ r]^T \qquad \text{(A.4)}$$

$$\eta_1=[x\ y\ z]^T,\quad \eta_2=[\varphi\ \theta\ \Psi]^T,\quad \eta=[\eta_1;\eta_2] \qquad \text{(A.5)}$$

The relation between the body and the inertia fixed reference frame is defined by a coordinate transform matrix which is as follows:

$$\dot{\eta}=J(\eta)v \qquad \text{(A.6)}$$

$$J(\eta)=\begin{bmatrix} J_1(\eta_1) & 0 \\ 0 & J_2(\eta_2) \end{bmatrix} \qquad \text{(A.7)}$$

$$J_1(\eta_2)=\begin{bmatrix} \mathrm{c}\theta\mathrm{c}\Psi & -\mathrm{s}\Psi\mathrm{c}\varphi+\mathrm{c}\Psi\mathrm{s}\theta\mathrm{s}\varphi & \mathrm{s}\Psi\mathrm{s}\varphi+\mathrm{c}\Psi\mathrm{c}\varphi\mathrm{s}\theta \\ \mathrm{s}\Psi\mathrm{c}\theta & \mathrm{c}\Psi\mathrm{c}\varphi+\mathrm{s}\varphi\mathrm{s}\Psi\mathrm{s}\theta & -\mathrm{c}\Psi\mathrm{s}\varphi+\mathrm{s}\theta\mathrm{s}\Psi\mathrm{c}\varphi \\ -\mathrm{s}\theta & \mathrm{c}\theta\mathrm{s}\varphi & \mathrm{c}\theta\mathrm{c}\varphi \end{bmatrix} \qquad \text{(A.8)}$$

$$J_2(\eta_2)=\begin{bmatrix} 1 & \sin\varphi\tan\theta & \cos\varphi\tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi/\cos\theta & \cos\varphi/\cos\theta \end{bmatrix} \qquad \text{(A.9)}$$

where $c(\cdot)$ and $s(\cdot)$ stand for trigonometric $\cos(\cdot)$ and $\sin(\cdot)$ respectively. The six degree of freedom equations of motion of AUVs are expressed in Eq. (A.10) to (A.15).
Force equation along with the x axis:

$$\begin{aligned}(m-X_{\dot u})\dot u+mz_G\dot q-my_G\dot r&=-(W-B)\sin\theta+X_{u|u|}u|u|+(X_{wq}-m)wq+(X_{qq}+mx_G)q^2-my_Gpq+(X_{vr}+m)vr\\&\quad+(X_{rr}+mx_G)r^2-mz_Gpr+Thrust\end{aligned}\tag{A.10}$$

Likewise, the force equation along with the y axis:

$$\begin{aligned}(m-Y_{\dot v})\dot v-mz_G\dot p+(mx_G-Y_{\dot r})\dot r&=(W-B)\cos\theta\sin\phi+Y_{v|v|}v|v|+Y_{r|r|}r|r|+my_Gr^2+(Y_{ur}-m)ur+(Y_{wp}+m)wp\\&\quad+(Y_{pq}-mx_G)+Y_{uv}uv+my_Gp^2+mz_Gqr+Y_{uu\delta_r}\delta_r\end{aligned}\tag{A.11}$$

Similarly for the force equation along with the z axis:

$$\begin{aligned}(m-Z_{\dot w})\dot w+my_G\dot p-(mx_G+Z_{\dot q})\dot q&=+(W-B)\cos\theta\cos\phi+Z_{w|w|}w|w|+Z_{q|q|}q|q|+(Z_{uq}+m)uq+(Z_{vp}-m)vp\\&\quad+(Z_{rp}-mx_G)rp+Z_{uw}uw-my_Grp+mz_G(p^2+q^2)+Z_{uu\delta_e}u^2\delta_e\end{aligned}\tag{A.12}$$

In parallel, the momentum equation along with the x axis:

$$\begin{aligned}-mz_G\dot v+my_G\dot w+(I_{xx}-K_{\dot p})\dot p&=+(y_GW-y_BB)\cos\theta\cos\phi-(z_GW-z_BB)\cos\theta\sin\phi+K_{p|p|}p|p|\\&\quad-(I_{zz}-I_{yy})qr+m(uq-vp)-mz_G(wp-ur)+KThrust\end{aligned}\tag{A.13}$$

The same for momentum equation along with the y axis:

$$\begin{aligned}mz_G\dot u-(mx_G+M_{\dot w})\dot w+(I_{yy}-M_{\dot q})\dot q&=-(z_GW-z_BB)\sin\theta-(x_GW-x_BB)\cos\theta\cos\phi+M_{w|w|}w|w|+M_{q|q|}q|q|\\&\quad+(M_{uq}-mx_G)uq+(M_{vp}+mx_G)vp+\left[M_{rp}-(I_{xx}-I_{zz})\right]rp\end{aligned}\tag{A.14}$$

and finally the momentum equation along with z axis is as follows:

$$\begin{aligned}-my_G\dot u+(mx_G-N_{\dot v})\dot v+(I_{zz}-N_{\dot r})\dot r&=(x_GW-x_BB)\cos\theta\sin\phi+(y_GW-y_BB)\sin\theta+N_{v|v|}v|v|+N_{r|r|}r|r|\\&\quad+(N_{ur}-mx_G)ur+(N_{wp}-mx_G)wp-my_G(vr-wq)\\&\quad+\left[N_{pq}-(I_{yy}-I_{zz})\right]pq+N_{uv}uv+N_{uu\delta_r}u^2\delta_r\end{aligned}\tag{A.15}$$

The value of parameters which are used in the depth plane equation of motion, Eq. (18), is presented here:

Table B.I

$a_1=0.0669$	$a_2=-4.2233\times10^{-}$	$a_3=-0.0146$	$a_4=0.0011$	$a_5=-1.9885$	$a_6=3.6413$	$a_7=2.7080\times10^{-4}$
$a_8=0.0692$	$a_9-1.1487\times10^{-8}$	$a_{10}=0.0430$	$a_{11}=6.6859\times10^{-4}$			
$b_1=-9.4496\times10^{-4}$	$b_2=-0.0327$	$b_3=1.3998\times10^{-6}$	$b_4=0.4716$	$b_5=9.4575\times10^{-8}$	$b_6=-0.0173$	$b_7=-0.0089$
$b_8=-0.1820$	$b_9=-0.7010$	$b_{10}=0.0149$	$b_{11}=-6.3966\times10^{-8}$			
$c_1=-0.0437$	$c_2=0.0143$	$c_3=6.4731$	$c_4=-0.0355$	$c_5=0.0044$	$c_6=-0.0123$	$c_7=-0.0091$
$c_8=-2.3354$	$c_9=0.0039$	$c_{10}=-9.4575\times10^{-}$	$c_{11}=-2.9580\times10^{-6}$			

References

[1] D.E. Kirk, "An Introduction to Optimal Control Theory", *Englewood Cliffs, N.I. Prentice Hall*, 1970.

[2] G.J. Lastman, "A shooting method for solving two-point boundary value problems arising from nonsingular bang-bang optimal control processes", *International journal of control*, vol. 27, (1978) 513-524.

[3] K.L. Teo, C.J. Goh and K.H. Wong, "A Unified Computational Approach to Optimal Control Problems", *Longman Scientific and Technical; New York: Wiley*, 1991.

[4] K.H., Wong, D.J. Clements and K.L. Teo, "Optimal Control Computation for Nonlinear Time-Lag Systems", *Journal of Optimization Theory and Applications*, vol. 47, (1985) 91-107.

[5] R.R. Mohler, "Bilinear Control Processes", *Chapter3, Academic Press, New York*, 1973.

[6] R.R. Mohler, "Nonlinear Systems", *Prentice Hall: Englewood Cliffs, N.J., Chapter 7,* 1991.

[7] E.B Meier and A.E. Brayson, "Efficient Algorithm for Time Optimal Control of a Two Link Manipulator", *Journal of Guidance, Control and Dynamics*, vol. 13, (1990) 859-866.

[8] C.Y. Kaya and J.L. Noakes, "Computations and Time Optimal Controls", *Optimal Control Applications and Methods*, vol. 17, (1996) 171–185.

[9] C.Y. Kaya and J.L. Noakes, "Computational Method for Time-Optimal Switching Controls", *Journal of Optimization Theory and Applications*, vol. 117, (2003) 69-92.

[10] C.Y. Kaya, S.K. Lucas and S.T. Simakov, "Computations for Bang-Bang Constrained Optimal Control Using a Mathematical Programming Formulation", *Optimal Control Applications and Methods*, vol. 25, (2004) 295–308.

[11] H. Modares, M.B. NaghibiSistani, "Solving Nonlinear Optimal Control Problems Using a Hybrid IPSO–SQP Algorithm"*, Journal of Engineering Applications of Artificial Intelligence*, (2010) 476–484.

[12] M. Varadarajan, K.S. Swarup, "Differential evolution approach for optimal reactive power dispatch", *Applied Soft Computing 8*, (2008) 1549–1561.

[13] M.M. Ali, C.T. Storey, A. Orn, "Application of stochastic global optimization algorithms to practical problems", *Journal of Optimization Theory and Applications 95*, (1997) 545–563.

[14] J.H. Li, P.M. Lee, "Design of an Adaptive Nonlinear Controller for Depth Control of an Autonomous Underwater Vehicle", *Ocean Engineering* vol. 32, (2005) 2165- 2181.

[15] R. Kumar, A. Dasgupta, C.S. Kumar, "Real-Time Optimal Motion Planning for Autonomous Underwater Vehicles", *Ocean Engineering*, vol. 32, (2005) 1431-1447.

[16] J. Kennedy, R.C. Eberhart, "Particle Swarm Optimization", *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4. (1995) 1942–1948.

[17] C.B.B. Costa, A.C. Maciel, R. Filho, "Mathematical Modeling and Optimal Control Strategy Development for an Acidic Crystallization Process", *Chemical Engineering and Processing*, vol. 44, (2005) 737–753.

[18] L. Bayo´n, J.M. Grau, M.M., Ruiz, M.M., Sua´rez, "Initial Guess of The Solution of Dynamic Optimization of Chemical Processes", *Journal of Mathematical Chemistry*, (2009) 1-10.

[19] EL. Sayed, M.E. Mostafa, "An Augmented Lagrangian SQP Method for Solving Some Special Class of Nonlinear Semi–Definite Programming Problems", *Computational and Applied Mathematics*, vol. 24, (2005) 461–486.

[20] M.V. Schwerin, O. Deutschmann, V. Schulz, "Process Optimization of Reactive Systems by Partially Reduced SQP Methods", *Computers and Chemical Engineering*, vol. 24, (2000) 89–97.

[21] A. Antoniou, Wu.Sh. Lu, "Practical Optimization Algorithms and Engineering Applications", *Springer*, 2007.

[22] S.P. Bank, K.J. Mhana, "Optimal control and stabilization of nonlinear systems", *IMA journal of mathematical control and information*, vol. 9, (1992) 179-196