

Diver Propulsion Vehicle using MATLAB: A survey

Mr. Swanand U.Mane¹, Mr. Amol B. Sase², Mr. Ashish Prasad³

¹M.E(Embedded System), ²M.B.A (H.R), ³M.B.A (Finance)

Corresponding Author: Mr. Swanand U.Mane

Abstract: DPVs have recreational, scientific and military applications. A diver propulsion vehicle (DPV), also known as an underwater propulsion vehicle or underwater scooter is an item of equipment used by scuba divers and rebreather divers underwater to increase the range. Range is restricted by the amount of breathing gas that can be carried, the rate at which that breathing gas is consumed, and the battery power of the DPV. Though this has been the common methodology for many years, there is an emerging technique that involves developing algorithms with graphical tools. This improves development time, decreases programming complexity, and facilitates sharing of the design with co-workers. A leading software package that fits this mold is MATLAB, from The Math Works.[1]With the use of MATLAB software this paper will also be useful to understand and study different parameters for data analysis and to design efficient machine for recreational diving industry according to the commercial point of view. The basic optimization steps is minimizing DPV drag force to reduce power required.

Keywords: DPV, MATLAB, DSP, GUIDE.

Date of Submission: 18-04-2019

Date of acceptance: 04-05-2019

I. Introduction

Manned torpedoes are a type of diver propulsion vehicle used as secret naval weapons in World War II. The name was commonly used to refer to the weapons that Italy, and later Britain, deployed in the Mediterranean and used to attack ships in enemy harbours. The first human torpedo was the Italian Maiale ("Pig"). In operation, it was carried by another vessel (usually a normal submarine), and launched near the target. It was electrically propelled, with two crewmen in diving suits and rebreathers riding astride. They steered the torpedo at slow speed to the target, used the detachable warhead as a limpet mine and then rode the torpedo away.

The PADI Diver Propulsion Vehicle (DPV) Specialty is one of the most fun specialties to participate in. While originally used only by technical divers and the military, DPV's are rapidly gaining popularity in the recreational diving sphere. DPVs offer a thrilling way to cover a lot of underwater territory in a brief amount of time. They cut down the need for the diver to exert himself by kicking or finning. However using a DPV does require some amount of mastery as the DPV has the potential to make buoyancy control difficult and cause barotrauma if the diver ascends or descends under power.[2]There are primarily 2 types of underwater transportation systems for recreational and technical divers, the diver propulsion vehicle and the under water scooter. Usually the difference between DPV and the scooter is that a DPV is a vehicle that one can physically ride with an integrated saddle while a scooter is the unit that tows the diver along by the arm. A more general rule is that scooter tends to have smaller capacity battery than the DPV.[3]

The applications of this multi-disciplinary URV [1-6] design have experienced tremendous growth in the last few decades. Based on the designed tasks and modes of operations, the URV are broadly classified as remotely operated vehicle (ROV) and autonomous underwater vehicle (AUV). The ROV and AUV have been used in offshore oil industry such as installation of structures and pipelines, salvage, mine hunting, fishery study and other applications where their endurance, economy and safety can replace divers. Due to its hard wire link or tethered design, the ROV are best suited for work that involves real-time image transmission, operating from a stationary point or cruising at relatively slow speeds such as pipeline inspection. [4] Though, a DPV usually consists of a propeller which is driven by DC electric motor. The design of DPVs should be satisfied all diver safety standards to ensure propeller disservice and its harmless to diver or other diving equipment.[5]

History and Types of DPV:

A master research work in designing and operating DPV as a next generation vehicles. The design of such submersible vehicle should be passed many safety standards and real tests, Allmendinder (1990). On the other hand, in designing of high performance DPVs good CFD methods should be used to evaluate their hydrodynamic force and moment.[6] A total models have been identified in Chinese, Russian, UAE and US.

China:

Table 1: Chinese DPV

Name	Weighting	Diameter	Length	Endurance	Depth	Speed
QY18DPV(1man)	<20Kg	0.385m	0.8m	>1hr	40m	2Kn
QY40DPV(1man)	<40Kg	0.32m	1.2m	>1.5hr	40m	2Kn
QX50DPV(1man)	<50Kg	0.23m	1.6m	>2hr	40m	2Kn
QJY001DPV(2man)	<90Kg	0.53m	2.3m	>2.5hr	30m	2Kn

Russian:

Table 2: Russian DPV

Name	Diameter	Length	Endurance	Depth	Speed
Sirena DPV	53cm	53cm	6hr	40m	4Kn
Project 907 Triton 1(2man)	-	5m	6hr	40m	6Kn
Project 908Triton2(6man)	-	9.5m	12hr	40m	6Kn

United Arab Emirates:

Table 3: UAE DPV

Name	Weight	Back to Power	Speed
Class 4(3man)	3Ton	8Kw motor	6Kn
Class 5(2man)	8Ton	8Kw motor	6Kn
Class 6(8man)	8Ton	8Kw motor	4Kn
Class 8(6man)	8Ton	8Kw motor	6Kn

United States: A total models have been identified in US services.

Table 4: US DPV[7]

Name	Weight	Depth	Speed
Piranha	1.63Ton	70m	8Kn
SDV-X Dolphin	2.75Ton	91m	8Kn
Sea Shadow SDV	107Kg	30m	3Kn
SEAL Delivery SDV	-	6m	6Kn
Shallow Water Combat Submersible	-	>6m	>6Kn

Characteristic of typical commercial DPV:

Table 5: Characteristic of DPV

Brand/Model	Battery Voltage/Capacity	Speed Km/hr	Run Time min	Current Range	I/P Power	Overall length
AUL Drako	24/36	6.0	120+	18-18	432	1.828
AUL Predator 2000	24/18	4.5	40-100	12-30	288-720	1.168
Farallon MK8	24/18	4.75	60-115	10-18	240-432	1.574
Torpedo2000	12/32	3.6	50-80	42-42	504	0.914
Ocenicmako	12/17	2.4-4.3	40-120	8.5-11.3	100-135	0.635

Different technologies used for DPV:

The preliminary dimensions of the first DPV had been selected as 2m×0.7m×0.7m (length ×width× height). It is based on real prototype that had been presented for production. That was an improvement vehicle similar to K-10 TNT Marine Enterprises diver training propulsion vehicle. The following restrictions have been considered in generating each computational model. In fact these restrictions have been imposed in Geometry Implementation module as fixed design parameters:

- a) Afrontal = Aref. (Fixing one of the main parameter that will be affected the drag force)
- b) Lmin ≥ k (Length limitation for introduced volume constraint of diver by coupling to frontal area)

The limitation of inner vehicle volume will be leaded to a minimum frontal area (Aref.) for diver accommodating on it with thruster unit. This value is the minimum area required to one normal person occupation in prone position ahead to flow stream in addition to thruster unit. Meanwhile the minimum length of vehicle is selected as the minimum required length k from vehicle nose to diver waistline. A calculating domain with suitable dimensions has been generated after implementing the first model. The length and width of geometric domain have been selected as three tenfold vehicle lengths for domain length and about tenfold the width of vehicle for domain width. The domain geometry built around the model.

Governing Equation And Turbulence Model: The Reynolds number is the primary nondimensional parameter in low speed flows concern under consideration. This parameter represents the ratio of inertial to viscous forces in the flow and defined as: $Re = \rho U L / \mu = U L / \nu$ Where U is the speed that simulated free-

stream velocity and ν is the fluid kinematic viscosity. The characteristic length (L) can be selected as the length of the vehicle. On the other hand the minimum length of the vehicle is obtained from the length of the real diver with open hand as in optimization process this value is the one of the basic limitation that must be considered in the overall calculation. In many cases, the drag coefficient is reasonably independent of Reynolds number upon a critical value is exceeded. Though in complex shape with different parts they may be exposed to different velocity condition there is less chance of this independence as different parts of the vehicle subjected to different Reynolds numbers. As the vehicle may be exposed to a range of flow speeds, it might be necessary to simulate a range of Reynolds numbers to get a more complete prospect of the loads that will be encountered in actual motion. The minimum Reynolds number according to minimum length of the vehicle in this study is about 15.95 that is assumed fully turbulent condition. The governing equations explained the conservation laws of flow in mathematical forms. These laws are as follow: Continuity eq.: $\nabla \cdot \mathbf{u} = 0$

$$\text{Momentum eq.: } \frac{\partial}{\partial x} \rho u_i u_j = \frac{-\partial p}{\partial x_i} + \frac{\partial t_{ij}}{\partial x_i} + \frac{\partial \tau_{ij}^t}{\partial x_i}$$

In this equation the stress tensor t_{ij} and Reynolds stress term τ_{ij}^t are as follow:

$$t_{ij} = \left[\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right], \tau_{ij}^t = -\rho u_i^t u_j^t$$

Since in turbulent flow the actual velocity of each particle is the sum of instantaneous and fluctuation component. The last term $\overline{\rho u_i^t u_j^t}$ that is named Reynolds stress should be obtained with favorable turbulent models. There are different turbulent models in computational fluid dynamic which should be selected according to nature of flow, required accuracy, calculating facilities and computational time. This is an active field as the researchers have been published many papers to demonstrate which one is more efficient relative to other models with the same conditions. For example, Zaidi et al. (2010) was presented a paper to specify which model of turbulence is the most adapted in order to predict drag forces. They showed that SST k- ω model is an accurate one. According to this model the k and ω are:

$$\frac{\partial}{\partial t} \rho k + \frac{\partial}{\partial x_i} \rho k u_i = \frac{\partial}{\partial x_j} \left[-k \frac{\partial k}{\partial x_j} \right] + G_k - Y_k + S_k$$

$$\frac{\partial}{\partial t} \rho \omega + \frac{\partial}{\partial x_i} \rho \omega u_i = \frac{\partial}{\partial x_j} \left[-\omega \frac{\partial \omega}{\partial x_j} \right] + G_\omega - Y_\omega + S_\omega$$

The parameter “ ω ” is the rate of energy dissipation per unit volume per time (or specific dissipation rate) and “k” is the turbulent kinetic energy per unit mass. Hence, with obtaining the drag forces by this method for a diver propulsion vehicle, the propulsion power related to require speed could be calculated: $\rho_p = D_t U$. On the other hand, if it was assumed that electrical power is going to be used for propelling him other parameters enter the problem, like propulsion efficiency (that is composed of propeller and motor efficiency) and weight geometry of such system. Thus the installed power P_i is defined for this condition as $p_i = \frac{P_p}{\eta - \rho}$

(η =propulsion efficiency))The hydrodynamic forces in underwater motion highly have been related to moving body shape and geometry.[8]

There are many design variables may be changed in optimization process but the most important one are the nose and body contour and propulsion unit level and its’ longitudinal position that each one would be changed to satisfy intimated constraints in sequential algorithm

Proposed System:

There are essentially six degrees of freedom that the vehicle has: three that indicate the position, and the other three that indicate the orientation. These six degrees of freedom identify the state of the vehicle.[9]

Use of MATLAB for development of DPV: MATLAB combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly. It includes the Live Editor for creating scripts that combine code, output, and formatted text in an executable notebook.[matlab math works].MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include: Application development, including Graphical User Interface building. MATLAB supports developing applications with graphical user interface (GUI) features. MATLAB includes GUIDE(GUI development environment) for graphically designing GUIs. It also has tightly integrated graph-plotting features. For example, the function *plot* can be used to produce a graph from two vectors x and y .

Simulink, developed by MathWorks, is a graphical programming environment for modelling, simulating and analysing multidomain dynamical systems. Simulink is an extension of MATLAB that enables user to design and simulate systems by creating block diagram.

There are two major categories of elements in Simulink: Blocks , Lines and Blocks are used to generate, modify, combine, output, and display signals. Lines, on the other hand, are used to transfer signals from one block to another. Blocks There are several general classes of blocks, some of which are: • Sources: Used to generate

various signals. Sources blocks have outputs but no inputs. One may want to use a Constant input, a Sine Wave, a Step, a Ramp, a Pulse Generator, or a Uniform Random number to simulate noise. The Clock may be used to create a time index for plotting purposes.

- Sinks: Used to output or display signals. Sinks blocks have inputs but no outputs. Examples are Scope, Display, To Workspace, Floating Scope, XY Graph, etc.
- Discrete: Discrete Filter, Discrete State-Space, Discrete Transfer Fcn, Discrete Zero Pole, Unit Delay, etc.
- Continuous: Integrator, State-Space, Transfer Fcn, Zero-Pole, etc.
- Signal routing: Mux, Demux, Switch, etc.
- Math Operations: Abs, Gain, Product, Slider Gain, Sign, Sum, etc. 315 Lines transmit signals in the direction indicated by the arrow. Lines must always transmit signals from the output terminal of one block to the input terminal of another block. One exception to this is that a line can tap off of another line. This sends the original signal to two (or more) destination blocks. Lines can never inject a signal into another line; lines must be combined through the use of block such as a summing junction. A signal can be either a scalar signal or a vector signal.

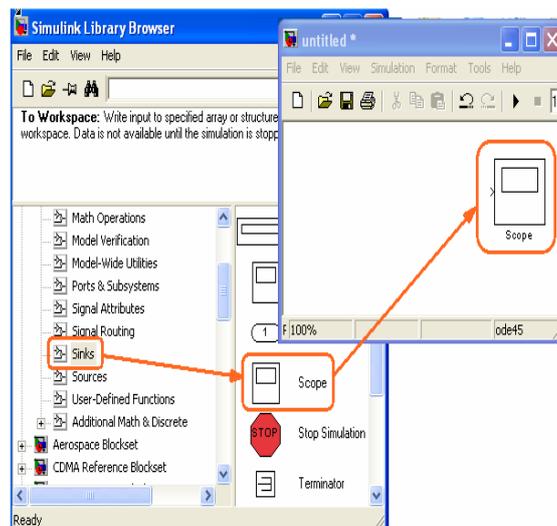


Figure 1: Simulink library Browser.

Model creation:

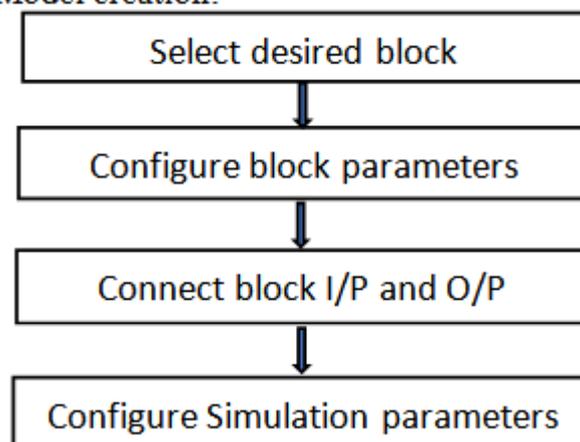


Figure 2: Model creation

Step 1: Select desired blocks –

Getting the blocks into the model window: Follow the steps below to collect the necessary blocks:

- Create a new model (New from the File menu). You will get a blank model window
- Click on the Sources icon in the Library Browser. This opens the Sources window which contains the Sources block library. Sources are used to generate signals.
- Drag the Sine Wave and Clock blocks from the Sources window into the left side of your model window.
- Click on the Sinks icon in the Library Browser to open the Sinks window.

- Drag the Scope and To Workspace blocks into the right side of your model window.
- Click on the Signal Routing icon in the Library Browser to open the Signal Routing window.
- Drag the Mux block into the your model window
- Click on the Math Operations icon in the Library Browser to open Math Operations window.
- Drag the Gain block into your model window.

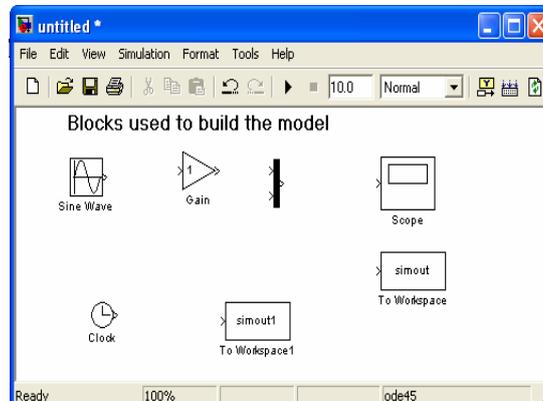


Figure 3: Block used to built.

Step 2: Configure block parameters - Resizing and moving blocks:

- First select the block by clicking with your left mouse button while the pointer is on the block. In each corner of the block a little filled square appears. Put the pointer on one of the corner, press the left mouse button and keep it pressed down. Move the mouse and release the mouse button when the block has the desired size.
- To move a block you first have to select it. Then put the pointer inside the block, press the left mouse button down and keep it pressed down. Drag the block to its new position and release the mouse button.

C. Step 3: Connecting block inputs and outputs Adding a branch line: To connect the output of the Sine Wave to the input of the Mux block, you will use a branch line. Drawing a branch line is slightly different in that to start, the branch line must be welded to an existing line. Position the pointer on the line that connects the Sine Wave block to the Gain block. Without moving the mouse, press and hold down the CTRL key and then press and hold the left mouse button. Drag the pointer to the input port of the Mux block and release the mouse button.

Step 4: Configure simulation parameters –

Step 4.1 Adding Annotation: Annotations add textual information about the designed model. One can add an annotation to any unoccupied area of your block diagram. To create a model annotation, double click in an unoccupied area. A small rectangle appears and the pointer changes to a vertical insertion bar. Start typing the annotation content. To start a new line, simply hit the Enter key. Each line is centered within the rectangular box that surrounds the annotation. To close the annotation, click with the mouse elsewhere in the window. In your block diagram you will place a title at the top and you will place your name and date at the lower right, as depicted below. To move an annotation, drag it to a new location. To change the font size of an annotation, select it and with the mouse select Format and then Font. In the pop-up window select the desired font size and then click OK. The font size will change when the annotation is deselected, to do so simply click elsewhere.

Step 4.2 Adding Signal Labels: Labels can be added o lines to further annotate your model. To create a signal label, double click on a line segment and type the label at the insertion point. When you click on another part of the model, the label fixes its location. To move a signal label, point at it with the mouse, press and hold the left mouse button and drag the label to a new location. To unselect a label, click elsewhere in the model.

First you will gather all the necessary blocks from the Library Browser. Then you will modify the blocks so that they correspond to the blocks of the desired model. Lastly, but not the least, you will connect the blocks with lines to form the complete system and set the overall simulation parameters. After this, you will simulate the complete system to verify that it works.

The control system in the vehicle is composed of power, communication, drive, and sensor modules. The control system functions in the following way: the computer accepts the speed command through the human-machine interface. The RS-232 signals output by the computer are converted to RS-485 signals for remote transmission to the vehicle via the communication module. The signals are converted to analogue voltage through the D/A module and input into the drive. The drive causes the actuator, together with the propeller, to produce the propelling force. The acceleration data obtained from the vehicle are converted to RS-

485 signals via the A/D module and transmitted to the communication module for conversion. The converted signals are then transmitted to and stored in the computer. Fig 6 shows the signal flow chart of the vehicle. In general, the control system can use analogue signals, digital signals and pulse width modulation (PWM) to control the actuator. In this study, uniform analogue signals for the control were used, which were convenient for the transmission and reception of data among the devices.[10]

To demonstrate the efficiency and performance of MATLAB, it was used to develop a motor speed control system. Both open-loop and closed-loop systems were designed and simulated with MATLAB. They were then implemented using a TMS320F2812 digital signal processor (DSP) from Texas Instruments. MATLAB has special tools included that allow it to compile code for this DSP, program it, and transfer data to and from it in real-time.

MATLAB encompasses numerous facets of engineering. There are built in functions dealing with video and image processing, filter design, communications, control system design, just to name a few. Rather than having to be an expert in every field, the engineer can focus on solving the problem at hand. All that is required is a basic knowledge of the MATLAB development environment, the help system, and Matlab's programming language. It's a simple matter to locate and integrate the appropriate functions to solve the current problem. Digital signal processing will always involve with signals by doing some mathematical operation on signals then it produces an output signal in the form of digital so the conversion may require whether to digital to analog or analog to digital.

The basic elements of DSP are ADC, Digital Signal Processor, and DAC.



Figure 4: Basic DSP kit

Usually, the Digital signal processing is mathematical manipulation on a DSP chip. All the algorithms will end with the mathematical equation whether a signal, image or video processing. We will start with the basic DSP algorithm to learn how to implementation of DSP algorithm on DSP kits. Before proceed to implement stage first we must be familiar with that algorithm.

Linear Convolution:

Convolution is a one of the basic algorithm in Digital signal processing. It is nothing but a mathematical operation, just as addition, subtraction, multiplication, and integration. Mathematical operation takes two numbers and produces a third number, while convolution takes two signals and produces a third signal. In linear systems, convolution is used to describe the relationship between three signals of interest: the input signal, the impulse response, and the output signal. If the input and impulse response of a system are $x[n]$ and $h[n]$ respectively, then the Convolution is given by the expression, $\sum_{k=0}^{n-1} x(k) * h(n - k)$ [11]. Texas Instruments TMS320 is a blanket name for a series of digital signal processors (DSPs) from Texas Instruments. It was introduced on April 8, 1983 through the TMS32010 processor, which was then the fastest DSP on the market. Finally the output of fluent are compared with constraints in MATLAB. According to the survey the use of MATLAB will be quite efficient and smooth working. It will actually help in dive centres for diving according to the adventure sports.

II. Conclusion

This paper has been demonstrated an optimization scenario to modifying an arbitrary DPV for diver hydrodynamic characteristic such as pitching moment might be considered. Since the DPV compact and modular design allow an easy assembly enhancing the portability even further. In general, the traditional method for DPV is not important because of time and fuel consuming.

Acknowledgement

We would like to say thanks to all who supported us at various stages to complete this piece of work.

Reference

- [1]. Development of a motor speed control system using MATLAB and Simulink, implemented with a digital signal processor. Andrew Klee, B.S. University of central florida., 2003.
- [2]. www.padi.com
- [3]. A practical BLDC motor design procedure diver propulsion vehicle applications. Yannis L Karnavas member IEEE, Ioannis D. Chasiotis, Student member IEEE and D.N Stavoulellis.
- [4]. Systematic modelling and model based simulation of a remotely operated vehicle using MATLAB and Simulink. C.S Chin – School of Marine and Technology, jan 2012

- [5]. Frederic, P. et al. (2010). Diver Propulsion Vehicle book, by VDM Publishing, ISBN 6130783159, 9786130783150
- [6]. Griffiths, G. and I. Edwards (2003). Designing and operating next generation vehicles, Elsevier Oceanography Series 69, 229–236.
- [7]. QY18 DPV "Archived copy". Archived from the original on 2014-01-08. Retrieved 2013-06-17. accessed 18 June 2013
- [8]. Optimization of a Diver Propulsion Vehicle Hydrodynamics Parameter and its' Shape Improvement by CFD Method for Improving Underwater Speed Record M. R. Sadeghizadeh¹, B. Saranjam¹ and R. Kamali² ¹Department of Hydro-Aerodynamic Research Center of MUT University, I. R. Iran ²School of Mechanic Engineering, Shiraz University, I.R. Iran. Volume no. 5 pp.1319-1328, April,2017.
- [9]. Autonomous underwater vehicle: Modelling, control design and simulation. SubihaWadoo, PushkunKachroo. CRC press 2011.
- [10]. Simulink tutorial.
- [11]. How to implement the DSP algorithms on TMS320C6745/C5505/F2812 DSP kits. 2013

IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) is UGC approved Journal with Sl. No. 4198, Journal no. 45125.

Mr. Swanand U.Mane " Diver Propulsion Vehicle using MATLAB: A survey." IOSR Journal of Electrical and Electronics Engineering (IOSR-JEEE) 14.3 (2019): 08-14.