Multicriterion Genetic Programming for Trajectory Planning of Underwater Vehicle

Jerzy Balicki,[†]**,**

Naval University of Gdynia, ul. Smidowicza 69, 81-103 Gdynia, Poland

Summary

An autonomous underwater vehicle is supposed to find its trajectory, systematically. It can be obtained by using genetic programming for multi-criterion optimization of the set of alternative paths. For assessment of an underwater vehicle trajectory, three crucial criteria can be used: a total length of a path, a smoothness of a trajectory, and a measure of safety.

Key words:
Genetic programming, remote operating vehicle, multi-criterion optimization.

Introduction

Path of an underwater vehicle can be determined by genetic programming that is capable to solve some multiobjective optimization problems [1]. For evaluation of a vehicle trajectory, three main criteria can be used: a total length of a path, a measure of safety, and a smoothness of a trajectory. A genetic algorithm and also evolutionary algorithms have been just applied for finding Pareto-optimal trajectory of underwater vehicles [1].

An algorithm implemented in the computer board of an underwater vehicle should find a path between two specified locations in a three dimensional space, which is collision-free and satisfies optimization criteria.

Evolutionary algorithms are the extended genetic algorithms by another chromosome representation, more complex operators, and a specific knowledge related to the optimization problem [7, 8]. On the other hand, evolution strategies give solution of high quality for some optimization problems. The up to date motivating approach is related to applying genetic programming [3].

Genetic programming is an appealing paradigm of an artificial intelligence [3]. Solutions to several problems have been found for instances from different areas like optimal control, planning and sequence induction. Genetic programming permits finding solutions to symbolic regression, automatic programming or discovering a game playing strategy. Furthermore, problems related to empirical discovering and forecasting, symbolic integration or differentiation, discovering mathematical identities or classification and decision tree induction can be solved by genetic programming. Evolution of emergent behavior and also automatic programming of cellular automata are on the list of problems that have been solved successfully by genetic programming.

Super Achille M4 and Koral 100 are remotely operated vehicles designed for underwater observation in hostile environment [9]. The foremost attributes of these vehicles are their power capability and their compactness. The vehicle Super Achille M4 consists of two divisions. The upper part ensures the vehicle positive buoyancy and houses the sonar head. The lower part consists of a watertight frame made of welded pressure-resistant tubular stainless steel. The underwater vehicle is equipped with four three-phase asynchronous thruster motors with propellers. There is a surface control unit with its power cable. It is possible to extend unit's capabilities by using the board computer to find the trajectory of the vehicle.

1. Criteria of anti-collision trajectory

If (x_1,y_1,z_1) is the starting point and (x_M,y_M,z_M) is the destination point, then the path can be represented as follows:

$$x=(M,x_1,y_1,z_1,\ldots,x_m,y_m,z_m,\ldots,x_M,y_M,z_M) \quad (1)$$

The point (x_m,y_m,z_m) for $m\in\{1,\ldots M\}$ is feasible, if both the segment from (x_{m1},y_{m-1},z_{m-1}) to (x_m,y_m,z_m) and the segment from (x_m,y_m,z_m) to $(x_{m+1},y_{m+1},z_{m+1})$ do not cut forbidden areas for the vehicle. A path x can be either feasible (collision-free) or infeasible. A path with at least one unfeasible point is non-feasible, too. We assume that at the time t the forbidden areas are given, and it is possible to determine if any trajectory is feasible or not. M – the number of turn points, that define a trajectory x, can be changed. There are given both the maximum number of points $M_{\max}$ and the minimum number of points $M_{\min}$.

Some formal constraints for including trajectories in the given water areas are formulated, as follows:

Manuscript received November 5, 2006.
Manuscript revised December 25, 2006.

$$\begin{aligned} & M_{\min} \le M \le M_{\max}, \\ & X^{\min} \le x_m \le X^{\max}, \quad m = \overline{1,M}, \\ & Y^{\min} \le y_m \le Y^{\max}, \quad m = \overline{1,M}, \\ & 0 \le z_m \le Z_m^{\max}, \qquad m = \overline{1,M}, \end{aligned} \tag{2}$$

where

$X^{\min}, X^{\max}$ – area constraints for the coordinate x_m,

$Y^{\min}, Y^{\max}$ – area constraints for the coordinate y_m,

$Z_m^{\max}$ – the maximal depth of water at (x_m, y_m).

Distinguish criteria can be used for evaluation the quality of the planning underwater vehicle path. The total length of the path is usually discussed because of the time and economical aspects of motion [8]. Let $p_m=(x_m,y_m,z_m)$ denotes an interior point of trajectory direction changing. The total length of the path x can be expressed, as follows:

$$F_1(x) = \sum_{m=1}^{M-1} d(p_m, p_{m+1}), \tag{3}$$

where $d(p_m,p_{m+1})$ is a distance between two adjacent path points $p_m=(x_m,y_m,z_m)$ and $p_{m+1}=(x_{m+1},y_{m+1},z_{m+1})$.

The distance $d(p_m,p_{m+1})$ between two adjacent path points can be calculated, as below:

$$d(p_m, p_{m+1}) = \sqrt{(x_m - x_{m+1})^2 + (y_m - y_{m+1})^2 + (z_m - z_{m+1})^2}. \tag{4}$$

The length of a trajectory is the same, if this path goes through a forbidden area or through a permitted field. So, a safe aspect of navigation is required to distinguish that situation.

2. Safety criterion of trajectory

Coordinates $(X^{\min}, X^{\max}, Y^{\min}, Y^{\max}, Z^{\max})$ determine the rectangular of the water space, where an underwater vehicle is supposed to omit some obstacles. Let regions of obstacles $\Omega_1,\ldots,\Omega_k,\ldots,\Omega_K$ in the water rectangular be recognized before the plan of a trajectory is determined. A set of obstacles is constant and obstacles do not move during the vehicle movement along determined trajectory.

The second criterion for evaluation the quality of a trajectory is a safety measure, which can be defined according to the following formula [1]:

$$F_2(x) = \max_{m=\overline{1,M-1}} b(p_m, p_{m+1}), \tag{5}$$

where $b(p_m,p_{m+1})$ denotes the penalty value for the line segment from the point p_m to the point p_{m+1}, if the segment cuts any forbidden area.

The penalty value $b(p_m , p_{m+1})$ is defined, as follows:

$$b(p_m, p_{m+1}) = \begin{cases} d_{\min} - r(p_m, p_{m+1}), & \text{if } r(p_m, p_{m+1}) \ge d_{\min}, \\ e^{\beta(d_{\min} - r_m(p_m, p_{m+1}))} - 1, & \text{otherwise,} \end{cases} \tag{6}$$

where

$r(p_m, p_{m.+1})$ – the smallest distance from the line segment connecting path points $(p_m, p_{m.+1})$ to an object from all detected objects that create forbidden areas,

$d_{\min}$ – a parameter defining a minimal safe distance from the underwater vehicle to another object,

β – a positive penalty coefficient.

If the smallest distance from the line segment connecting path points $(p_m, p_{m.+1})$ to an object from all detected objects is non-smaller than the save distance $d_{\min}$, then the penalty $b(p_m, p_{m.+1})$ is negative. When the distance between a path segment and the closest obstacle is smaller than $d_{\min}$, then the penalty $b(p_m, p_{m.+1})$ is positive and it grows exponentially. The function F_2 is defined as a maximum of $b(p_m, p_{m.+1})$ for all segments to make sure that if a certain segment of a path is dangerously close to an obstacle, i.e. within distance $d_{\min}$, then the path is penalized strongly even if all other path segments are safe. The safety criterion F_2 should be minimized to obtain a trajectory as safe as possible.

3. Smoothness of trajectory

The third criterion F_3 should maintain a smooth trajectory to avoid sudden turns of direction according to the following formula [1]:

$$F_3(x) = \max_{m=\overline{2,M-1}} s(p_m), \tag{7}$$

where $s(p_m)$ denotes the measure of a trajectory "curvature" at the point p_m.

The trajectory curvature at the point p_m can be defined as follows:

$$s(p_m) = \frac{\alpha_m}{\min\{d(p_{m-1}, p_m), d(p_m, p_{m+1})\}}, \tag{8}$$

where $\alpha_{m.}$ is the angle between the extension of the line segment (p_{m-1}, p_m) and the line segment $(p_m, p_{m.+1})$ on a plane determined by above segments.

We assume, that $\alpha_m \in [0,\pi]$. For the same distances the trajectory is smoother, if the maximum angle for it is smaller. If the minimum length from distances $d(p_{m-1}, p_m)$ and $d(p_m, p_{m+1})$ is longer, then there are less points p_m, where the direction of trajectory is changed. The criterion F_3 is supposed to be minimized.

Another approach for improving the smoothness of trajectory is related to the sum of all trajectory curvatures at points $p_m, m = \overline{1,M}$, as follows:

$$\overline{F}_3(x) = \sum_{m=2}^{M-1} s(p_m). \tag{9}$$

What is more, the minimization of sum-squared function can be carried out, as below:

$$\widetilde{F}_3(x) = \sqrt{\sum_{m=2}^{M-1} s(p_m)}. \tag{10}$$

4. Three-criterion optimization problem

There are several classes of multiobjective optimal solutions related to the preferences for criteria. If partial criteria are ordered from the most important criterion to the least important criterion, then a hierarchical solution can be found. In a multicriteria navigation of the underwater vehicle, the safety criterion seems to be the most important.

If criteria have the equal priority, then Pareto-optimal solutions can be considered [1]. Because of the great number of Pareto-optimal solutions some reducing techniques can be used. For instance, the compromise solutions with the parameter p equal to 1, 2 or ∞ may be extracted from the Pareto set. Moreover, an additional criterion can be used. If the anti-collision situation permits on a dialog with the navigator, then some dialog techniques can be introduced, where the navigator chooses the best trajectory from the proposed set of trajectories during several iterations. An expert knowledge can be respected, too.

Let the multicriteria optimization problem be considered for finding trajectory of the underwater vehicle as the Pareto-optimal solution:

$$(X, F, R), \tag{11}$$

where
X – the set of admissible trajectories,
F – the vector criterion,
R – the relation for finding Pareto-optimal solutions [2].

Because of the variable number of points in trajectory, a set of all trajectories (admissible or non-admissible) consists of vectors with no more than $3M_{\max}$ coordinates. It can be denotes as $\mathbb{X} = 2^{\mathbb{T}}$, where $\mathbb{T} = \mathbb{R}^{3M_{\max}}$ and $\mathbb{R}$ is a set of real numbers.

The set of feasible trajectories is defined, as follows:

$$\begin{aligned} X = \{ & x \in \mathbb{X} \mid x = (x_1, y_1, z_1, \ldots x_m, y_m, z_m, \ldots x_M, y_M, z_M), \\ & M_{\min} \le M \le M_{\max}, \\ & X^{\min} \le x_m \le X^{\max}, \ m = \overline{1,M}, \\ & Y^{\min} \le y_m \le Y^{\max}, \ m = \overline{1,M}, \\ & 0 \le z_m \le Z_m^{\max}, \quad m = \overline{1,M} \} \end{aligned} \tag{12}$$

Obstacles are respected by increasing the safety criterion of the trajectory. The vector criterion $F : \mathbb{X} \to \mathbb{R}^3$ has three partial criteria, as follows:

$$F(x) = [F_1(x), F_2(x), F_3(x)], \ x \in \mathbb{X}, \tag{13}$$

where $F_1(x), F_2(x), F_3(x)$ are calculated according to (2), (5) and (7).

5. Evolutionary methods

For solving continues optimization problems, an evolution strategy was proposed [6]. An extension of evolution strategy on multi-objective optimization was introduced by Kursawe [4]. An evolution strategy for finding Pareto-optimal trajectory for an underwater vehicle has been developed in [1]. Chromosome in evolution strategies consists of two main parts, as follows:

$$\overline{X} = (x, \sigma), \tag{14}$$

where σ is a deviation standard vector for trajectory x

Evolutionary algorithms for finding Pareto-optimal solutions are an alternative approach to evolution strategies [5]. An overview of evolutionary algorithms for multiobjective optimization problems is presented by Fonseca and Fleming [3]. The ranking procedure for non-dominated individuals is applied to avoid the discrimination of the interior Pareto solutions.

The evolutionary algorithm for finding Pareto-optimal trajectories of the underwater vehicle is presented in [1]. At the beginning, L randomly chosen trajectories from the given point A to B are generated. Each initial trajectory performs the coordinate constraint, according to the formula (2).

If there are some feasible solutions in a population, then the Pareto-optimal trajectories are sought, and they get the rank 0. Then, they are temporary eliminated from the population. From reduced population, the new Pareto-optimal trajectories are found and get the rank 1. This procedure with increasing of the rank is repeated until the set of feasible solutions will be exhausted. That is why, all non-dominated solutions have the same rank and the same fitness to reproduction.

6. Genetic programming

Although, several different problems have been taken into account, finding trajectory of the underwater vehicle by genetic programming is a new scientific challenge. Figure 1 shows an example of a tree of the computer program performance. This tree corresponds to the program written in the LISP language, as follows:

(GT (* -1 x) (* y (SQRT y)))

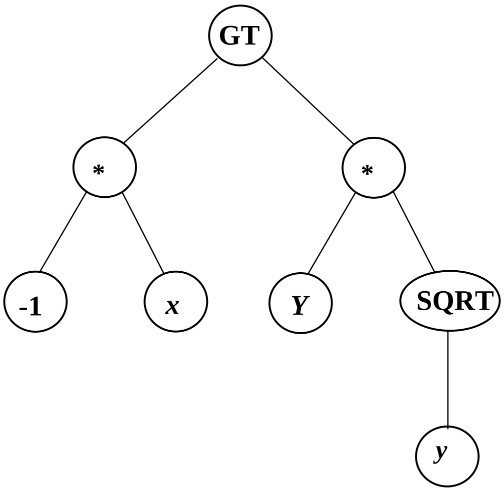

Fig. 1. Tree as a model of the computer program

Above program calculates both the value $-x$ and $y\sqrt{y}$, and then compares $-x$ to $y\sqrt{y}$. If $-x$ is greater than $y\sqrt{y}$, then an outcome of the LISP procedure is equal to 1. However, in the other case, the result is –1, because the function GT is defined in such a way.

This tree is equivalent to the parse tree that most compilers construct internally to represent the given computer program. If a computer program was represented by any algorithm form, genetic operators like reproduction, crossover or mutation would be complicated to implement.

Despite the data structure representing chromosomes in an evolution strategy or an evolutionary algorithm, a chromosome for genetic programming is the tree of a computer program. Even the simplest procedure differs from a complex data structure significantly because the procedure can calculate that gives ability to represent not only knowledge about a problem, but also it gives possibility to draw conclusions or to process data in the way difficult to discover. That is, a computer program may model a solution to the problem as an intelligent procedure.

Generation of the tree is an important step for finding Pareto-optimal trajectories. The size of the generated tree is limited by the number of nodes or by the number of the tree levels. Nodes in the tree are divided on functional nodes and terminal ones. A functional node represents the procedure randomly chosen from the primary defined set of functions:

$$\mathsf{F} = \{f_1, \ldots, f_n, \ldots, f_N\} \tag{15}$$

Each function should be able to accept, as its arguments, any value and data type that may possible be returned by the other procedure [3]. Because a procedure is randomly chosen from the set, and then it is returned, each function should be able to accept, as its arguments, any value and data type that may possible be returned by itself. Moreover, each procedure should be able to accept any value and data type that may possible be assumed by any terminal in the terminal set:

$$\mathsf{T} = \{a_1, \ldots, a_m, \ldots, a_M\} \tag{16}$$

An above property of procedure is called a closure property because each function should be well defined and closed for any arrangement of arguments that it may come across.

Another property, called the sufficiency property, requires that the solution to the problem be expressed by the any combination of the procedures from the set of functions and the arguments from the set of terminals. For example, the set of functions $\mathsf{F} = \{AND, OR, NOT\}$ is sufficient to express any Boolean function. If the logical operator *AND* is removed from this set, the remaining procedure set is still satisfactory for realizing any Boolean function. A sufficient set is $\{AND, NOT\}$ as well.

Let the following set of procedures be considered:

$$\mathsf{F} = \{IF\_OBSTACLE, MOVE, IF\_END, +, -, *, /\} \tag{17}$$

The procedure *IF_OBSTACLE* takes two arguments. If the obstacle is recognized ahead the underwater vehicle, the first argument is performed. In the other case, the second argument is executed. The function *MOVE* requires three arguments. It causes the movement along the given direction with the velocity equals the first argument during assumed time Δt. The time Δt is the value that is equal to the division a limited time by $M_{\max}$. The direction of the movement is changed according to the second and third arguments. The second argument is the angle of changing this direction up if it is positive or down if it is negative. Similarly, the third argument represents an angle of changing the direction to the left if it is positive or – to the right if it is negative. The last procedure *IF_END* ends the journey of the underwater vehicle if it is in the destination region or the expedition is continued if it is not there.

The set of arguments consists of the real numbers generated from the interval (-1; 1).

A genetic algorithm has been applied for operating on the population of the computer procedures written in the Matlab language. Our initial numerical experiments confirm that feasible, sub-optimal in Pareto sense, trajectories can be found by genetic programming. Although, the quality of obtained trajectories is a little lower than the trajectories determined by an evolutionary algorithm [1], a paradigm of genetic programming gives opportunity to solve the control problems for changing environment.

7. Parameters of underwater vehicles

An underwater vehicle can be used for supporting an outer inspection of the ship hull, a remote carrier of the cameras, testing of the bottom shape, finding obstructions, or planning the gas pipeline on the ocean bottom. These classes of tasks pressure on the technical parameters of submarine means of transport like maximal speed or maximal depth of draught. Furthermore, a size and also a weight of the mini-vehicle play important role for task performing. The catalog of foremost factors ends the number of electrical engines that permits its to penetrate the water space.

Table 1 shows some parameters of two selected underwater vehicles. These universal mini-vehicles can perform distinguish tasks to substitute the human being motion and to decrease the risk of the underwater actions. Both Koral 100 and Super Achille M4 are prepared for underwater observation in hostile environment. They are remotely operated devices by an operator from the surface deck, the land center or even helicopter.

Table 1. Technical parameters of underwater vehicles [9]

Technical parameter	Koral 100	Super Achille M4
maximal speed	1.5 m/s	6 m/s
maximal depth of draught	100 m	500 m
number of electrical engines	5	4
size [m]	0,7 meter in each dimension	length 0.72 width 0.6 height 0.52
weight [kilos]	90	120

The first main characteristic of Super Achille M4 is its power capability. It is four times faster than Koral 100 as well as a maximal depth of draught is fifth times greater. It is possible because of the higher quality of electrical engines than five engines of Koral 100. This underwater vehicle is equipped with four three-phase asynchronous thruster motors with propellers. There are two horizontal motors for forward and also backward propulsion. They can be used for rotation and direction. Moreover, it is equipped with a lateral motor as well as a vertical motor.

The second main characteristic of Super Achille M4 is its compactness. Although, a vehicle weight is 120 kg, its length is 0.72 mm, the width – 0.60 m, and the height – 0.52 m. So, we call it a mini-vehicle to distinguish it from micro-vehicles and vehicles determined by size that is larger than 1.5 m in at least one dimension. Because of its compactness Super Achille M4 uses less electrical energy and also is more responsive than larger vehicles.

Figure 2 shows main elements of the underwater vehicle Koral 100. The vehicle consists of two parts. The upper part ensures the vehicle positive buoyancy and houses the sonar head. The lower part consists of a watertight frame made of welded pressure-resistant tubular stainless steel. There is a surface control unit with its power cable.

Better quality of a vision system (TV-camera, photo-camera, stronger reflectors) characterizes Super Achille M4 than Koral-100. Moreover, a board computer can be programmed easily. Therefore, it is possible to extend unit's capabilities by finding trajectory of the underwater vehicle. Super Achille M4 is equipped with a fluxgate compass, completed with gyrocompass. Furthermore, there is a pressure sensor.

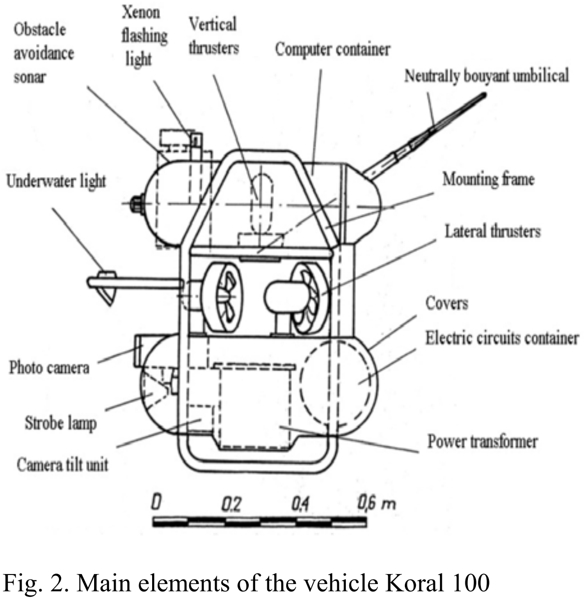

Fig. 2. Main elements of the vehicle Koral 100

8. Concluding remarks

Genetic programming can be applied for control an underwater vehicle. A computer program as a chromosome is a subject of genetic operators such as recombination, crossover and mutation. It gives possibility to represent knowledge that is specific to the problem in more intelligent way than for the data structure. That is, we process the potential ways of finding solution not the possible solutions.

Our future works will focus on comparing obtained results to the other sets of procedures and terminals to find the Pareto-optimal trajectories of an underwater vehicle for different environment.

References

[1] J. Balicki: *Finding Trajectory of Underwater Vehicle by Multi-Criterion Genetic programming*, Proceedings of the 9th IEEE International Conference on *Methods and Models in Automation and Robotics*, (Ed. R. Kaszyński), Vol. 1, Międzyzdroje, Poland, 25-28 August 2005, pp. 237-242.

[2] C.M. Fonseca, P.J. Fleming, *An overview of evolutionary algorithms in multiobjective optimization*, Evolutionary Computation, Vol. 3, No. 1, 1995, pp.1-16.

[3] J.R. Koza, Genetic programming. The MIT Press, Cambridge 1992.

[4] E. Kursawe, *A variant of evolution strategies for vector optimization.* In H.-P. Schwefel, R. Manner (Eds.): *Parallel problem solving from nature*, 1st Workshop, Lecture Notes in Computer Science, Vol. 496, 1991, pp. 193-197.

[5] J.D. Schaffer, *Multiple objective optimization with vector evaluated genetic algorithm,* Proc. Int. Conf. on Genetic Algorithms and Their Applications, 1985, pp. 93-100.

[6] H.P. Schwefel: *Evolution and optimum seeking.* John Wiley&Sons, Chichester 1995.

[7] J. Xiao, Z. Michalewicz, L. Zhang, K. Trojanowski: *Adaptive Evolutionary Planner/Navigator for Mobile Robots.* IEEE Transactions on Evolutionary Computation, Vol. 1, No. 1, April 1997, pp. 18-28.

[8] C.K. Yap, *Algorithm motion planning.* In the Advances in Robotics, vol. I: Algorithmic and Geometric Aspects of Robotics, J.T.Schwartz and C.K.Yap, Eds., Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 95-14.

[9] *Remotely operated vehicle Super Achille. Operator manual*, Comex-Pro, Marseille, France, 2000.

Jerzy Balicki received the M.S. and Ph.D. degrees in Computer Science from Warsaw University of Technology in 1982 and 1987, respectively. During 1982-1997, he stayed in Computer Center of High School of Gdynia to study management systems, mobile systems, and decision support systems. Then, he achieved habilitation from Technical University of Poznan in 2001. He was admitted as a professor at Naval University of Gdynia in 2002.