Applying Particle Swarm Optimization to the Motion-Cueing-Algorithm Tuning Problem

Sergio Casas
IRTIC, University of Valencia
Spain
Sergio.Casas@uv.es

Cristina Portalés
IRTIC, University of Valencia
Spain
Cristina.Portales@uv.es

Inmaculada Coma
IRTIC, University of Valencia
Spain
Inmaculada.Coma@uv.es

Marcos Fernández
IRTIC, University of Valencia
Spain
Marcos.Fernandez@uv.es

ABSTRACT

The MCA tuning problem consists in finding the best values for the parameters/coefficients of Motion Cueing Algorithms (MCA). MCA are used to control the movements of robotic motion platforms employed to generate inertial cues in vehicle simulators. This problem is traditionally approached with a manual *pilot-in-the-loop* subjective tuning, based on the opinion of several pilots/drivers. Instead, this paper proposes applying Particle Swarm Optimization (PSO) to solve this problem, using simulated motion platforms and objective indicators rather than subjective opinions. Results show that PSO-based tuning can provide a suitable solution for this complex optimization problem.

CCS CONCEPTS

• **Theory of computation ~ Optimization with randomized search heuristics** • *Computing methodologies ~ Real-time simulation* • *Computer systems organization ~ Robotic control*

KEYWORDS

Optimization, particle swarm, PSO, MCA, tuning, simulation.

1 INTRODUCTION AND RELATED WORK

Motion Cueing Algorithms (MCA) are used in vehicle simulators to control the behavior of motion platforms. They take (as input) the simulated physical state of the vehicle and they provide (as output) the desired pose for the motion platform in the form of rotational and translational degrees of freedom (DOF) [1].

The evaluation of these algorithms is based on a measure of *motion fidelity* [2]. Since the motion platform is unable to replicate the motion of the simulated vehicle, the self-motion system needs to control what parts of this motion are eliminated and what parts are not. This is almost always controlled by a series of parameters/coefficients of the MCA that need to be tuned before it is used in the simulator. These parameters substantially modify the behavior of the MCA, so it is essential to find a method to tune them. In the case of the *classical MCA* algorithm [3], which is the one used in the experiments of this work, the parameters and their meanings can be found in [4], [5].

The tuning of MCA is usually performed with the *pilot-in-the-loop* approach. This method implies executing successive tests on the simulator with a pilot/driver and an expert changing, in real-time, the parameters of the MCA in reaction to the comments of the pilot/driver. The process is not systematic and is repeated until the user is satisfied with the result (if that happens). For this reason, a systematic automatic solution based on objective evaluations is proposed in [6]. This proposal substitutes the subjective evaluation by the calculation of objective motion fidelity metrics/indicators, which are calculated upon the execution of a simulation of the motion platform movements with a Genetic Algorithm (GA). This solution represents a promising approach, but it would be advisable to test if better optimization strategies can be found. For this reason, this work proposes to use a different heuristic that has also proven to be capable of solving optimization problems with vast search spaces: Particle Swarm Optimization (PSO) [7]. PSO has been used for parameter tuning in other algorithms [8] and even for MCA [9] with a different approach. PSO is easy to implement, and the adaptation of this heuristic to this problem is feasible.

2 MATERIALS AND METHODS

The PSO heuristic can be adapted to the MCA tuning problem. Let us consider each t-uple t of parameters of the MCA as a particle, which contains possible values for each of the parameters of the MCA. The particle can be represented by a sorted sequence of the MCA parameter values: $t[1], t[2], \ldots t[i], \ldots t[n]$. Thus, the

GECCO '17 Companion, July 15-19, 2017, Berlin, Germany
ACM 978-1-4503-4939-0/17/07.
http://dx.doi.org/10.1145/3067695.3075990

values of t can be seen as a vector representing the position of the particle in the parameter space in which the optimization algorithm searches. To perform the optimization and search for the best set of values for the parameters, an evaluation/fitness function $f(w, t)$ is needed, where w is the MCA used, and t is the t-uple of parameters. The evaluation function returns a motion fidelity indicator for the MCA with this configuration (i.e., with this set of values for the parameters). The rationale and details of the motion fidelity indicators used in this paper are explained in [5]. Although an actual motion platform can be used to evaluate its performance by means of objective indicators, the proposed method is to use a virtual motion platform instead of a physical one. With this decision, the motion fidelity indicators can be calculated faster than real-time. The details of the motion platform simulator can be found in [10] and are not the focus of this paper.

The PSO-based solution implements the PSO algorithm as described in [7]. It evaluates each particle and stores the following information: the position of the particle that provides the best global indicator, the indicator itself, the best local indicator and its position. The positions of the particles are updated following the best rule proposed by Kennedy & Eberhart in [7].

3 EXPERIMENTS AND RESULTS

After performing successful correctness tests that demonstrate that the PSO implementation is correct and analyze the effect of population size, this value was set to 30 particles. This is the only parameter that needs to be setup for PSO, compared with four, in the case of the GA implementation in [6].

In order to draw conclusions on the performance of the PSO and analyze its behavior for different motion fidelity indicators, a comparative test is presented. For this experiment, a virtual 6-DOF motion platform, with 96-second MCA input signals, is used. These signals (only specific force and angular velocity are recorded) are extracted from an open-wheel Formula 3 vehicle in one lap of the Monaco Grand Prix circuit using the *rFactor* racing simulator. 18 parameters of the MCA are allowed to be varied/tuned, and the search time is set to 1000 seconds. The computer running the experiments is an Intel Pentium G840 at 2.80 GHz, with 8 GB of RAM and Windows 7 operating system. Several motion fidelity indicators are used: NPC, NAAD, AAS, ED and a multiplicative combination of AAS, NAAD and NPC [5]. The 6-DOF motion platform capabilities can also be seen in [5]. For this experiment, PSO is compared against the GA (set-up as in [6]) and a Monte-Carlo algorithm (MC). The results of the experiment are depicted in Table 1. Since the three optimization algorithms are probabilistic, 50 repetitions of each test are performed in order to calculate average values.

Results show that the PSO-based solution is the preferred choice in every case. With the combined indicator, the PSO algorithm reveals itself as a much more efficient solution than the GA-based solution, something that can be said for the rest of motion fidelity indicators but gets clearer with this one.

As a final note, it is important to emphasize that further experiments, not shown here for the sake of brevity, reveal that neither the robotic motion platform seems to have influence in the performance of the optimization algorithm, nor does the motion fidelity indicators, since the results are rather consistent for the different configurations tested. They also show that this approach converges faster than the GA for this problem.

Table 1: Compared 6-DoF performance.

Motion Fidelity Indicator	MC	GA	PSO
NPC	1.291992	1.265862	1.263993
NAAD	1.171382	1.168722	1.162953
AAS	5.852526	4.695831	3.169515
ED	1.181666	1.115000	1.100000
Combined	9.638290	8.171634	5.844439

4 CONCLUSIONS AND FUTURE WORK

The MCA tuning problem can be approached and solved, in certain circumstances, by using optimization techniques, instead of the traditional subjective pilot-in-the-loop solution. The major benefit of this approach is to accomplish an automatic and unattended method to obtain the best values for the parameters of a motion cueing algorithm. In this regard, PSO provides better results than a previously published GA, as the performed experiments have proven. It is true that the amount of improvement in terms of average indicators is sometimes small, but PSO is easier to tune, as only one parameter (the number of particles) needs to be set. In addition, PSO can provide a satisfactory solution earlier than GA, which can be extremely important if the motion platform is used for entertainment or unregulated training, where the MCA parameters may have to be adjusted if a different track, vehicle or even simulator is used.

Future work includes looking for different strategies, such as expensive black-box optimization. It would be also interesting to compare the performance of PSO for different vehicles, for different simulators, or objective indicators.

REFERENCES

[1] Reymond, G. and A. Kemeny, *Motion Cueing in the Renault Driving Simulator.* Vehicle System Dynamics: International Journal of Vehicle Mechanics and Mobility, 2000. **34**: p. 249-259.

[2] Sinacori, J.B., *The Determination of Some Requirements for a Helicopter Flight Research Simulation Facility*. 1977, Moffet Field: CA, USA.

[3] Reid, L.D. and M.A. Nahon, *Flight Simulation Motion-Base Drive Algorithms: Part 1 - Developing and Testing the Equations*. 1985, UTIAS: University of Toronto.

[4] Reid, L.D. and M.A. Nahon, *Flight Simulation Motion-Base Drive Algorithms: Part 2 - Selecting the System Parameters*. 1986, UTIAS: University of Toronto.

[5] Casas, S., et al., *Motion-Cuing Algorithms: Characterization of Users' Perception.* Human Factors: The Journal of the Human Factors and Ergonomics Society, 2015. **57**(1): p. 144-162.

[6] Casas, S., et al., *Towards a simulation-based tuning of motion cueing algorithms.* Simulation Modelling Practice and Theory, 2016. **67**: p. 137–154.

[7] Kennedy, J. and R. Eberhart. *Particle Swarm Optimization*. in *IEEE International Conference. on Neural Networks*. 1995. Perth, WA, Australia: IEEE Service Center, Piscataway, NJ, USA.

[8] Shih-Wei, L., et al., *Particle swarm optimization for parameter determination and feature selection of support vector machines.* Expert Systems with Applications, 2008. **35**: p. 1817–1824.

[9] Asadi, H., et al. *A Particle Swarm Optimization-based washout filter for improving simulator motion fidelity*. in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*. 2016. IEEE.

[10] Casas, S., et al., *Towards an extensible simulator of real motion platforms.* Simulation Modelling Practice and Theory, 2014. **45**(0): p. 50-61.