

UNIVERSITY COLLEGE LONDON (UCL)

ELEC0036

---

# Optimizing switching speeds of SOAs using Particle Swarm Optimization

---

*Author*

Vasileios VAMVAKAS

*Student Number*

18062354

*Supervisor*

Dr./ Prof ZERVAS

*Second Supervisor*

Dr./Prof LAST NAME

October 21, 2020  
**A BEng Project Final Proposal**

Contents

1	Introduction	2
2	Goals and Objectives	3
3	Preliminary Assessment of Risks	5

### Abstract

This paper studies particle swarm optimization as a method of producing an electronic drive signal to turn on a semiconductor amplifier, which will act as an ultra-fast switch for data transfer based on existing research [1]. Improvements in the algorithm [2] used will be made by experimenting with changing the sampling-rate, the shell used to formulate the minimum and maximum of the signal and the transfer function of the equivalent circuit used to emulate the SOA. The experimental setup will also be tampered with by implementing the use of cascaded SOAs and observing how the gain profile of the system varies while also employing the FPGA as an 8-bit DAC to provide the control signal instead of an arbitrary waveform generator that was used in the main research.

## 1 Introduction

The third year project I am aiming to work on started as an interest in high-frequency trading and how its implemented using the FPGA. Expressing this interest to my project supervisor he immediately redirected me to an existing research which had to do with Optimal Control of SOAs using Artificial Intelligence for Sub-Nanosecond Optical Switching. This paper investigates methods that could be used to minimize switching speeds of SOA for effective use in data centre networks where it is essential that the switching time of the SOA is considerably less than the forwarding time of the data packet.

In this paper, the disadvantages associated with using SOAs as optical switches are investigated in depth. These are, most importantly, their non-constant transfer function which varies from each piece of hardware to the next, their high overshoot, settling and guard times. Other than altering the physical characteristics of the SOA, electronic signals have been used to counter these previous disadvantages while also keeping the rise time low. The electronic drive signal can be used to allow the optical input to switch on and off without delays.

The step impulse is the most basic form but also PID control has been used and more recently 'Pre-impulse step injection current' and 'Multi-impulse step injection current' have proved to be most effective when attempting to minimize the switching speed and introduce SOA as a method to transfer data [3]. The last method seems to have been the most effective but the settling time still remains an issue according to the paper. Therefore, AI control for sub-nanosecond switching is introduced, which would optimize the control signal

in simulations so that the desired switching speed is achieved with the appropriate characteristics aforementioned. The three AI control methods investigated are the Particle Swarm Optimization, the Ant Colony Optimization and the Genetic algorithm.

The necessity of understanding past methods used is underlined in this paper as the PISIC is utilized to provide a shell for the control signal which significantly improved the settling time of the signal. The paper is more focused in the methods used to configure the electronic drive signals but in my project I intend to look into the efficiency of the control signal generation and what modifications or methods would need to be used for real-world application of this technology. Another important aspect of this research is finding an equivalent circuit transfer function for the SOA which simulates its behaviour as closely as possible.

## 2 Goals and Objectives

In this project I aim to understand the research done by the paper on AI algorithms for control and which method was found to be most ideal for the electronic drive signal of the SOA. The research done clearly suggests that the *dynamic PSO algorithm* used for ultra-fast control is the most efficient and gives the best output result therefore my focus will be on this configuration, most likely. As the PISIC shell and the embedded step used seem to be essential to minimizing the switching time, it could also prove useful for me to look into past methods used and investigate how they can be utilized so that the PSO algorithm [4] can be optimized further. This could allow me to *formulate the shell* differently which decrease the settling time. A shell means that I will confine the input signal between a minimum and a maximum for any given moment in the time period.

My initial objective has been to run simulations of current algorithm, with the given sample size of 240 and utilizing the current method to view results and use them as benchmark for any future changes to compare if they had a net positive result. The first step into altering the use of the algorithm would be to reduce the *sample rate* (240 samples) as current drive signal was over-sampled, as the -3dB frequency response of the SOA is at 0.5 GHz whereas 240 dimensions in a 20 ns time period results into a 6 GHz input. Manipulating dimensions, which correspond to points in signal, could allow for further optical gain and a better output result. Identification of the minimum number of samples that can be used before the performance is affected due to the increased number of iterations needed

or reduced accuracy of output signal will also be needed. The balance between performance of the algorithm and output result needs to be found for optimal switching of the SOA. found to optimize the system.

Another aspect I intend to research upon is improving the *equivalent circuit transfer function* to better emulate the SOA behaviour and thus allow the drive signal found from the simulated transfer function to be more effective in real world applications. One of the extensions recommended is the use of *cascaded SOA* cascaded SOAs. Through experimentation I will attempt to determine whether individual or combinational optimization would be ideal by finding how ASE noise will be affected and whether additional work will need to be done on the circuit to reduce distortion which comes with additional gain. This application allows for a flatter gain curve despite the input signal and has a lot of room for optimization as the number of cascaded SOAs[5] also entails a number of electronic driving signals for each which, if working in tandem, may not affect the switching speed significantly while providing a much more robust result. This could also provide the chance for me to look further into the SOAs gain profile and examine its functionality as a power amplifier[6]. As aforementioned, the purpose of the SOA in data transfer is not to just act as a switch but also to change its optical gain depending its properties such as the input wavelength, the input power and the bias current of signal. By further understanding its operation the variety of applications it can be used in will increase. As part of my initial target I also aim to use the *FPGA's embedded DAC* to input the driving signal into the SOA shifting from the focus of initial research partially onto actual wave generation rather than just the method used to find the drive signal.

The plan I will follow will be to first understand the theory behind the method used and the SOA in general and then run the simulations to enhance my understanding. Then I will be looking towards manipulating the dimensions of the PSO algorithm and then possibly manipulate the transfer function of the equivalent circuit of the SOA and compare the results with the benchmark of the existing documentation. In tandem, I will also be examining how the shell can be improved. After that I will look into the principles of cascaded SOAs and use the new transfer function by distorting it across the different units. Then the optimized algorithm can be used for each to get the best possible results. Throughout the process I will be keeping track of the gain profile of the individual and combinational SOA systems. This is a process I plan to do regularly. I will also need to understand how to implement the DAC of FPGA to perform regular testing.

To organize my tasks I will be using a Gantt Chart made in python using the plotly [7] library where it will be easy for me to update the tasks and their status. The time frames defined will be relatively flexible as new issues may arise that I am not currently aware of.

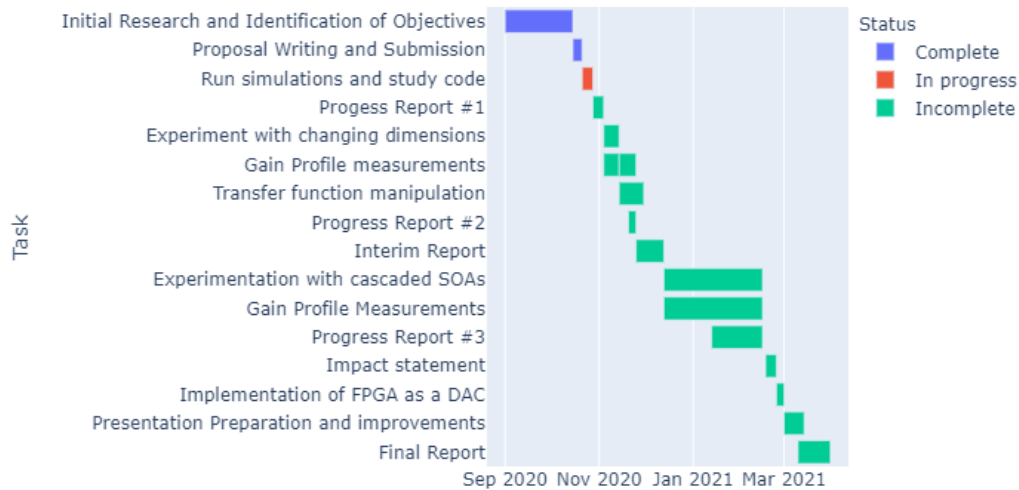


Figure 1: Re-configurable Gantt Chart

### 3 Preliminary Assessment of Risks

**Safety Risks:** The obvious safety risk associated with the current circumstances will be to access the FPGA unit. I am unsure whether it can be used remotely to get the input electronic drive signal or a different method will need to be used. Other than that I believe that the safety risks will be very limited as most of the project was done in simulations in the first place. Therefore, there will be no loss in the accuracy of the procedure while also not being exposed to any safety hazards.

**Failure Risks:** The failure risks would be mostly associate to the variety of aspects that this project involves. This is a multi-fledged project with many interconnected parts so I am uncertain on how changing one variable or characteristic of the system will affect the rest. Also I am not very experienced with the experimental use of

SOAs, therefore performing simulations to determine performance may prohibit me from identifying some of the limitations of the physical performance. The most intimidating part of the project, however, is working on code that has been written and is currently being modified by members of the research group. I will have to communicate with the initial members to understand the fundamentals of the algorithm and how it was structured so I know exactly what effect my changes will have. My lack of access to computational power other than my personal computer may prove stifling at times as performing an experiment with many iterations and different transfer functions is very time-consuming and may limit the amount of testing I can perform. My focus could also be shifted towards hardware limitations in real life experimentation but the current situation may not provide the ideal circumstances for this kind of project to be pursued.

## References

- [1] C. W. F. Parsonson, Z. Shabka, W. K. Chlupka, B. Goh, and G. Zervas, "Optimal control of soas with artificial intelligence for sub-nanosecond optical switching," *Journal of Lightwave Technology*, vol. 38, pp. 5563–5573, Oct 2020.
- [2] C. W. F. Parsonson, Z. Shabka, W. K. Chlupka, B. Goh, and G. Zervas, "Optimal Control of SOAs With Artificial Intelligence for Sub-Nanosecond Optical Switching," *Journal of Lightwave Technology*, vol. 38, no. 20, pp. 5563–5573, 2020.
- [3] T. Sutili, P. Rocha, C. M. Gallep, and E. Conforti, "Energy efficient switching technique for high-speed electro-optical semiconductor optical amplifiers," *Journal of Lightwave Technology*, vol. 37, no. 24, pp. 6015–6024, 2019.
- [4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948 vol.4, 1995.
- [5] T. Berceci and E. Udvary, "Transmission challenges of cascaded semiconductor optical amplifiers," vol. 2005, pp. 129 – 132, 11 2005.
- [6] J. Wei, A. Hamié, R. Giddings, and J. Tang, "Semiconductor optical amplifier-enabled intensity modulation of adaptively modulated optical ofdm signals in smf-based imdd systems," *IEEE/OSA Journal of Lightwave Technology - J LIGHTWAVE TECHNOLOGY*, vol. 27, pp. 3678–3688, 08 2009.
- [7] P. T. Inc., "Collaborative data science," 2015.