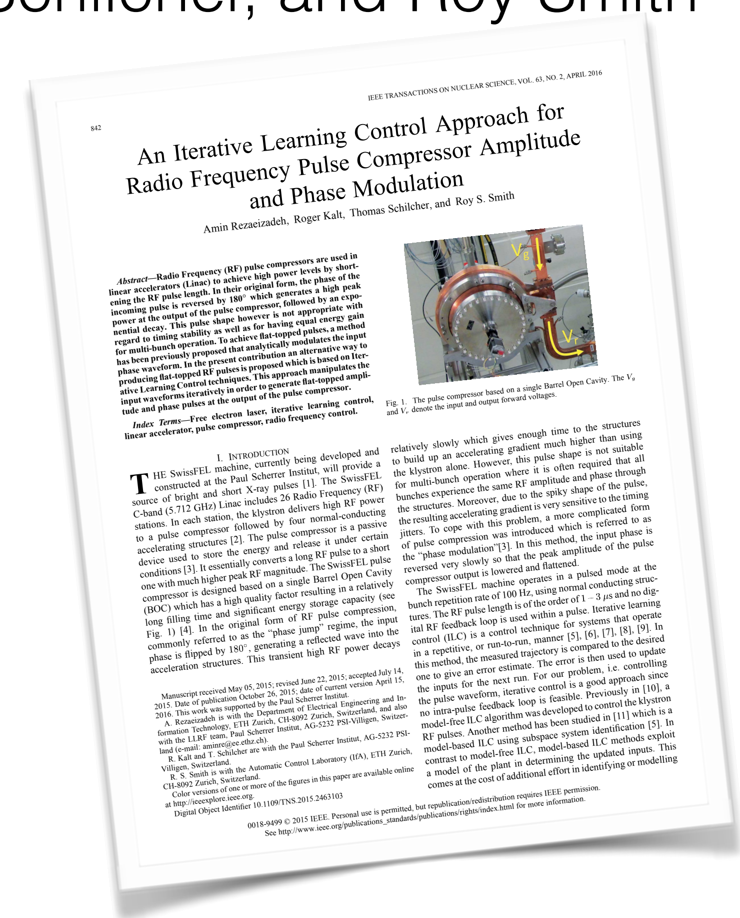
An Iterative Learning Control Approach for Radio Frequency Pulse Compressor Amplitude and Phase Modulation

Paper by Amin Rezaeizadeh, Roger Kalt, Thomas Schilcher, and Roy Smith
Discussion by Matt Gibbs



An Iterative Learning Control Approach for Radio Frequency Pulse Compressor Amplitude and Phase Modulation

Amin Rezaeizadeh, Roger Kalt, Thomas Schilcher, and Roy S. Smith

***Abstract*—Radio Frequency (RF) pulse compressors are used in linear accelerators (Linac) to achieve high power levels by shortening the RF pulse length. In their original form, the phase of the incoming pulse is reversed by 180° which generates a high peak power at the output of the pulse compressor, followed by an exponential decay. This pulse shape however is not appropriate with regard to timing stability as well as for having equal energy gain for multi-bunch operation. To achieve flat-topped pulses, a method has been previously proposed that analytically modulates the input phase waveform. In the present contribution an alternative way to producing flat-topped RF pulses is proposed which is based on Iterative Learning Control techniques. This approach manipulates the input waveforms iteratively in order to generate flat-topped amplitude and phase pulses at the output of the pulse compressor.**

***Index Terms*—Free electron laser, iterative learning control, linear accelerator, pulse compressor, radio frequency control.**

Fig. 1. The pulse compressor based on a single Barrel Open Cavity. The V_g and V_r denote the input and output forward voltages.

I. Introduction

THE SwissFEL machine, currently being developed and constructed at the Paul Scherrer Institut, will provide a source of bright and short X-ray pulses [1]. The SwissFEL C-band (5.712 GHz) Linac includes 26 Radio Frequency (RF) stations. In each station, the klystron delivers high RF power to a pulse compressor followed by four normal-conducting accelerating structures [2]. The pulse compressor is a passive device used to store the energy and release it under certain conditions [3]. It essentially converts a long RF pulse to a short one with much higher peak RF magnitude. The SwissFEL pulse compressor is designed based on a single Barrel Open Cavity (BOC) which has a high quality factor resulting in a relatively long filling time and significant energy storage capacity (see Fig. 1) [4]. In the original form of RF pulse compression, commonly referred to as the "phase jump" regime, the input phase is flipped by 180°, generating a reflected wave into the acceleration structures. This transient high RF power decays relatively slowly which gives enough time to the structures to build up an accelerating gradient much higher than using the klystron alone. However, this pulse shape is not suitable for multi-bunch operation where it is often required that all bunches experience the same RF amplitude and phase through the structures. Moreover, due to the spiky shape of the pulse, the resulting accelerating gradient is very sensitive to the timing jitters. To cope with this problem, a more complicated form of pulse compression was introduced which is referred to as the "phase modulation"[3]. In this method, the input phase is reversed very slowly so that the peak amplitude of the pulse compressor output is lowered and flattened.

The SwissFEL machine operates in a pulsed mode at the bunch repetition rate of 100 Hz, using normal conducting structures. The RF pulse length is of the order of $1-3\,\mu s$ and no digital RF feedback loop is used within a pulse. Iterative learning control (ILC) is a control technique for systems that operate in a repetitive, or run-to-run, manner [5], [6], [7], [8], [9]. In this method, the measured trajectory is compared to the desired one to give an error estimate. The error is then used to update the inputs for the next run. For our problem, i.e. controlling the pulse waveform, iterative control is a good approach since no intra-pulse feedback loop is feasible. Previously in [10], a model-free ILC algorithm was developed to control the klystron RF pulses. Another method has been studied in [11] which is a model-based ILC using subspace system identification [5]. In contrast to model-free ILC, model-based ILC methods exploit a model of the plant in determining the updated inputs. This comes at the cost of additional effort in identifying or modelling

Manuscript received May 05, 2015; revised June 22, 2015; accepted July 14, 2015. Date of publication October 26, 2015; date of current version April 15, 2016. This work was supported by the Paul Scherrer Institut.
A. Rezaeizadeh is with the Department of Electrical Engineering and Information Technology, ETH Zurich, CH-8092 Zurich, Switzerland, and also with the LLRF team, Paul Scherrer Institut, AG-5232 PSI-Villigen, Switzerland (e-mail: aminre@ee.ethz.ch).
R. Kalt and T. Schilcher are with the Paul Scherrer Institut, AG-5232 PSI-Villigen, Switzerland.
R. S. Smith is with the Automatic Control Laboratory (IfA), ETH Zurich, CH-8092 Zurich, Switzerland.
Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.
Digital Object Identifier 10.1109/TNS.2015.2463103

Paper available at: http://control.ee.ethz.ch/index.cgi?page=publications&action=details&id=5288

Pulse compression to achieve higher accelerating gradients

- AKA “SLED” - SLAC Linac Energy Doubler
- Idea: Take the relatively long RF pulse generated by a klystron (around 3 μs), and compress it to achieve a higher amplitude pulse over a shorter time (around 0.7 μs).
- How? Feed the klystron pulse into an RF cavity, then find some way to extract the energy from the cavity quickly, and deliver it to the beam.

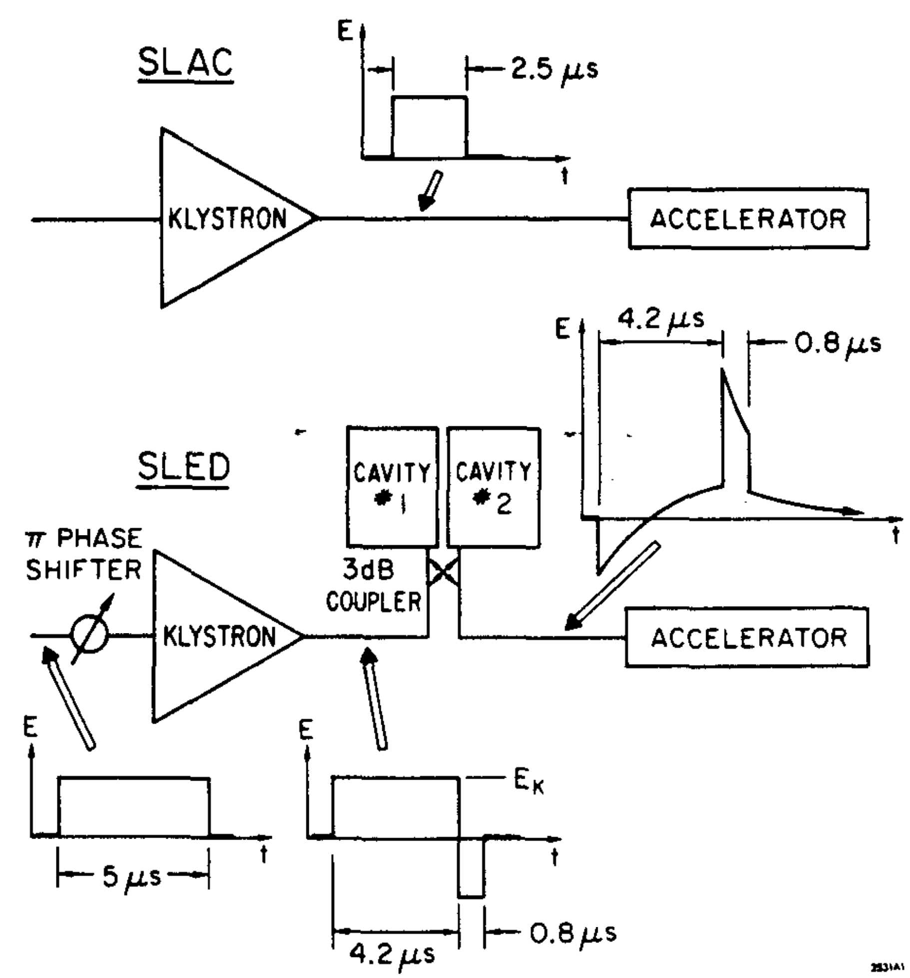


FIG. 2--A comparison of the present SLAC and SLED RF systems.

SLED: A Method of Doubling SLAC’s Energy,
Z. D. Farkas, H. A. Hogg, G. A. Loew, P. B. Wilson.
1974

Great, but…

- Output amplitude has a steep slope.

- What if you want to accelerate two bunches with ~100 ns separation? They will end up with significantly different energies, which might not be what you want. For example, in an FEL, wavelength is proportional to electron bunch energy, so it must be set precisely to meet user requirements.

- Bunches with energy error aren't matched to the linac lattice, causing emittance blow-up.

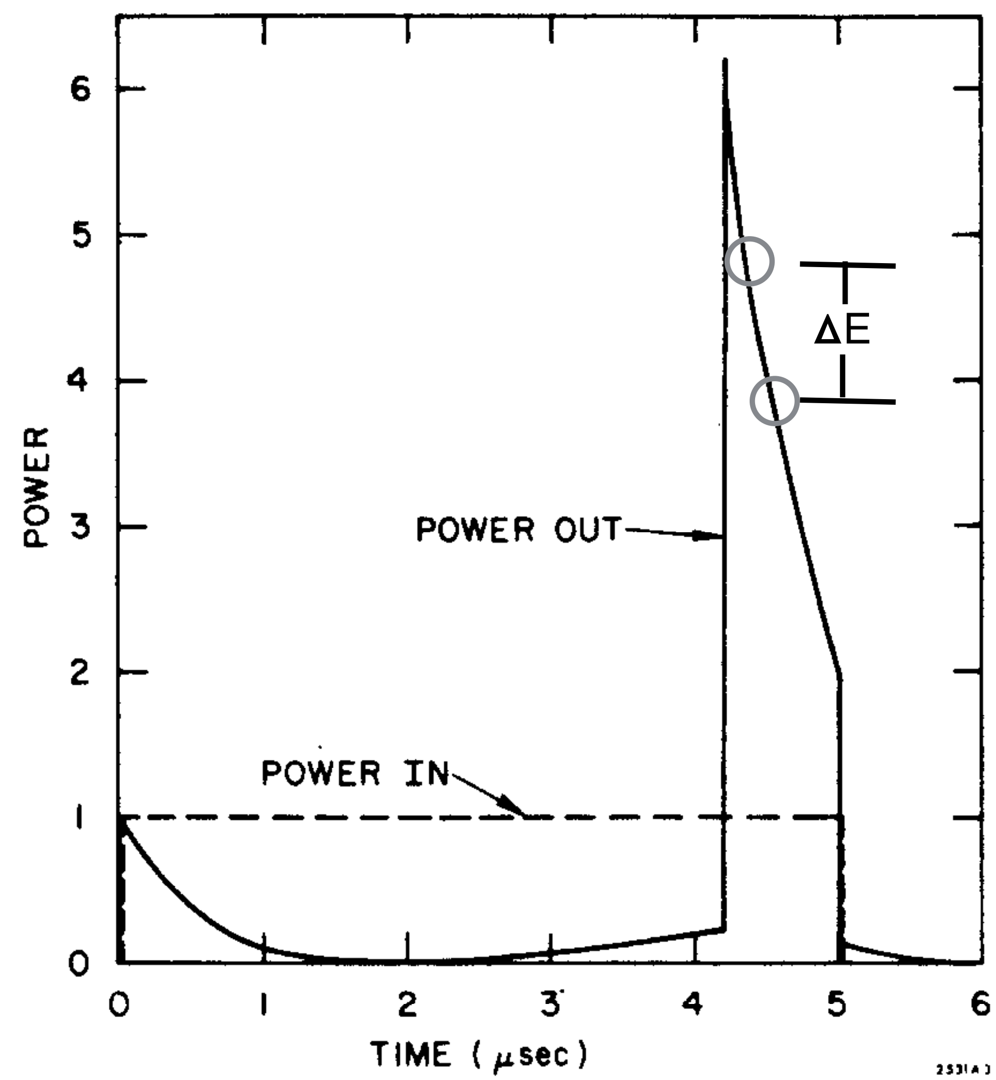


FIG. 7--Input power to and output power from the SLED microwave network as a function of time.

Controlling the output pulse shape by modulating the input to the klystron

SwissFEL's solution: build a feed-forward system that measures the output from the cavity, digitizes I and Q waveforms, then determines a correction to apply to the klystron drive pulse.

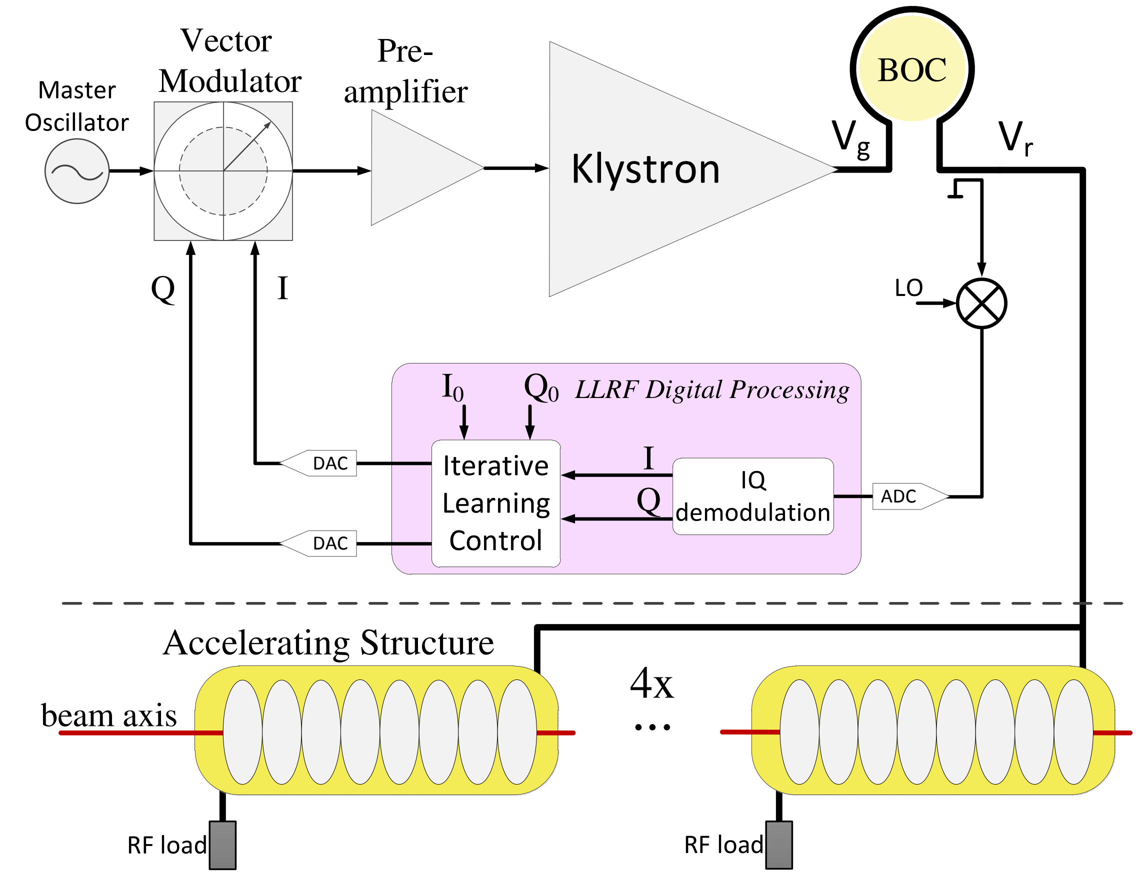


Figure 1: The simplified RF layout of a C-band RF station in the SwissFEL beamline.

Hardware setup

- Looks pretty standard: convert the cavity output to IF, digitize, and demodulate to an I and Q representation digitally.
- An “Iterative Learning Control” system calculates a function by comparing the measured I and Q to reference I and Q waveforms (they just want to make phase and amplitude constant over the output pulse duration, but could probably design a custom shape too)
- Transfer function for RF amplifier chain + cavity is used to calculate new I and Q waveforms for klystron drive pulse.

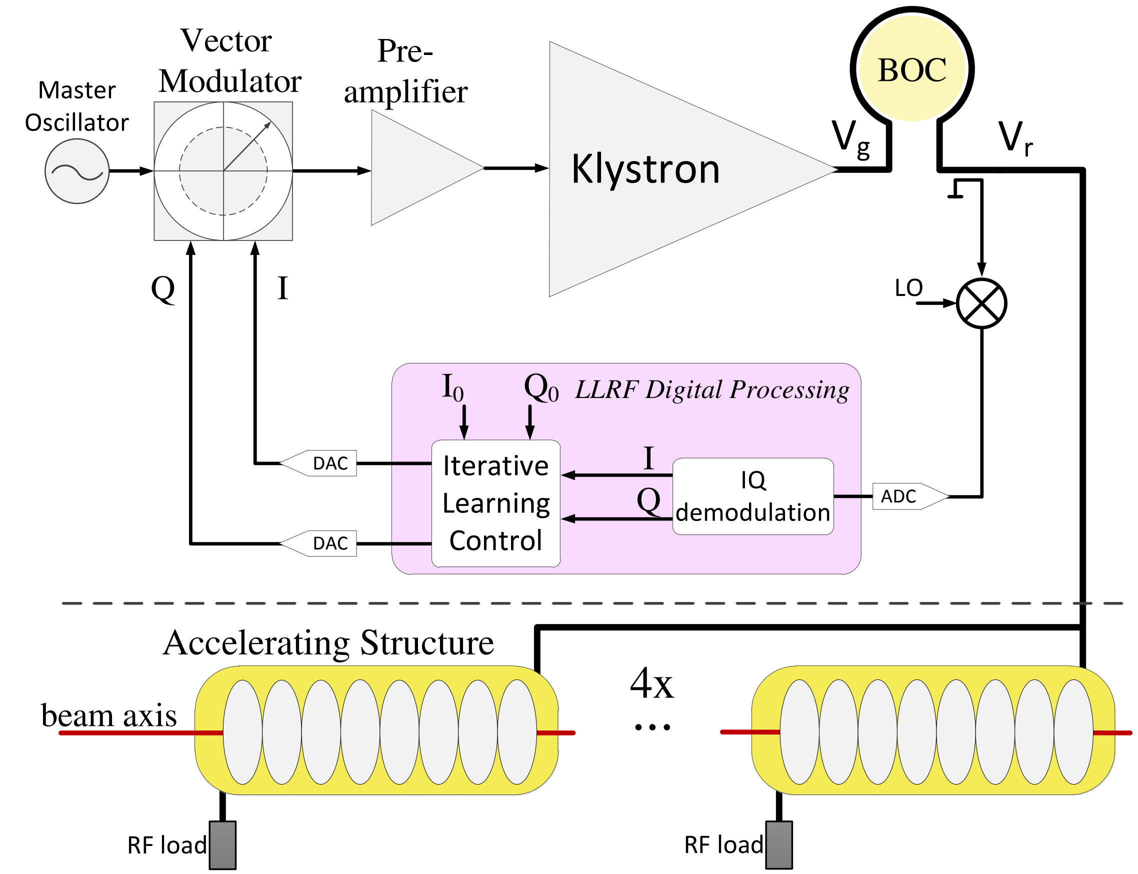


Figure 1: The simplified RF layout of a C-band RF station in the SwissFEL beamline.

Finding the transfer function for the system

Relationship between klystron output voltage and compressor cavity voltage is:

$$\alpha V_g = V_c(1 + j\tau\Delta\omega) + \tau\dot{V}_c,$$

Where α is related to the cavity coupling coefficient, τ is the filling time of the pulse compressor.

The reflected wave from the pulse compressor is given by:

$$V_r = V_c - V_g$$

Discretize the first equation and do the Z-transform (sort of a discrete Fourier transform), and you can find the transfer function for the cavity:

$$G_{BOC}(z) = \frac{V_r(z)}{V_g(z)} = \frac{T_s(\alpha - 1) - \tau - jT_s\tau\Delta\omega + \tau z^{-1}}{T_s + \tau + jT_s\tau\Delta\omega - \tau z^{-1}}$$

RF drive (modulator, pre-amplifier, and klystron) are modeled as a 1st-order low pass system. So, overall transfer function from DAC inputs to cavity output is:

$$G(z) = K\frac{1-\gamma}{1-\gamma z^{-1}}G_{BOC}(z).$$

I and Q representation

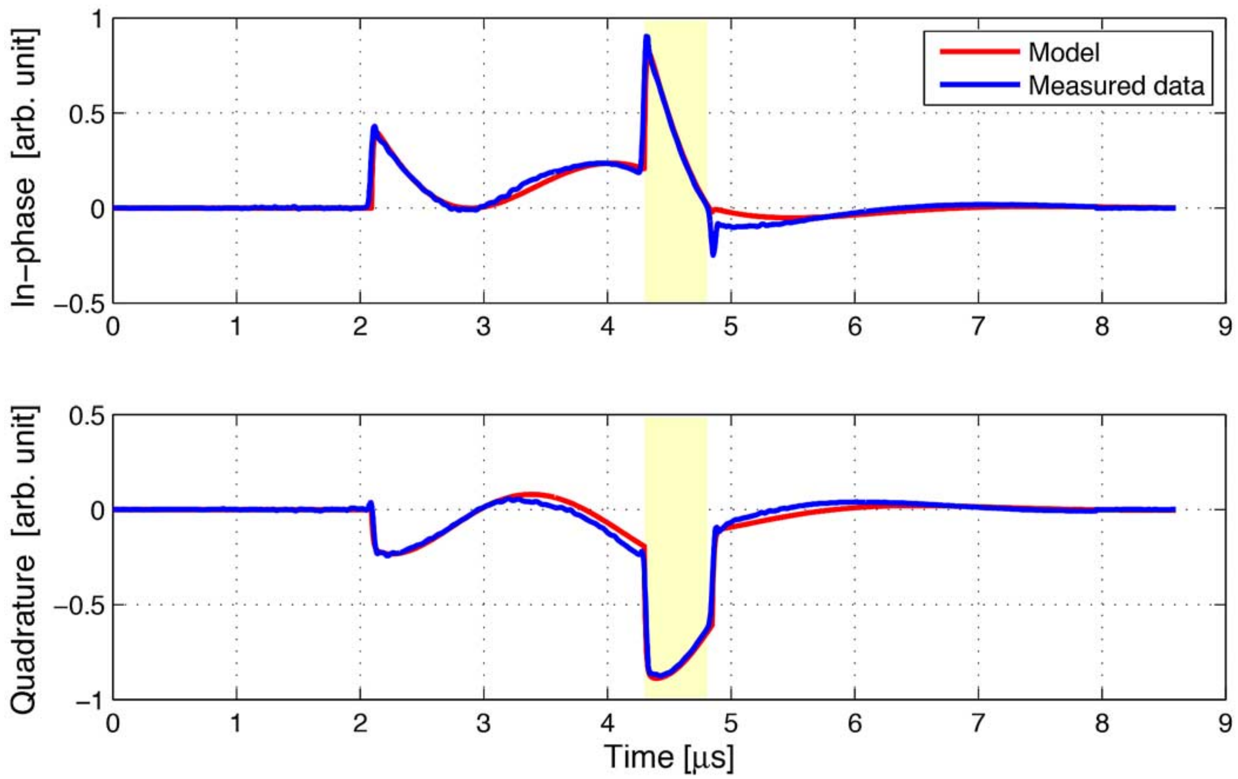
Using "lifted system representation" the equations can be written in an I and Q representation:

$$y_I + jy_Q = G_{IQ}\left(u_I + ju_Q\right)$$

where G_{IQ} is a NxN "lower-triangular Toeplitz matrix of the impulse response h(k)" derived from the transfer function.

G_{IQ} is then split into real and imaginary parts G_r and G_i such that $G_{IQ} = G_r + jG_i$:

$$y_I = G_r u_I - G_i u_Q$$
$$y_Q = G_i u_I + G_r u_Q.$$



Comparison of the model's predicted output vs. measured I and Q waveforms.

Cost Function

Stacking the I and Q terms, you get the following expression for the relationship between input and output signals:

$$y = Gu + d, \quad y, u \in \mathbb{R}^{2N}, \quad G \in \mathbb{R}^{2N \times 2N}$$

Where d is the output disturbance, which captures the uncertainty about the system. The algorithm will make several iterations. Measured output at an iteration 'i' is:

$$y_i = Gu_i + d$$

The optimization algorithm calculates the input for the next iteration (u_{i+1}) as the solution of an optimization problem that minimizes the following cost function:

$$J_{i+1}(u_{i+1}) = \|y_d - y_{i+1}\|_X^2 + \|u_{i+1} - u_i\|_R^2$$

X and R are positive NxN diagonal matrices. How are they determined?

Calculating the Optimal Input

The disturbance term can be estimated from the current iteration:

$$d \simeq y_i - Gu_i$$

Which you can plug into the cost function to give:

$$\begin{aligned}\tilde{J}_{i+1}(u_{i+1}) = u_{i+1}^T \left(R + G^T X G\right) u_{i+1} \\ - 2\left(u_i^T R + \left(y_d - y_i + Gu_i\right)^T X G\right) u_{i+1}\end{aligned}$$

From which the optimal input for the next iteration can be calculated:

$$u_{i+1} = u_i + \left(R + G^T X G\right)^{-1} G^T X \left(y_d - y_i\right), \forall i \geq 0$$

Finally, The Algorithm

1: **Initialize** Phase jump regime: constant amplitude, phase jump of 180°

2: **Do**

3: Measure the output I and Q waveforms

4: Compare to the reference trajectories, $e_i = y_d - y_i$.

5: Update the input waveforms

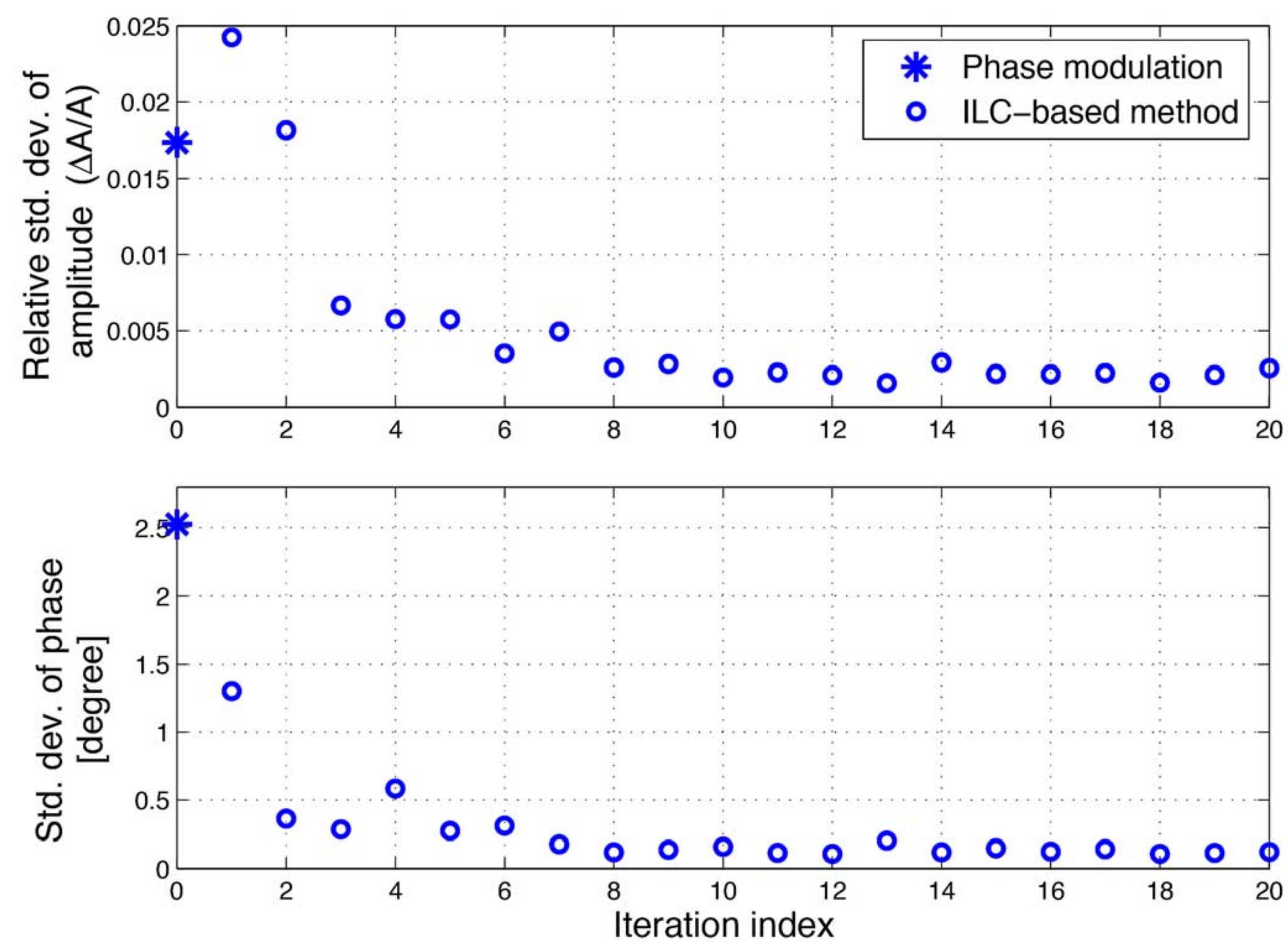
$$u_{i+1} = u_i + (R + G^T X G)^{-1} G^T X e_i \forall i \geq 0.$$

6: Check the limits $u_{\text{low}} \leq u_{i+1} \leq u_{\text{up}}$

7: **If** Convergence achieved

8: **Stop**

9: **Repeat**

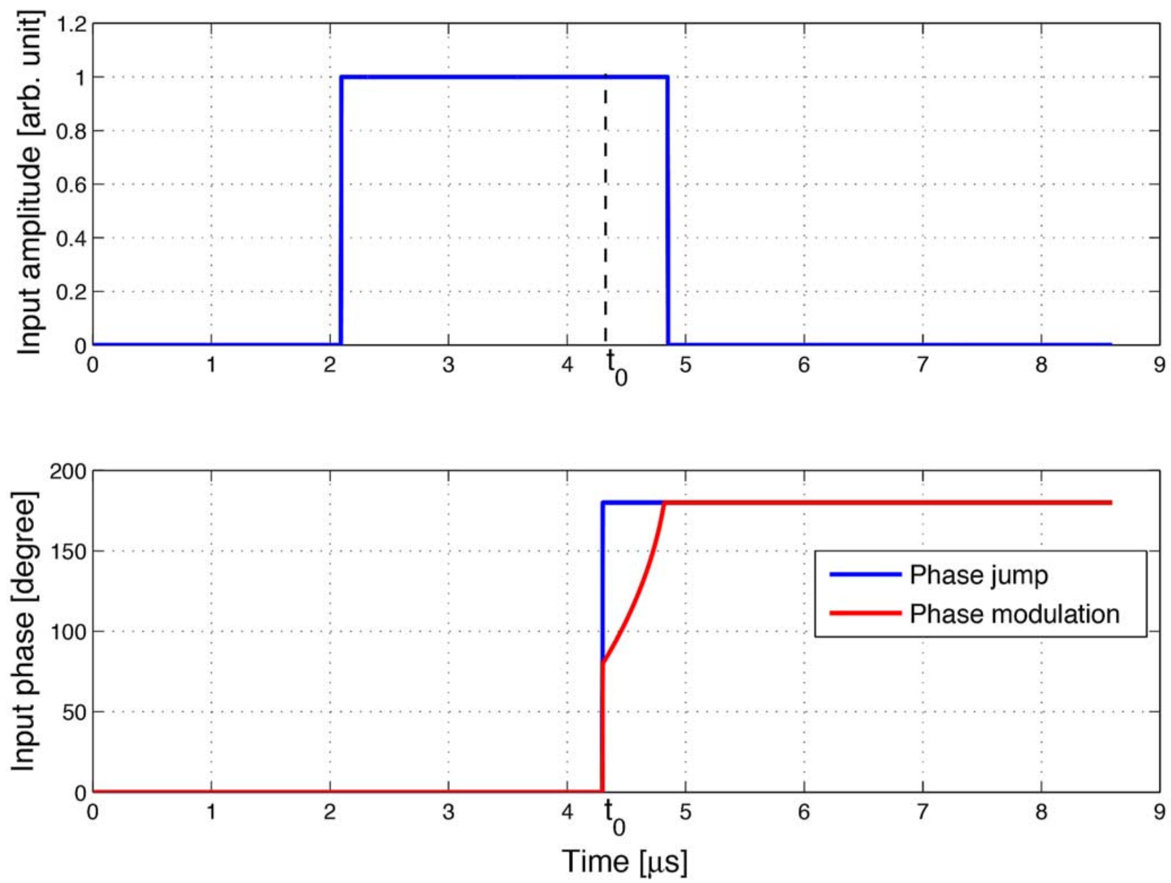


For each iteration, I and Q is measured over ten beam pulses, to suppress noise. After the input waveform is updated, the algorithm pauses for a while to let the cavity temperature re-equilibrate.

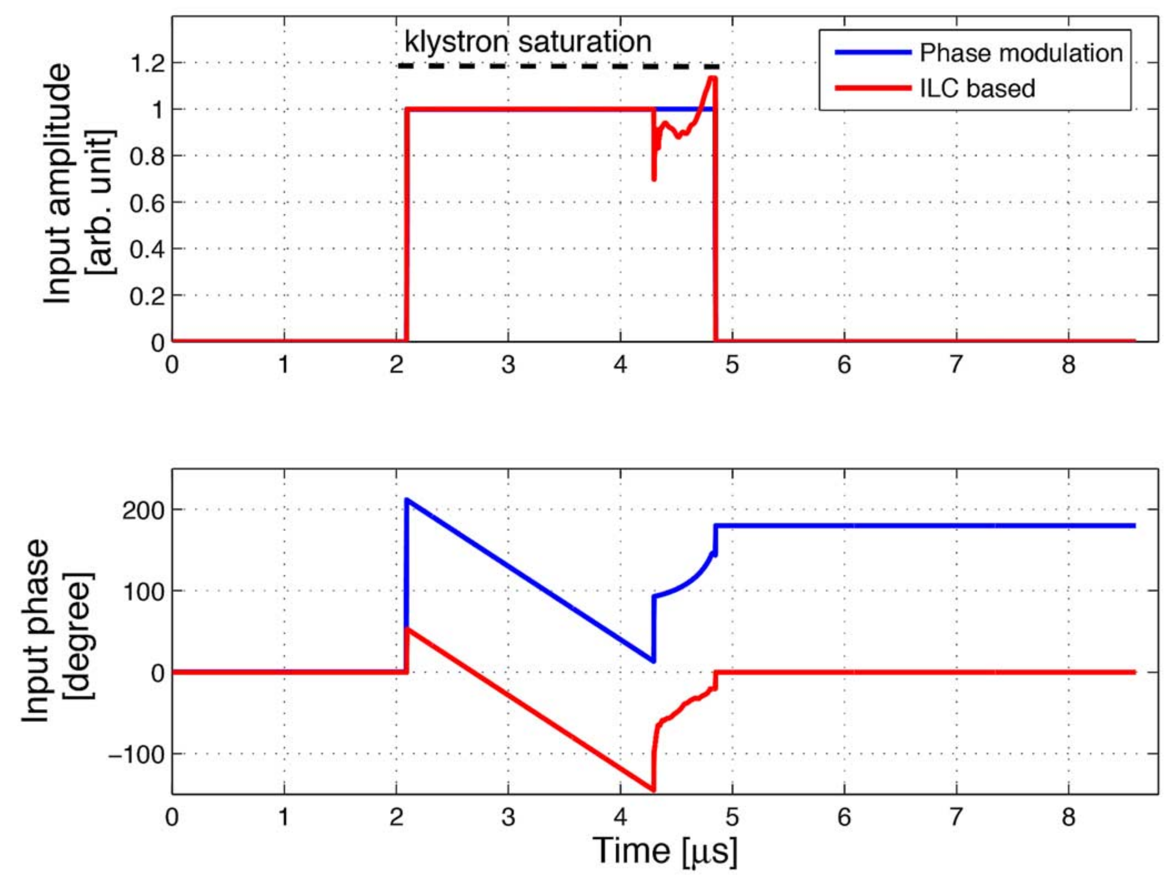
Klystron Input Waveform Before and After Optimization

'Phase jump' waveforms are used as a starting point for the algorithm.

Before



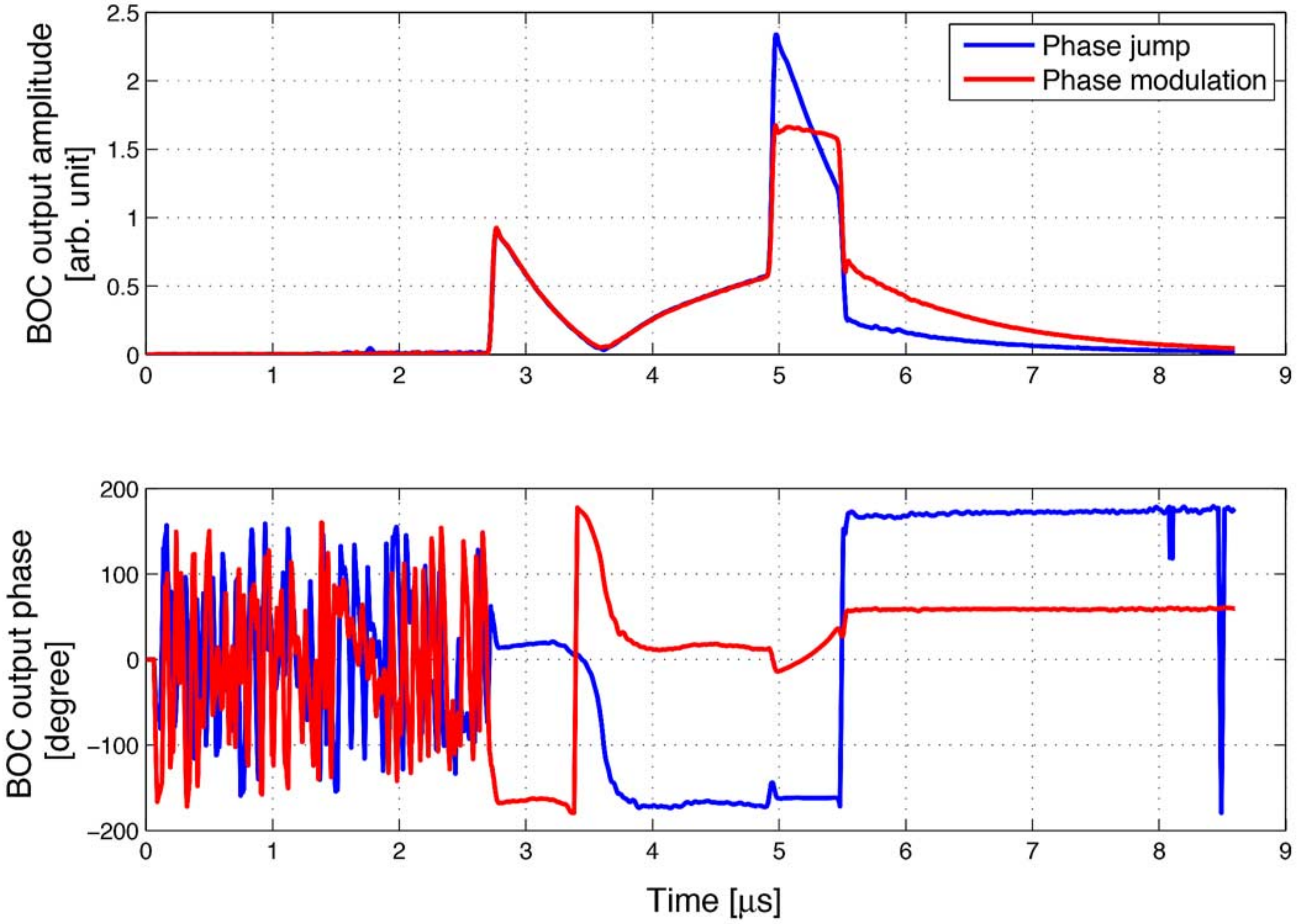
After



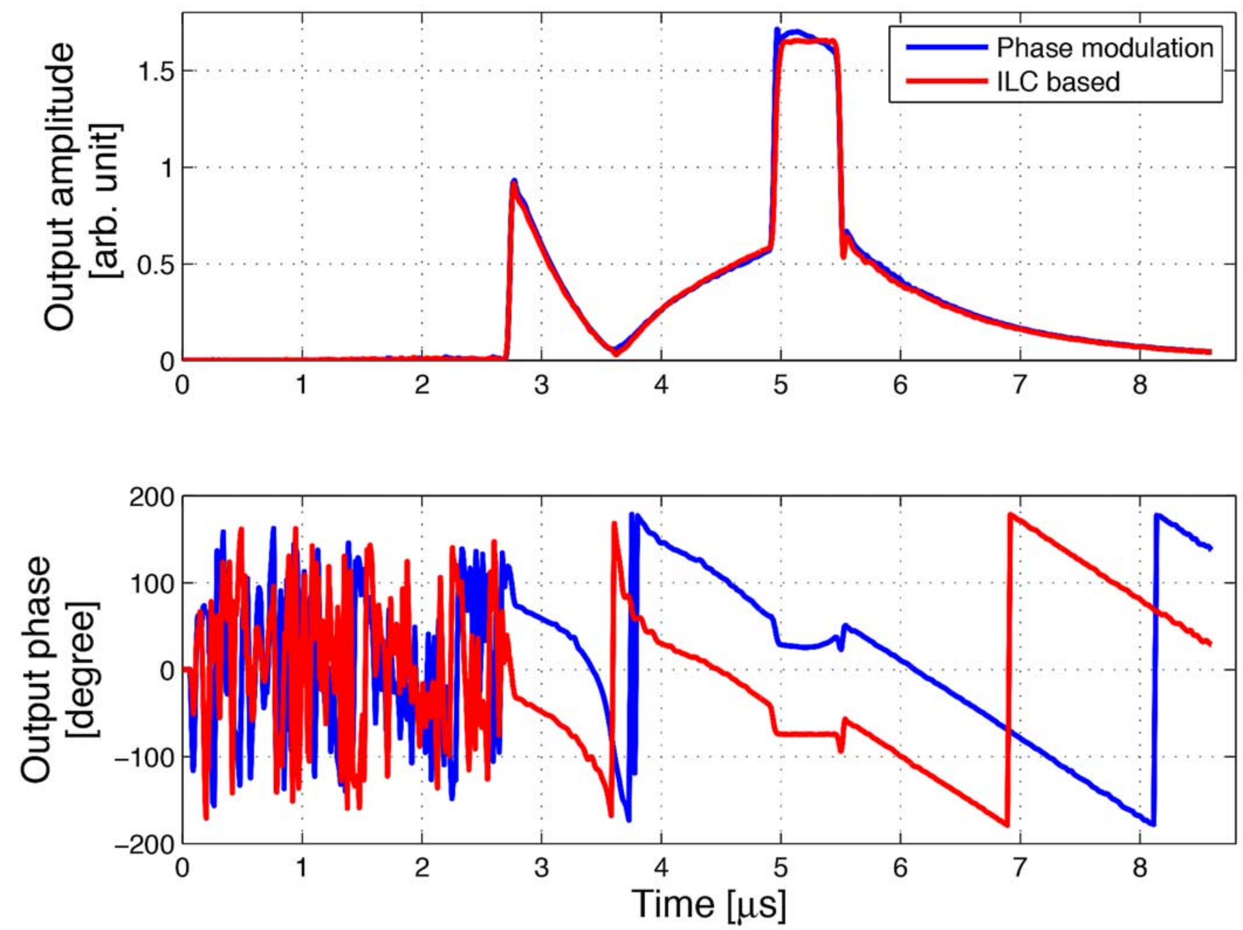
(Phase modulation is an alternative scheme mentioned in the paper, where the phase waveform is analytically determined in advance, and programmed into the modulator. It is shown for comparison to the ILC approach.)

Cavity Output Waveforms After Optimization

Before



After



(Compare 'phase jump' before to 'ILC based' after)

Remarks and Questions

- Authors report losing about 20% of the potential energy gain by using this flattening scheme, which is a pretty steep cost.
- "10 output waveforms are captured and filtered to suppress both random noise as well as the repetitive non-IQ demodulation patterns." What are the details of this filtering? Just averaging?
- Not much discussion of the stability of this system, besides "the delay must be precisely measured, otherwise the algorithm may run into instability". With fancy digital control, how easy is it to determine the stability conditions? Is it even possible?
- The algorithm seems to be something that runs until it converges, and then it stops. How often does it need to run? Will drifting environmental conditions for klystron and cavity invalidate the learned pulse shape?

Extensions

- They use reference I and Q waveforms which give a flat phase and amplitude. How well will this scheme work for arbitrary phase and amplitude waveforms? Could you create custom waveforms with finely-tuned bunch energy and/or phase differences? (You'd still have some issues with bunch energies not being matched to the lattice, so there are some external limits to how far you can go.)