Improving the Electronically Variable Interactive Lockbox (EVIL)

Semester thesis

Martin Stadler

Vlad Negnevitsky
Supervisor

Prof. Jonathan Home
Trapped Ion Quantum Information Group
ETH Zürich

November 1, 2017

Contents

1 Introduction

For trapped ion experiments pulsed, highly stable, coherent light beams are crucial as atomic electron transitions need to be excited as accurately as possible. In this context, feedback loops are absolutely necessary to stay within error tolerances. An error signal of some sort (e.g. based on frequency or intensity error) is extracted and processed by the controller. The output is a correction signal, which is fed back into the system in order to reduce the error. The most commonly used controller type is the Proportional-Integral-Derivative (PID) controller.

An EVIL can process two analogue signals digitally on a Xilinx Spartan-3E FPGA with a clock speed of 96 MHz. The input analogue-to-digital converters (ADCs) are 10 bits wide and work at the same speed as the FPGA. Each EVIL has one fast channel with a 14 bit DAC also working at 96 MHz and a slow channel with a 12 bit DAC at 5 MHz. Both channels are processed on the same FPGA and therefore can be coupled. EVILs are connected to a Raspberry Pi server via USB that connects to a client software that allows convenient configuration of numerous EVILs over the network at the same time. An EVIL can be used to implement a digital PID controller which stabilizes an analogue feedback loop.

There are multiple applications for PID loops in the laboratory. For example, the EVILs are used in a Pound-Drever-Hall(PDH)-setup to stabilize the laser sources used to address trapped ions. The error signal from the PDH lock is fed back through a PID controller to the frequency generator of an acousto-optic-modulator (AOM) or to a piezoelectric crystal of a cavity. Additionally, EVILs are used for beatnote stabilisation and intensity stabilisation. In another configuration one error signal is used for both channels of the EVIL and adjusts the frequency of the AOM and the laser current at the same time.

In the lab, there currently are around 30 PID loops in use. The exact number depends on which experiments are running.

With the goal of improving the stabilizing experience of the EVILs a differential term with median filtering was added to the existing PI controller. Usually low pass filters are used in this context. The median filter was chosen as a novel approach to introduce non-linearity to the otherwise linear channel. Its main advantage is the ability to completely block very short (few cycles) errors regardless of their amplitudes.

In an effort to meet timing constraints, the already existing PI-module and other modules of the firmware were optimized. Icarus Verilog was used to verify modules and Xilinx ISE to synthesize the Verilog code.

2 Theory

PDH locking is a powerful scheme to lock the frequency of a laser source to a reference cavity. The electric error signal generated from the photo diodes detecting the reflected light from the cavity can be used as an error-signal which, properly processed, controls the frequency of AOMs to lock the laser frequency. For an explanation of the PDH scheme have a look at [1]. As long as the error is small, the error signal is linearly dependent on the frequency error of the laser compared to the reference cavity. In this regime, a PID controller can be used. When disturbance causes the frequency error to leave this regime before the PID controller can counteract the disturbance the system will go out of lock.

2.1 Control theory

Since functionality of a PID controller can be fully described within the framework of control theory, a quick introduction will be given. All signal processing is done digitally. Therefore, only discrete methods will be described. The following is just a wrap up of undergraduate electrical engineering lectures. Scripts can be found at [2] and [3].

A PID controller can easily be described in the time domain because all parts can be implemented separately and added together in the end.

$$y_k = K_i T_s \sum_{i=0}^{k} u_i + K_p u_k + K_d \frac{u_k - u_{k-1}}{T_s} \tag{1}$$

Here y_k is the controller output and u_k the error signal. The index k specifies a time slice where the difference between slices is T_s. The constants K allow individual tuning of the three terms.

However, there are more efficient ways for this task. They can be found by transforming to frequency space. In the continuous case the Laplace transform is used to go from the time domain to the frequency domain. Its discrete analogue is the z-transform:

$$\mathfrak{Z}\{(f_k)\} = \tilde{f}(z) = \sum_{k=0}^{\infty} f_k z^{-k}$$

Where f_k is an infinite sequence e.g. the signal in the time domain. It is a linear transformation. To arrive at the necessary transformations for a PID controller the time shift property is needed:

$$z^{-1}\tilde{f}(z) = z^{-1}\sum_{k=0}^{\infty} f_k z^{-k} = \sum_{k=0}^{\infty} f_{k-1} z^{-k}$$

f_k is zero for $k < 0$. This means that multiplication by z^{-1} represents one cycle of delay of the signal.

A useful way to describe linear signal processing are transfer functions. They describe the relationship between input and output signal:

$$G(z) = \frac{\mathfrak{Z}\{(y_k)\}}{\mathfrak{Z}\{(u_k)\}} = \frac{\tilde{y}(z)}{\tilde{u}(z)} \tag{2}$$

Now one can z-transform (1) and divide by $\tilde{u}(z)$ to arrive at the transfer function of a PID controller. The time shift property allows intuitive transformation of the derivative term. The proportional term is trivial. The integration term can easily be derived by using recursion:

$$\begin{aligned} y_k &= \sum_{i=0}^{k} u_i \\ y_k &= y_{k-1} + u_k \\ \tilde{y}(z) &= \tilde{y}(z)z^{-1} + \tilde{u}(z) \\ \tilde{y}(z) &= \frac{\tilde{u}(z)}{1 - z^{-1}} \end{aligned}$$

This leads to the full transfer function:

$$G(z) = K_p + K_i T_s \frac{1}{1 - z^{-1}} + K_d \frac{1 - z^{-1}}{T_s} \tag{3}$$

The three terms in (3) can be converted to a common denominator. This yields:

$$G(z) = \frac{(K_p + K_i T_s + \frac{K_d}{T_s}) + (-K_p - \frac{2K_d}{T_s})z^{-1} + \frac{K_d}{T_s}z^{-2}}{1 - z^{-1}} \tag{4}$$

(4) can be translated to one accumulator where the error signal itself, its first and its second derivative are added every cycle. Additional delay can be modelled by multiplying $G(z)$ by z^{-l} where l is the number of delay cycles. This formalism allows discrete frequency response analysis.

Let $u_k = e^{i\omega k T_s}$ be the input signal and $k \gg 1$ i.e. the system is in a steady state. Then the output y_k is

$$y_k \approx G(e^{i\omega T_s}) e^{i\omega k T_s} \tag{5}$$

This means the output is the same as the input weighed with the factor $G(e^{i\omega T_s})$ [3]. It is the discrete frequency response. As an example the response of a PID channel is shown in Fig. 1.

You can clearly see the three different sections in the frequency spectrum caused by different terms dominating the PID transfer function: For low frequencies the integral part with 90° phase lag, for middle frequencies the proportional term with no phase lag and for high frequencies the derivative term with 90° phase lead. The phase lead of the derivative term drops at the high frequency end, meaning it is not effective anymore. Additional delay of the signal leads to an earlier occurrence of this phase drop and thus lowers the maximum frequency in the error signal that can be counteracted by the PID controller.

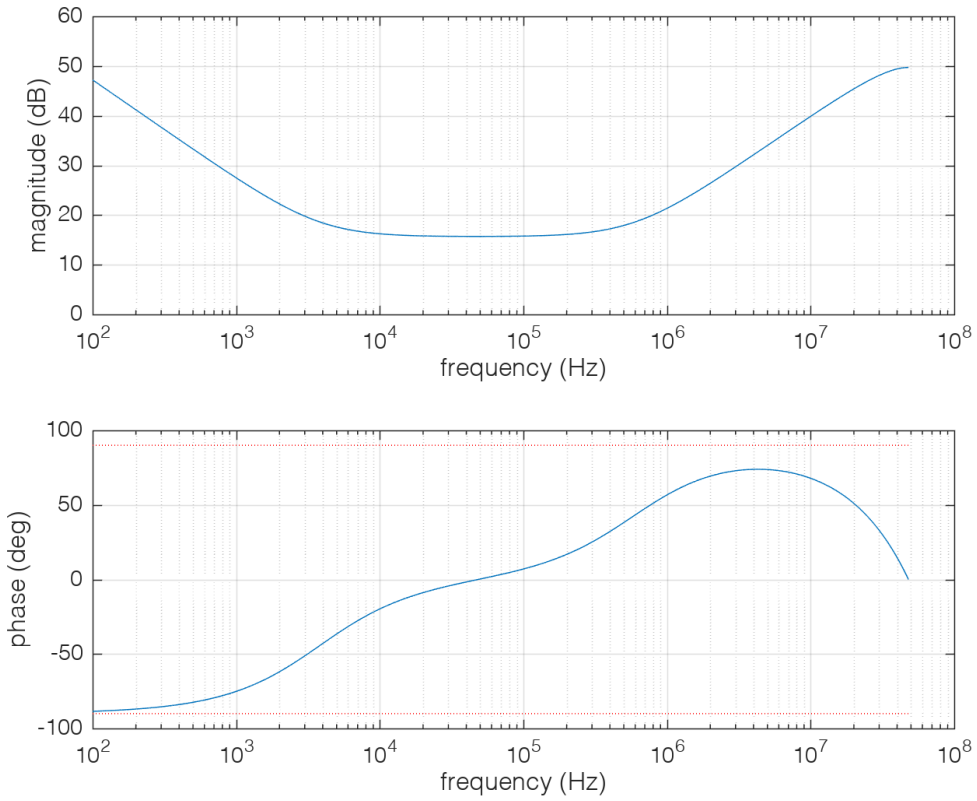

Figure 1: Frequency response of a PID channel for a sample rate of 96 MHz, $p = K_p = 6.1$, $i = K_i T_s = 1.5 \cdot 10^{-3}$ and $d = \dfrac{K_d}{T_s} = 1.5 \cdot 10^2$.

3 Implementation

The EVIL handles two separate PID channels which can be controlled through parameters written into registers via UART. Significant effort has already been put into making streaming data out of the EVIL possible. This makes debugging and calibrating of the PID controls much easier. It was implemented as part of a semester thesis [4].

When it comes to functionality only the PID channel module was modified in this project. Hence, only this module will be described in detail.

3.1 PID channel structure

In Fig. 2 the structure of the `pid_channel` module is displayed. Controls for the submodules are omitted as they don't help to understand the functionality of the channel. The module `ramp_gen` is used to find the linear part of the PDH feedback signal and lock the PID controller to its zero-crossing. It generates a sawtooth signal which can be modified in range, frequency and centre of the interval it sweeps over. The peak detector prevents the system to go out of lock if the PID controller overreacts to strong sudden changes that might be caused by short lived disturbances. In principle the system should return to its previous state on its own after the short disturbance is over. Although the low pass filter does work in principle, it is not used for the laser locks in the lab at the moment. It was implemented as part of a semester thesis as well [5].

Additionally, some of the internal workings of the `digi_pid` module are shown to make clear how the filtered signal from the low pass filter is passed through the `digi_pid` module to the final output `data_o` if a peak is detected. As you can see the register `ramp_centre_r` serves two purposes: Firstly, it initializes the accumulator of the `digi_pid` module (`u_full`). This way the system won't jump out of lock if the user switches from sweeping with the ramp generator to PID control because the ramp centre should be set to the zero-crossing of the linear part of the PDH feedback. The second purpose is to serve as a pass through for the smoothed signal from the peak detector if a peak is detected. At the same time the accumulator is set to a value that's not influenced by the disturbance that caused the peak detector to trigger. Since `ramp_centre_i` is a 16-bit integer it has to be zero padded before it is written to the accumulator (43-bits) which in turn is truncated before it's written to the 16-bit output.

All of this was already in place and running before this semester project began.

3.2 Original PI controller

The original `digi_pid` module, which calculates the correction signal, was already programmed with the implementation of a differential term in mind. So the D coefficient was already connected to the respective slider in the GUI. However, the design of the module had to be significantly changed to meet timing. As you can see in Fig. 3 the polarity of the error signal is changed directly, which takes one cycle combined with the addition of the input offset. In the new design the polarity flip is included in the P, I and D gain, which shifts some computational load from the signal path to less time critical

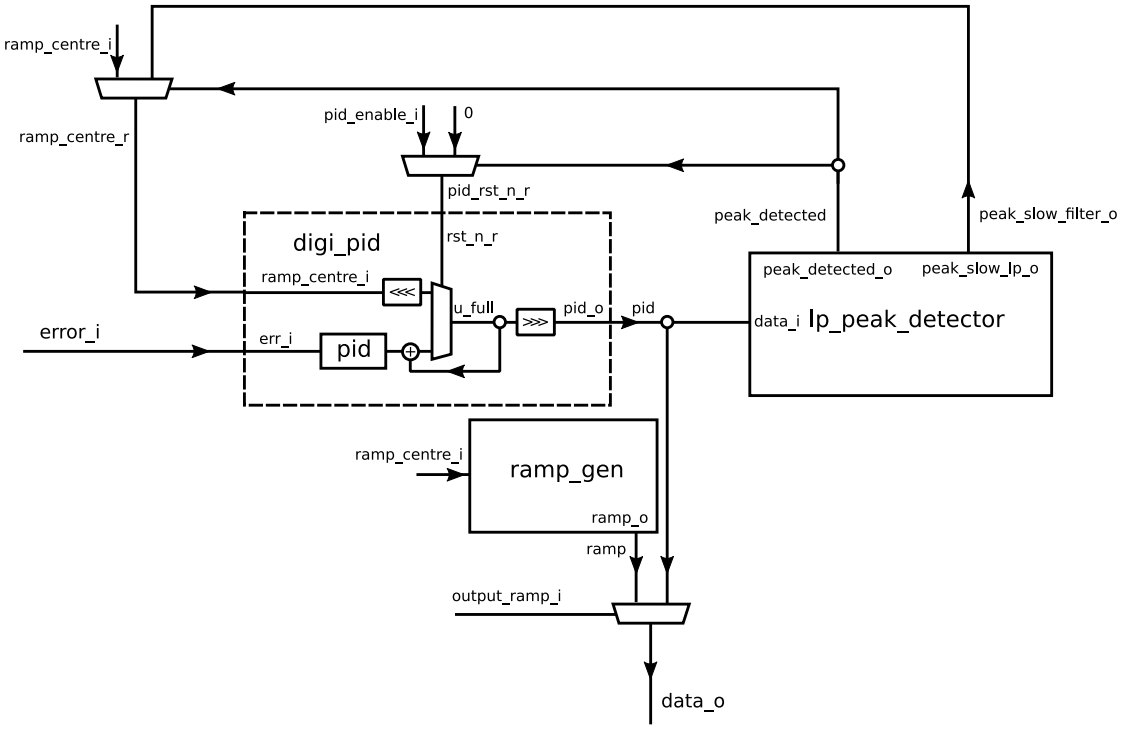

Figure 2: Internal structure of module `pid_channel` leaving out most of the wires and registers used for configuration of PID, low pass filter and ramp module. At multiplexer (trapezoidal) symbols, left or upper input is used if control pin is logical 0. All connections are labelled the same as in the code. Boxes symbolize submodules in the PID channel, where the pins are labelled according to the code as well.

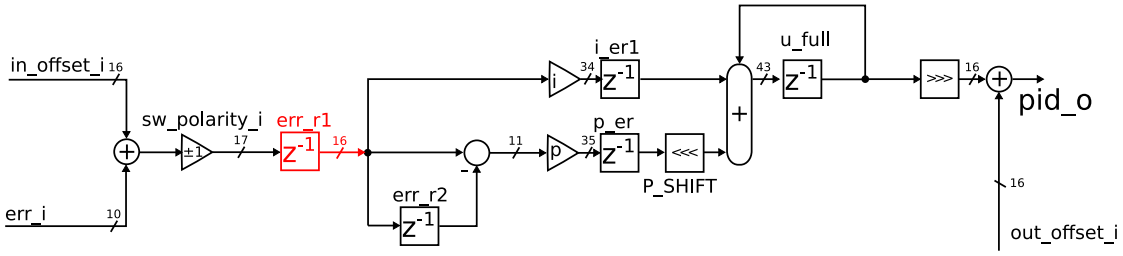

Figure 3: Block diagram showing the transfer function of the original PID controller. All connections are named as in the code. Bit widths of registers are indicated. The red lag term could be removed in the first step by including the sign change from `sw_polarity_i` in p and i.

parts of the module. After all, the hardware multipliers in the Spartan-3E FPGA are 18 by 18 bits wide, so that sign extension of the unsigned 16 bit gains are straightforward. As a side effect, the delay is reduced by one cycle. The final addition depends on the PID reset (see Fig. 2) which means that there is an additional logic level to the already resource intensive addition.

The bit shift of the P term in Fig. 3 compensates the amplification of the I term caused by the short cycles. Since the I term is a discrete-time integration, the accumulative sum of the input signal has to be multiplied by the sample time T_s (see (3)). Using an actual multiplication at this point would be a waste of resources. It is more efficient to use bit shifts to reach the right magnitude. Remaining scaling can be balanced out by the gains. Instead of truncating the I term, the P term is scaled up to keep full precision for the I term. Finally, the accumulator is truncated to 16 bits, which is equivalent to a downscaling of the whole output signal sent to the DAC.

3.3 New PID controller

The straightforward addition of the derivative term led to timing issues. There are a few crucial points in the path which are prone to cause problems with timing. To allow faster reactions one addition and multiplication has to be done in one cycle for the individual terms. In principle the FPGA is able to calculate this operation within the required time. However, the accumulator (`u_full`) at the end has to be updated every cycle as well which involves the addition of P, I and D term. In one cycle 4 registers which are 43 bits (accumulator, I term), 28 bits (P term) and 30 bits (D term) wide are added. Two complex steps in row have to be carried out, which is challenging to synthesize.

While the performance of the P and D term depend on fast signal processing, the I term doesn't suffer from any significant drawbacks because it counteracts the low frequency part of the error signal. Therefore, an additional step was added to the path of the I term which takes the input offset of the P term into account, as you can see in Fig. 4. Theoretically, the input offset could even be added at the same point as the output offset. If the input offset is high, the accumulator would be more likely to overflow in one direction which could destabilize the lock because of the missing overflow protection.

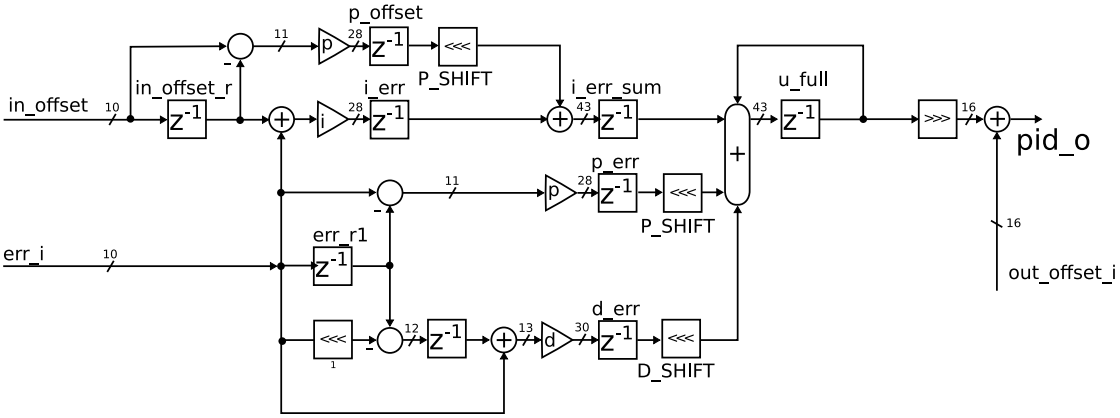

Figure 4: Flow chart showing the transfer function of the new PID

In this case the DC offset in front of the ADC should be adjusted, though. If the input offset is taken into account right at the beginning, the system will leave the linear error regime and go out of lock before any overflow occurs.

When timing criteria were finally met and the new firmware could be tested, a major error in the configuration of the firmware was encountered. When the PID controller was switched on, the accumulator just kept overflowing regardless of the input. This unexpected behaviour could be traced back to unset clock uncertainty in the configuration file of the firmware. When the synthesis tools routes the circuit it has to make sure that the result of an operation is written to the target register before the next flank of the clock signal. If it fails to accomplish this requirement after several attempts it will be denoted as a setup error. Such errors occurred frequently at the multiplication step of one of the terms and their summation with `u_full`. Since clocks are imperfect some tolerance has to be subtracted from the optimal per cycle computation time. This means that the circuit has to be optimized for a shorter cycle. At first the clock uncertainty was set to a high value of 0.5 ns, which led to the desired behaviour of the PID controller. At the same time, the optimization effort had to be increased to meet timing, however. Later on, the uncertainty was lowered to 0.1 ns which didn't cause any unexpected behaviour.

The firmware was tested on the EVIL controlling the 866 nm (fast channel) and 854 nm (slow channel) laser frequencies, stabilizing each to a reference cavity via a PDH lock. After adjusting the scaling of P and D term by changing the size of the bit shift for the respective register, `p_err` and `d_err`, it could be observed that the D term picks up noise and passes it on to the output. The upscaling of the D term of the fast channel had to be set to 22 bits to generate oscillatory behaviour for high D gain. For the slow channel, 20 bits were sufficient. This is due to the much higher output delay of the slow channel caused by the serial interface between the FPGA and the DAC.

As expected, the D term on its own without filter didn't have any advantage over the old system.

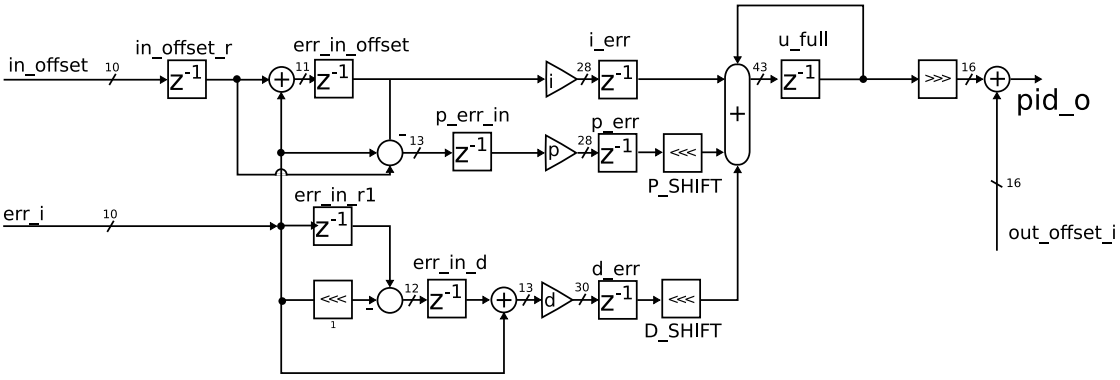

Figure 5: Flow chart showing the transfer function of the PID implementation used together with the median filter

3.4 New PID controller with median filter

In the second step of the project a median filter was implemented to tackle this problem. Usually a simple low pass filter is used to eliminate noise before the error signal is passed on to the PID controller. While a low pass filter is a linear element which can easily be described with tools of control theory, a median filter is non-linear i.e. it doesn't obey the superposition principle.

For complexity reasons a 5 point median filter was chosen. The first approach was to store the order of the last 5 inputs in a list and maintaining the inverse of this list as well. An if tree can be constructed which uses at most three comparisons to add a new element. In the Verilog code, arrays are indexed via entries of the list inverse. The synthesis tool didn't find an efficient way to implement conditional indexing, so timing problems occurred which couldn't be overcome. Additionally, maintaining the lists turned out to be quite cumbersome. A simpler approach led to the desired outcome. For each of the 24 orders of 4 elements an if tree with a maximum depth of 3 was written which determines the rank of a new element. The median is written to the output register and the order of the last 4 elements is updated for the next cycle. The order is stored as an integer between 0 and 23. This leads to a finite state machine dependent on the order which can be efficiently implemented on the FPGA. After routing a copy of the error signal to the median filter module and splitting comparisons and writing to registers between the copies, timing could be met.

The PID module itself had to altered as well. In Fig. 5 you can see that an additional cycle had to be added to the P term. The input offset is included in the P term again which saves one multiplication per channel. The D term remains as direct as possible. On average 3 cycles of delay are added to the whole channel by the median filter.

4 Results and discussion

4.1 Diode laser

The final firmware was flashed on the EVIL controlling the 866 nm (fast channel) and 854 nm (slow channel) laser. After thorough stability testing the new firmware was kept for these locks.

The lock was configured as it would be for actual experiments and data was recorded. In Fig. 6 the output with D term in use is displayed while Fig. 7 shows the same configuration with deactivated D term. Please note that the signal is rounded to 8 bits because of limited streaming capabilities. The received data is just multiplied by 4 to make up for the lost 2 bits. The PID output is multiplied by 4 as well although the output sent to the DACs is 12 or 14 bits wide.

You can see, that the D term introduces random jumps to the PID output which are amplified fluctuations of the error signal. A jump corresponds to a fluctuation of the least significant bit of the error signal. The multiplication by 32133 corresponds to a bitshift of approximately 15 bits. An additional bit shift of 22 (`D_SHIFT`) happens, leading to 37 bits. Only the first 8 bits of the 43 bit accumulator are streamed. Therefore a change of the LSB of the D term leads to a change of the third LSB of the streamed output signal. This change of 8 is displayed as a jump of 32 in the plot because of the multiplication in the streaming interface. In Fig. 7 you see that there are no random jumps in the error signal which is due to the much lower sensitivity of P and I term to noise.

Because of the low transfer speed over the network compared to the internal clock, the plots contain only one sample per 900 cycles. The estimated time difference between transferred samples is 9.5 µs while the cycle length is 10.4 ns. Hence, the plots lack detail and can only be seen as a rough overview. By connecting the EVIL directly to a PC via USB, a gapless measurement can be obtained which allows further analysis of the D term. Since the stability of the PDH locks was already sufficient with the existing PI controller, the usefulness of the D term remains limited in this application.

The 854 nm laser exhibits similar behaviour if the bit shift of the differential term is reduced by 2.

4.2 Frequency doubling lock

For excitation of a raman transition of ^{9}Be+ ions, coherent light with a wavelength of 313 nm is needed [6]]. This is achieved by doubling the frequency of a 626 nm diode laser. A stable lock for this setup is hard to achieve. In this case the two channels of the EVIL are coupled and control the frequency of the AOM with the fast channel and the current of the pumping laser with the slow channel. Only minor changes in the firmware have to be done to make this possible.

After the firmware was adapted to this setup, it was tested on the frequency doubling lock. Additionally to the error signal on the EVIL, the intensity of the output shaped to pulses was monitored on an oscilloscope. In this configuration the limited accuracy and data rate of the stream makes calibration difficult. After a certain point, the frequency of intensity dips seen on the oscilloscope are not reduced anymore although the error signal

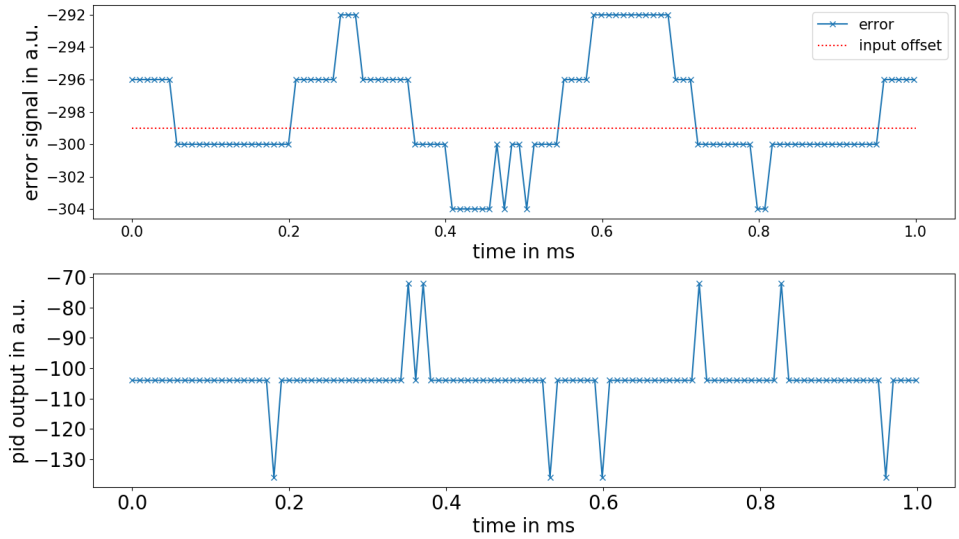

Figure 6: Error signal and pid output stream for active lock of 866 nm laser used for experiments. p = 5919, i = 52005, d = 32133, input offset = -299. Absolute times for upper and lower plot are different.

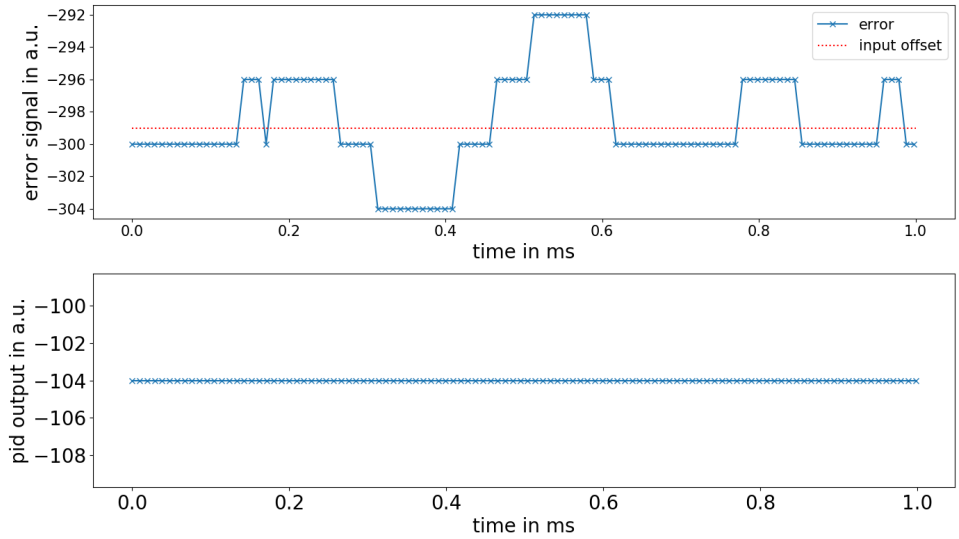

Figure 7: Error signal and pid output stream for active lock of 866 nm laser. p = 5919, i = 52005, d = 0, input offset = -299. Absolute times for upper and lower plot are different.

seemingly improves by increasing the P parameter for the channel controlling the AOM. High D coefficients lead to peaks in the PID output for the slow channel. No changes in the error signal could be observed until too high D gains lead to oscillatory behaviour.

The new firmware was tested merely superficially with this lock. Therefore no profound evaluation can be given. Although the D term seemingly doesn't improve the stability, it certainly adds a degree of freedom for attenuating high-frequency noise present in the error signal.

5 Conclusion

A differential term and 5-point-median filtering was successfully added to the firmware of the EVIL. Functionality was verified using several testbenches for single modules and for the top level module as well. Experimentally, the new firmware could successfully be used to stabilize PDH locks and a raman frequency doubling lock. No detailed measurements of the error signals with and without a D term were conducted. The overview measurements show no significant improvements. Nevertheless, the working PID channel is a useful building block and will certainly be used in future FPGA-based PID loops with a better signal-to-noise ratio than the EVILs.

A low pass filter might work better than the 5-point-median filter to keep the differential term from picking up noise. Alternatively, increasing the number of points the median is taken over could also help. However, this doesn't seem feasible.

References

[1] E. D. Black. *An introduction to Pound–Drever–Hall laser frequency stabilization.* American Journal of Physics **69** (2001) 79.

[2] TUGraz. Control Systems 1, 2017.
https://www.tugraz.at/fileadmin/user_upload/Institute/IRT/Aufgabensammlung/Control_Systems_1/Control_Systems_1.pdf

[3] TUGraz. Control Systems 2, 2017.
https://www.tugraz.at/fileadmin/user_upload/Institute/IRT/Skripten/Control_Systems_2.pdf

[4] C. Fischer. Implementation of a Digital Lock-in Amplifier on a Field Programmable Gate Array and its Remote Control in a Local Area Network, 2015.
https://www.ethz.ch/content/dam/ethz/special-interest/phys/quantum-electronics/tiqi-dam/documents/semester_theses/semesterthesis-christoph_fischer.pdf

[5] A. Hungenberg. Automatic relocking of an FPGA-based PID controller using a bandpass-filtering approach, 2013.
https://www.ethz.ch/content/dam/ethz/special-interest/phys/quantum-electronics/tiqi-dam/documents/semester_theses/semesterthesis-alex_hungenberg.pdf

[6] H.-Y. Lo, J. Alonso, D. Kienzler, B. C. Keitch, L. E. de Clercq, V. Negnevitsky, J. P. Home. *All-solid-state continuous-wave laser systems for ionization, cooling and quantum state manipulation of beryllium ions.* Applied Physics B **114** (2014) 17.