Experimental Nuclear Physics Department

Faculty of Physics and Technology

Karazin Kharkiv National University

Deutsches Elektronen Synchrotron

Photo Injector Test Facility in Zeuthen

Photo Injector Cathode Laser Beam Intensity and Position Monitoring System

Diploma Thesis

by Yevgeniy Ivanisenko

Supervised by Dr. Sergey Korepanov

Zeuthen 2007

Contents

1 Introduction

The radio frequency electron photo injector is a type of injectors, where the formation of the electron bunch is based on the photocathode illumination with short light pulse followed by electrons photoemission and acceleration by a rf-field. For the purpose intensive laser radiation is used.

To use a photoinjector with maximal efficiency one should reduce the cycle dead time, that is defined by the time needed for system components temperature relaxation - quasi-isothermal operation. Limitation factor is the cooling system productivity. To investigate the subject of the optimal operating mode the facility should be equipped with stability monitoring systems of all components.

Investigations of such type are done at the PITZ (Photo Injector Test facility in Zeuthen) in Zeuthen, DESY, Germany. The repetitional rate of the RF-power pulses is 10 Hz, and of laser pulses is 1MHz. The RF-power flat top pulse duration is 0.8 ms (klystron output power 4 MW), that gives opportunity to produce up to 800 bunches. The gun cavities for the FLASH experiment (DESY-Hamburg, free electron laser) are comissioned at the PITZ facility. FLASH operates at RF-power repetitional rate 5 Hz with 300 pulses of 1 MHz laser pulses repetitional rate at the moment.

This thesis desribes the cathode laser intensity and pointing position stability monitoring system for the PITZ facility. The system was designed and produced, and the task of this thesis was setting into the operation, creating software for analysis and developing a tool based

on the system described for the shift crew to monitor the parameters continuously during the facility operation.

The order of sections is following: influence of intensity and pointing position instabilities on the electron beam and the beam diagnostics is discussed; experimental approach and consist of four parts: photoinjector test facility, cathode laser system, photodetectors and the laser beam diagnostics; measurement results, discussion and conclusion.

2 Theory Basis

2.1 Photo Injector Beam Parameters

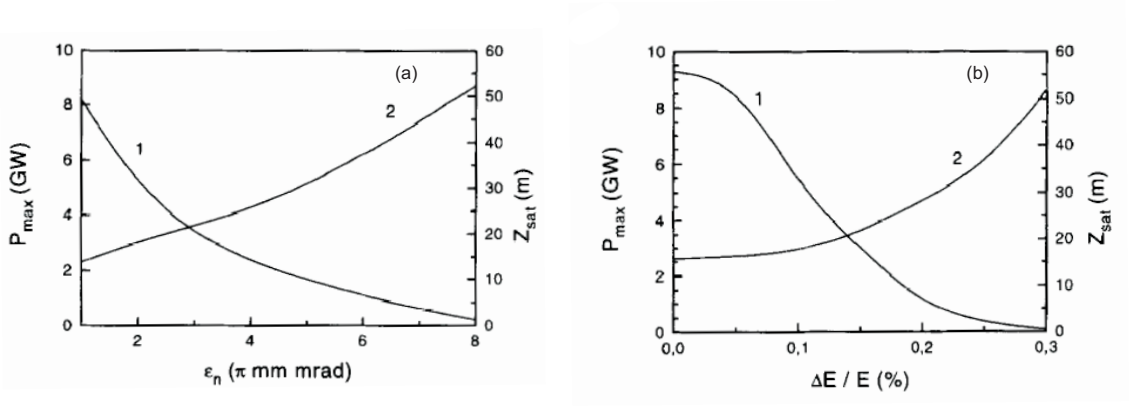

Figure 1: Output peak radiation power (1) and undulator saturation length (2) dependence on (a) - emittance; (b) - energy spread

The aims of high brilliance low emittance photoinjector developing includes their use at high gain FELs. It is shown in Fig.1 that gain and saturation length depends on transverse emittance and energy spread of the electron beam, which produces the radiation [3, 4]. Decreasing saturation length of undulators is economically more efficient, because of technical complexity and high price of the undulator construction and installation.

Transverse emittance is mainly increased by space-charge forces. There is also a thermal emittance, that depends only on the photocathode properties and emission process. Transverse emittance diagnostics and optimisation are developed at PITZ. Close collaboration with producers of rf-cavities and cathodes is aimed to reduce thermal emittance and to suppress emittance growth during acceleration. Also

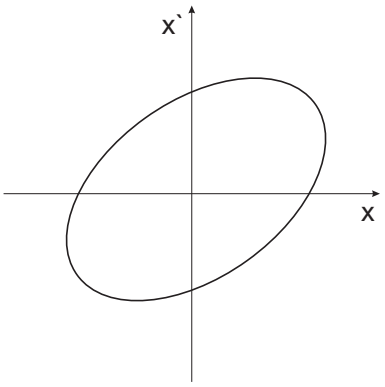

Figure 2: Volume occupied by particles in phase space

measurement methods and diagnostics are investigated at the Photo Injectors Test Facility. There is no strict demand for exact emittance value, but the current decision is to be capable to produce electron beams of at least 1 π mm mrad transverse emittance [4].

Transverse emittance is a parameter depending on both size and opening angle of the particle beam. Transverse emittance provides an important indication of the beam quality. The lower the emittance, the easier it is to focus the beam. In terms of beam dynamics transversal emittance corresponds to ellipse area of particls distribution in phase space (Fig. 2) within factor of π [5]. Several methods exist for measuring beam transverse emittance [7]. Single slit scan methodics is typically used at PITZ.

Another parameter, which is important in the work, is electron beam displacement off the axis. Numerical analysis gives value of 20 μm off the undulator axis as a upper limit for the successful lasing of the FEL. There are still some issues opened concerning beam alignment technique [9].

2.2 Stability Studies

The work contains two subjects of investigation: photoinjector laser intensity stability (IS) and pointing position stability (PPS) of the laser beam. IS is obviously influencing the bunch charge stability, which consequently influences the FEL radiation gain stability [1]. Emittance measurements are influenced by the PPS. For understanding of this problem numerical simulations were performed.

The first task is the laser operation monitoring. Any desynchronization, power drop of pumping elements or optical element failure will cause changes in the longitudinal intensity profile of the train. In the next section we will take a closer look to the laser system and talk about some possible reasons, which lead to stability degradation.

Pointing position stability depends on external factors like optical element vibration caused by acoustic noise, air pressure or temperature fluctuations. Also a reason can exist internally in laser system. This subject is studied in relations with different conditions of the measurements. Setting into operation of the monitoring system and first measurements were done during the facility shutdown. That gives a chance to observe the "clear" monitoring system behavior without external noises. In comparison with the results from the operating facility one can deduce the displacement of the laser beam corresponding to acoustic noise, occured along the optical transfer line.

3 Experiment Approach

3.1 The Photoinjector

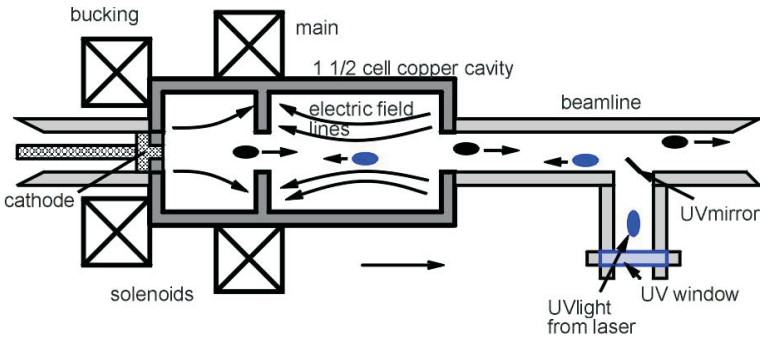

Figure 3: Injector layout: gun cavity, rf-coupler, main and bucking solenoids, cathode system and laser light input

The role of the photoinjector is the generation of high brilliance and small emittance electron bunches for FEL application. For this purpose the generation is based on the photoemission of the Cs2Te cathode [13] induced by the incident light of 262 nm wavelength. Formation of the 18 psec flat-top laser pulse is an initial process. Then the laser light is transmitted from the laser table to the tunnel through a 22 m optical line and enters the vacuum throug the input window. It is directed by the vacuum mirror to the cathode. Just before the entrance to the vacuum there is a beam splitter, which directs 2% of the light to the laser beam diagnostics system.

The cathode material is chosen because of short response time and it is also blind to visible and heat radiation, in spite of poor knowledge existing about this material. The sensitive area radius of the photocathode is 5 mm. By the time laser impulse arrives to the cathode

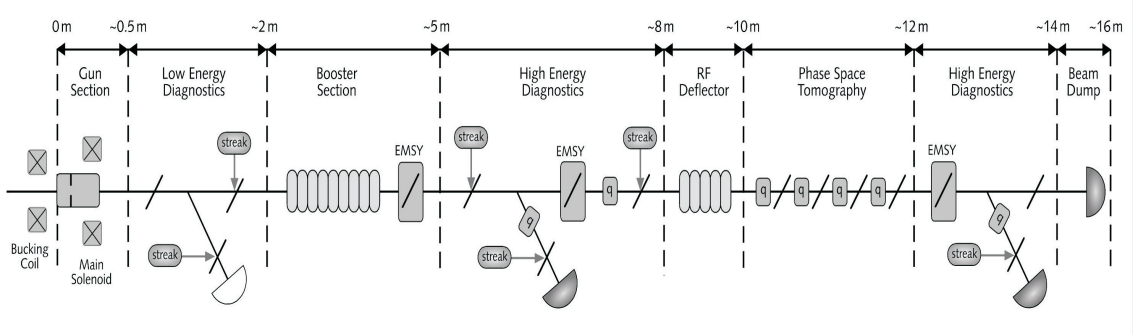

Figure 4: PITZ facility layout scheme

surface the electrical accelerating field is applied in the cavity volume (see Fig.3). During the process of the photoemission electrons get not only a momentum perpendicular to the surface, but also a transverse component. This component is known as thermal emittance. Then the electron bunch is accelerated by the electrical field of the cavity to $\approx 5MeV$, focused by the longitudinal magnetic field of the main solenoid (see Fig. 3). The RF-system is designed to provide up to 1 msec long rf-power pulses. The bucking solenoid is compensating the main solenoid field at the cathode position. Electron bunches are leaded by steering magnets to go around the vacuum mirror. Also there is a dipole magnet (low energy dispersive arm) after the gun, which is used for bunch energy analysis. The next step of acceleration is performed by a booster cavity up to $\approx 12MeV$. After the booster one can see the first EMittance measurement SYstems. The method of emittance measurement is based on the previously described single slit measurements. On EMSY position there are a screen, vertical and horizontal slits of different width. About two meters down-stream there is a screen for the measurement of the slit projection. One mea-

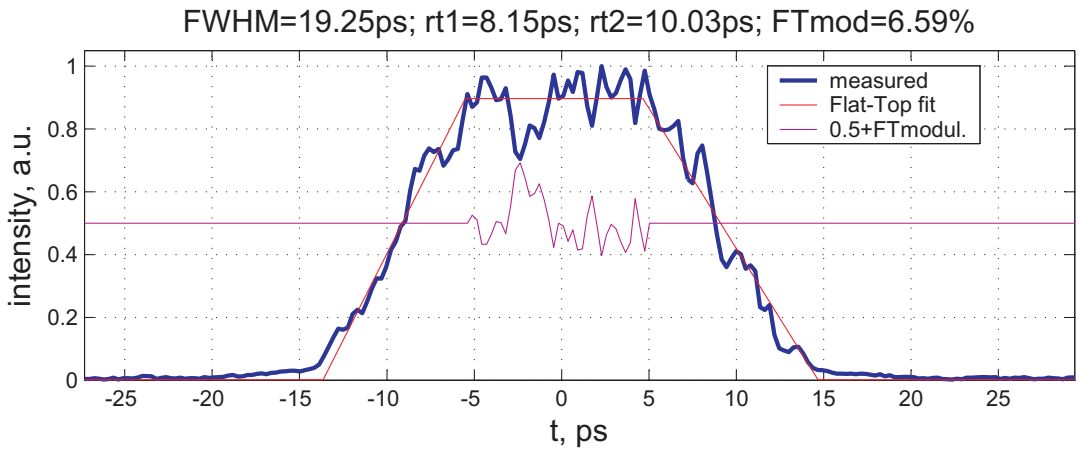

Figure 5: Laser pulse longitudinal profile

sures beam size at EMSY, then a slit is inserted and the projection dimensions are measured at the distant screen. The slit is moved along the beam transverse distribution and the slit projection sizes are measured for the different slit positions . It gives beam density distribution in the phase space, then emittance is calculated.

3.2 Detailed Laser Description

The laser system generates ultraviolet about 20 psec long flat-top pulses at 262 nm wavelength. The repetition rate of the trains is 10 Hz and for the pulses inside is 1 MHz. The laser system can produce up to 800 pulses in the train with output power of each pulse ≈30 μJ. It illuminates the photocathode of the RF gun to generate photoelectrons which are subsequently accelerated by the electrical rf-field. The main oscillator (1065 nm) produces pulses with 27 MHz frequency, what corresponds to full round trip inside the optical cavity. Then the light passes through the pulse shaper, where the Gaussian (longitudinal)

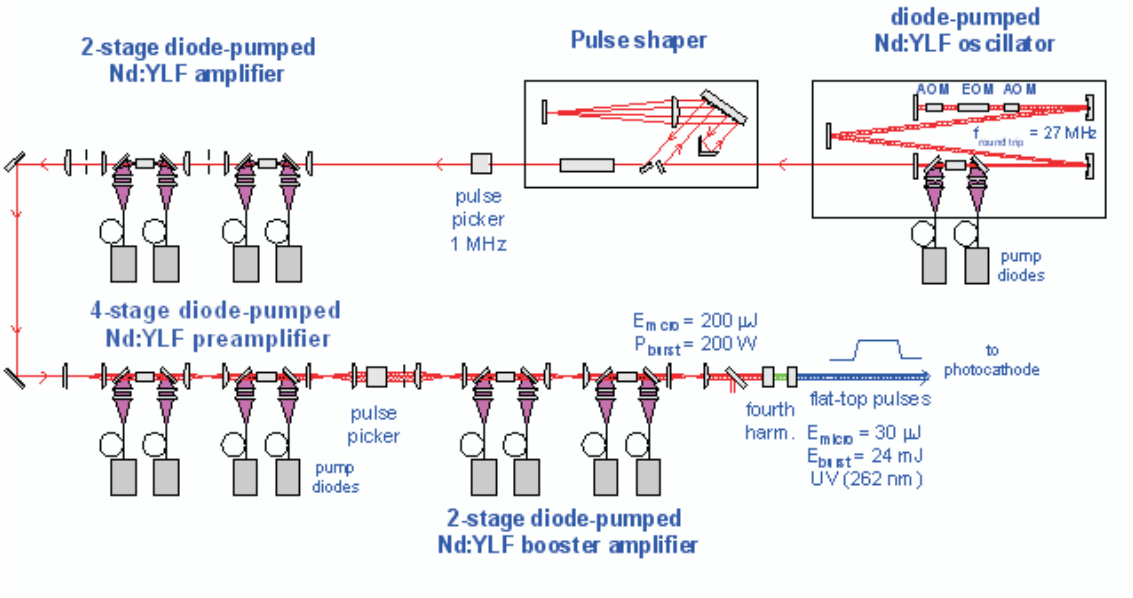

Figure 6: Laser system schematic diagram

pulse is transformed into the flat top pulse by the phase modulation of the different harmonics. After shaping the pulses are amplified and then the pulse series with the frequency of 1 MHz are picked out and sent to the pre-amplifier and booster amplifier. The fourth harmonics (262 nm) is produced by passing through LBO (Lithium Triborate LiB_3O_5) and BBO (beta BaB_2O_4) crystals. Finally, on the output there are flat-top pulses with energy of about 30 μJ and duration full width at half maximum of $\approx$ 20 psec (Fig. 5). Two polarizers with adjustable rotational angle in regard to each other are used to attenuate the intensity. Different beam size limitation apertures are installed in the tunnel and can be inserted to the laser beam line. The laser beam is imaged from the laser to the cathode by two telescopes. It creates a geometrically not moving spot at the cathode. As the main differences from the FLASH cathode laser system the PITZ laser system

has diode pumping of the active media ofoscillator and all the amplifiers and the described pulse shaper system. For the FLASH laser system details see [27].

3.3 Transmission Factor

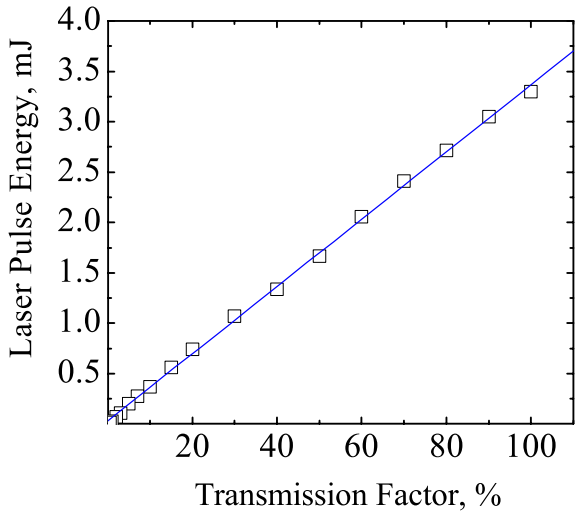

Figure 7: Transmission facor calibration

As it was described previousely for the laser attenuation rotating polarizer is used. The value of the attenuation is controlled by the polarizer angle and is called transmission factor. The calibration was done using calibrated powermeter. It shows that dependence of the light pulse energy on transmition factor is linear (Fig. 7).

3.4 Photodetectors

For controlling the laser beam position and intensity one should carefully choose proper photodetectors. Let's look through variety of the industry proposed solutions.

3.4.1 Light intensity measurements

There are two widely-spread detectors:

- Photomultiplier tube (PMT)
- Semiconductor photodiode

PMT The device is based on conversion of the input light signal to the electrons at the photocathode. Photocathodes are semitransparent or backscattered types. The first one is deposited directly to the input window and is $\sim 10nm$ thick. Photocathodes are made from different materials depending on the interesting wavelength range (from 120 nm to 1000 nm) and conditions[15]. The electrons are accelerated toward the nearest dynode by the applied voltage, where they are multiplied in secondary emission back scattering processes. The process repeats between each subsequent pair of dynodes. Finally electrons reach anode. The number of input photons is proportional to the output quantity of the electrons. The current can be integrated after the PMT if working in pulse mode. There is also a PMT current mode,

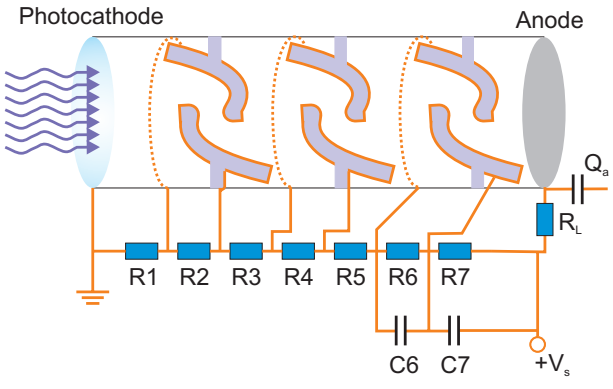

Figure 8: PMT voltage supply scheme

that is used for continuous light illuminations measurements and not described here. Next expression is used to obtain the input photon numbers in a pulse:

$$N_{ph} = \frac{Q}{g \cdot \varepsilon \cdot \delta \cdot e} \tag{1}$$

Q - charge arrived to the anode as a result of a pulse light illumination, g - electron multiplication process gain, ε - the photocathode quantum efficiency, δ - collection efficiency of the first dynode. Electron multiplication is a probabilistic process, hence gain has some distribution and can be specified by the distribution parameters. The width of the distribution describes the quality of the detector - large width can be caused by situation when one of the first dynodes has a very low collection efficincy. Calibration of the g-factor can be done by measuring the PMT one photoelectron response – g-distribution is just the charge distribution at the anode in this case [16, 17, 18].

Advantages of the PMT are: high gain due to the internal multiplication, linearity in dynamical range $\approx 10^4$, fast response (< 10 nsec), low noise, big active area;

Disadvantages: high cost, large dimensions of the multiplying system, gain strongly suffers from external magnetic fields (solved for special types), needed high voltage supplied, easy to destroy the photocathode layer by an intense light illumination.

Photodiode Photodiode is a semiconductor device similar to a simple diode with reverse bias and junction exposed to light. We will talk about common p-n diodes, but everything is true for p-i-n modifications or some more rare hybrids. Due to the reverse bias the diode is

closed for a current. The p-n junction region have no free charge carriers, beside some thermal generated pairs. Absorbed in the junction light generates pairs of electrons and holes, which then tend to reach p or n layers correspondingly. This is a photocurrent, that is integrated, measured and light is obtained [19]. The response is defined as the current response relative to incoming light and for the most photodiodes is $\sim 0.1 A/W$ and covers range from 190 to 1100 nm. One should regard as photodiode advantages: cheap, 5-12V power supply; but slow response (>50 nsec), small active area, high noise-signal ratio.

APD It has a more complicated semiconductor layered structure. A charged particle accelerated by the internal field generates an avalanche towards the electrode. The APD has gain of ≈ 100.This device combines all the features of the PMT with smaller gain and dynamic range, it is more sensitive than a photodiode, and still needs about 100 volts of power supply [20, 19].

Detectors for the monitoring system The structure of the pulse train defines that the response time should be much less than 1 μsec, and there is a 10^4 dynamical range to cover. That is why PMT was chosen for the intensity measurements. Moreover we were not suggesting to use some special features of reconfiguration of the field defining divider circuits, that is why complete PMT module H6780 (Hamamatsu) with intrinsic high-voltage converter was chosen [21].

This PMT module has a bialkali type cathode with quantum efficiency of 0.8 % at 260 nm and an effective area diameter of 8 mm.

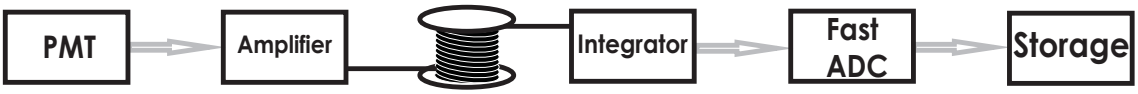

Figure 9: PMT signal line

Another UV-sensitive type H6780-03 has quantum efficiency $\approx 4.5\%$. The high-voltage converter should have +12V of power supply. The gain of the PMT can be regulated externally in range from 100 to 10^6. Maximal linear mode output current is 100 μA. The cathode saturation current is 0.1 μA for the bialkali. If the gain is less then 10^3 the linear output is limited by the cathode saturation, otherwise by the space charge forces in the dynode system during the multiplication.

The signal of the PMT is transmitted to the pre-amplifier with the gain of 14, then it is transmitted to the electronics room, integrated, digitized and send to a data storage (Fig. 9).

3.4.2 Position measurement equipment

Light beam intensity gravitational center position measurement devices are based position sensitive PMT or position sensitive photodiodes. For PMT method one exchanges the anode by two perpendicular non electrically contacted wire arrays, which create a grid, and the current is measured for each wire. The position of the light gravitational center is obtained. Position sensitive photodiodes have additional uniform resistive layer, two side electrodes connected to the p-layer(one dimension) and the central point electrode connected to the n-layer. The current passing a resistance depends on the distance

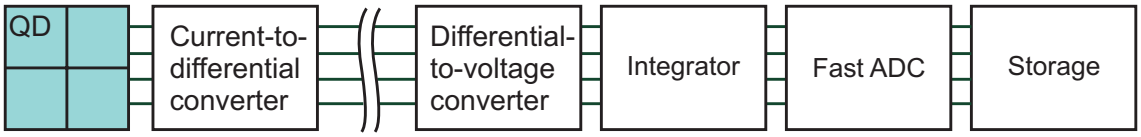

Figure 10: Quadrant diode signal line

to the electrode. If one can measure two currents independently then the position of the gravitational center is obtained.

The first one is expensive and the second one is not fast enough to measure position of each pulse at 1MHz repetitional rate. There is also charge coupled discrete elements detector which is used for the laser transverse beam intensity distribution measurements. It is not sensitive and fast enough, to satisfy demands for parallel to facility operation measurements.

There are two examples when fast controlling of the incident light beam position is required: automatic alignment of the laser beam lines (for example [22]) the position detectors for wavefront sensors of adaptive optics [23]. Both of them use a quadrant diode application. The quadrant diode is essentially a normal photodiode split into four quadrants, output current of each is measured. Since the current is proportional to the incident energy one can easily do alignment or measure cylinder-symmetric beam spot, if the dimensions of the spot are smaller than the detector. For measurements in our case the PIN quad photodiode S4349 (Hamamatsu) was chosen[24]. The response time is $\approx$ 100 ns. The QD has 40% quantum efficiency at 262 nm. Four signals are transmitted to the electronics racks room, integrated, digitized and stored at a hard drive (Fig. 10). The sensitive area is a

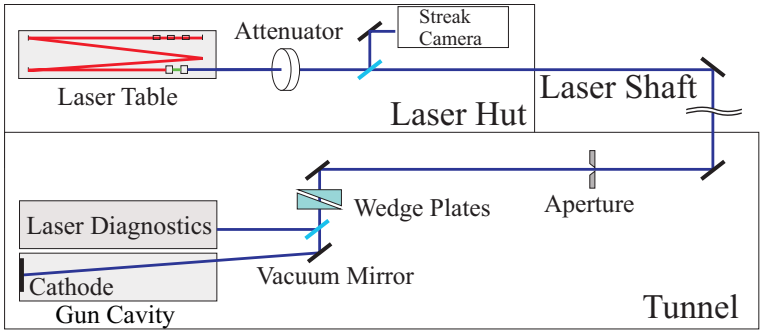

Figure 11: Laser beam line

square with 3 mm side. Gap between the quadrants is 0.1 mm.

3.4.3 Transverse profile of intensity distribution

The position analysis should take it into account the complex beam transverse profile . The algorithm of the quadrant diode signals analysis involves a transverse laser beam intensity distribution, which is measured by the camera (Jai CV-M10CX). The active area is 575x767 pixels, pixel dimensions is 8.3 μm x 8.3 μm. The minimal exposition is 1/917000 sec.

It is impossible to extract the exact laser beam profile from the camera, because it is always will be modified by the binning effect and effects in the transmition line [11].

3.5 Laser Beam Diagnostics

At the facility several diagnostics tools are used. Among them there is the laser beam profile measurement system. The existing system allows to insert a mirror to the laser beam pass way and direct it to the camera (CCD) for transverse intensity profile measurements.

This camera is shielded by 10 cm of lead against the bremsstrahlung produced by the back accelerated dark current striking the walls of the gun cavity. It is demanded to control the transverse laser profile parallel to the electron beam production. The splitter is installed, that separates a percent of the main beam intensity intended for photoelectron production and directs it to a similar camera. It is also situated inside the lead box, constructed on the movable platform - laser trolley (Fig. 11).

New detectors for intensity and position monitoring were decided to be installed at the laser trolley . Quartz plates are used as splitters along the optical line (Fig. 12). On the figure one can find the incident polarization of the light on the reflectors. It is important for the quartz plates where for perpendicular light polarization the reflection factor is ≈ 0.1 and transmission factor is ≈ 0.81, and for parallel it is ≈ 0.02 and ≈ 0.96 correspondingly. These values are obtained from the Fresnel's formulations [14] for $n_{quartz} = 1.50$ at 263 nm. The planes of the quartz plate are not parallel. Therefore the beams reflected from the first and second are not parallel.

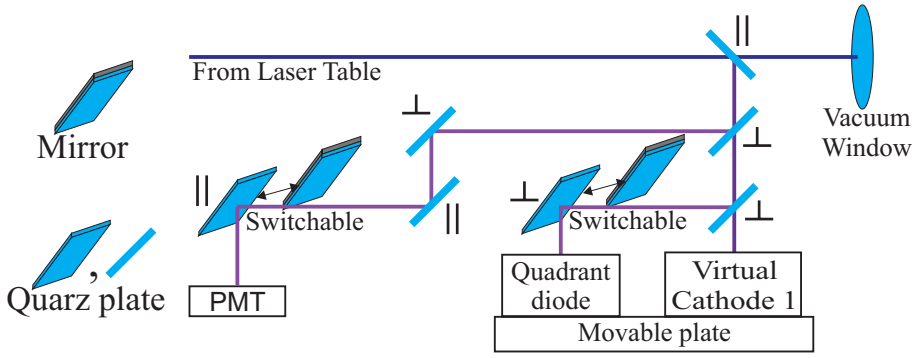

Figure 12: Laser Trolley

The suggestions for the light level attenuation at the input of the detectors come from linearity range limitation. The maximal linear output current of the PMT is 100 μA. As far as minimal response time is $\sim$ 10 nsec the maximal pulse energy on the photocathode should not exceed $10^{-13}\mu$J when gain is $\approx 10^4$ times[1] . For the laser energy of 30 μJ and transmission factor of transport laser beam optics from the laser table to the laser trolley splitter of $\approx 10\%$ the attenuation should be $\approx 10^{-7}$. Nevertheless for the calibration purposes there is a possibility of $8*10^{-8}$ attenuation and $4*10^{-6}$ for the lower attenuator transmission values measurements only.

4 Data Analysis

4.1 Intensity Measurements

As it was shown above that PMT has a wide dynamical range due to the adjustable gain. For a gain value the measurements are limited by the saturation at higher signals, by small signal to noise ratio at lower signals. To find the optimum one should foresee a calibration of the range.

In Fig. 26 such calibration is shown. The linear part for the 0.45V of the control voltage is shorter but gives twice better noise to signal ratio up to 3% transmission factor.

One can see that saturation comes at the same level of the output signal for different control voltages, that means the saturation of the

[1] 100 μA at the anode corresponds to 10 nA at the cathode. This current transports $\approx 10^{-16}$ C in 10 nsec time. To produce this amount of electrons one should illuminate the cathode with 125 000 photons of 4.73 eV energy (quantum efficiency 0.8%)

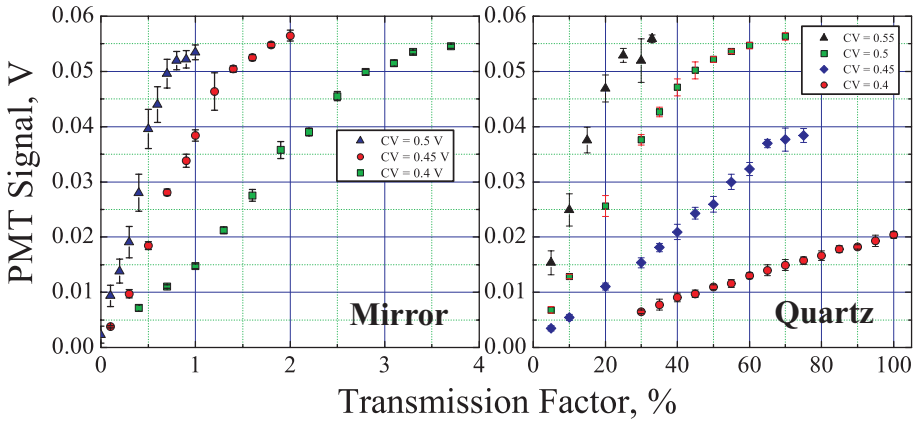

Figure 13: PMT Calibration curves for different control voltage

dynode multiplication system occurs, what corresponds to the value in the PMT specification: for the gain 20000 the limit of 100 μA corresponds to 200 photoelectrons, each of them contributes as ≈0.2 mV, what results at ≈ 40 mV saturation signal. A photoelectron signal contribution is calculated from the measured single photoelectron spectrum of this PMT. With the gain of $2 * 10^6$ one photo electron contributes a Gaussian signal of 20 mV and RMS value of the distribution is 10 mV. Similarity rule can be applied in this case. Then for the gain of 20000 the center of the single electron distribution is 0.2 mV and a sigma of the distribution is 0.1 mV. The charge multiplication process is assumed to be independent for each photoelectron. It points to the fact that N photoelectron signal distribution center is N times bigger, but width is $\sqrt{N}$ times bigger (Appendix A).

The factor of the transmitted light to the PMT between the mirror and quartz reflectors is 2% from Fresnel formulations. On practice it is 1-2% and it corresponds to not exactly perpendicular alignment, possible surface coverage of the reflectors with films. One should always

perform a new calibration of this factor each time after the optical line readjustment.

For the quartz reflector four control voltages are considered: 0.4, 0.45, 0.5, 0.55 V. A range from 6% to 100% of transmission factor can be covered. From the first one should see no saturation, from the second, the third and the fourth there is dynode system saturation at the same level as for the mirror signal calibration curve. The linear parts are observed and the signal amplification between the curves can be easily calculated.

There is a possibility to increase the control voltage further, but one should not forget about the PMT internal signal distribution growth. This growth is proportional to the voltage applied between the subsequent electrodes in the power of 3/2. As the gain is growing proportional to the control voltage, then the same is true for the growth of the width of the distribution.

As far as it is clear now how to choose the control voltage, one can perform some measurements using the PMT.

On Fig. 14 one can see values[2] of the laser pulse intensity from the pulse train stored by Fast ADC. A value is obtained each microsecond after the PMT pulse signal was integrated. These data is an analysis result of averaging of each pulse value for 300 trains. As well in the figure one can see intensity slope in the train. Also standard deviation is given on the figure. The value is caused by electronics noise in the circuit, "natural" width of the PMT signal distribution and the

[2]Only every eighth pulse value is presented at the figure

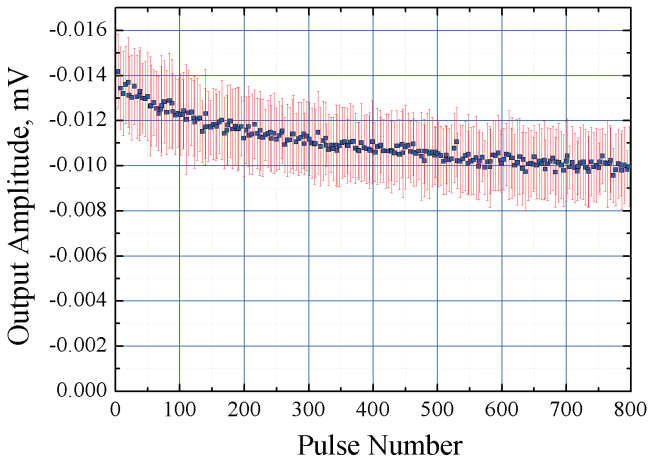

Figure 14: Laser pulse train intensity

distribution width of the laser beam intensity. To extract the real intensity variations distribution width, the next formulation is used:

$$\sigma^2 = \sigma_{en}^2 + \sigma_{sd}^2 + \sigma_{iv}^2 \tag{2}$$

where σ - signal standard deviation, σ_{en} - electronics noise distribution standard deviation, σ_{sd} - PMT signal distribution standard deviation and σ_{iv} - intensity variation distribution standard deviation.

The standard deviation of the electronics noise is 1.3 mV. The standard deviation of the PMT signal distribution comes out from the single photoelectron measurement (Appendix A) and equals 1 mV for the 100 photoelectrons signal. Fig. 15 shows the intensity variation fraction[3] for different light levels (CV = 0.5 V). The first graph shows homogeneous variations all over the train. The second one has lower variations at first pulses, and the last one shows smaller variations for

[3]Intensity variation is defined by fraction of standard deviation of the intensity variations distribution to the averaged signal value

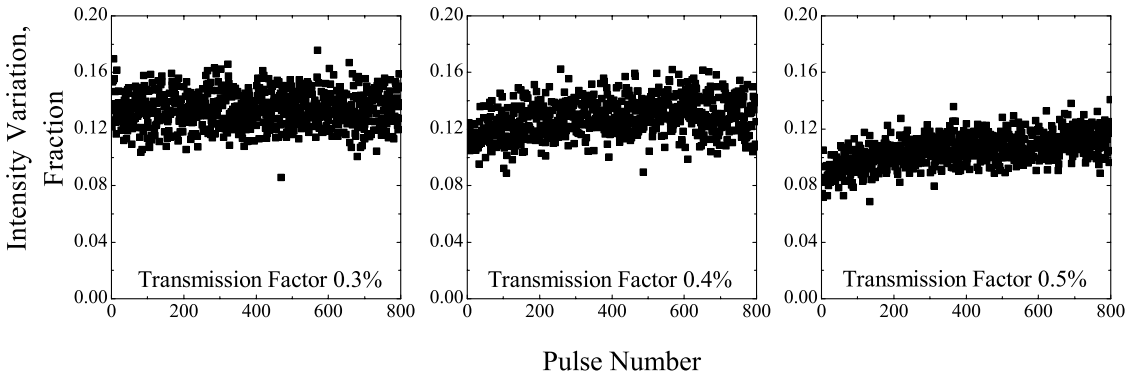

Figure 15: Intensity variations for different light levels. Control voltage is 0.5 V

all the pulses. It can be explained by the dynode system saturation, which limits the distribution width from the side of larger charges.

4.2 Position measurements

For the position measurements it is important to have symmetry of the four channels of the quadrant diode. If the condition is not matched at the level of the equipment then it can be corrected by the analysing software. To check this a test was made for equality of the quads and their electronics circuits. The integrator channels after the test and the recalibration(is done in analysis software) are found to be with the accuracy of 1%. Test of the quads showed equality with the same accuracy as the previous test, what means if difference exist it is much smaller than the value of 1%.

The results of the initial procedure of output signal versus transmission factor of the attenuator calibration is shown on Fig. 29. The mirror gives saturation at 8% of the transmission factor. Interesting

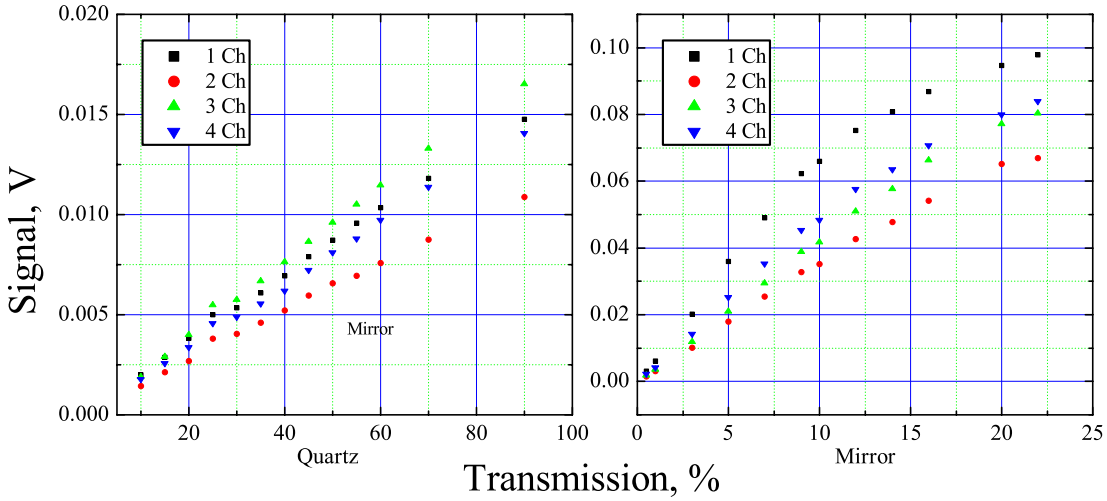

Figure 16: Calibration of quadrant diode signal versus transmission factor

area[4] is not covered with the linear region. But nevertheless one could use it in a range from 10% to 20% of the transmission factor taking into account that the response is not linear and is less sensitive to the displacements[5].

The factor of attenuation by quartz is 0.06 (theoretically 0.1). For quartz the light intensity is too low at the level of the cathode illumination needed for producing 1 nC. That means the displacement is mostly defined by the quadrants noise.

The aim of the laser pointing stability measurement tool is to measure pulse-to-pulse laser beam position. One should know the laser beam intensity distribution to be able to convert signal change into the position change. It has been easily done with Gaussian profiles using the signal to position calibration curve. Such calibration curve corresponds to the numerical integration of the gauss distribution im-

[4]Area of the intensiting light intensities needed for producing 1 nC

[5]Displacements are rather small to consider response function linear in the range of the output signal

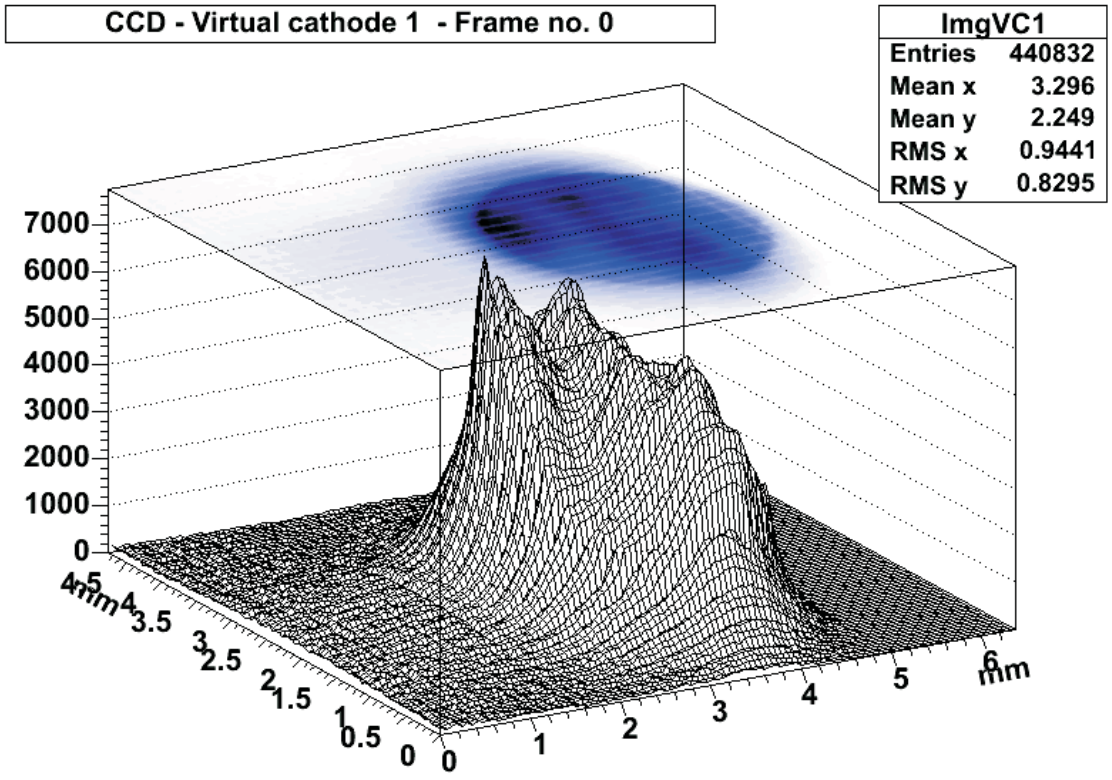

Figure 17: Laser beam transversal profile

itating the integration of the signal over each quadrant. In our case the transverse profile can differ from cylindric symmetric. Then the only possibility to make such convertion is to use the two dimensional distribution profile. The matrix of the intensity distribution is taken by the camera (VC1). This matrix is divided into 4 parts and intensity is integrated and compared with 4 signals of the quad diode.

The complex transverse profile of the laser beam was chosen see Fig. 17. Ability to move the quadrant diode makes it possible to check whether the real displacement correspond to measured one. The results of this check are presented at Fig. 18. Strong discrepancy is observed in this measurement between the readout position change

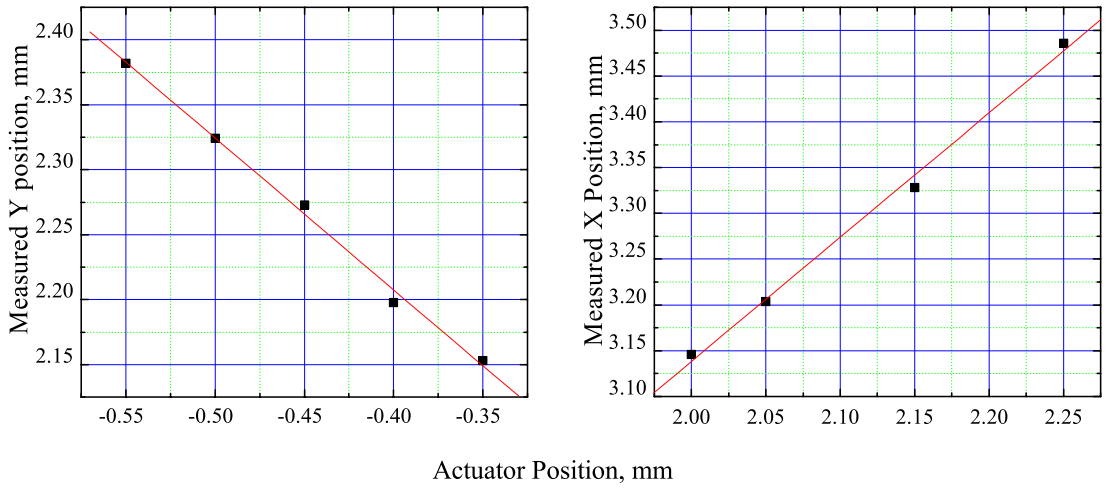

Figure 18: Correspondence between quadrant diode displacements and obtained from the analysis values of the position

and the actual one. The actuator positioning accuracy in the range of 0.2 mm displacement is less than 10 μm. There is a possibility to align a geometrical center of the laser beam at the center of the quadrant diode with accuracy of 0.05 mm. In this position the analysis program did not give the right position (0.5 mm off the center). The reason is in one more degree of freedom — the angle between the quadrant cross and camera pixels rows and columns which was considered to be 0 in analysing software.

First results are shown in Fig. 19. The position of each pulse was averaged over 300 trains, and also the standard deviation was calculated. Trains of 800 pulses are used. One can see a drift along the Y-axis. This drift is $\approx 140\mu$m during the train. The drift along the X-axis is smaller than 40 μm. There is a doubling of the curve, which come from the synchronized 500 kHz hindrance and can be easily neglected.

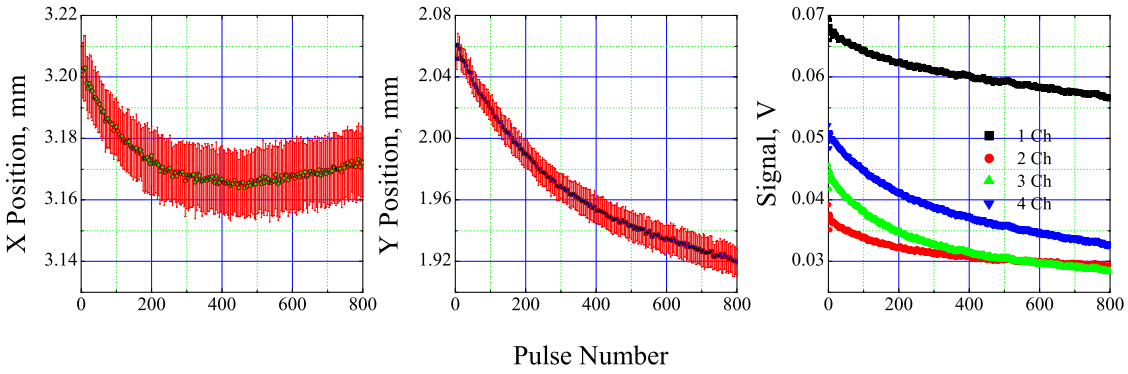

Figure 19: Position measurement results. Mirror reflector. Transmission factor = 10%

It was observed, that drift doesn't depend on position in the position displacement from the geometric center range ± 100 μm, and on intensity in the range of linear response. One can assume, that it can occur because of heat lensing [27] in the laser light amplifier systems. To check that the level of pumping was decreased two times, but the same drift existed anyway.

Another thought was to prove that the drift is real. As far as the aperture makes the laser beam geometric spot stable, the beam displacement before it will lead to the laser intensity distribution gravitational center displacement at the cathode (and virtual cathode). If that is true, one should observe a mean position variation at the camera with increasing the number of integrated pulses. The results are shown in Fig. 21. The drift exists along the y-axis of the camera, which has opposite to the y-axis of the quadrant diode due to the optics line. The drift is approximately twice smaller than for the quadrant diode, and it is obviously due to the integration.

The next step was an attempt to find out: is it a profile drifting, or

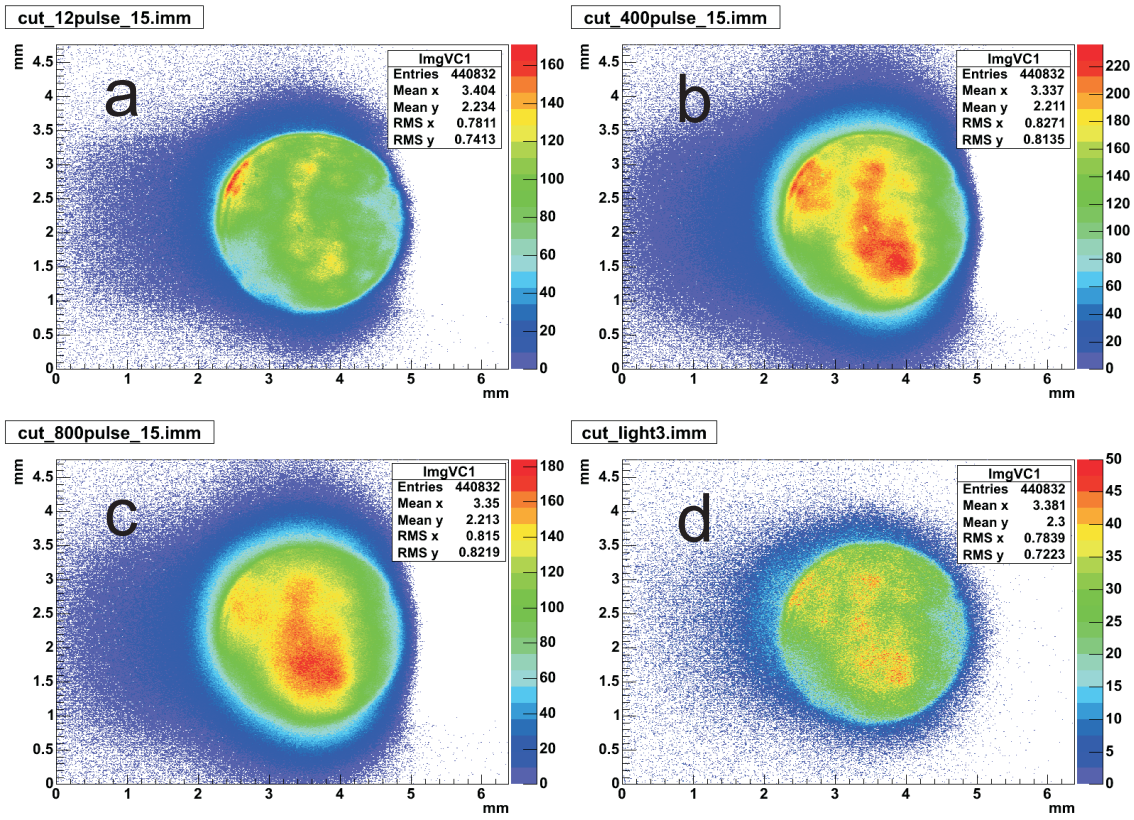

Figure 20: Transverse profiles of laser beam. Time stamp 0 μsec corresponds to the beginning of the first pulse: a - 0 μsec; b - 400 μsec; c - 800 μsec; d - 914 μsec

is the distribution changed. For this purpose we used short exposition camera mode to make pictures of the laser beam profile of the first pulse, then 400th and then the 800th. One should see from these shots if distribution is changing. According to the camera specification the shortest exposition is 1/917000 sec, it mean one can capture not more than 2 pulses, but with the appropriate synchronization only one. It was discovered that with the exposition we could capture even three pulses. The measurements were done with the virtual cathode 2, which obtains more intensity. Three pictures were taken, and a blurring was found at 400th and 800th pulses picture. Then the picture was taken

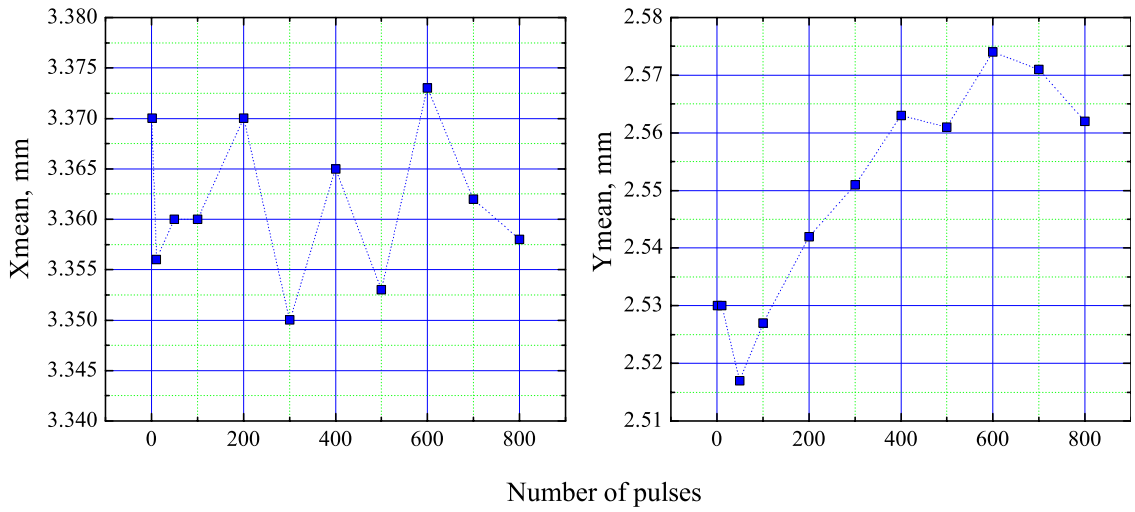

Figure 21: Average of the distribution measured by camera versus number of integrated pulses

114 μsec (light3) after the train and there still was a signal (Fig. 20).

The stability of position is referred to the value of position standard deviation, that consists of few main parts: electronics interference, photodiode signal distribution and position variation. The standard deviation is calculated for each pulse from 300 trains. As far as there was no analysis of the variation components done, lets consider the value of 15 μm as a high limit of the standard deviation of the position variation distribution.

5 Conclusions

Obtained Results

Laser Beam Intensity Measurements

The system is implemented and first results have been obtained. Due to the PMT single photo electron calibration we could find the response signal variation of the PMT. The deconvoluted value of the laser beam intensity variations is 14%. For comparison declared intensity stability of the Gaussian longitudinal profile laser system at FLASH is 5%. The intensity slope inside a train can be observed. The acuracy of the measurement is about 7% and comes from the fast ADC channel width of 0.6 mV.

There is a demand to be able to monitor and control the intensity slope in the train, because for the 800 pulses mode the heating of the cathode can lead to quantum efficiency changes and then it could be compensated by the laser beam intensity slope. This can not be solution, but a good tool for the investigations. Slope control can be performed changing the timing of the laser pumping elements [27].

Laser beam pointing position measurements

The system using a quadrant diode is implemented, analysis software was written and checked. The first results have shown drift of the laser spot of the position inside a train of 30 μm and 140 μm for X and Y axis correspondingly. The geometrical sum of both axes jitter does not exceed 15 μm. This value should not be treated as a final one, because strong discripency between the real position change and

read out one was found. The measurements for a symmetric profile should give a real value which is definetely smaller.

A single pulse camera shot could not resolve the beam transverse intensity distribution changings because of a shadowing effect by afterglowing. The secondary radiation is excited by the intense UV and could not be avoided by simple intensity reduction, what will lead to very slow camera signals.

First of all one should exclude the angle dependence of the laser beam pointing position value or find and make a correction inside the software. If beam can be positioned at the center of the quadrant diode with the accuracy of 10 μm then one should fasten position parameters in the program and vary only the angle. Few measurements should be done and the angle should be defined. The quadrant diode could be rotated once by the angle to match the camera orientation.

The reason of the drift is believed to be inside the laser system. Advanced experiments with the system can be performed together with laser specialists. Till now drift adds 1% more to the electron beam transverse size RMS at 800 pulses train length.

After starting of the PITZ facility operation one should repeat all the measurements: electronics noise distribution, drift and jitter in noisy environment. It could be performed in stages - a measurement after each component was started (RF system, pumping stations, water cooling of the gun cavity and etc.). These measurements should reveal the contribution of each element to the laser parameters stability.

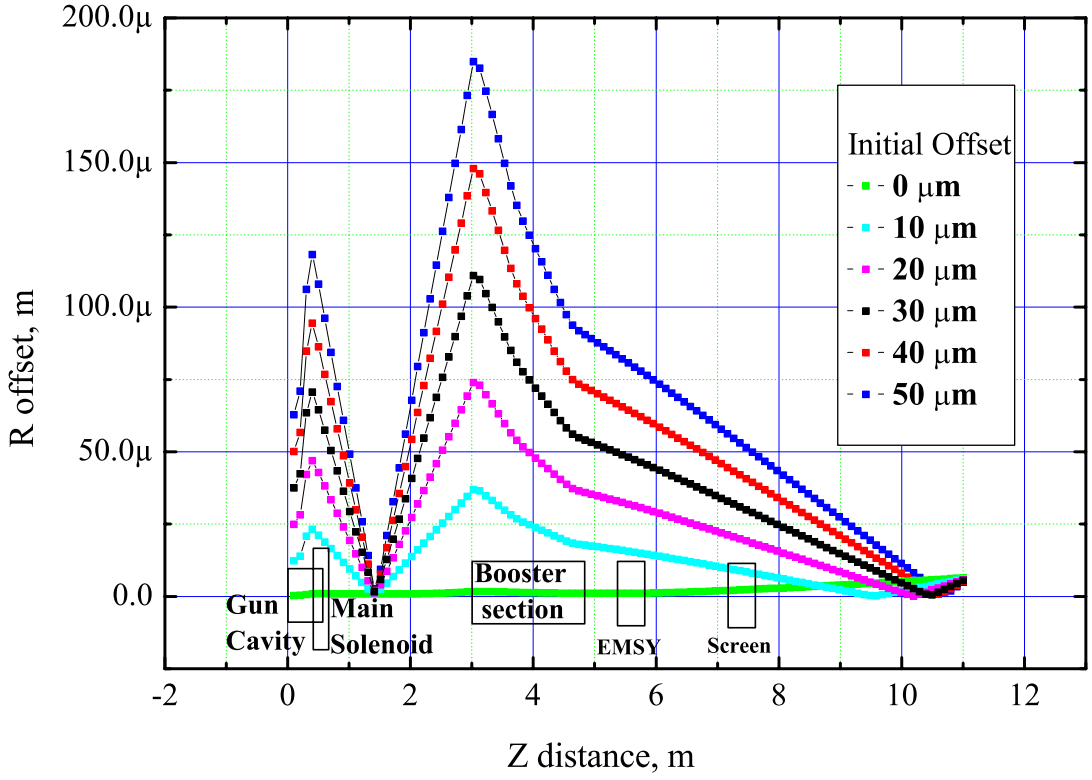

Figure 22: electron beam offset along the accelerator line for different initial values

Simulations

Electron beam behavior simulations are performed to determine the limits for the laser beam displacement from the axis at the cathode. ASTRA code is used [12]. Initially some laser beam displacements at the cathode are given - 10, 20, 30, 40, 50 μm. The gun cavity, main and bucking solenoids and the booster(Fig.22) are involved in simulations. Then the positions of the elements and phases are tuned to optimize the mean bunch energy and emittance at the output of the booster cavity. As result one obtains beam size and distance from the axis of symmetry as a function of passed distance.

Results are shown in Fig.22 . The electron beam transverse dis-

placement was investigated along the beam line. One can see beam line elements at the bottom of the figure. Offset is given in cylindric coordinates only by radius. From the results one should conclude that at the EMSY position electron beam displacement is approximately 7/4 of that at the cathode. It leads to slit projection mixing. In this case 35 μm position jitter at the EMSY position of the electron beam stretches the transverse beam size root mean square less than 1%.

References

[1] Avi Gover et al., Coherence of E-Beam Radiation Sources and FELS - A theoretical Overview, FEL2006 conference talk, Germany, Berlin, 27^{th} August - 1^{st} September.

[2] W. Ackermann et al., Operation of a Free Electron Laser in theWavelength Range from the Extreme Ultraviolet to the Water Window, not published.

[3] W. Brefeld et al., Parameter study of the VUV FEL at the TESLA Test Facility, Nucl. Instr. and Meth. A375(1996), pp. 295-298.

[4] J. Rossbach, A VUV free electron laser at the TESLA test facility at DESY, Nucl. Instr. and Meth. A375(1996), pp. 269-273.

[5] Klaus Wille, The Physics of Particle Accelerators, [Oxford, Oxford University Press, 2000].

[6] J.-P. Carneiro et al., Emittance Measurements At The A0 Photoinjector, XX International Linac Conference, Monterey, California, 2000, 21-25 August

[7] K.T. McDonald and D.P. Russell, Methods of Emittance Measurements, [Joseph Henry Laboratories, Princeton University,Princeton, *NJ 08544*], Oct.20 1988.

[8] S.C. Hartman et al., Emittance Measurements of the 4.5 MeV UCLA RF Photo-Injector, Particle Accelerator Conference, Washington D.C., 17-20 may, 1993.

[9] B. Faatz and S.Schreiber, Commissioning of the VUV-FEL

[10] R. Brinkmann, The European XFEL Project, FEL2006 conference talk MOBAU03, Germany, Berlin, 27^{th} August - 1^{st} September.

[11] R. Spesyvtsev, Beam Size measurement uncertainties, diploma thesis, 2007.

[12] K. Floettmann, ASTRA User Guide, internal user instructions.

[13] R.A. Loch, Cesium-Telluride and Magnesium for high quality photocathodes, Masterdiploma thesis, University of Twente, 2005.

[14] Max Born and Emil Wolf, Principles of Optics, Published by Pergamon, 1993.

[15] Hamamatsu corp., Photomultiplier tube: principle of application, 1994.

[16] E.H. Bellamy, I. Chirikov-Zorin, S. Tokar and et al., Absolute Calibration and Monitoring of a Spectrometric Channel Using a Photomultiplier, Nucl. Instr. and Meth. A339(1994), pp. 468-476.

[17] Razmick Mirzoyan, Conversation Factor Calibration for MAGIC Based on the Use of Measured F-factor of PMT, internal MAGIC memo, 12.04.2000.

[18] A. G. Wright, Absolute Calibration of Photomultiplier Based Detectors - Difficulties and Uncertainties, technical preprint RP/091, Electron Tubes Limited.

[19] Silvano Donati, Photodetectors: Devices, Circuits, and Applications, Chapter 5: Semiconductor Photodetectors, Prentice-Hall PTR, 2000.

[20] A. Stoykov and R. Scheuermann, Silicon Avalanche Photodiodes,

http://www.neutron-eu.net/jra8/files/APD_intro_JRA.pdf

[21] See Hamamatsu webpage,

http://sales.hamamatsu.com/en/products/electron-tube-division/detectors/photomultiplier-modules/h6780.php

[22] A. Dunster et al., Automatic Alighnment System Testing for Vulcan, Central Laser Facility Annual Report 2004/2005, CCLRC Rutherford Appleton LAboratory, Chilton, Didcot, Oxon., 0X11, 0QX, UK.

[23] A. Tokovinin, http://www.ctio.noao.edu/∼atokovin/tutorial/part3/wfs.html.

[24] Hamamatsu webpage,

http://sales.hamamatsu.com/assets/pdf/parts_S/S4349.pdf

[25] E. Schneidmiller, Lasing at 13 nm of the SASE FEL at FLASH, FEL2006 conference talk, Germany, Berlin, 27^{th} August - 1^{st} September.

[26] See papers at http://adweb.desy.de/pitz/web/scientific_infos.html

[27] I. Will, G. Koss, I. Templin, The upgraded photocathode laser of the Tesla Facility, Nuclear Instruments and methods in physics Research A 541(2005) pp. 467-477

A Single photoelectron technique

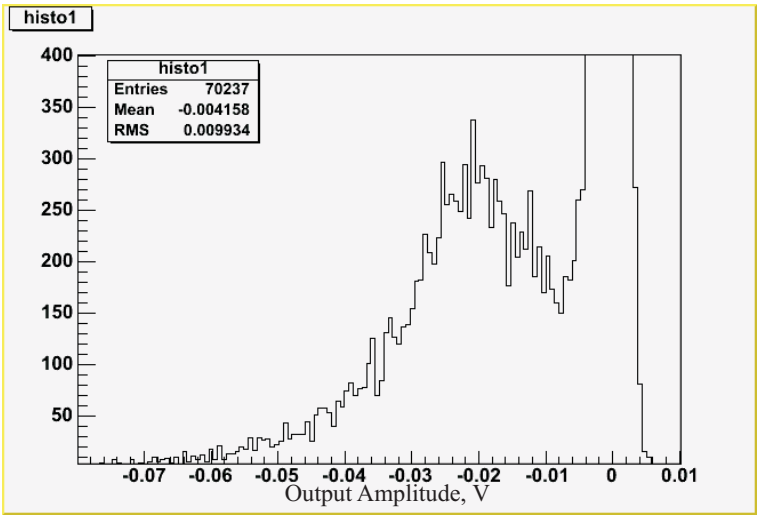

Figure 23: Single Photoelectron Distribution

The idea of the single photoelectron method of PMT characterization lies in measuring the response of PMT to light levels near to the one-electron creation level. As far as the light source, such as LED, produces number of photons in a pulse distributed by Poisson law, one can lower the intensity to suppress the probability of two and more photoelectrons events, but still have single photoelectrons. It means that the probability of zero will be much higher and it leads to a significant statistics of the noise events. A good PMT should easily resolve the single photoelectron signal level. Let's consider, that this signal value is distributed by the Gauss law. Then it is defined with two parameters: mean output charge Q_1, standard deviation of the distribution σ_1 as a width parameter. Then a probability to have charge x at the cathode caused by one photoelectron event is formulated:

$$G_1(x) = \frac{1}{\sigma_1\sqrt{2\pi}} \cdot \exp\left(-\frac{(x-Q_1)^2}{2\cdot\sigma_1^2}\right) \tag{3}$$

Finally total PMT response is expressed by a sum of different photoelectron quantity responses weighted by the possibility of occuring C_n.

$$R = \sum_{n=1}^{\infty} C_n \cdot \exp\left(\frac{-(x-n\cdot Q_1)^2}{2\cdot n\cdot\sigma_1^2}\right) \tag{4}$$

On the Fig. 23 there is a thin high peak, that corresponds to "zero" events distribution. On the left side there is obviously the assymetric distribution and it is clearly separated from the zero-distribution. It is not Gaussian, because of more probability of low charge

noise events. One of them is autoemission from the last dynodes. It can be seen comparing zero peak of high voltage supplied not illuminated PMT and much lower voltage supplied without illumination. The zero-peak of the first one has a "tail".

Result analysis is shown in Fig. 24. For the PMT gain $2*10^6$ the mean response signal is $\approx 21mV$ and the standard deviation is 10 mV.

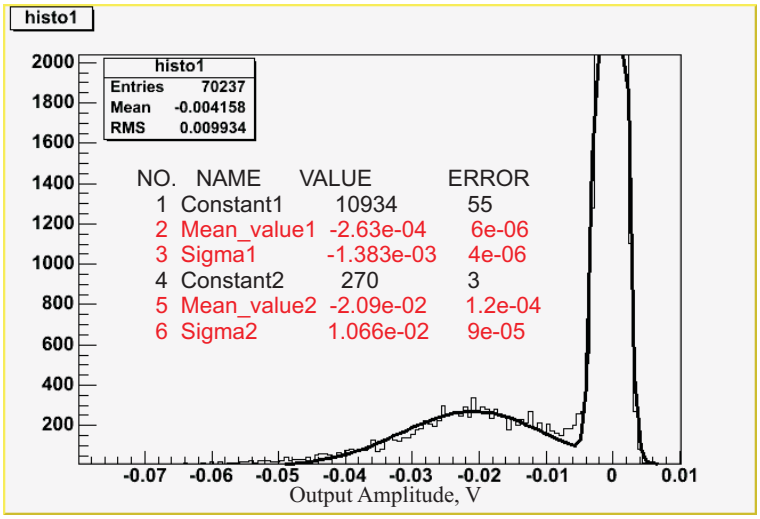

Figure 24: Parameters defined from the fitting

B Software Listing

Profile Matrix Creation

```
#include <TRandom.h>
#include <TObject.h>
#include <TH2.h>
#include <TProfile2D.h>
#include <fstream>
#include <iostream>
#include <TFile.h>
#include "math.h"
#include <string>
#include <TROOT.h>
#include <stdio.h>
#include <stdlib.h>

int IntegralRowN (int y, int x);
int IntegralRowS (int y, int x);
int IntegralColumnN (int x,int y);
int IntegralColumnS (int x,int y);
int IntegrateQ(int x,int y, int QuarterNum);
float ABS (float x);
int ReadImage(char []);
float ABS(float);
bool NewPalette(int Levels, Double_t *Red,Double_t *Green, Double_t *Blue, Double_t
*Stops);
const char *f2s(float num);
const char *i2s(int num);

using namespace std;

const int Cross = 6; // Number of pixels from center to the edge of gap
int binQx,binQy; // Horizontal and vertical Bin quantity. Each bin is a pixel of camera.
This var-les are worked out with external function ReadImage, that is why they are de-
```

clared as globals

float Endx, Endy, Scale,Startx,Starty; // Range of x and y in milimeters from 0 to end—x/y— and miliimeters->scale*pixels

int shift = 6; // Shift while searching for stable solution

char* File;

int Matrix[768][574];

float SumM=0;

float q1A,q2A,q3A,q4A; // Signals from QD in consequent process of position searching

float q1,q2,q3,q4, SignalSum, pmt; // Simulated division by QD

float SI, SII; // Vertical and horizontal sum of qiA

//int x1,y1,x2,y2,x3,y3,x4,y4;

float ds1,ds2,ds3,ds4,Stran;

float dSdown = 0, dSup = 0; // Variables for defining the most presice position

int op,on; // variables for defining the direction of where to shift virtualQD coordinates

int x,y; // For imitation of real position

int Sum; // Total Sum of values of all pixels

int hxbinN,hybinN; // Number of bins where the virtualQD center is lying

int MeanxBin, MeanyBin; // Position of Gravity center of the intensity

int i,j,k; // Counters

float pi = 3.141592654; float HandVar,HandVar1; // assistant variable

double Meanx, Meany; // Position of Gravity center of the intensity distribution in mm

char charData[10250], Answer;

float arg[6];

float val[4];

float ar1[2048];

float ar2[2048];

float ar3[2048];

float ar4[2048];

float ar5[2048];

int Signal12[768-2*Cross][574-2*Cross];

int Norm34[768-2*Cross][574-2*Cross];

int Signal14[768-2*Cross][574-2*Cross];

int pr1,pr2;

```
int xmin,xmax,ymin,ymax;
float FAr[768][574];
float HSig, VSig, MinH, MinV, Minx, Miny; // HSig = 1+4; VSig = 1+2
int cntr;
bool CM[767][573];
char Buffer[60];

int main(int argc,char** argv){

File = argv[1];

        ifstream interest("interest.ing");
        if (!interest.is_open()) cerr << "File interest.ing is missing";
        interest.getline(Buffer,60);
        interest.getline(Buffer,60);
        interest.getline(Buffer,60);
        interest.getline(Buffer,60);
        interest >> xmin;
        cout << xmin << endl;
        interest.getline(Buffer,60);
        interest.getline(Buffer,60);
        interest.getline(Buffer,60);
        interest.getline(Buffer,60);
        interest >> xmax;
        cout << xmax << endl;
        interest.getline(Buffer,60);
        interest.getline(Buffer,60);
        interest.getline(Buffer,60);
        interest.getline(Buffer,60);
        interest >> ymin;
        cout << ymin << endl;
        interest.getline(Buffer,60);
        interest.getline(Buffer,60);
```

```
        interest.getline(Buffer,60);
        interest.getline(Buffer,60);
        interest >> ymax;
        cout << ymax << endl;
        interest.close();
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
//Defining integral signal 1+2 and 3+4 and then 1+2 normilized.
ofstream opn(”output12.txt”);
//ifstream ipn(”JQO.txt”);
//if (!ipn.is_open()) cerr << ”Sorry, can’t open source file ” << endl;
pr1 = ReadImage(File);

// First element initialization for 1&2
Signal12[xmin][ymin] = IntegrateQ(xmin,ymin,1) + IntegrateQ(xmin,ymin,2);

// First element initialization for 4&3
Norm34 [xmin][ymin] = IntegrateQ(xmin,ymin,3) + IntegrateQ(xmin,ymin,4);
// Limits for cross movements
//pr1 = 768 -2*Cross;!!
//pr2 = 574 - 2*Cross;!!ne nuzhno dlja oblasti interesa
for(i=xmin+1; i<=xmax; i++)
{
        j=ymin;
        cout << i << endl;

        Signal12[i][j] = Signal12[i-1][j] + IntegralColumnS(i-Cross, j+Cross) - IntegralColumnS(i+Cross, j+Cross);
        Norm34 [i][j] = Norm34[i-1][j] + IntegralColumnN(i-Cross, j-Cross+1) - IntegralColumnN(i+Cross, j-Cross+1);

for( j = ymin+1; j <= ymax; j++)
```

```
{
        Signal12[i][j] = Signal12[i][j-1] - IntegralRowN(i-Cross+1,j+Cross) - IntegralRowS(i+Cross,j+Cross);
        Norm34[i][j] = Norm34[i][j-1] + IntegralRowN(i-Cross+1,j-Cross) + IntegralRowS(i+Cross,j-Cross);
}};

// Output to file
for( i = xmin + 1; i <= xmax; i++)
for( j = ymin + 1; j <= ymax; j++)
{
        if ((Signal12[i][j]+Norm34[i][j])==0) FAr [i][j] = 1;
        else FAr[i][j] = Signal12[i][j]*1.0/(Signal12[i][j]+Norm34[i][j]);

};
//ipn.close();
opn.close();
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
//Defining integral signal 1+2 and 3+4 and then 1+2 normilized.
ofstream opn1(”output14.txt”);
//ifstream ipn(”JQO.txt”);
//if (!ipn.is_open()) cerr << ”Sorry, can’t open source file ” << endl;

// First element initialization for 1&2
Signal14[xmin][ymin] = IntegrateQ(xmin,ymin,1) + IntegrateQ(xmin,ymin,4);
// First element initialization for 4&3 is not needed - we know norm from 12+34

//Consequent creation of matrix for 1&4 signal
for( i = xmin + 1; i <= xmax; i++)
{
        j=ymin ;
```

```
            cout << i << endl;
            Signal14[i][j] = Signal14[i-1][j] - IntegralColumnS(i+Cross, j+Cross) - Integral-
ColumnN(i+Cross, j-Cross+1);



    for( j = ymin + 1; j <= ymax; j++)
{
            Signal14[i][j] = Signal14[i][j-1] - IntegralRowS(i+Cross,j+Cross) + IntegralRowS(i+Cross,j
- Cross);

}};

// Output to file
for( i = xmin + 1; i <= xmax; i++)
for( j = ymin + 1; j <= ymax; j++)
{
            if ((Signal12[i][j]+Norm34[i][j])==0) FAr [i][j] = 1;
            else FAr[i][j] = Signal14[i][j]*1.0/(Signal12[i][j]+Norm34[i][j]);
            opn1 << FAr[i][j] << " ";
};
//ipn.close();
opn1.close();
cout << endl;
cout << endl;
cout << "output12.txt, output14.txt, interest.ing were created!" << endl;
cout << "That's it" << endl;
return pr1;
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
                    };




// Returns integrated signal from CCD picture for defined cross coordinates and part
```

```
number
int IntegrateQ(int x,int y, int QuarterNum) { //QuaterNum: 1 - x>0,y>0; 2 - x<0,y>0;
                        //3 - x<0,y<0; 4 - x>0,y<0
                        // Gap 12x12 pixels


            int i,j;
            int Integral = 0;


            switch (QuarterNum) {
            case 1:
                        for(i = x+Cross+1; i <= binQx; i++)
                        for(j = y+Cross+1; j <= binQy; j++) Integral += Matrix[i][j];
                        break;
            case 2:
                        for(i = 0; i < x-Cross+1; i++)
                        for(j = y+Cross+1; j <= binQy; j++) Integral += Matrix[i][j];
                        break;
            case 3:
                        for(i = 0; i < x-Cross+1; i++)
                        for(j = 0; j < y-Cross+1; j++) Integral += Matrix[i][j];
                        break;
            case 4:
                        for(i = x+Cross+1; i <= binQx; i++)
                        for(j = 0; j < y-Cross+1; j++) Integral += Matrix[i][j];
                        break;
            default:
                        cout << "Error while switch" << endl;
                        break;};
            return Integral;



}
// Returns integrated signal from one row excluding cross
```

```
int IntegralRowS (int x, int y){

        int i;
        int integralRow = 0;

        //for (i = 0; i < x - Cross; i++) integralRow += Matrix[i][y+Cross+1];
        for (i=x+1; i < binQx; i++) integralRow += Matrix[i][y];


return integralRow;
}

// Returns integrated signal from one row excluding cross
int IntegralRowN (int x, int y){

        int i;
        int integralRow = 0;


        for (i = 0; i < x - 1; i++) integralRow += Matrix[i][y];
        //for (i=x+Cross+1; i < binQx; i++) integralRow += Matrix[i][y+Cross+1];


        return integralRow;
}
// Returns integrated signal from a column part over horizontal cross line
int IntegralColumnS (int x,int y){

        int i;
        int integralColumn = 0;


        //for (i=0; i < y-Cross;i++) integralColumn += Matrix[x+Cross+1][i];
```

```
        for ( i = y + 1; i < binQy; i++) integralColumn += Matrix[x][i];

return integralColumn;
}

// Returns integrated signal from a column part under horizontal cross line
int IntegralColumnN (int x,int y){

        int i;
        int integralColumn = 0;



        for (i=0; i < y-1;i++) integralColumn += Matrix[x][i];
        //for (i=y+Cross+1; i<=binQy-1;i++) integralColumn += Matrix[x+Cross+1][i];

        return integralColumn;
}

// Returns absolut value
float ABS (float x){

        if (x>0) return x;
        else return x=-x;}
int ReadImage(char filename[])
gROOT->SetStyle("Plain");

ifstream examplefile(filename,ios::in — ios::binary—ios::ate);

if (! examplefile.is_open())
        cout << "Error opening file" <<endl;
        exit (1);
```

```
long size;
size = examplefile.tellg();

char *buffer;
buffer = new char [size];

examplefile.seekg (0, ios::beg);
examplefile.read (buffer, size);

cout << "the complete file is in a buffer, size = " << size << endl;

examplefile.close();

int n[8],inc, frameno;
unsigned int t;

int w= * (int *) &buffer[0];
int h= * (int *) &buffer[sizeof(w)];

double scale=* (double *) &buffer[(w*h)+8];

if (size==((w*h)+8+sizeof(scale))) frameno=0;
else
        frameno=* (int *) &buffer[(w*h)+8+sizeof(scale)];

binQx=w;
binQy=h;
Scale = scale;

Startx= 0;
Endx= w*scale;
Starty= 0;
Endy= h*scale;
```

```
cout << "WxH = " << w << "x" << h << endl;
cout << "scale = "<< scale << " frame no.= " << frameno << endl;

string title;
title="CCD - Virtual cathode 1 - Frame no. ";
title+=i2s(frameno);

TH2F *_histo = new TH2F("OutH",title.c_str(),w,0,w*scale,h,0,h*scale);

inc=0;
_histo->Reset("");
for (int j = 0; j <h; j++)
    for (int i = 0; i <w ; i++)
        t= 0xff&buffer[inc+8];
        inc++;

        _histo->Fill((i+0.5)*scale,(j+0.5)*scale,t);
        Matrix[i][j]=t;



_histo->GetXaxis()->SetTitle("mm");
_histo->GetYaxis()->SetTitle("mm");

/* TLatex *comment;
comment =new TLatex;

comment->SetTextAlign(11);
comment->SetTextSize(0.05);
comment->SetTextColor(9);
comment->DrawLatex(0.6,0.87,"");*/
```

```
//c1->Update();

cout << endl;
delete[] buffer;

//c1->Update();
return 0;



const char *i2s(int num)
char buf[20];
sprintf(buf,"
string b=buf;
return b.c_str();



const char *f2s(float num)
char buf[20];
if (num >1000) sprintf(buf,"else if (num <=0) sprintf(buf,"n.a.",num);
else sprintf(buf,"
string b=buf;
return b.c_str();



bool NewPalette(int Levels, Double_t *Red,Double_t *Green, Double_t *Blue, Double_t
*Stops)
int halfLevels=(int) (Levels/2);
for (int i=0;i<halfLevels;i++)
        Red[i]=1- ( (Double_t) i)/halfLevels;
        Green[i]=1-( (Double_t) i)/halfLevels;
        Blue[i]=1;
        Stops[i]=((Double_t) i)/Levels;
```

```
for (int i=halfLevels;
i<Levels;i++)
        Red[i]=0;
        Green[i]=0;
        Blue[i]=1- ( (Double_t) (i-halfLevels))/halfLevels;
        Stops[i]=(((Double_t) i)/Levels);

return 0;
```

Position Calculation

#include <TRandom.h>

#include <fstream.h>

#include <iostream.h>

#include <TFile.h>

#include <stdlib.h>

#include <stdio.h>

#include "math.h"

#include <TH2.h>

#include <TProfile2D.h>

#include <fstream>

#include <iostream>

#include <string>

#include <TROOT.h>

#include <StatAn.cxx>

float ABS(float);

int IntegralRowP (int y, int x);

int IntegralColumnN (int x,int y);

int IntegralColumnS (int x,int y);

int IntegrateQ(int x,int y, int QuarterNum);

float ABS (float x);

int ReadImage(char filename[]);

bool NewPalette(int Levels, Double_t *Red,Double_t *Green, Double_t *Blue, Double_t *Stops);

float min(float fParam1,float fParam2);

const char *f2s(float num);

const char *i2s(int num);

using namespace std;

int StopPiedestal = 807; // Last Channel without signal. All Channels 0...20471 int NPulses = 700; // Number of Laser pulses1 const int Cross = 6; // Number of pixels from

center to the edge of gap

int binQx,binQy; // Horizontal and vertical Bin quantity. Each bin is a pixel of camera. This var-les are worked out with external function ReadImage, that is why they are declared as globals

float Endx, Endy, Scale,Startx,Starty; // Range of x and y in milimeters from 0 to end—x/y— and miliimeters->scale*pixels

//TCanvas *c1 = new TCanvas("c1","Hello!",1);

int shift = 6; // Shift while searching for stable solution

char FileName[20], OFile[20],ResultFile[20], Answer;

int Matrix[768][574];

float SumM=0, Garbage;

float q1A,q2A,q3A,q4A; // Signals from QD in consequent process of position searching

float q1,q2,q3,q4, SignalSum, pmt; // Simulated division by QD

float SI, SII; // Vertical and horizontal sum of qiA

//int x1,y1,x2,y2,x3,y3,x4,y4;

float ds1,ds2,ds3,ds4,Stran;

float dSdown = 0, dSup = 0; // Variables for defining the most presice position

float qdSignal12, qdSignal14;

int op,on; // variables for defining the direction of where to shift virtualQD coordinates

int x,y; // For imitation of real position

int Sum; // Total Sum of values of all pixels

int hxbinN,hybinN; // Number of bins where the virtualQD center is lying

int MeanxBin, MeanyBin; // Position of Gravity center of the intensity

int i,j,k; // Counters

```
float pi = 3.141592654;
float HandVar,HandVar1; // assistant variable
double Meanx, Meany; // Position of Gravity center of the intensity distribution in mm
float QDPiedestal[4]; // Each channel has own piedestal, must be substracted from the signal. One must always Extract data from DAQ with one channel before laser to save this piedestal
string filename;
char charData[132000];
float arg[6];
float val[4];
float ar1[2048];
float ar2[2048];
float ar3[2048];
float ar4[2048];
float ar5[2048];
float Sig12[768][574],Sig14[768][574],Sig12m[768][574],Sig14m[768][574];
int pr1,pr2;
int xmin,xmax,ymin,ymax;

float MinValy, MinValx;
int Minx, Miny, MarkerWhile=1, Train=0;

//100100 100100 100100

int main(int argc, char** argv){

char* File;
char ConfFile[20];
char Buff[60];
File = argv[1];

//!!!!!!!!!!!!ReadOut Area of Interest!!!!!!!!!!!!!!!!!
```

```
ifstream interest("interest.ing");
if (!interest.is_open()) cerr << "There is no file interest.ing or it is corrupted" << endl;
interest >> xmin >> xmax >> ymin >> ymax;
interest.close();

//cout<< "Enter file name with raw data. It must be output file of DaqBrowser! Linux-
Time,Ch3,Ch2,Ch1,Ch4" << endl;
//cin >> File;
//while (MarkerWhile){
//cout << "Enter a number of the last empty channel (0..2047)" << endl;
//cin >> StopPiedestal;
//cout << "Enter a number of pulses" << endl;
//cin >> NPulses;
//cout << endl;
//cout << "Last empty channel is " << StopPiedestal <<" and Number of Pulses is "
<< NPulses << endl;
//cout << "Is it ok? (y/n): y - It's fine/ n - redefine" << endl;
//cin >> Answer;
//if (Answer == 'y') MarkerWhile = 0;
//};

//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
ifstream fp(File);
    if (!fp.is_open()) cerr << "Sorry, can't open source file " << endl;
    cout << "Creation of ChannelToChannel " <<fp.good()<< endl;
    fp.getline(charData, 200);
    fp.getline(charData, 220);
    fp.getline(charData, 220);
    fp.getline(charData, 220);
    fp.getline(charData, 132000);
    sprintf(FileName, "Channel_     ofstream fo(FileName);
    while (fp.good()){
    fp >> ar1[0];
```

```
Train++;
cout << ”Number of Trains = ” << Train <<endl;
for (i=0; i<2048; i++) fp >> ar1[i]; //Q1    for (i=0; i<2048; i++) fp >> ar2[i];
//Q2    for (i=0; i<2048; i++) fp >> ar3[i]; //Q3    for (i=0; i<2048; i++) fp >>
ar4[i]; //Q4
for(int i=0; i<2048; i++) fo << ar1[i] << ” ” << ar2[i] << ” ”<< ar3[i]<< ” ”<<
ar4[i] << endl;
    };
ifstream conf(”config.ing”);
sprintf(ConfFile, ”Config_    ofstream conffile(ConfFile);
for (i = 0; i < 7; i++ ){
conf.getline(Buff,60);
conffile << Buff << endl;};
conffile << endl;
conffile << ”Number of trains = ”<< endl;
conffile << Train;
conffile.close();

conf.seekg(0);
conf.getline(Buff,60);
conf.getline(Buff,60);
conf.getline(Buff,60);
conf >> NPulses;
conf.getline(Buff,60);
conf.getline(Buff,60);
conf.getline(Buff,60);
conf >> StopPiedestal;
//cout << StopPiedestal << endl;
//cin >> i;

fp.close();
fo.close();
//!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!! //This block converts from DAQ
```

```
file to column->(physical channels) format file. 5 column: for each quater and PMT signal
ifstream qdsig(FileName);
ifstream ipn12("output12.txt");
ifstream ipn14("output14.txt");

sprintf(ResultFile,"RES_
ofstream res(ResultFile);
if (!ipn12.is_open()) cerr << "Sorry, can't open source file ipn12.txt" << endl;
if (!ipn12.is_open()) cerr << "Sorry, can't open source file ipn14.txt" << endl;
if (!qdsig.is_open()) cerr << "Sorry, can't open source file JQO.txt" << endl;
while (qdsig.good()){
for ( i = 0; i <= StopPiedestal; i++){
qdsig >> q1 >> q2 >> q3 >> q4;
QDPiedestal[0] += q1;
QDPiedestal[1] += q2;
QDPiedestal[2] += q3;
QDPiedestal[3] += q4;};
QDPiedestal[0] /= -(StopPiedestal+1.);
QDPiedestal[1] /= -(StopPiedestal+1.);
QDPiedestal[2] /= -(StopPiedestal+1.);
QDPiedestal[3] /= -(StopPiedestal+1.);
cout <<QDPiedestal[3]<<endl;

qdsig >> q1 >> q2 >> q3 >> q4;
cout << "q1=" << q1 << endl;
cout << "q2=" << q2 << endl;
cout << "q3=" << q3 << endl;
cout << "q4=" << q4 << endl;
q1 = -q1 - QDPiedestal[0];
q2 = -q2 - QDPiedestal[1];
q3 = (-q3 - QDPiedestal[2])*0.93;
q4 = -q4 - QDPiedestal[3];
cout << "q1=" << q1 << endl;
```

```
cout << "q2=" << q2 << endl;
cout << "q3=" << q3 << endl;
cout << "q4=" << q4 << endl;
SignalSum=q1+q2+q3+q4;
q1=q1/SignalSum;
q2=q2/SignalSum;
q3=q3/SignalSum;
q4=q4/SignalSum;
qdSignal12 = q1 + q2;
qdSignal14 = q1 + q4;
MinValy=1;
MinValx=1;
pr1 = 768 - 2 * Cross;
pr2 = 574 - 2 * Cross;
for( i = xmin + 1; i <= xmax; i++)
for( j = ymin + 1; j <= ymax; j++)
{
        ipn12 >> Sig12[i][j];
        ipn14 >> Sig14[i][j];
        Sig12m[i][j] = ABS(Sig12[i][j]-qdSignal12);
        Sig14m[i][j] = ABS(Sig14[i][j]-qdSignal14);
        //cout << FAr[i][j] << endl;
        if (Sig12m[i][j] < MinValy){
            MinValy = Sig12m[i][j];
            Miny=j;
                    };
        if (Sig14m[i][j] < MinValx){
            MinValx = Sig14m[i][j];
            Minx=i;
        };
};

res << Minx << " " << Miny << endl;
```

```
ipn12.close();
ipn14.close();

for (k = 1; k < NPulses; k++)
    {
        //cout << "i=" << i << endl;
        ifstream ipn12("output12.txt");
        ifstream ipn14("output14.txt");
        //cout << "qdsig.good()"<< qdsig.good() << endl;
        qdsig >> q1 >> q2 >> q3 >> q4;
        //cout << "qdsig.good()"<< qdsig.good() << endl;
        q1 = -q1 - QDPiedestal[0];
        q2 = -q2 - QDPiedestal[1];
        q3 = (-q3 - QDPiedestal[2])*0.93;
        q4 = -q4 - QDPiedestal[3];
        SignalSum=q1+q2+q3+q4;
        q1=q1/SignalSum;
        q2=q2/SignalSum;
        q3=q3/SignalSum;
        q4=q4/SignalSum;
        qdSignal12 = q1 + q2;
        qdSignal14 = q1 + q4;
        MinValy=1;
        MinValx=1;

        for( i = xmin + 1; i <= xmax; i++)
        for( j = ymin + 1; j <= ymax; j++)
        {

        Sig12m[i][j] = ABS(Sig12[i][j]-qdSignal12);
        Sig14m[i][j] = ABS(Sig14[i][j]-qdSignal14);

        if (Sig12m[i][j] < MinValy){
```

```
            MinValy = Sig12m[i][j];
            Miny=j;
            };
        if (Sig14m[i][j] < MinValx){
            MinValx = Sig14m[i][j];
            Minx=i;
            };
        };
        res << Minx << " " << Miny << endl;

        ipn12.close();
        ipn14.close();
};
for (i = 0; i <= 2047-(StopPiedestal+1)-NPulses; i++)
        { qdsig >> Garbage; // Ostal'noe v musor
        qdsig >> Garbage; // Ostal'noe v musor
        qdsig >> Garbage; // Ostal'noe v musor
        qdsig >> Garbage; // Ostal'noe v musor
        };
};

qdsig.close();
res.close();

return StatAn(File);
}



//Functions
int IntegrateQ(int x,int y, int QuarterNum) { //QuaterNum: 1 - x>0,y>0; 2 - x<0,y>0;
            //3 - x<0,y<0; 4 - x>0,y<0
            // Gap 12x12 pixels
```

```
        int i,j;
        int Integral = 0;

        switch (QuarterNum) {
        case 1:             for(i = x+Cross+1; i <= binQx-1; i++)
            for(j = y+Cross+1; j <= binQy-1; j++) Integral += Matrix[i][j];
            break;
        case 2:             for(i = 0; i < x-Cross; i++)
            for(j = y+Cross+1; j <= binQy-1; j++) Integral += Matrix[i][j];
            break;
        case 3:             for(i = 0; i < x-Cross; i++)
            for(j = 0; j < y-Cross; j++) Integral += Matrix[i][j];
            break;
        case 4:             for(i = x+Cross+1; i <= binQx-1; i++)
            for(j = 0; j < y-Cross; j++) Integral += Matrix[i][j];
            break;
        default:             cout << "Error while switch" << endl;
            break;};
        return Integral;
}

int IntegralRowP (int x, int y){

        int i;
        int integralRow = 0;

            for (i = 0; i < x - Cross; i++) integralRow += Matrix[i][y+Cross+1];
            for (i=x+Cross+1; i < binQx; i++) integralRow += Matrix[i][y+Cross+1];

        return integralRow;
}

int IntegralColumnS (int x,int y){
```

```
        int i;
        int integralColumn = 0;

            //for (i=0; i < y-Cross;i++) integralColumn += Matrix[x+Cross+1][i];
            for (i=y+Cross+1; i<=binQy-1;i++) integralColumn += Matrix[x+Cross+1][i];

        return integralColumn;
}

int IntegralColumnN (int x,int y){

        int i;
        int integralColumn = 0;

            for (i=0; i < y-Cross;i++) integralColumn += Matrix[x+Cross+1][i];
            //for (i=y+Cross+1; i<=binQy-1;i++) integralColumn += Matrix[x+Cross+1][i];

        return integralColumn;
}

float ABS (float x){

        if (x>0) return x;
        else return x=-x;}

float min(float fParam1,float fParam2) {

if(fParam1 < fParam2)          return fParam1;
else
        return fParam2;
}
```

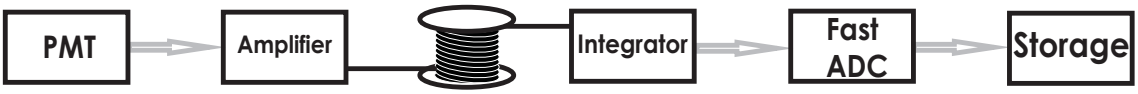

Figure 25: PMT Signal Line

C Measurement Instructions

D PMT

PMT is a tool intended for relative laser pulse intensity measurements. Full PMT line consists of the pre-amplifier (gain 14), a long transer line (tunnel-rackroom), Integrator (PMT power supply and control voltage supply are placed in the same crate module; internal amlification about 4) and a Fast ADC (14-bit, -5..+5 V or -1..+1 V input range). After these steps it is stored on a hard drive. PMT signal from the tunnel can be read out at the Rack14_PP1_CH2. Crate over VMEPITZ04 computer:

position 10 – Delay (DL5-integrator, DL1 - Fast ADC(PMT))

position19_20 – Integrator -¿ replaced to crate under the VMEPITZ04 computer.

position 11 – Fast ADC (1 channel)

D.1 PMT signals observation

Check whether all connections are done in the tunnel. Check, that power cable of the PMT is unplugged (TU_LBL_PM_01 RR_R14_PP_3C). Start laser system and set attenuation of 10% and 100-pulses train (just to see that signals exist). Go to the PITZ Control - Laser Beam Line - Swith the PMT reflector to quarz.

Factor between the mirror and the quartz is 1-2 % now (depends on alignment).

Choosing the appropriate control voltage the signal calibration curve should be used to cover the area of interesting transmission factors with linear part of the characteristic(Fig.26).

In the rackroom: set gain voltage to 0.45 V (or choose needed range from the calibration curve Fig. 26). Plug in PMT power supply cable (Number !!!).

To check whether the signal exists follow: PITZ Control - ADC Modules - Diag.ADC0 - CH00 TD (time domain) - Adjust the range to be able to resolve 5-10mV signal.

Attention! Be aware: Do not exceed 2% of transmission, if mirror is inserted.

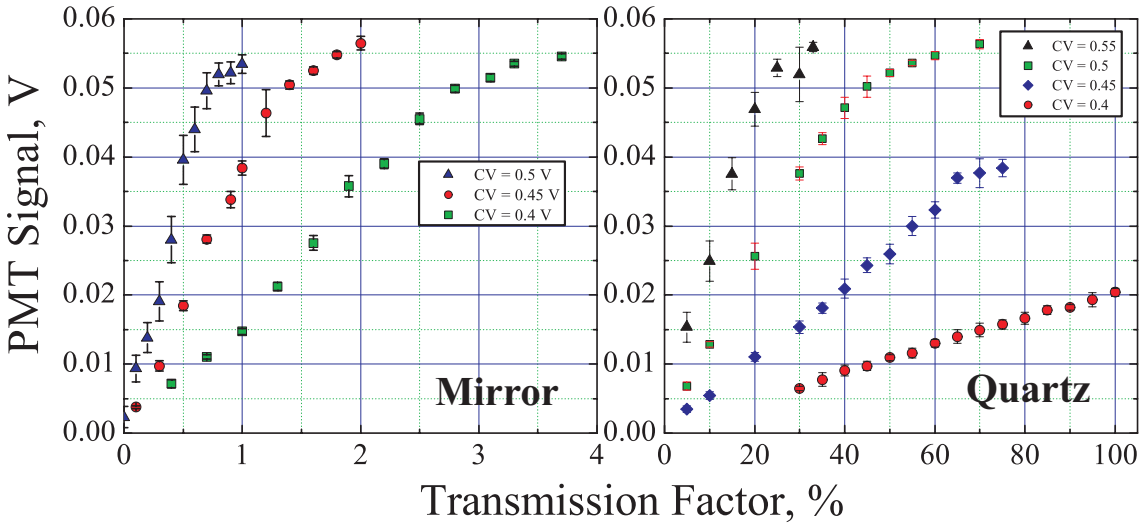

Figure 26: PMT Calibration curves for different control voltage

D.2 Extracting signals from the DAQ

To start extracting data from DAQ you shoud first of all make ssh (under the Linux system) to picus4 or other 64-bit computer.

Then type in daqbr in the following directory and execute:

/afs/ifh.de/group/pitz/data % daqbr

After DAQBR widow has opened, push ctrl+o to open extraction script InitPMT.c (InitQD.c): folder /afs/ifh.de/group/pitz/data/Ivanis/

You should change values only in time fields(otherwise consult with Galina Assova).

To produce output press ctrl+r, then give the destination folder and name of the file (it is not allowed to create new directories from save dialog, please create outside of daqbr if needed).

Extracted file will have following structure:

//name of extracted value 1

//name of extracted value 2 ...

//name of extracted value N

timestamp _0[0] ... _0[2047] _1[0] ... _(N-1)[2047]

123456789098 0.8674 0.8273

... ...

For quadrant diode N=4.

In case of loosing the file for quadrant diode to use standard analysis program one

should extract data in following consequence:

1. 3 channel 0-2047
2. 2 channel 0-2047
3. 1 channel 0-2047
4. 4 channel 0-2047

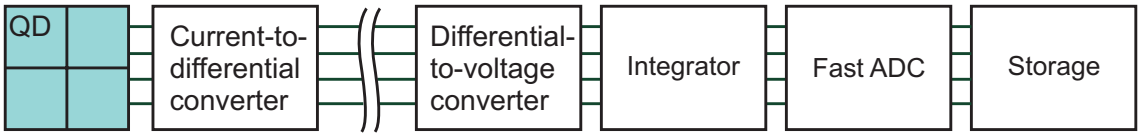

Figure 27: Quadrants signals transfer line

D.3 QD

QD is the instrument for laser beam pointing position measurements. Four signal lines of the quad diode consist of 4 channel current-to-differential converter, long transfer line (tunnel rackroom), 4-channel differential-to-voltage signal, integrator (4-channels, internal amplification), Fast ADC(4 channels, 14-bit -1..+1 V input range). Crate over VMEPITZ04 crate:

position 10 – Delay (DL3-integrator, DL4 - Fast ADC(PMT))

position 14 – Integrator

position 13 – Fast ADC (1 channel)

D.4 Observation of the signals

Check if everything is connected in the tunnel, and rackroom. Check that the higher rack is switched on and the QD cable on the backward side is plug in. Check all connections on the front panel of the crate.

Set the laser light transmission to 10% and use the mirror as the last reflector. Follow this to open the QD signals display windows:

PITZ Control – ADC Modules – Diag.ADC2 – first four channels belong to the QD.

If there are no signals it could be due to the bad alignment. One should try to find signals moving the position of the quad diode in regard to the laser beam.

To find geometrical center of the beam (to insure it is not cutted by the edges), one should find gravitational center (signals equality), then find a coordinate moving the actuator along one axis, where beam is on two quads and the spot edge goes between the quads as shown at Fig 28. After the distance between those two positions is obtains one should use the middle value as a center position. The same procedure should be done for the second axis.

Factor of attenuation between the mirror and quartz reflector is $\approx 6\%$ (depends on alignment).

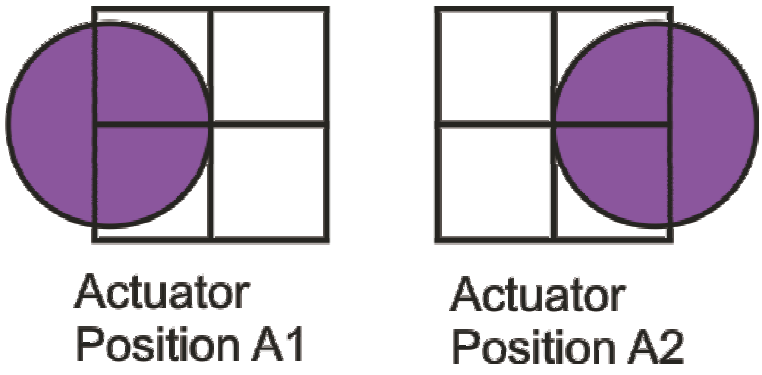

Figure 28: Geometric center alignment

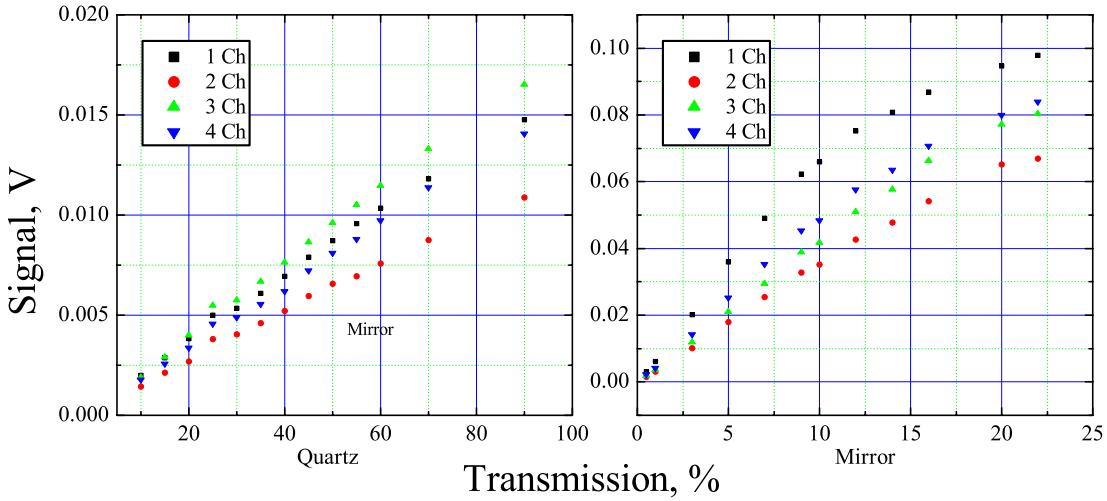

Figure 29: Calibration of Quadrant Diode Signal versus Transmition factor

Before starting a position measurement procedure one should capture single pulse transverse intensity distribution at VC2 and save it in the **imm** format. See futher instructions how to substract the background.

After the alignment is done one can start measurements.

D.5 DAQ data extraction

One should input time interval. See similar chapter for the PMT.

D.6 Data Analysis

There are few methods of data analysis. If the task consists of a lot of similar data files with the same statistics it is more convenient to use pc-farm evaluation. If each file has different parameters (Pulse number, statistics) then it is reasonable to use ROOT and separate scripts.

D.7 Timing

To obtain the correct value one should fully integrate the signal and digitize it at the intgrated signal maximum. There are always internal delays in the devices. One should find appropriate values by searching for maximums at oscilloscope (integrated value) or PITZ CONTROL/ADC modules/Diag.ADC0/CH00/TD (digitized value)

E Analysis Software

E.1 PMT signals processing script

This script takes DAQ output file with the signals and makes statistical analysis over all given data. To introduce the input parameters, number of pulses in the train and number of prelast piedestal channel before the trainshould be inserted directly in the script in the user defined values section.

As output a file will be generated, which consists of two columns: average output signal for each pulse and standard deviation of each signal variation distribution.

Copy the script Hist.C (/afs/ifh.de/group/pitz/data/Ivanis/PMT/Measurement/28_01_2006/) to the folder containing the data and run it with ROOT:

root

.x Hist.C

Then it will ask for the input file name.

E.2 QD signals processing scripts

E.2.1 Profile Image Preparation

To cut background from the *.imm file, a VB_Analyser.cxx script can be used.

Run ROOT at the folder containing File.imm and File.bkg. Load the script.

.L VB_Analyser.cxx

CutBck("File.imm","File.bkg", 10)

A file will be generated Cut_File.imm - this is a cut picture, which can be used for position analysis.

10 is a threshold value below wich all pixels values are set zero.

The script was written by Vittorio Boccone and revised a bit by Yevgeniy Ivanisenko. All further questions You can direct to Ye.Ivanisenko (e-mail jayq@mail.ru).

E.2.2 2D Analysis

Profile Analyser 2D Input file is interest.ing (should have strict structure, that is performed in the example): There one can define area of interest to reduce time needed for analysis. There can problems occur, if area is choosed incorrectly.

Input argument *.imm (transverse intensity distribution) image: ProfileAnalyser 40VC2.imm

ProfileAnalyser converts imm picture to a couple of matrices, which then are used for position definition:

output12.txt

output14.txt

Source code is in file SKrestomFileRe.cxx

All the files are at /afs/ifh.de/group/pitz/data/Ivanis/Products/1/, or at the CD-ROM.

Position analysis Input data:

Before using the program, please, fill in next parameters in the file. Change only the following values !!!:

config.ing - Number of Pulses, Last Piedestal Channel Number before the signal;

. (for example File.txt) - DAQ output file with QDInit.C. To start analysis: Position File.txt;

Output:

Config_File.txt - Number of Pulses, Last Piedestal, Number of trains for the file;

RES_File.txt - Position for every pulse of each train;

AnRes_File.txt - Averaged positions for each pulse over all trains with standard deviation;

Parallel Run at PITZ Kluster for multi-file analysis:

1. Edit RunAn.C : one should change to an appropriate number of tasks (file to analyze). Save.

2. Edit Analysis.C : Each case corresponds to a new file, that must be analyzed. Change the names and activate sufficient amount of cases. If number of tasks in RunAn.C is bigger, than number of cases - there will occur a lot of tasks involving last file, what leads to an unpredictable results.

Change the name of the directory with data files in the text of the script (destination folder) and place there RunAn.C Analysis.C

Make ssh to satyr46, nereide16 or kili; Open the folder; run root; type in and run: .x RunAn.C

then quit ROOT. Type in the command line: qstat -u YourUsername, to see whether tasks are submitted for execution.

All the files are in /afs/ifh.de/group/pitz/data/Ivanis/Products/ComprAn/1/ or on the CD-ROM.

E.2.3 1D Analysis

Profile Analyser 2D /afs/ifh.de/group/pitz/data/Ivanis/Kluster/QD/

Everything is the same as for the 2D, but name is ProfileAnalyser1D.

Output: output1D12.txt output1D14.txt

Source code SKrestomFileCreate.cxx

Position Analyser 1D Source Total.cxx (1D Analysis) Y. Ivanisenko 2007 /afs/ifh.de/group/pitz/data/Iv

Input data: Before using the program, please, fill in next parameters in the file.
Change only the following values !!!:
config.ing - Number of Pulses, Last Piedestal Channel Number before the signal;
interest.ing - matrices indeces of the output1D12.txt output1D14.txt

. (for example File.txt) - DAQ output file with QDInit.C. To start analysis: Position File.txt;

Output:

Config_File.txt - Number of Pulses, Last Piedestal, Number of trains for the file;

Res_File.txt - Position for every pulse of each train;

Analysed_File.txt - Averaged positions for each pulse over all trains with standard deviation;

Sig_File.txt - Contains signals for each pulse of all the train

Pied_File.txt - Contains signals of the piedestal

SigAn_File.txt - Statistics result average pulse signals over all trains.

Parallel Run at PITZ Kluster for multi-file analysis:

1. Edit RunKluster1D.C : one should change to an appropriate number of tasks (file to analyze). Save.

2. Edit Analysis.C : Each case corresponds to a new file, that must be analyzed. Change the names and activate sufficient amount of cases. If number of tasks in RunAn.C is bigger, than number of cases - there will occur a lot of tasks involving last file, what leads to an unpredictable results.

Change the name of the directory with data files in the text of the script (destination folder) and place there RunAn.C Analysis.C

Make ssh to satyr46, nereide16 or kili; Open the folder; run root; type in and run: .x RunKluster1D.C

then quit ROOT. Type in the command line: qstat -u YourUsername, to see whether tasks are submitted for execution.

qdel ProcessNumber - command for killing the job.

Acknowledgements

First of all I would like to thank Dr. Sergiy Khodyachykh for the invitation to come to Zeuthen and a great assistance in all accompanying tasks.

Special thanks I devote to Dr. Frank Stephan, who gave me this opportunity to advance my professional skills and accomplish my diploma thesis at PITZ.

This work became possible because of efforts of Oleg Kalekin, who carried out the idea of creating the system and organized components design and production.

I would like to express my gratitude to Dr. Sergey Korepanov, who coordinated my diploma project and contributed much with ideas of developing the system and the software.

Dr. Juergen Baehr spent a lot of time organizing and supervising my experimental work and I appreciate it much.

I wish to thank following individuals for the fruitful discussions of the subjects: Sergey Khodyachykh, Mikhail Krasilnikov, Sven Lederer, Anna Oppelt, Lazar Staykov, Frank Stephan.

I am very grateful for the help of Galina Asova and Bagrat Petrosyan in a field of data acquisition and Stephan Weisse for the help concerning camera tasks.

Thanks to engineers involved in hardware realization: Hans Henschel, Hartmut Luedecke, Mario Pohl, Hans Scholz, Frank Tonisch.

I could hardly imagine success without a friendly climate of the group. I had a great experience of well coordinated team work, and that is why I adress my respect to all of you, who has contributed to the project.

I would like to sincerely thank all of my university teachers, who did their best to deliver knowledge to me.

Finally there were a lot of support and encouragement from my family, which I should mention here to complete the picture of contributions.