Appendix A - Coincidence.vi

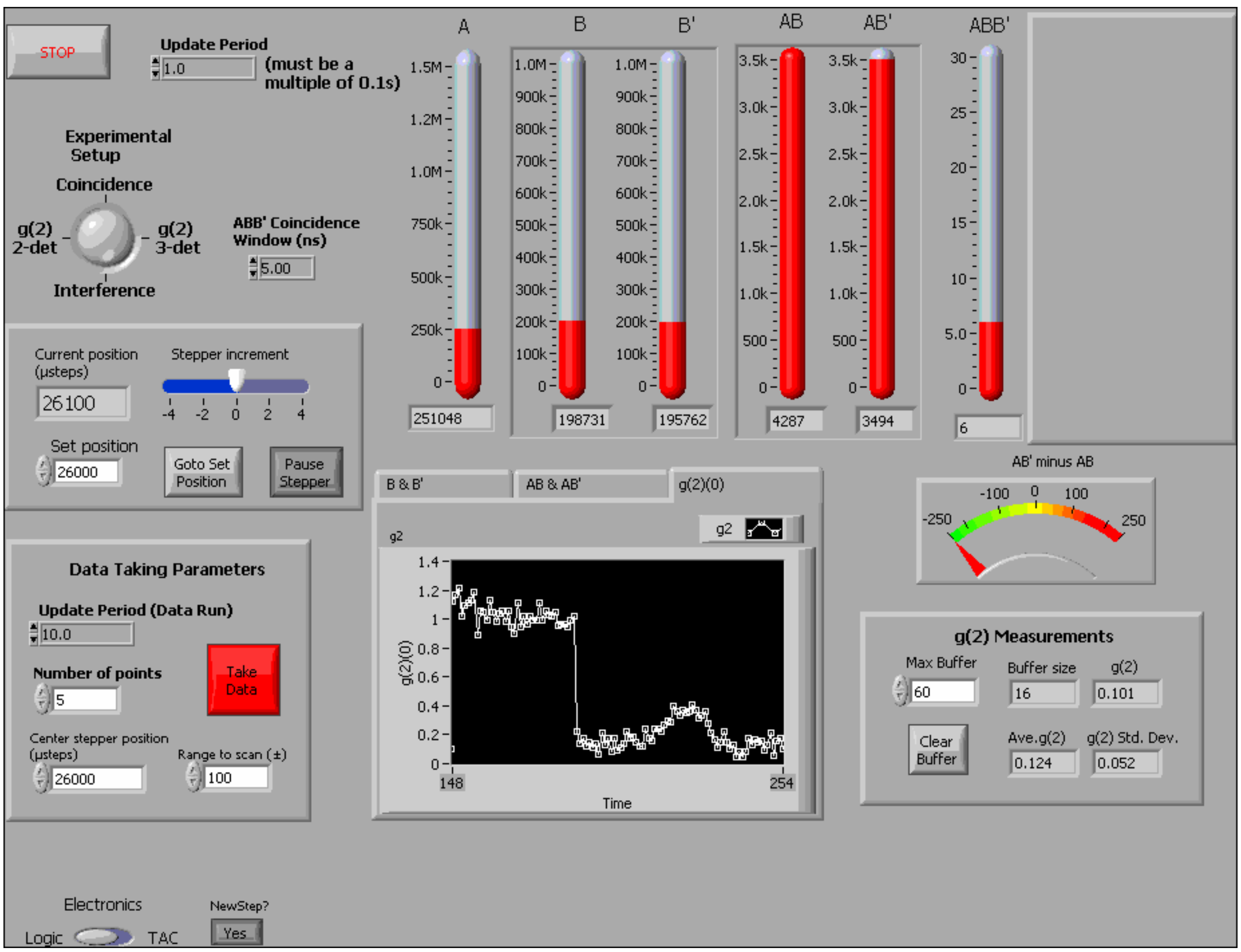
This appendix describes the LabView vi used to acquire data for several experiments: coincidence measurements, proving the existence of photons, and single photon interference.

I. Starting Out

The appearance of this vi will change depending on the experiment you are doing (as determined by the **Experimental Setup** knob on the vi.) The experiment with the most controls and indicators is the interference experiment, and the front panel for this experiment is shown below. If you are doing a different experiment, some of the objects may not be visible.

Coincidence.vi

Front Panel



Description

This is the program you use to run one of 4 different experiments:

1) Simply measuring coincidences between two beams.

2) Measuring g(2)(0) for 2-beams using 3-detectors.

3) Measuring single photon interference patterns.

4) Measuring g(2)(0) for 1-beam using 2-detectors.

Note that g(2) is calculated differently depending on whether you're making 3-detector or 2-detector measurements. The calculation also depends on whether you're using the logic circuit (default) or TAC's to do the coincidence detection. See details in the description of the "g(2) Measurements" parameter.

The program does not record data to a file right away, but displays the counters in real time so that you can align things.

Once everything is aligned and the parameters are set, you press the "Take Data" button. This transfers control to another program which records a data set and saves it to a file.

Uses a National Instruments PCI-6602 and reads counters 0 thru 7. The device number for this board should be set to 2. Input to this board is via the Whitman/Trinity coincidence circuit (or via a BNC-2121 connector block for TAC measurements).

Note that there are two controls: "Electronics" and "NewStep?" that tell the computer what hardware you have present in your experiment. These controls are not visible normally. They are located below the "Data Taking Parameters", so you'll need to scroll the window down to access them. These two parameters are only read once at the beginning--so they must be properly set BEFORE you run the program.

The “Electronics” control determines which signals are present on which counters, and also determines how g(2) is calculated (more details below). By default “Electronics” is set to “Logic” for the coincidence circuit.

The “NewStep?” control tells the computer whether or not you have a NewStep stepper motor attached to your computer to control phase adjustments of the interferometer. This control is useful so that the computer won't try to communicate with an instrument you don't have, causing a hang-up. By default “NewStep?” is set to "No"--if you do have a NewStep, you'll want to change the default to "Yes". For more info see the description of this control below.

If using the logic circuit the clock should be set to 10Hz for proper gating of the counters. If you don’t use the Whitman/Trinity circuit you will need to have a 0-5V, 10Hz square wave in order to gate the counters. Connect it to the Gate 0 (PFI 38) connector on the PCI-6602 (via the BNC-2121).

For interference measurements, a Newport NewStep actuator (inexpensive stepper motor) is used to tilt a beam displacing polarizer to adjust the phase of a polarization interferometer. This actuator is controlled by a NewStep controller, connected to the serial port of the computer with an RS-485 to RS-232 converter. If you do not have this controller, set the "NewStep?" parameter to "No".

The inputs connect to the sources (SRCn) for the individual counters. This is automatically accomplished with the Whitman/Trinity circuit.

Electrical connections for counters:

When using Logic: (there are automatically made with the Whitman/Trinity circuit)

A (Counter 0): PFI 39

B (Counter 1): PFI 35

B' (Counter 3): PFI 27

AB (Counter 4): PFI 23

AB' (Counter 6): PFI 15

ABB' (Counter 5): PFI 19

BB' (Counter 7): PFI 11

Unused A’ (Counter 2): PFI 31

When using TACs:

A (Counter 0): PFI 39

B (Counter 1): PFI 35

B' (Counter 3): PFI 27

AB (Counter 4): PFI 23

AB' (Counter 6): PFI 15

ABB' (or just BB' for 2-det measurement) (Counter 5): PFI 19

A Valid Start (from back of AB TAC) (Counter 2): PFI 31

B Valid Start (from back of ABB' TAC) (Counter 7): PFI 11s

Note that for the coincidence experiment, only 3 counters are needed: A, B, and AB. For the other 3 experiments, 7 counters are needed. When making TAC measurements the A & B Valid Start measurements are used to determine the normalization for g(2), not the raw A & B counts.

Note that if you’re using the BNC-2121 connector block, the A, B, and B' outputs should be terminated on the block into 50 ohms. The coincidence and Valid Start outputs are terminated straight into the BNC-2121.

On startup, the program initializes the counters.

After initialization the program simply loops and displays the counts in a given time window (determined by the "Update Period" control in the upper left.) This is useful for tweaking the alignment and adjusting parameters.

Nothing is written to disk until the parameters are chosen and the "Take Data" button is pressed. This loads a second VI that records and saves data to disk. Parameters for this data acquisition phase are set in the "Data Taking Parameters" box.

When performing an interference experiment, the stepper motor will automatically adjust the tilt of the beam displacing prism, changing the phase.

Help for each of the controls and indicators can be obtained from the Contextual Help window <cntrl - H> by mousing over each control or indicator. Full documentation for each control and indicator can be obtained by printing using: File>Print>Custom, and then checking "All controls" and "Descriptions".

Controls and Indicators

Coincidence Measurements

Experimental Setup

Which measurement to perform.

Coincidence: Measures A, B, and AB counts.

g(2) 3-det: Make g(2) measurements on 2 correlated beams using 3-detectors. This is the single photon measurement.

Interference: interference measurement [with simultaneous g(2) 3-detector measurement]

g(2) 2-det: Make g(2) measurements on a single beam using 2-detectors. This is the classical field measurement.

Stop

Use this to stop. If you stop some other way you'll probably need to quit LabView and restart; you may even need to reboot the computer.

Electronics

Which coincidence electronics are being used.

This is important because different electronics have different signals connected to the various counters. It also influences the formulas that are used to calculate g(2).

Update Period

[Must be a multiple of 0.1s]

Time window (in s) for the counters during setup phase (i.e., before the "Take Data" button is pressed.) Readings update once each time window if "Status" reads "Reading Counters".

Counts A, B

Singles counts in the time window specified by “Update Period” (upper left)

Counts AB

Coincidence counts in the time window specified by “Update Period” (upper left)

AB Plot

Chart displaying history of AB coincidence counts

Update Period (Data Run) [Data Taking Parameter]

[Must be a multiple of 0.1s]

Time window (in s) for counters during data acquisition.

This applies after the "Take Data" button has been pressed.

Number of points [Data Taking Parameter]

Number of measurements that are made during data acquisition.

Error measurements are essentially useless if this is less than 5. 10 is a better minimum number. For interference measurements, you will want to use many more than this so you can better see the pattern.

Take Data [Data Taking Parameter]

Leave the setup "tweaking" mode and switch to data acquisition mode.

Func Gen?

Control function generator w/ GPIB?

If No, no GPIB commands are sent. Useful so that the program won't hang if you don't have a GPIB controlled function generator. However, in this case you still need to have a 0-5V, 10Hz square wave input to PFI38 (Gate 0) to use as a clock.

If Yes, the vi uses GPIB to sets the function generator to output a 0-5 V, 10Hz square wave to use as the clock. I still recommend using the SYNC output from the generator--then only the frequency matters.

This parameter is only read once, when the vi first starts to run. Therefore, it needs to be set BEFORE you run the program.

NewStep?

Is the NewStep controller present?

If No, there is no attempt to communicate with the NewStep controller. Useful if you don't have a NewStep controller. In this case, the "Interference" experiment is unavailable. If you set "Experimental Setup" to "Interference", the program will act as though it's set to "g(2) 3-det".

If Yes, the NewStep functions normally.

This parameter is only read once, when the vi first starts to run. Therefore, it needs to be set BEFORE you run the program.

3-Detector g(2) Measurements

All of the above parameters apply, plus the following:

ABB' Coincidence window

Coincidence window for ABB' measurements.

Used to determine expected g(2) for 3-detector (single-photon) measurements. Basically, this parameter determines the number of expected accidental ABB' counts.

Counts A, B, B'

Singles counts in the time window specified by "Update Period" (upper left)

Counts AB, AB'

Coincidence counts in the time window specified by "Update Period" (upper left)

Counts ABB'

Three-fold coincidence counts in the time window specified by "Update Period" (upper left)

Counts A VS

A Valid starts.

Only used for TAC measurements. Comes from valid start output of AB TAC, and is connected to Counter 2.

This is useful for measuring how many starts get triggered from the A detector. For A rates of about 100k/S or less, the Avs will be slightly less than A (within about 10%). For larger A rates, this can be significantly less than A. Really should keep this rate so that Avs is within 10% of A.

Counts B VS

B Valid starts.

Only used for TAC measurements. Comes from valid start output of ABB' or BB' TAC, and is connected to Counter 7.

On 3-detector g(2) measurements, when measuring three-fold ABB', this is useful for determining whether the start gate is functioning properly. If the start gate delay is set right, then Bvs should be greater than the AB coincidences.

On 2-detector g(2) measurements, when measuring two-fold BB', this is useful for measuring how many starts get triggered from the B detector. For B rates of about 100k/S or less, the Bvs will be slightly less than B (within about 10%). For larger B rates, this can be significantly less than B. Really should keep this rate so that Bvs is within 10% of B.

AB' minus AB

Displays the difference in counts between the AB' and AB counters. Useful for getting the beamsplitter to split 50/50 by balancing the these counts.

g(2) Measurements

The program calculates g(2) for every loop--and this value is displayed as **g(2)**.

Calculated differently depending on which experiment you're doing.

Formulas:

T: update period (counting window, in S)

dt: BB' coincidence window (in nS)

3-detector, 2-beam experiment:

g(2)=(Avs^2*ABB')/(A*AB*AB') TAC

g(2)=(A*ABB')/(AB*AB') Logic

2-detector, 1-beam experiment:

g(2)=(BB'/(Bvs*B'))*(T/(dt*1.0e-09)) TAC

g(2)=(BB'/(B*B'))*(T/(dt*1.0e-09)) Logic

These g(2) measurements are then stored in an array, which contains the most recent measurements.

g(2) Ave. and **g(2) Std. Dev** are the mean and standard deviation of these stored measurements.

Max Buffer is the maximum size of this array (i.e., the largest number of measurements that will be averaged).

Buffer size is the current length of the array that is being averaged.

Clear Buffer clears out the array, and new measurements begin repopulating it.

Charts

Click on the tabs to display running plots of different measurements: **B & B'**, **AB & AB'**, or **g(2)**.

Interference Measurements

Interference measurements assume a 3-detector setup for g(2) calculations.

All of the above parameters apply, plus the following:

Current position (µsteps)

Current position of the stepper motor.

26000 is about the center of the range with the mount we are using.

Stepper increment

Amount that the stepper increments on each loop if the "Pause Stepper" button is off.

Set position

Force the stepper motor to go to this position by pushing the "Goto Set Position" button.

26000 is about the center of the range with the mount we are using.

Go To Set Position

Forces the stepper motor to go to the "Set position".

Pause stepper

If this is in, the stepper motor is stopped. If it is out, the stepper increments by "Stepper increment" on each loop.

Center stepper position (µsteps) [Data Taking Parameter]

Center of the range for the stepper during data acquisition.

Range to scan (±) [Data Taking Parameter]

Range for the stepper during data acquisition.

At each point of data acquisition, the stepper moves evenly between ("Center stepper position" - "Range to scan") and ("Center stepper position" + "Range to scan")

2-Detector g(2) Measurements

All of the above parameters apply, plus the following:

BB' Coincidence window

Coincidence window for BB' measurements.

Needed to determine g(2) for 2-detector (classical) measurements.

2D window stdev

Standard deviation of coincidence window for BB' measurements.

Needed to determine the error in g(2) for 2-detector (classical) measurements.

BB' BB' coincidences.

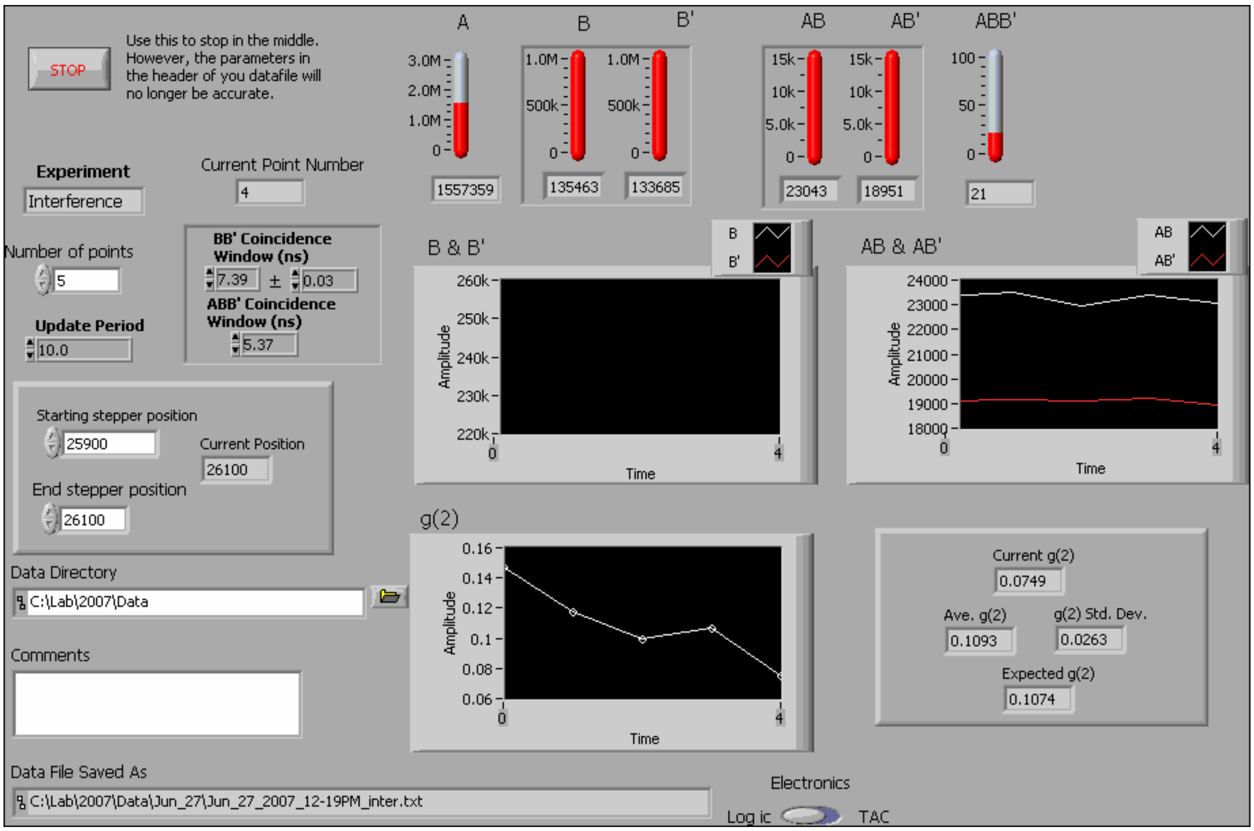
Only used for 2-detector g(2) measurements. Comes from counter 5 for TAC measurements, and counter 7 for Logic measurements.

II. Taking Data

When you move to data taking mode, the program you see will look something like that shown below. Again, the appearance of this vi will change depending on the experiment you are doing.

Coincidence Recorder.vi

Front Panel



Description

This VI should ONLY be called from the "Coincidence" VI--necessary parameters are set there.

This program records data (using parameters set in the "Coincidence.vi") and saves the data to a file.

For a more detailed description of the parameters, see "Coincidence.vi" .

Controls and Indicators

Stop

Use this to stop in the middle of a data run. However, the parameters in the header of you datafile will no longer be accurate (e.g., you won't have as many points as the header says.)

Also, if you stop some other way you'll probably need to quit LabView and restart; you may even need to reboot the computer.

Experiment

Displays which experiment is being performed: Coincidence, g(2), or Interference.

Number of points

Number of measurements that are made during data acquisition.

Current Point Number

The data point that the computer is currently acquiring.

ABB' Coincidence window

Coincidence window for ABB' measurements.

Used to determine expected g(2) for 3-detector (single-photon) measurements. Basically, this parameter determines the number of expected accidental ABB' counts.

BB' Coincidence window

Coincidence window for BB' measurements.

Needed to determine g(2) for 2-detector (classical) measurements.

BB' 2D window stdev

Standard deviation of coincidence window for BB' measurements.

Needed to determine the error in g(2) for 2-detector (classical) measurements.

Update Period

Time window (in s) for counters during data acquisition.

Data Directory

Path to directory where the data will be saved.

Comments

A space where you can enter comments about the data run. These get saved in the header of the data file.

Once you've entered the text, note that you must hit "Enter" on the NUMERIC KEYPAD, not the "Enter" (or "Return") on the regular keyboard for this text to be saved.

Data File Saved As

Path to the data file. The data file is automatically named using the date and time.

Counts A, B, B'

Singles counts in the time window specified by "Update Period" (upper left)

Counts A VS, B VS

Valid Start counts (only used with TAC)

Counts AB, AB'

Coincidence counts in the time window specified by "Update Period" (upper left)

Counts ABB'

Three-fold coincidence counts in the time window specified by "Update Period" (upper left)

Graphs

Displays graphs of B and AB [Coincidence measurement], or **B & B'**, **AB & AB'**, and **g(2)**. [g(2) and Interference measurement.]

Ave. AB [Coincidence measurement only]

Average value of AB coincidences. Updated when program ends.

AB Std. Dev. [Coincidence measurement only]

Standard deviation of AB coincidences. Updated when program ends.

---

g(2) Measurements (both 2-Detector and 3-Detector)

All of the above parameters apply (except "Ave. AB" and "AB Std. Dev."), plus the following:

Current g(2)

g(2) measurement for the current data point.

DBL **Ave. g(2)**

Average of the g(2) measurements.

DBL **g(2) Std. Dev.**

For 3-detector measurements this is the standard Deviation of the g(2) measurements.

For 2-detector measurements during data acquisition it is the standard deviation of the g(2) measurements. At the end this error is added in quadrature with the error in g(2) due to the uncertainty of the BB' coincidence time window ("BB' 2D window stdev").

DBL **Expected g(2)**

Calculate the expected value of the parameter g^(2) (0). Based on number of expected accidental coincidences.

Calculated differently depending on which experiment you're doing, and which electronics you're using.

For 2-detector, 1-beam measurements, this is assumed to be 1.0 (a classical beam).

For all other measurements, when the vi first starts running it is set to 0.0, and is updated with the expected g(2) at the very end.

Formula (from Appendix A of Thorn AJP)

T: update period (counting window, in S)

dt: ABB' coincidence window (in nS)

g(2)=(dt*1.0e-09/T)*Avs*((B/AB)+(B'/AB')) TAC

g(2)=(dt*1.0e-09/T)*A*((B/AB)+(B'/AB')) Logic

Interference Measurements

Interference measurements assume a 3-detector setup for g(2) calculations.

All of the above parameters apply (except "Ave. AB" and "AB Std. Dev."), plus the following:

U32 **Starting stepper position**

Position of stepper motor at start of scan.

U32 **End stepper position**

Position of stepper motor at end of scan.

Over the course of a data run, the motor steps evenly between the starting and ending positions.

I32 **Current Position**

Position of stepper motor for the current data point.

Appendix B - Hardy-Bell.vi

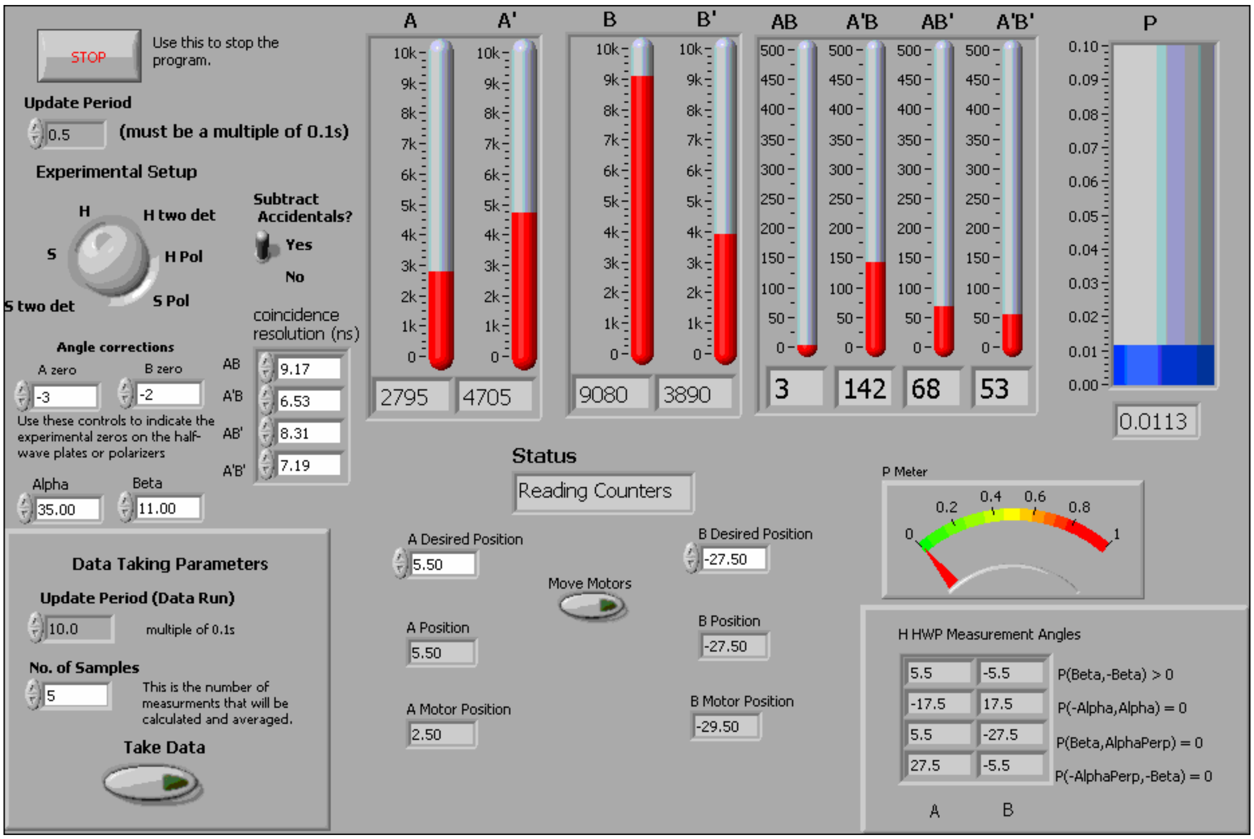
This appendix describes the LabView vi used to acquire data for Hardy's test of local realism.

I. Starting Out

When you first run the vi, the program that you see is:

Hardy-Bell.vi

Front Panel



Description

This is the program you see first when you start to run the experiment. It can be used for either Hardy-type or Bell-type measurements. It does not record data to a file right away, but displays the counters in real time so that you can align things. It also gives you interactive control over such things as the waveplate angles.

Once everything is aligned and the parameters are set, you press the "Take Data" button. This transfers control to another program which records a data set and saves it to a file.

Uses a National Instruments PCI-6602 and reads counters 0 thru 7. The device number for this board is should be set to 2. Input to this board is via the Whitman/Trinity coincidence circuit. Alternatively you could use TAC's for coincidences, and a BNC-2121 connector block (note that the detector outputs must be terminated into 50 ohms at the connector block, while the TAC outputs do not need this termination.)

Counters are gated by the clock in the coincidence circuit. For proper gating the internal clock should be set to 10 Hz. Alternatively the gate could come from an external source. This gate should be 0-5V at 10Hz. The gate pulse goes to the Gate 0 (PFI 38) connector on the PCI-6602.

Waveplates (or polarizers) are controlled by a Newport ESP300 controller at GPIB address 1. On program start, during initialization, the motors home themselves. The motors then set the waveplates to 0 (using the corrections in "A zero" and "B zero").

The inputs connect to the sources (SRCn) for the individual counters.

Electrical connections for counters: (automatically connected with the Whitman/Trinity circuit)

A (Counter 0): PFI 39

B (Counter 1): PFI 35

A' (Counter 2): PFI 31

B' (Counter 3): PFI 27

AB (Counter 4): PFI 23

A'B (Counter 5): PFI 19

AB' (Counter 6): PFI 15

A'B' (Counter 7): PFI 11

On startup, the program initializes the motors and the counters. During this time the "Status" indicator reads "Initializing".

After initialization the program simply loops and displays the counts in a given time window (determined by the "update Period" control in the upper left.) Status reads "Reading Counters". This is useful for tweaking the alignment and adjusting parameters. Waveplates (or polarizers) are moved by first setting the desired waveplate angles in the " A(B) Desired Position" controls, and then pressing the "Move Motors" button. Status changes to "Moving Motors" while the motors are in motion.

Nothing is written to disk until the parameters are chosen and the "Take Data" button is pressed. This loads a second VI that records and saves data to disk. Parameters for this data acquisition phase are set in the "Data Taking Parameters" box.

During data acquisition the motors will automatically set the waveplates to the correct angles. Angles for a Bell measurement are fixed. Angles for a Hardy measurement are determined from the controls "Alpha" and "Beta".

Help for each of the controls and indicators can be obtained from the Contextual Help window <cntrl - H> by mousing over each control or indicator. Full documentation for each control and indicator can be obtained by printing using: File>Print>Custom, and then checking "All controls" and "Descriptions".

Controls and Indicators

TF **Stop**

Use this to stop. If you stop some other way you'll probably need to quit Labview and restart; you may even need to reboot the computer.

DBL **Update Period**

[Must be a multiple of 0.1s]

Time window (in s) for the counters during setup phase (i.e., before the "Take Data" button is pressed.) Readings update once each time window if "Status" reads "Reading Counters".

U8 **Experimental Setup**

Which Measurement to perform.

H: Hardy measurement (4 detectors)

H two det: Hardy measurement (2 detectors - waveplates)

H Pol: Hardy measurement (2 detectors - polarizers)

S: Bell measurement (4 detectors)

S two det: Bell measurement (2 detectors - waveplates)

S Pol: Bell measurement (2 detectors - polarizers)

DBL **A zero**

Motor angle at which A waveplate axis is 0.

DBL **B zero**

Motor angle at which B waveplate axis is 0.

DBL **Alpha** (Hardy measurements only)

Angle Alpha used in the Hardy measurement.

DBL **Beta** (Hardy measurements only)

Angle Beta used in the Hardy measurement.

TF **Subtract Accidentals?**

Determines whether or not accidental coincidences are subtracted. Controls data taking mode as well as tweaking mode.

DBL **coincidence resolution (ns)**

The coincidence time resolutions (used in subtraction of accidentals).

Status

Program Status:

Initializing: initializing the counters and motors.

Reading Counters: Program is looping, reading the counters and updating the screen.

Moving Motors: Waveplates (polarizers) are rotating.

A Desired Position

Angle to set A waveplate to.

Movement occurs after "Move Motors" button is pressed.

B Desired Position

Angle to set B waveplate to.

Movement occurs after "Move Motors" button is pressed.

Move Motors

Set the motors to the "Desired Positions"

A Position

Current angle of the A waveplate.

B Position

Current angle of the B waveplate.

A Motor Position

Actual position of the motor for the A waveplate.

Not the same as "A Position" because of the 0 position correction specified in "A zero".

B Motor Position

Actual position of the motor for the B waveplate.

Not the same as "B Position" because of the 0 position correction specified in "B zero".

DBL **H HWP Measurement Angles** (Hardy measurements only)

HWP angles at which probabilities that determine H will be measured. Determined from Alpha and Beta settings.

Useful so that you know which angles to set waveplates to when tweaking. Of course, for 2 detector measurements you need perpendicular combinations as well.

DBL **H Pol Measurement Angles** (Hardy measurements with polarizers only)

Polarizer angles at which probabilities that determine H will be measured. Determined from Alpha and Beta settings.

Useful so that you know which angles to set waveplate to when tweaking. Of course, you need perpendicular combinations as well.

U32 **Counts A, A’, B, B’**

Singles counts in the time window specified by “Update Period” (upper left)

U32 **Counts AB, A’B, AB’, A’B’**

Coincidence counts in the time window specified by “Update Period” (upper left)

DBL **P** (4-detector Hardy measurements only)

Probability of AB.

DBL **P Meter** (4-detector Hardy measurements only)

Probability of AB

Data Taking Parameters

DBL **Update Period (Data Run)**

[Must be a multiple of 0.1s]

Time window (in s) for counters during data acquisition.

This applies after the "Take Data" button has been pressed.

I32 **No. of Samples**

Number of independent measurements (of H for Hardy measurements, or S for Bell Measurements) that are made during data acquisition. These are averaged to get the mean and error of the measurement.

Error measurements are essentially useless if this is less than 5. 10 is a good number for reasonable statistics.

Take Data

Leave the setup "tweaking" mode and switch to data acquisition mode.

Bell Measurements

E Meter (4-detector Bell measurement only)

Expectation Value

E (4-detector Bell measurement only)

Expectation Value

S HWP Measurement Angles

HWP angles at which probabilities that determine S will be measured.

Useful for 4 detector measurements so that you know which angles to set waveplates to when tweaking. Angles labeled + correspond to expectations you want to be as positive as possible, while those labeled - should be as negative as possible.

Not particularly useful for 2 detector measurements.

S Pol Measurement Angles

Polarizer angles at which probabilities that determine S will be measured.

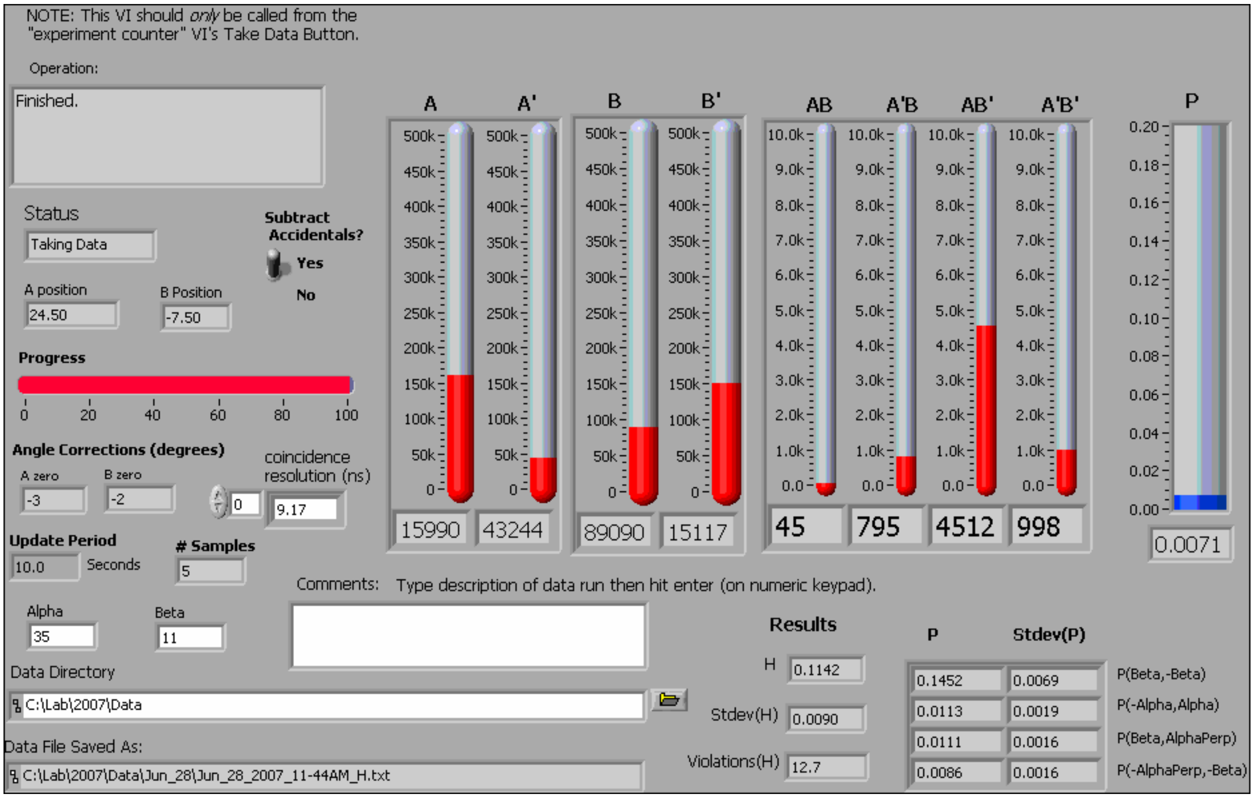
Not a particularly useful parameter for 2 detector measurements.

II. Taking Data

When you move to data taking mode, the program you see is:

Hardy-Bell-Recorder.vi

Front Panel



Description

This VI should ONLY be called from a "Hardy-Bell" VI--necessary parameters are set there.

This program records data (using parameters set in the "Hardy-Bell" VI) and saves the data to a file.

Controls and Indicators

Operation

Displays what the program is trying to do. Normally this shows what the angles of the motors are.

There are 2 angles shown -- the waveplate angles are given first and the actual motor angles are in parentheses (they are different by the 0 offset displayed in "A zero" and "B zero".)

When the program is done this reads "Finished." Values for probabilities (expectations for Bell) and H (S for Bell) are not updated until this happens.

Status

Program Status:

Moving Motors: waveplate (polarizer) motors are in motion.

Waiting: Clearing out the counter buffers after motors move.

Taking Data: Data is being collected.

A position

Current position of A motor.

B Position

Current position of B motor.

Subtract Accidentals?

Determines whether or not accidental coincidences are subtracted.

coincidence resolution (ns)

Array of the coincidence time resolutions (used in subtraction of accidentals).

Progress

Rough indication of what percentage of the data run is complete.

A zero Motor angle at which A waveplate axis is 0.

B zero Motor angle at which B waveplate axis is 0.

Update Period

Time window (in s) for counters.

 # Samples

Number of independent measurements (of H for Hardy measurements, or S for Bell Measurements) that are made during data acquisition. These are averaged to get the mean and error of the measurement.

Error measurements are essentially useless if this is less than 5. 10 is a good number for reasonable statistics.

Comments

A space where you can enter comments about the data run. These get saved in the header of the data file.

Note that you must hit "Enter" on the NUMERIC KEYPAD, not the "Enter" (or "Return") on the regular keyboard for this text to be saved.

Data Directory

Path to directory where the data will be saved.

Data File Saved As:

Path to the data file. The data file is automatically named using the date and time.

Alpha

Angle Alpha used in the Hardy measurement.

Beta

Angle Beta used in the Hardy measurement.

H (Hardy measurement only)

Mean value of H. Updated when "operation" reads "Finished".

Stdev(H) (Hardy measurement only)

Standard Deviation of H. Updated when "operation" reads "Finished".

Violations(H) (Hardy measurement only)

Number of standard deviations that H is above 1. Updated when "operation" reads "Finished".

P **Stdev(P)**

Probabilities and standard deviations of the probabilities for the indicated angles. Udated when "Operation" reads "Finished".

P (4 detector Hardy only)

Probability of AB

Bell Measurements

S (Bell measurement only)

Mean Value of S. Updated when "Operation" reads "Finished".

Stdev(S) (Bell measurement only)

Standard deviation of S. Updated when "Operation" reads "Finished".

Violations(S) (Bell measurement only)

Number of standard deviations that S is above 2. Updated when "Operation" reads "Finished".

E (4 detector Bell only)

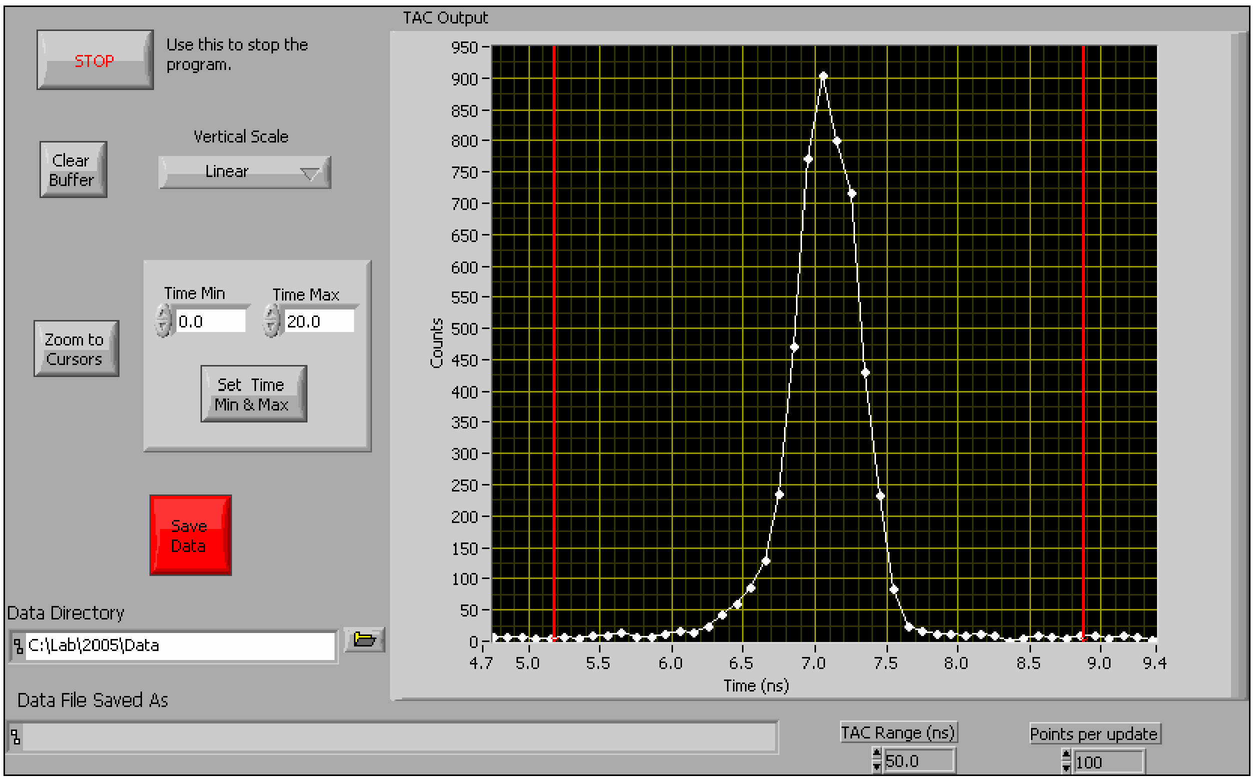
Expectation value.

Appendix C - MCA.vi

This appendix describes the LabView vi used to mimic the behavior of a multichannel analyzer (MCA). This is useful for measuring the arrival time of photons.

MCA.vi

Front Panel



Description

This VI emulates a multi-channel-analyzer (MCA). An MCA essentially digitizes a series of voltage pulses, and creates a histogram of the pulse heights. The purpose of this VI is to analyze at the voltage pulses from a Time-to-Amplitude Converter (TAC) [This VI was tested using the TAC in an ORTEC 567 TAC/SCA module].

The start and stop pulses going into the TAC come from photon counters, and the TAC output voltage is proportional to the time between the start and stop. Thus, the MCA is creating a histogram of the time interval between the arrivals of the two photons.

The main difference between this VI and a real MCA is that an MCA takes only one input-the voltage pulses. This VI also needs timing information about when the pulses are arriving. Fortunately, this can be obtained from the Valid Conversion output of the TAC. The Valid Conversion output is an approximately 3 microsecond long TTL pulse coincident with the TAC output pulse.

We have tested this VI with 2 different National Instruments A/D cards: 6036E and 6052E. With these cards there are two clock signals that are needed to perform an A/D conversion: the Scan Clock and the Channel Clock.

The Scan Clock (input through PFI0) comes directly from the Valid Conversion output of the TAC. The program is configured to initialize a scan on a low-to-high transition of the Scan Clock (edit the program if you want a high-to-low transition) -initializing a scan arms the A/D, but does not actually cause an A/D conversion.

The Channel Clock (input through PFI2) follows the Scan Clock, and a the program is configured to perform an A/D conversion on a high-to-low transition of the Channel Clock (edit the program if you want a low-to-high transition). The Channel Clock is obtained from the output of an HP33120A waveform generator. The 33120A is externally triggered by the Valid Convert output of the TAC, and on each trigger it outputs a 0.5 microsecond pulse. The 33120A is operated in burst mode to accomplish this-the VI programs the 33120A with the correct parameters when it first starts. The 33120A is programmed over GPIB, and must be at GPIB address 9.

The TAC output to be digitized is input though Analog Input Channel 0.

Unlike many VI's you'll encounter, the screen updates after a specified number of data points are acquired (determined by the parameter "Points per update"), not after a specific time interval. Because of this, if the detectors are off, nothing will happen.

NOTE: The A/D boards we use are just barely fast enough to catch the 3 microsecond long TAC output--indeed, they're probably technically too slow. Because of this, the accuracy of the measurement of the TAC output is not very good (can be off by as much as 10 or 20%), which means that the time scale is poorly calibrated. However, the measurement is repeatable, so the time scale could be calibrated much more accurately. Despite this problem, the VI works quite well enough to accomplish its main goals-to display a bump in the histogram of arrival times between the photons from two detectors, and to allow one to set the SCA window to select coincidences.

ALSO NOTE: This vi has only been seriously tested for START-STOP times ranging from a few up to about 30 ns. With much longer intervals than this the output pulses may be delayed, and the timing may change.

Rough Timing diagram:

```
          _____________
_________|             |____________   Scan CLK (PFI0)

                 __
_______________|  |________________   Channel CLK (PFI2)
              _________
___________/          \_____________   A/D Ch 0
```

Help for each of the controls and indicators can be obtained from the Contextual Help window <cntrl - H> by mousing over each control or indicator. Full documentation for each control and indicator can be obtained by printing using: File>Print>Custom, and then checking "All controls" and "Descriptions"

Controls and Indicators

Stop

Use this to stop.

Clear Buffer

Clears the histogram memory, and the accumulation starts over again.

Vertical Scale

Change the scaling of the vertical scale. Options are Linear and Logarithmic.

Zoom to cursors

Horizontal scale will zoom in to the region specified by the cursor positions.

Time Min

Set minimum of horizontal axis to this value after pressing "Set Time Min & Max".

Time Max

Set maximum of horizontal axis to this value after pressing "Set Time Min & Max".

Set min & max

Set the minimum and maximum values of the horizontal axis to those specified.

Save Data

Save the current data to a file. Program exits after this is done.

Data Directory

Path to directory where the data will be saved.

Data File Saved As

Path to the data file. The data file is automatically named using the date and time.

TAC Range (ns)

The full scale range of the TAC output. Used to scale the horizontal axis.

Points per update

Number of data points acquired before updating the screen. Remember, the screen updates after a specified number of data points are acquired, not after a specific time interval.

If you have very low count rates, you might want to decrease this from it's default value of 100.

TAC Output

Histogram of the time interval distribution.

Appendix D – Coincidence_time_res.vi

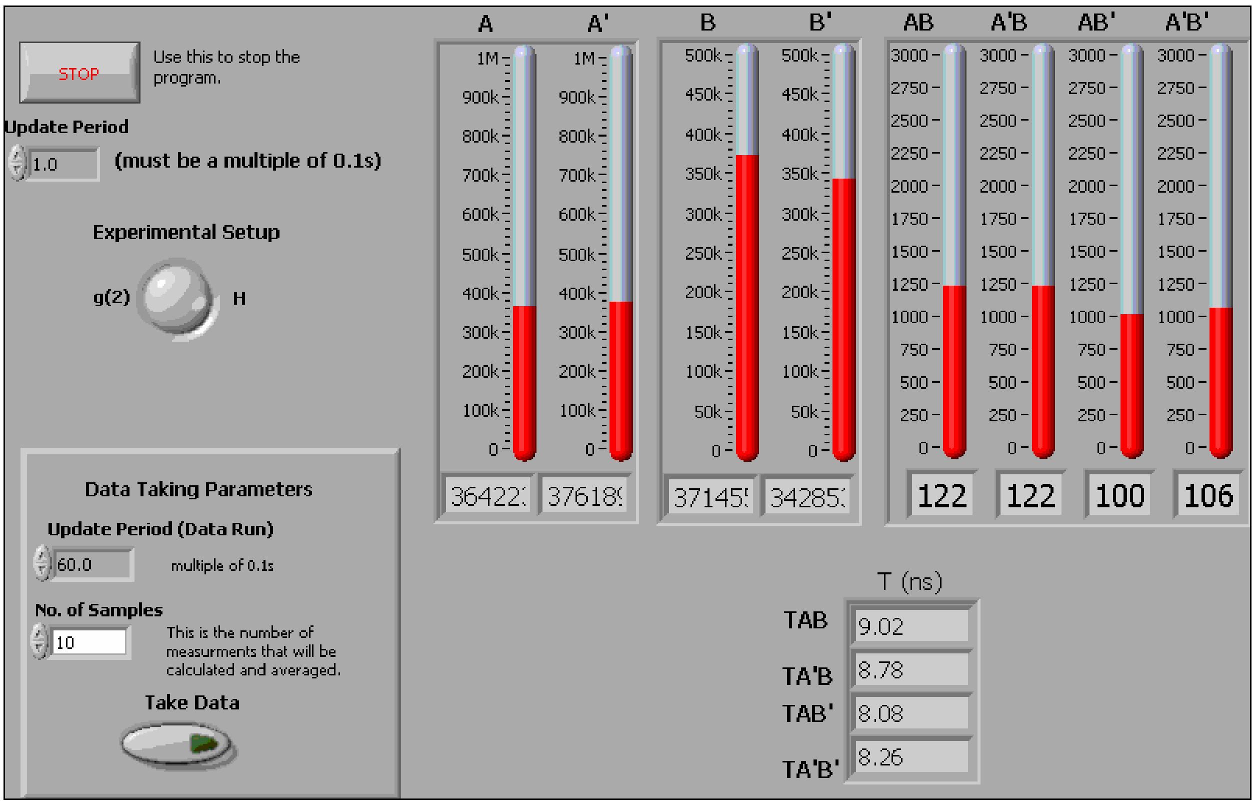
This appendix describes the LabView vi used to calibrate the coincidence time resolution.

I. Starting Out

The appearance of this vi will change slightly depending on the experiment you are doing (as determined by the **Experimental Setup** knob on the vi.)

Coincidence_time_res.vi

Front Panel



Description

This is the program you use to calibrate the coincidence time resolution of your coincidence circuit. It was specifically written with the Whitman/Trinity coincidence circuit in mind.

In order to get accurate readings, the SPCMs must be illuminated by uncorrelated random streams of photons. One way to do this is to simply place a white card after the downconversion crystal to scatter blue pump light. There are plenty of scattered photons to get through the RG780 blocking filters. The intensity level can be adjusted somewhat by moving the card closer or further away from the exit face of the crystal. Adjusting iris diameters can help as well. 500k to 1M singles cps on each detector works well.

For example, with random photons the coincidence time resolution TAB for AB coincidences is

TAB=(AB*T)/(A*B) , where T is the counting time, AB the number of coincidences, and A and B the number of singles.

There are 2 modes:

1) g(2)measures the resolution for AB, AB' and BB' coincidences. (random photons do not give an accurate measurement of the ABB' triple time because for those measurements you WANT at least one valid coincidence).

2) H measures the time resolution for AB,. A'B, AB' and A'B' coincidences.

NOTE: The switch on the coincidence circuit must match the "Experimental Setup" switch here on the vi.

The program does not record data to a file right away, but displays the counters in real time so that you can align things.

Once everything is aligned and the parameters are set, you press the "Take Data" button. This transfers control to another program which records a data set and saves it to a file.

Uses a National Instruments PCI-6602 and reads counters 0 thru 7. The device number for this board is should be set to 2. Input to this board is via the coincidence circuit.

Counters are gated by the clock signal coming from the circuit. For this vi to operate properly this clock signal must be at 10Hz. [Note that this clock signal is connected to Gate 0 (PFI 38) on the PCI-6602].

Electrical connections between the circuit and the counters on the PCI-6602:

When set to g(2):

A (Counter 0): PFI 39

B (Counter 1): PFI 35

B' (Counter 3): PFI 27

AB (Counter 4): PFI 23

AB' (Counter 6): PFI 15

ABB' (Counter 5): PFI 19

BB' (Counter 7): PFI 11

Unused A' (Counter 2): PFI 31

When set to H:

A (Counter 0): PFI 39

B (Counter 1): PFI 35

A' (Counter 2): PFI 31

B' (Counter 3): PFI 27

AB (Counter 4): PFI 23

A'B (Counter 5): PFI 19

AB' (Counter 6): PFI 15

A'B' (Counter 7): PFI 11

On startup, the program initializes the counters.

After initialization the program simply loops and displays the counts in a given time window (determined by the "Update Period" control in the upper left.) This is useful for tweaking the alignment and adjusting parameters.

Nothing is written to disk until the parameters are chosen and the "Take Data" button is pressed. This loads a second VI that records and saves data to disk. Parameters for this data acquisition phase are set in the "Data Taking Parameters" box.

Help for each of the controls and indicators can be obtained from the Contextual Help window <cntrl - H> by mousing over each control or indicator. Full documentation for each control and indicator can be obtained by printing using: File>Print>Custom, and then checking "All controls" and "Descriptions".

Controls and Indicators

Experimental Setup

Which calibration to perform.

1) g(2)measures the resolution for AB, AB' and BB' coincidences.

2) H measures the time resolution for AB, A'B, AB' and A'B' coincidences.

Update Period

[Must be a multiple of 0.1s]

Time window (in s) for the counters during setup phase (i.e., before the "Take Data" button is pressed.)

Stop

Use this to stop. It takes a little longer, but this way, the board gets reset. If you stop some other way you'll probably need to quit Labview and restart; you may even need to reboot the computer.

Counts A, A', B & B'

Singles counts in the time window specified by "Update Period" (upper left)

Coincidences

AB, A’B, AB’, A’B’ for H measurements

AB, ABB’, AB’, BB’ for g2 measurements.

Coincidence counts (doubles or triples as indicated) in the time window specified by "Update Period" (upper left)

T (ns)

Values of coincidence time resolution (in ns) for the current iteration.

No. of Samples

Number of independent measurements that are made during data acquisition. These are averaged to get the mean and error of the measurement.

Error measurements are essentially useless if this is less than 5. 10 is a good number for reasonable statistics.

Update Period (Data Run)

Time window (in S) for counters during data acquisition.

This applies after the "Take Data" button has been pressed.

Take Data

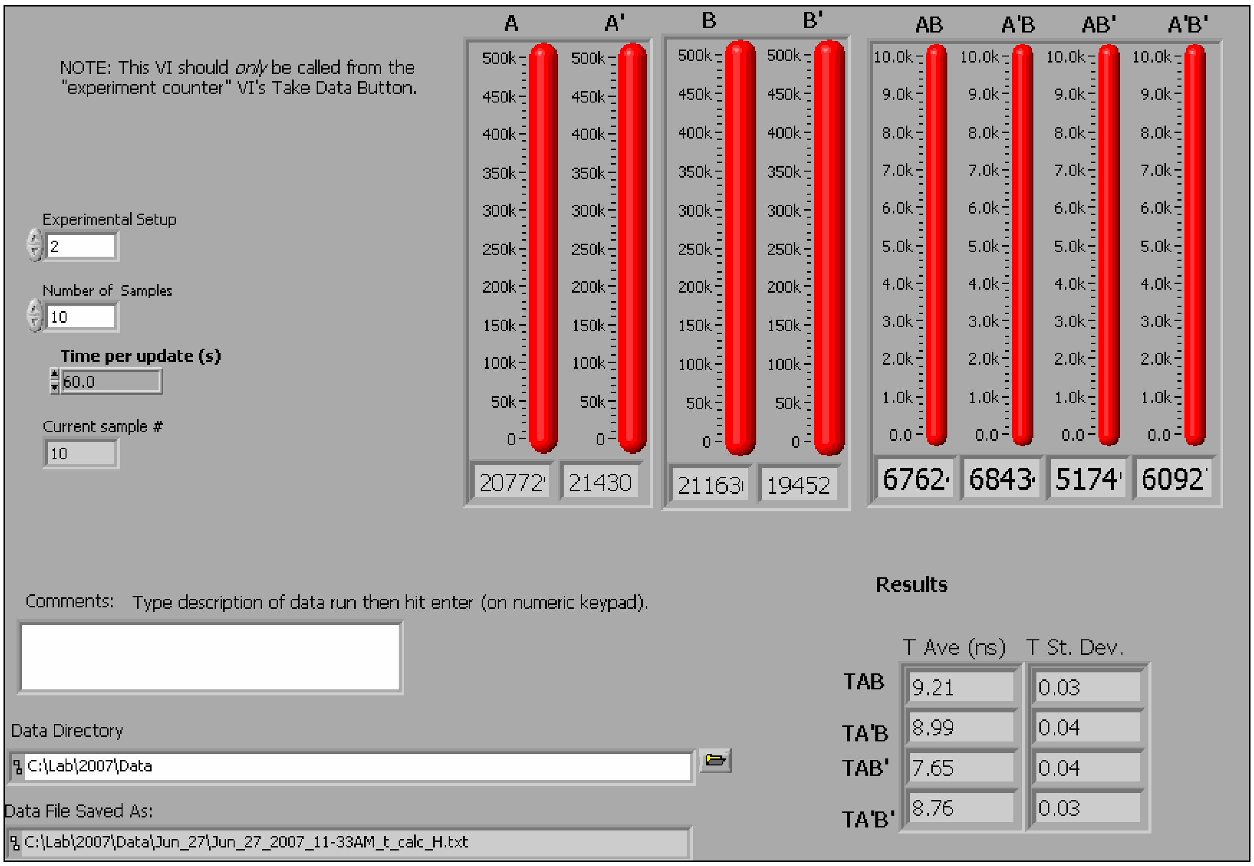
Leave the setup "tweaking" mode and switch to data acquisition mode.

II. Taking Data

When you move to data taking mode, the program you see will look something like that shown below. Again, the appearance of this vi will change slightly depending on the calibration you are doing.

coincidence_time_res_recorder.vi

Front Panel



Description

This VI should ONLY be called from an "coincidence_time_res" VI--necessary parameters are set there.

This program records data (using parameters set in the "coincidence_time_res" VI) and saves the data to a file.

Controls and Indicators

Experimental Setup

Which calibration is being performed.

1: g2 calibration

2: H calibration

Number of Samples

Number of independent measurements that are made during data acquisition. These are averaged to get the mean and error of the measurement.

Time per update (s)

Time window (in s) of counters during data acquisition.

Current sample #

Which sample number is presently being measured.

Counts A, A', B & B'

Singles counts in the time window specified by "Time per update"

Coincidences

AB, A’B, AB’, A’B’ for H measurements

AB, ABB’, AB’, BB’ for g2 measurements.

Coincidence counts (doubles or triples as indicated) in the time window specified by "Time per update"

Comments

A space where you can enter comments about the data run. These get saved in the header of the data file.

Note that you must hit "Enter" on the NUMERIC KEYPAD, not the "Enter" (or "Return") on the regular keyboard for this text to be saved.

Data Directory

Path to directory where the data will be saved.

Data File Saved As:

Path to the data file. The data file is automatically named using the date and time.

[DBL] **T Ave (ns)**

Measured average time resoltion in ns.

[DBL] **T St. Dev.**

Standard deviation of the measured time resolutions in ns.