Ruhr-Universität Bochum
Lehrstuhl für Automatisierungstechnik und Prozessinformatik
Prof. Dr.-Ing. J. Lunze

Versuch 3: Programmierung in LabVIEW

Autoren:

Dr.-Ing. J. Dastych, Dipl.-Ing. C. Ortmann,
Dipl.-Ing. C. Maul, M. Sc. Daniel Vey, M. Sc. Philipp Welz

08.05.2017

Inhaltsverzeichnis

1 Einleitung **2**

1.1 Was ist LabVIEW? 2

1.2 Ziel des Versuchs 2

2 Grundbegriffe in LabVIEW **3**

2.1 Start 3

2.2 Projekte und SubVI's 3

2.3 Frontpanel und Blockdiagramm 4

2.4 Die Menüleiste und die Toolbar 4

2.5 Hinzufügen von Elementen 5

2.6 Die Werkzeugpalette 6

3 Debugging und Fehlersuche in LabVIEW **7**

4 Vorbereitungsaufgaben **9**

4.1 Umgang mit LabVIEW 9

4.2 Zweipunktregelung eines Tanksystems 10

5 Praktikumsaufgaben **11**

5.1 Aufgabe 1: Umgang mit LabVIEW 11

5.2 Aufgabe 2: Zweipunktregelung eines Tanks 13

6 Nachbereitung des Praktikumsversuches **15**

1 Einleitung

1.1 Was ist LabVIEW?

LabVIEW steht für **Lab**oratory **V**irtual **I**nstrument **E**ngineering **W**orkbench. Es ist eine Programmierumgebung für Laborinstrumente, welche die Fähigkeit hat, die meisten realen Laborinstrumente mittels einer Benutzeroberfläche auf dem Rechner darzustellen. LabVIEW kann mit gängiger Hardware kommunizieren und so eine komplette Laborumgebung simulieren.

Dies ist auch mit üblichen textbasierten Programmiersprachen möglich, jedoch ist LabVIEW mit seiner grafischen Oberfläche, in Einfachheit und Übersichtlichkeit diesen weit überlegen. LabVIEW liefert zudem viele Standardkomponenten als vorgefertigte „Virtual Instruments (VI)“ mit, so dass „das Rad nicht immer wieder neu erfunden“ werden muss.

LabVIEW wird in vielen Bereichen der Automatisierungstechnik angewendet. Hiermit lassen sich Messdaten erfassen, physikalische Phänomene simulieren und Anlagen regeln.

1.2 Ziel des Versuchs

Der Versuch soll eine Einführung in die Benutzung von LabVIEW geben. Es wird natürlich nicht auf jeden Menüpunkt und jede Besonderheit eingegangen. Es soll vor allem die Arbeitsoberfläche des Programms sowie einige wichtige Grundfunktionen und Einstellungen vorgestellt werden. Darüber hinaus ist LabVIEW sehr intuitiv zu bedienen und verfügt über eine gute Hilfe-Funktion. Für die LabVIEW-Einführung stehen zudem sogenannte Flash-Animationen zur Verfügung, die in den einzelnen Kapiteln weitergehende Informationen bieten. Diese sind in dem Praktikumsordner „M_Praktikum“ auf „ATP-Common auf Frodo“ unter „Versuch 3 – Programmierung in LabVIEW“ zu finden.

2 Grundbegriffe in LabVIEW

2.1 Start

Nach dem Start von LabVIEW erscheint ein Menüfenster, welches mehrere Auswahlmöglichkeiten zulässt. Dieses Auswahlfenster kann je nach verwendeter Version von LabVIEW leicht variieren. Im Normalfall soll ein neues „Virtual Instrument“ (in Zukunft mit „VI“ abgekürzt) erstellt werden. Dazu wird im Menüpunkt *Neu* der Unterpunkt ein *Leeres VI* ausgewählt.

2.2 Projekte und SubVI’s

SubVI’s sind VI’s, die mit Ein- und Ausgängen versehen werden, um sie in größeren komplexeren VI’s wieder zu verwenden. So sind auch die erwähnten, vorgefertigten VI’s bereits erstellte VI’s, die durch ihre Einbindung in ein neues, übergeordnetes VI zu einem SubVI werden. Durch SubVI’s kommt eine Modularisierung in LabVIEW zustande. Es stehen drei **Flash-Animationen** „SubVIa“ , „SubVIb“ und „SubVIc“ von National Instruments zur Verfügung, die den Prozess verdeutlichen.

Werden SubVI’s in VI’s verwendet, auch die SubVI’s die bereits zu LabVIEW gehören, ist es wichtig zu beachten, dass alle VI’s und SubVI’s, die im Zusammenhang stehen, zu einem gemeinsamen Projekt zusammengefügt werden, damit LabVIEW bei der Ausführung die Zusammengehörigkeit erkennt. Die geschieht entweder direkt im Auswahlfenster beim Start von LabVIEW, wenn im Menüpunkt *Neu* anstelle des Unterpunktes *Leeres VI* der Unterpunkt *Leeres Projekt ausgewählt wird*. Dann öffnet sich ein Projekt-Fenster (siehe Abb. 1), indem wiederum über den Menüpunkt *Neu* der Unterpunkt ein *Leeres VI* ausgewählt werden kann, welches dann dem Projekt angehört.

Abbildung 1: Projektfenster

Alternativ kann auch später ein Projekt erzeugt werden, wenn im Fenster des VI’s über den Menüpunkt *Neu* der Unterpunkt ein *Leeres Projekt* aufgerufen wird. LabVIEW fragt dann, ob das aktuelle VI gleich dem neuen Projekt hinzugefügt werden soll. SubVI’s, die in diesem VI eingebunden werden, werden ab diesem Zeitpunkt automatisch dem Projekt hinzugefügt. Zuvor eingefügte SubVI’s müssen entweder neu eingebunden oder manuell hinzugefügt werden.

2.3 Frontpanel und Blockdiagramm

Nach dem Erstellen eines leeren VI's werden in LabVIEW automatisch zwei Fenster geöffnet. Diese stellen zwei verschiedene Arbeitsoberflächen, das `Blockdiagramm` und das `Front-Panel`, dar.

Das Front-Panel in LabVIEW ist die Visualisierungsoberfläche. Auf dem Front-Panel werden alle Elemente zur Bedienung und Beobachtung dargestellt, wie z.B. virtuelle Messinstrumente, Lampen, Schalter etc..

Das Blockdiagramm ist die eigentliche Programmierumgebung. Hier werden die Messinstrumente und Eingabeelemente, die auf dem Front-Panel zu sehen sind, sowie externe Signale über Funktionen und Filter miteinander verknüpft. Im Blockdiagramm wird im Prinzip eine Ablaufsteuerung erstellt, die Ein- und Ausgaben miteinander verknüpfen.

Obwohl das Blockdiagramm und das Front-Panel zwei unterschiedliche Ansichten in unterschiedlichen Fenstern darstellen, ist zu beachten, dass diese keinesfalls unabhängig voneinander existieren. Z.B. besitzt ein Radio von außen betrachtet lediglich ein paar Rädchen und Schalter um Lautstärke- und Frequenzeinstellungen vorzunehmen und eine Skala, die die aktuelle Frequenz anzeigt. Dies entspricht der Ansicht im Front-Panel, während die Elektronik im Inneren, über die die Skalen, Rädchen und Schalter miteinander verbunden sind, dem Blockdiagramm entspricht. Dementsprechend wird in LabVIEW, für jedes Element, dass im Front-Panel eingefügt wird, automatisch ein neues Element – ein sogenannter „Terminal" – im Blockdiagramm erzeugt.

Mit der Tastenkombination **STRG-E** kann beliebig zwischen Blockdiagramm und Front-Panel gewechselt werden. Über den Menüpunkt *Fenster* oder mit der Tastenkombination **STRG-T** können die beiden Fenster automatisch nebeneinander angeordnet werden.

2.4 Die Menüleiste und die Toolbar

Die Menüleiste (Abb. 2) von LabVIEW ist wie in jedem anderen Programm zu bedienen. Spezielle Menüpunkte und die Buttons der Toolbar in LabVIEW, sind in der **Flash-Animation** „Menüleiste" ausführlich beschrieben.

Abbildung 2: Menüleiste im Frontpanel

Ausführen von VI's: Der „Ausführen-Button"  startet eine Simulationsdurchlauf in LabVIEW, der „Wiederholt Ausführen-Button"  startet eine automatisch wiederholte Ausführung des VI's. Der rote „Abbrechen-Button"  bricht die Simulation ab. Ein Abbruch ist etwas anderes als ein kontrolliertes Beenden der Simulation. Deshalb sollte dieser Button nicht zum stoppen der Ausführung genutzt werden. Wie die Simulation kontrolliert gestoppt wird, wird in Abschnitt 5.1 erklärt. Der „Pause-Button"  pausiert die Ausführung des VI's. Die Glühbirne ist ein Debug-Tool und wir in Abschnitt 3 erklärt.

2.5 Hinzufügen von Elementen

Ein VI besteht aus einzelnen Bedien- und Anzeigeelementen. Um in LabVIEW ein solches Element hinzuzufügen, reicht sowohl im Front-Panel, als auch im Blockdiagramm ein Klick mit der rechten Maustaste auf eine freie Stelle.

Im Front-Panel erscheint ein Fenster mit dem Titel `Bedienelemente` (siehe Abb. 3). Hier kann aus einer Auswahl von vorgefertigten Schaltern, Drehknöpfen und Anzeigeelementen ausgewählt werden. Ein solches Element kann per Drag&Drop Verfahren platziert werden. Die Eigenschaften der Elemente können mit einem Klick mit der rechten Maustaste auf das Element bearbeitet werden. Bei einigen Elementen wird das Eigenschaften-Fenster direkt bei der Platzierung des Elements aufgerufen.

Im Blockdiagramm erscheint ein Fenster mit dem Titel `Funktionen` (siehe Abb. 4). Diese können, genau wie im Front-Panel, per Drag&Drop gesetzt und über einen Rechtsklick auf das Element bearbeitet werden. Auf ausgewählte Funktionen wird später eingegangen.

Abbildung 3: Bedienelemente

Abbildung 4: Funktionen

2.6 Die Werkzeugpalette

Die `Werkzeugpalette` (siehe Abb. 5) wird benötigt, um bestimmte Operationen auszuführen, die im Folgenden erklärt werden. Sollte die Werkzeugpalette nicht angezeigt werden, kann sie im Menüpunkt **Ansicht** aufgerufen werden.

Abbildung 5: Werkzeugpalette

Es stehen folgende Funktionen zur Verfügung (zeilenweise von links nach rechts):

- **Automatische Werkzeugauswahl**  : Ist diese Funktion aktiviert, wählt LabVIEW zwischen den meisten der folgenden Tools austomatisch, je nach Maus-Position und Kontext. Einige Tools, wie z.B. die Debugging-Funktionen müssen allerdings manuell gewählt werden.
- **Wert einstellen**  : Mit dieser Funktion kann bei bestimmten Terminals ein Wert manipuliert werden, um bestimmte Anfangsbedingungen zu setzen.
- **Position/Größe/Auswahl**  : Diese Funktion ist im Prinzip das in jedem Programm verwendete Zeigerwerkzeug. Mit diesem Werkzeug können Elemente ausgewählt, ihre Größe oder auch ihre Position verändert werden.
- **Text bearbeiten**  : Selbsterklärend.
- **Verbinden**  : Mit diesem Tool werden die einzelnen Blöcke im Blockdiagramm verbunden.
- **Objekt-Kontextmenü**  : Öffnet bei Linksklick die Bedienelemente-/Funktionenpalette.
- **Fenster verschieben**  : Erlaubt das Scrollen durch das Fenster, ohne die Scrollbar zu nutzen.
- **Setzen/Löschen des Haltepunktes**  : Erlaubt das Setzen bzw. Löschen eines Haltepunktes. Der Haltepunkt verursacht in der Ausführung einen Halt, wo auch immer er eingesetzt wird. Das Programm wird erst auf Anweisung weiter ausgeführt. Diese Funktion eignet sich zum Debuggen.
- **Sondenwerte**  : Dieses Werkzeug lässt sich an jede Verbindung im Blockdiagramm anbringen und zeigt während der Ausführung den aktuellen Wert an dieser Stelle an. Auch dieses Werkzeug ist ein Debugging-Tool.
- **Farbe ermitteln**  **/ Farbe setzen**  : Selbsterklärend.

3 Debugging und Fehlersuche in LabVIEW

Bei komplexeren Programmen bzw. Ablaufsteuerungen kann es vorkommen, dass trotz langer Überlegungen Verbindungen falsch positioniert werden, oder Funktionen nicht das tun, was eigentlich von ihnen verlangt wird. Zu diesem Zweck besitzt LabVIEW einige Debugging Tools, die die Suche nach Fehlern deutlich vereinfachen.

Gebrochener Ausführen-Pfeil: Sollte ein Programm nicht lauffähig sein, weil entweder eine Verbindung unterbrochen ist, oder andere Fehler vorliegen, wird direkt der „Ausführen-Pfeil“ in der Toolbar, wie in Abb. 6, gebrochen dargestellt.

Abbildung 6: Gebrochener Ausführen-Pfeil

Dass der Pfeil gebrochen dargestellt ist heißt nicht unbedingt, dass ein syntaktischer Fehler gemacht wurde. Der Pfeil ist normalerweise, während der kompletten Entwurfsphase gebrochen dargestellt, bis die Verdrahtung ausgeführt ist.

Fehlerliste: Sollte trotz vermeintlich richtiger Verdrahtung ein gebrochener Pfeil dargestellt werden, deutet dies tatsächlich auf einen Fehler hin. Durch Klicken auf den gebrochenen Pfeil, wird die Fehlerliste (siehe Abb. 7) geöffnet und die Fehlerquelle und Beschreibung kann angesehen werden. Wird darin ein Fehler markiert und dann der Button „Fehler anzeigen“ angeklickt, wird in aller Regel der Fehler im Blockdiagramm angezeigt, so dass dieser behoben werden kann.

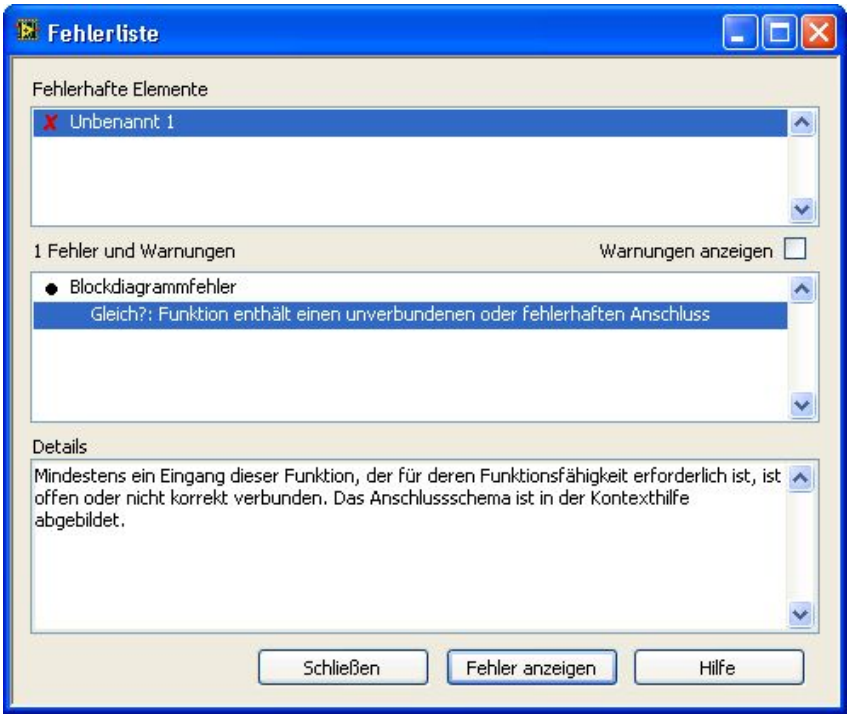

Abbildung 7: Fehlerliste

Wichtiger Hinweis zu ungültigen Verbindungen: Beim Löschen von Objekten im Blockdiagramm kann es passieren, dass Reste von Verbindungen überbleiben die kaum sichtbar sind. Diese Reste erzeugen ebenfalls Fehler. Um diese Fehler schnell zu beheben kann im Blockdiagramm im Menüpunkt *Bearbeiten* die Funktion *Ungültige Verbindungen entfernen* verwendet werden. Das Tastaturkürzel für diese Funktion ist **STRG-B**.

Haltepunkte und Sondenwerte: Sollte das VI ohne Probleme ausgeführt werden, aber dennoch nicht die gewünschten Ergebnisse liefern, gibt es drei weitere Funktionen, die zum Debugging genutzt werden können. Die beiden Funktionen „Haltepunkt setzen“ und „Sondenwerte“ sind bereits in Abschnitt 2.6 zur Werkzeugpalette angesprochen und erklärt worden. Für weitere Informationen zu diesen Tools ist die LabVIEW Hilfe zu nutzen.

Highlight-Funktion: Die dritte Funktion ist die sogenannte „Highlight-Funktion“ . Die Funktion wird im Blockdiagramm in der Menüleiste als Glühlampe angezeigt (siehe Abb. 8).

Abbildung 8: Highlight-Funktion (Glühbirne)

Diese Funktion verlangsamt die Ausführung des VI und zeigt an, wie die Daten durch das Blockdiagramm geleitet werden. Diese Funktion kann helfen, strukturelle Fehler im Blockdiagramm aufzudecken.

4 Vorbereitungsaufgaben

Die folgenden Aufgaben dienen zur Vorbereitung des Praktikumsversuches. Die Lösungen sind in handschriftlich und in Stichworten zu notieren und zum Versuchstermin mitzubringen. Liegen die Lösungen der Vorbereitungsaufgaben am Versuchstermin nicht vor, kann das zum Ausschluss von der Versuchsdurchführung führen.

4.1 Umgang mit LabVIEW

In der ersten Aufgabe soll der Umgang mit LabVIEW und seinen elementaren Funktionen vertraut gemacht werden. Der Aufbau der Simulationsumgebung wird sehr detailliert beschrieben, da der Fokus auf dem Umgang mit dem Programm LabVIEW liegen soll. Damit Sie sich am Versuchstag schnell in das Programm einfinden und Fehler zügig beseitigt werden, sollen Sie sich bereits im Vorfeld mit Hilfe der Abschnitte 1-3 mit dem Aufbau der Arbeitsoberfläche, den wichtigsten Werkzeugen, ein paar wichtigen Eigenschaften und den Debugging-Tools auseinandersetzen. Beantworten Sie daraufhin die folgenden Fragen:

1. Wie heißen die Arbeitsoberflächen in LabVIEW und was sind ihre Funktionen?
2. Welche Debugging-Tools gibt es in LabVIEW (mindestens 4) und was bewirkt die Tastenkombination **STRG-B**?
3. Worauf ist bei der Verwendung von SubVI's zu achten und wie werden Ein- und Ausgänge zur Erstellung von SubVI's definiert?

LabVIEW nutzt die aus Programmiersprachen bekannten For- und While-Schleifen, welche für die Simulation eine wichtige Rolle spielen. Zur Programmierung dieser beiden Funktionen stehen auch zwei **Flash-Animationen** „WhileLoop“ und „ForLoop“ zur Verfügung.

4. Informieren Sie sich mit Hilfe der Flash-Animationen über Schleifen in LabVIEW.

4.2 Zweipunktregelung eines Tanksystems

In der zweiten Aufgabe soll die Regelung eines Tanksystems simuliert werden. Die Regelung erfolgt mittels eines Zweipunktreglers, der die Zulaufpumpe ein- bzw. ausschaltet. Das Tanksystem in Abb. 9 dargestellt, wobei u der Stellwert für die Zulaufpumpe ist, q_P die Leistung der Zulaufpumpe, h die Füllhöhe, A_T den Querschnitt des Behälters, s_V die Stellung des Ausflussventils, A_V den Querschnitt des Ausflussventils und q_0 den Ausfluss beschreibt.

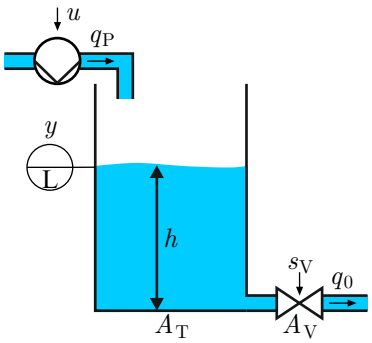

Abbildung 9: Tanksystem

Tanksystem:

1. Beschreiben Sie den Ausfluss q_0 des Tanksystems aus Abb. 9 mit Hilfe des Toricelli Ausflussgesetzes in Abhängigkeit des Füllstands h, des Querschnitts des Ausflussventils A_V und der Stellung des Ausflussventils s_V.

2. Stellen Sie die Volumenbilanzgleichung für das Tanksystem aus Abb. 9 in Abhängigkeit des Zuflusses q_P, des Stellwerts für die Zulaufpumpe u und des Abflusses q_0 auf.

3. Wie sieht der Differenzenquotient aus und was beschreibt er für $\Delta t \to 0$? Stellen Sie die Differenzengleichung zur Berechnung der Tankhöhe h auf.

Regelung:

1. Wie funktioniert ein Zweipunktregler? Geben Sie auch die Formel für das Stellgesetz und eine Skizze der Kennlinie an.

2. Zeichnen Sie das Blockschaltbild des Regelkreises. Benennen Sie alle Signale.

5 Praktikumsaufgaben

In der folgenden Aufgabe soll ein eigenes VI entworfen und dabei die beschriebenen Grundlagen von LabVIEW genutzt und vertieft werden. Zusätzlich werden durch diesen Versuch noch einige neue Funktionen eingeführt. Das VI soll ein Signal erzeugen, das Signal filtern und die Ergebnisse in einer Datei speichern.

5.1 Aufgabe 1: Umgang mit LabVIEW

1. **Einfügen einer Signalquelle:** Starten Sie LabVIEW und öffnen Sie ein leeres VI. Da das Frontpanel kein Symbol für eine Signalquelle hat wechseln Sie in das Blockdiagramm. Fügen Sie aus der Gruppe *Express* unter *Eingabe* den Block *Signal simulieren* ein. Der Block kann vergrößert werden um zu sehen welche Ein- und Ausgänge für diesen Block zur Verfügung stehen. Achten Sie darauf, dass in den Eigenschaften des Blocks unter *Timing* „Timing für Erfassung simulieren“ eingestellt ist, da sonst die Sinusschwingung viel zu schnell angezeigt wird.
2. **Darstellung auf dem Frontpanel:** Das Signal soll auf dem Frontpanel angezeigt werden. Wechseln Sie daher in das Frontpanel. Wählen Sie aus der Gruppe der *Graphen* einen *Graph* aus und fügen Sie ihn ein. Da im Blockdiagramm noch keine Verbindung zwischen Signalquelle und Anzeige besteht, erfolgt keine Darstellung der Signale. Erstellen Sie im Blockdiagramm die Verbindung. Speichern Sie die Aufgabe unter *Aufgabe11.vi*! Starten Sie das VI.

Es werden einige Sinusschwingungen angezeigt, jedoch keine kontinuierliche Simulation. Der Grund dafür ist, dass das VI nur ganz kurz ausgeführt wird. Für eine dauerhafte Simulation, muss eine Schleife implementiert werden.

3. **Implementierung einer Schleife:** Zunächst muss überlegt werden, welcher Schleifentyp gebraucht wird. Eine „For-Schleife“ würde sich hier nicht anbieten, da keine vorgegebene Zahl von Iterationen vorbestimmt werden soll. Im Blockdiagramm kann in der Gruppe *Ausführung* die „While-Schleife“ ausgewählt werden, welche eine dauerhafte Ausführung ermöglicht. Diese kann um beide Blöcke gelegt werden. Es fehlt eine Abbruchbedingung.
4. **Implementierung eines Stop-Buttons:** Auf dem Frontpanel wird ein „Stop-Button“ eingefügt. Dieser wird mit dem „Abbruch-Symbol“ für die Schleife verbunden. Dieser Stop-Button beendet jetzt das VI kontrolliert. Speichern Sie die Aufgabe unter *Aufgabe12.vi*!
   Hinweis: Es sollte die Nutzung des roten Knopfes in der Menüleiste vermieden werden. Dieser hält zwar das VI auch an, beendet es jedoch anders als der neu hinzugefügte Stop-Button und kann somit in der weiteren Bearbeitung Probleme verursachen.
5. **Veränderung eines Signals:** Im Frontpanel soll aus der Gruppe *Modern*, *Numerisch* ein *Drehregler* eingefügt werden. Dieser ist mit dem Eingang *Amplitude* der Signalquelle zu verbinden. In den Eigenschaften des Drehreglers ist die „Skalierung“ so einzustellen, dass in einem Wertebereich zwischen 0 und 15 eingestellt werden kann. Passen Sie nicht nur die Skalierung sondern ebenfalls die „Grenzwerte“ der Eingabeelemente an, um ungültige und unerwünschte Eingaben zu verhindern. Speichern Sie die Aufgabe unter *Aufgabe13.vi*! Das Programm ist nun zu testen.

6. **Hinzufügen von Rauschen:** Funktioniert das Programm, soll dem Signal ein Rauschen hinzugefügt werden. Diese Funktion ist in den Eigenschaften der Signalquelle zu finden. Hierdurch bekommt die Signalquelle zwei weitere Eingänge. Es ist ein weiterer Drehregeler einzubauen, um die Rauschamplitude zwischen 0 und 1 einstellen zu können. Der „Startwert“ des Rauschsignals wird auf 0 gesetzt. Die Regler sind am Frontpanel entsprechend zu bezeichnen, um sie nachher auseinander halten zu können. Speichern Sie die Aufgabe unter *Aufgabe14.vi*!

7. **Signalfilterung:** Bei Versuchen ist es im Allgemeinen nicht möglich das störende Rauschsignal ein- oder auszuschalten und ein Messrauschen ist vorhanden. Da dies meist hochfrequente Spektren enthält, werden Tiefpassfilter eingesetzt, um das eigentliche Nutzsignal zu erhalten. Zwischen Signalquelle und Anzeige soll ein Tiefpassfilter aus der Gruppe *Express*, *Signalanalyse* in das Blockdiagramm eingefügt werden. Über einen Drehregler soll die „Grenzfrequenz“ zwischen 0,1 Hz und 100 Hz logarithmisch variiert werden können.
   Wird das Programm nun ausgeführt, ist nur das gefilterte Signal auf der Anzeige zu sehen. Mit der Funktion *Signale kombinieren* aus *Palette Signalmanipulation* können das ungefilterte und gefilterte Signal zusammengefügt werden und dann auf das Anzeigeelement gegeben werden, was einen Vergleich der beiden Signale möglich macht. Testen Sie nun diese Stufe des VI. Speichern Sie die Aufgabe unter *Aufgabe15.vi*!

8. **Verwendung boolescher Operatoren:** Es soll eine Grenzwertüberwachung des gefilterten Signals implementiert werden und durch eine LED visualisiert werden. Nutzen Sie die Funktion *Amplituden- und Pegelmessung* aus der Gruppe *Express*, *Signal-Analyse* um den „Spitze-zu-Spitze-Wert“ des gefilterten Signals zu messen. Die Funktion *Vergleich* in *Express*, *Arithmetik/Vergleich*, kann genutzt werden um das Signal mit dem Grenzwert, hier auf ≥ 8, zu vergleichen. Fügen Sie im Frontpanel eine LED ein und verbinden Sie die Elemente im Blockdiagramm. Nun sollte die LED ab einer Amplitude von 4 eine Grenzwertüberschreitung anzeigen. Speichern Sie die Aufgabe unter *Aufgabe16.vi*!

9. **Datenspeicherung:** Die gemessenen Daten sollen in eine Datei abgespeichert werden. In der Gruppe *Express*, *Ausgabe* ist die Funktion *Messwerte in Datei schreiben* enthalten. Dieses Express VI ist einzufügen und ein Dateiname in ihrem „Homeverzeichnis“ anzugeben. Achten Sie darauf, dass die Funktion „Daten an Datei anhängen“ aktiviert ist. Verbinden Sie das gefilterte Signal ist mit dem „Signaleingang“ von *Messwerte in Datei schreiben* zu verbinden und das VI für einige Sekunden auszuführen. Die gespeicherte Datei kann mit Excel oder Word betrachtet werden. Um die Anzahl der auf die Datei geschriebenen Werte zu verkleinern, kann der *Aktivieren* Eingang getriggert werden. Im Frontpanel ist ein boolescher Schalter einzubauen, der die Datenerfassung triggert. Speichern Sie die Aufgabe unter *Aufgabe17.vi*!

5.2 Aufgabe 2: Zweipunktregelung eines Tanks

In der zweiten Aufgabe ist die Zweipunktregelung des Tanksystems aus Abschnitt 4.2 in LabVIEW zu simulieren.

1. Stellen Sie, ausgehend von der Volumenbilanzgleichung aus den Vorbereitungsaufgaben, das nichtlineare Zustandsraummodell für das Tanksystem unter Verwendung der folgenden Parameter auf (beachten Sie die Einheiten):

Erdbeschleunigung	:	$g = 9,81\ \frac{\mathrm{m}}{\mathrm{s}^2}$
Querschnitt des Tanks	:	$A_\mathrm{T} = 1\ \mathrm{dm}^2$
Leistung der Zulaufpumpe	:	$q_\mathrm{P} = 5\ \frac{\mathrm{l}}{\mathrm{s}} = 5\ \frac{\mathrm{dm}^3}{\mathrm{s}}$
Querschnitt des Ausflussventils	:	$A_\mathrm{V} = 0,05\ \mathrm{dm}^2$
Stellung des Ausflussventils	:	$s_\mathrm{V} \in [0; 1]$

   Die Öffnung des Ausflussventils kann kontinuierlich zwischen 0 (geschlossen) und 1 (offen) variieren.

2. Überführen Sie die zuvor erstellte zeitkontinuierliche Differentialgleichung in eine zeitdiskrete Differenzengleichung unter Verwendung der Ergebnisse aus den Vorbereitungsaufgaben.

3. Realisieren Sie das Tanksystem mittels der Differenzengleichung in einem separaten VI mit dem Eingängen u, h_k, s_V und Δt und dem Ausgang h_{k+1} und speichern Sie das VI als *System.vi*. Neben der Differenzengleichung sollen die folgenden Funktionen umgesetzt werden:

   - Ein Stop-Button.
   - Zwei E/A-Felder für die obere und untere Grenze des Zweipunktreglers.
   - Ein E/A-Feld zur Eingabe der Ventilstellung im Ausfluss.
   - Drei Anzeigeelemente für den Füllstand des Tanks:

     - Standard-Tankgrafik
     - Numerisches Anzeigefeld
     - Verlaufsdiagramm über die Zeit

4. Instanziieren Sie das zuvor erstellte *System.vi* und das *Regler.vi* mit dem Zweipunktregler als Sub-VI's in der in Abb. 10 dargestellten Grundstruktur des Programms und testen Sie die Funktion der Simulation. Die Programmvorlage *Aufgabe2.vi* sowie das *Regler.vi* finden Sie in dem Praktikumsordner „M_Praktikum“ auf „ATPCommon auf Frodo“ unter „Versuch 3 – Programmierung in LabVIEW“ .

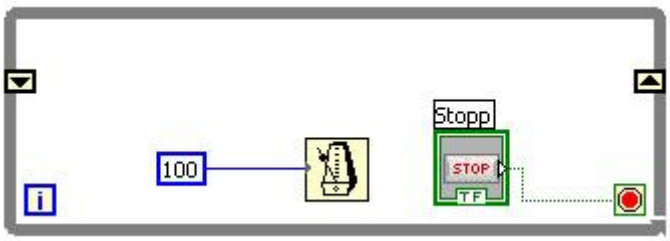

Abbildung 10: Grundstruktur des Programms

Wichtige Hinweise zur der Programmstruktur:

- Der Taktgenerator  hat die Bezeichnung: „Bis zum nächsten Vielfachen von ms warten.“ Dies bedeutet, dass in der Konfiguration, wie sie in Abb. 10 gegeben ist, die While-Schleife alle 100 ms ausgeführt wird. Das entspricht einer zeitdiskreten Ausführung der Rechnungen alle 100 ms, d.h. einer Abtastzeit von 100 ms.
- Die auf der While-Schleife links und rechts liegenden Kästchen sind sogenannte Schieberegister. Diese fehlen in der Vorlage. Es können Schieberegister erstellt werden indem mit der rechten Maustaste auf die While-Schleife geklickt wird und der entsprechende Menüpunkt ausgewählt wird. Diese Schieberegister sichern die Werte von einem Schleifenumlauf zum nächsten.
  Das heißt: Der Wert, der am Ende eines Schleifenumlaufs an der rechten Seite anliegt ist im nächsten Umlauf auf der linken Seite zu finden. Kombiniert mit der durchgeführten Zeitdiskretisierung kann auf diese Weise **mit einer einfachen Addition eine Integration** durchgeführt und der aktuelle Füllstand h berechnet werden.

Speichern Sie die Aufgabe unter *Aufgabe2.vi*!

6 Nachbereitung des Praktikumsversuches

Zur Nachbereitung des Praktikumsversuches ist ein Protokoll nach den Richtlinien anzufertigen, wie sie auf der Lehrstuhl-Homepage zu dieser Veranstaltung angegeben sind. Beschreiben Sie in dem Protokoll kurz die Durchführung der beiden Versuche, wobei in Aufgabe 1 alle Teilschritte zu einem gemeinsamen Ergebnis führen und nicht alle separat zu behandeln sind. In der zweiten Aufgabe sollte die Modellierung des Prozesses, die Umsetzung des Regelkreises zur Simulation in LabVIEW und die Realisierung des integralen Füllstandsverhaltens beschrieben werden.