



Using LabVIEW to Design User Interface and Control Real-Time System

Khanh Hieu Ngo and Quynh Nhu Le

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

October 24, 2022

Using LabVIEW to design User Interface and control real-time system

Ngo Khanh Hieu^{1,2} and Le Quynh Nhu^{1*}

¹ Aerospace Engineering Department, Ho Chi Minh City University of Technology

² VNU-HCM Key Lab. for Internal Combustion Engine

ngokhanhhieu@hcmut.edu.vn, nhu.le.h2t.3t@hcmut.edu.vn

Abstract

LabVIEW - Laboratory Virtual Instrument Engineering Workbench is a graphical application development environment developed by National Instruments Corporation based on the dataflow representation of the “G” language [1][2]. LabVIEW is implicitly multithreaded and has high level functions for communication/synchronization, allowing it to be used as a programming language for control/command and soft real-time applications. From the above features, LabVIEW is a useful tool for designing User Interface and real-time system control.

The purpose of this paper is to present a case-study in the application LabVIEW to control real-time system by using the feature of LabVIEW that is multitasking, parallel processing of many operations, many data sources to control multiple devices at the same time. Beside that, using LabVIEW to design an user interface which is friendly, easy to use, fully functional such as control, measure, operation, ... is also a topic of this paper. Understanding and using typical features of LabVIEW helps designers and operators easily manipulate, save time, and are convenient in use.

1 Introduction

One of the common problems of controlling a real-time system is multitasking, parallel processing of many operations, many data sources. The “concurrency” is one of the problems that we have to face frequently in real time systems. A concurrent system has many activities (or tasks) occurring in parallel.

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical application development environment in the G language. LabVIEW is very well suited for data-acquisition, signal processing, and (soft real-time) control/command of process. The LabVIEW programming language is naturally parallel: when parts of the data flow are independent, the runtime can map them in several

* Corresponding author

system threads. LabVIEW has multi-processing functions, an intuitive interface, and many features to support research and learning [3]. For these reasons, LabVIEW was chosen for control the real-time system. In this paper, LabVIEW is used to control the CVT testbench.



Figure 1.1: CVT Testbench System

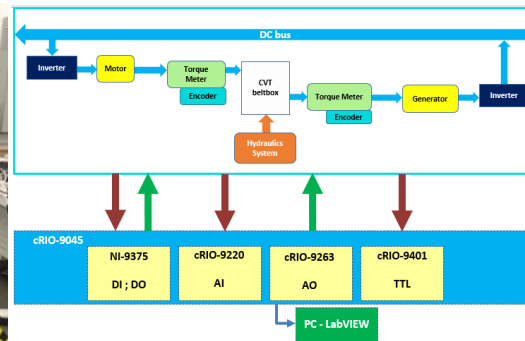


Figure 1.2: Diagram of CVT testbench

A Continuous Variable Transmission, best known as a CVT. A CVT does not have any gears. Instead, it has two pulleys. One pulley connects to the engine, and the other connects to the wheels. A flexible belt connects the two pulleys. CVT L-box is a system that simulates the CVT transmission in a car. CVT L-box has all the functions of a CVT transmission, designed for educational purposes. This system is funded by a lead plant based in the Netherlands. CVT testbench is a system that simulate the operation of CVT.

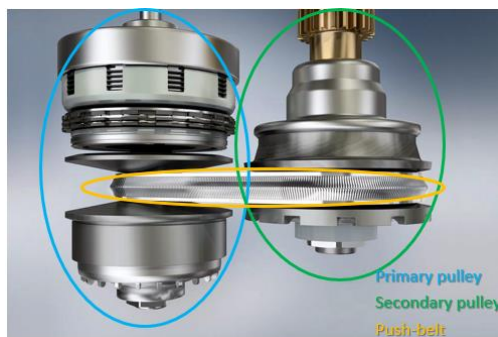


Figure 1.3: Structure of CVT

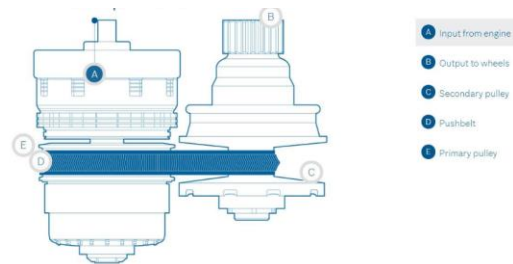


Figure 1.4: Structure diagram of CVT

2 Structure of CVT testbench and design requirement

Structure of CVT testbench: 1 Motor control by inverter (inverter no.1), 1 Generator control by inverter (inverter no.2), 1 CVT belt-box, 2 Inverter, 2 Torque Meter, 2 Encoder , 1 Hydraulic system include 2 proportional valve, 3 directional valve and 2 pumps.

Control CVT testbench means control inverter no.1 to reach at desired speed, control inverter no.2 to reach at desired torque and control both pumps and proportional valves to create desired

transmission ratio. These control activities had to happen at the same time, so it is suitable with LabVIEW function.

Beside that, multitasking on LabVIEW is used to design system states.

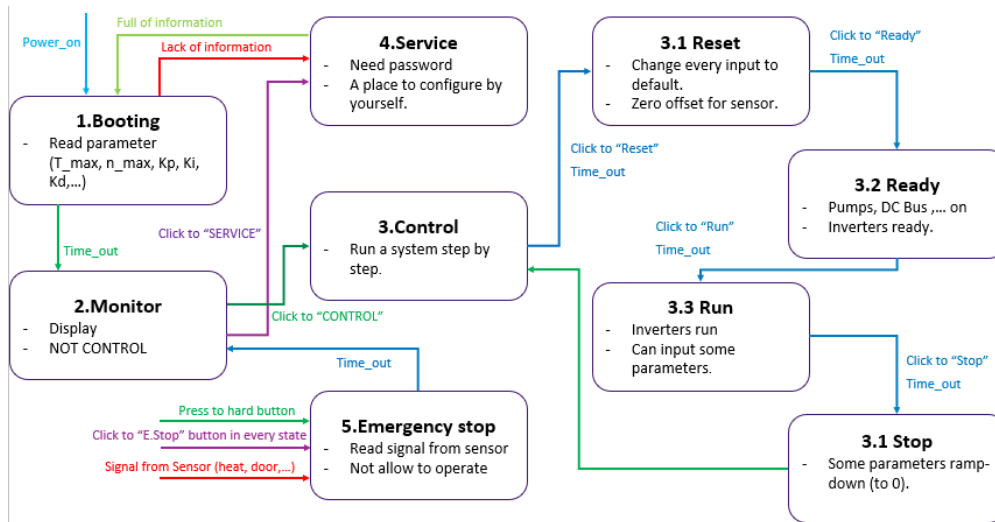


Figure 2.1: State of CVT testbench system

The system has 5 major states and sub-states (time_out).

State 1: Booting. This state will read some parameters (such as T_{max} , n_{max} , K_p , K_i , K_d , ...). If lack information, lack specs, UI will display the lack field => Request password to go to state 4 "Service" to fill the field. If full information, the system will message that booting done and change to state 2 "Monitor".

State 2: Monitor. User cannot control in this state. User interface will display the status of whole system.

Sub-states 2.1: Time out 2.1. This is a delay state to change status of system. The system has to meet the necessary conditions to change to state 3 or state 4.

State 3: Control This is a control state with 4 steps, each step will have 1 sub-step.

State 3.1: Reset: Reset whole system (reset status of inverter, reset current value to zero).

Sub-step 3.1: Time out 3.1.1: Ready 2 inverters, turn on DC bus.

State 3.2: Ready.

Sub-step 3.2: Time out 3.2.1: change inverters state to run, turn on pumps.

State 3.3: Run: The system run to reach at setpoints (speed, torque and transmission ratio setpoint).

Sub-step 3.3: Time out 3.3.1: First, ramp-down torque to zero. Second, ramp-down speed to zero. Third, turn off pumps.

State 3.4: Stop.

State 4: Service. This state allow user to configure the system by themselves.

State 5: Emergency stop. The system has 3 ways to change to state 5 emergency stop. First, user can use hard button on the system. Second, user can click to emergency stop button on user interface. Third, when errors occurs (such as: overheat, door open when operating ,...) the system will emergency stop.

3 Design user interface and control algorithm

Base on design requirement, the user interface and control algorithm were built-up.

3.1 Graphical User Interface (GUI)

Section Graphical User Interface (GUI) was designed not only for users who want to control the system but also for the next generation students who want to develop a system and do some experiments or configure the system by themselves. So that, the GUI need to have a friendly interface, easy to control, display fully information.

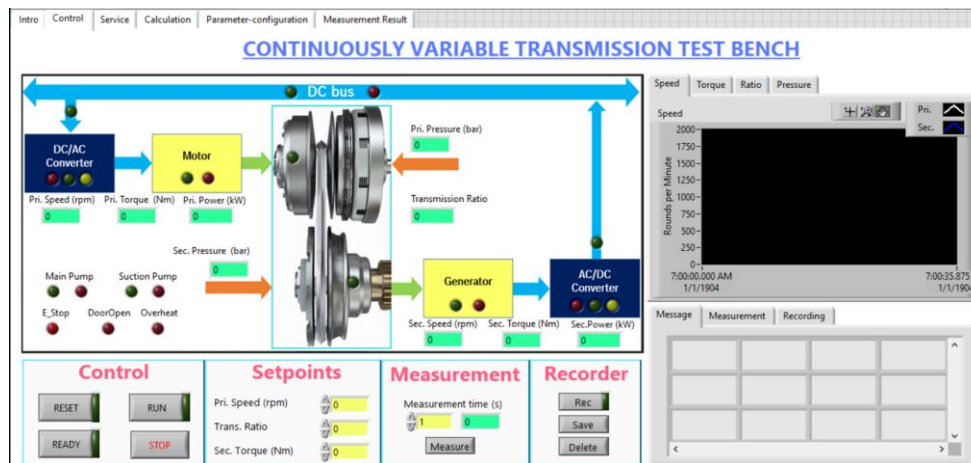


Figure 3.1: Graphical User Interface

The first one is “Control tab” for controllers. this tab has 7 main areas with different functions.

The area Number 1 is called “Monitoring area”. in this area we have some led lights to monitor the system. User can see the status of DC bus – if it is ON or OFF, similar to the motor and the Inverter (AC/DC converter), ... And the system has some sensors to measure speed and torque of these motors, so user can see some data in this area.

The area no.2 is called “control area”, students can select the status you want to run.

The area no.3 is “setpoint area”, students can put a desired number student want system to run.

The area no.4 is measurement area, this allows students to measure data.

The area no.5 is recording area, this allows user to students to record data.

The area no.6 is message area, this area will display real-time message. It helps students to read the operation’s diary. Moreover, in case of having errors, it will give users a warning.

The area no.7 is graph area, it will draw a graph when the system works.

Besides, there is also an extra area for next generation students and developers. This area allow students to observe all signals and configure the signal they want. Students can convert the signal,

change some gains or offsets number... each number students put in this area can affect the whole system, that is the reason why this area need to set a password.

3.2 Control algorithm

This software has 3 main control systems.

Speed control, torque control and transmission ratio control. To design these 3 controllers, using PID controller algorithm. These controllers had to happened at the same time and LabVIEW has a tool for PID so designer need to define input, output and suitable K_p , K_i , K_d .

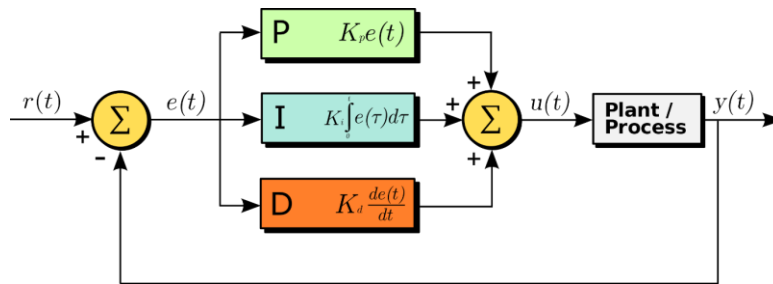


Figure 3.2: PID control algorithm

The input of speed controller is speed demand for motor 1, then PID control will process the signal and convert the set speed in rounds per minute into signal from 0 to 10 voltage, then through belt-box and flywheel, it also has a feedback signal, then the system will receive the output is actual speed of motor 1.

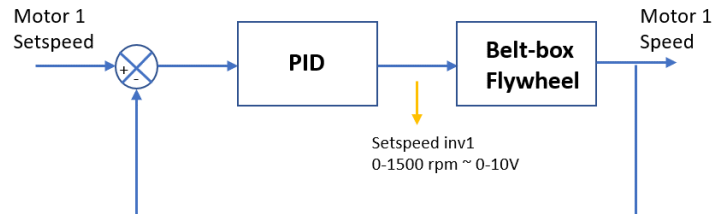


Figure 3.3: Principle diagram of Speed controller

The input of torque controller is torque demand for motor 2, then PID control will process the signal and convert the set torque in Newton-meter into signal from -10 to 10 voltage, then through belt-box and flywheel, it also has a feedback signal, then the system will receive the output is actual torque of motor 2.

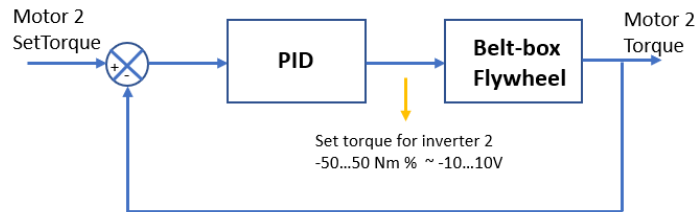


Figure 3.4: Principle diagram of Torque controller

The input of transmission ratio controller is ratio demand for whole system, then PID control will process the signal and convert the set ratio into signal of valve opening then convert into -10 to 0 voltage, then through belt-box, flywheel and hydraulic system, it also has a feedback signal, then the system will receive the output is actual transmission ratio.

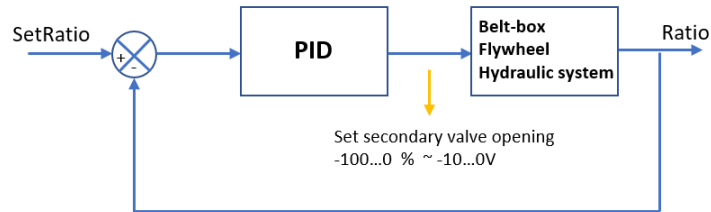


Figure 3.5: Principle diagram of Transmission ratio controller

4 Using LabVIEW to control system

LabVIEW allow multitask, use multiple conditions at the same time so the design of the control algorithm becomes easier and more convenient.

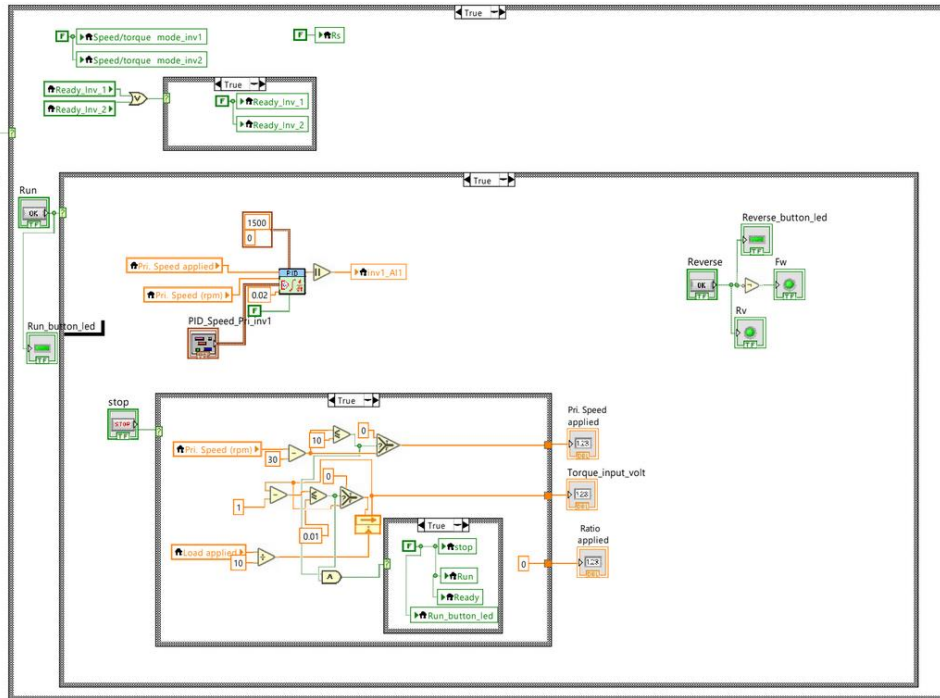


Figure 4.1: Example for LabVIEW code

For example: This is a diagram for “Area no.2 – Control area”.

First, create a “Ready” button. LabVIEW permits both work in the same time such as: design a “Front panel” – User interface and build a “Block diagram”. In this case, making a “Ready” button like a condition for “True/False case structures”. When “Ready” is ON, it will be “True condition”, then system will do all activities that controller set-up in true case. When “Ready” is OFF, it will be “False condition”, then system will do all activities that controller set-up in false case. About some task which appear in “True case” but don’t appear in “False case” (or vice versa), these tasks will maintain the previous state.

5 Conclusion and evaluation the results

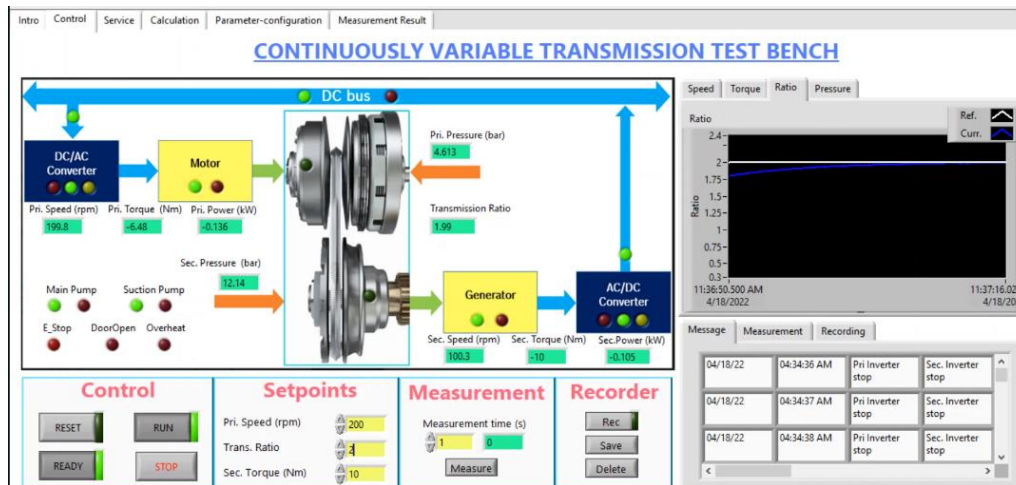


Figure 5.1: Result of operation

When input setpoints about 200 rounds per minute for primary speed, 2 for transmission ratio, 10 Newton meter for secondary torque. The result is:

About speed: reach desired speed after 5 seconds, maintain that speed. Maximum error is 0.2 percent.

About torque: reach desired torque after 15 seconds, maintain that torque. Maximum error is 2 percent.

About ratio: reach desired ratio after 30-60 seconds, maintain that ratio. Maximum error is 0.2 percent.

The monitoring system can monitor all equipment and monitor the operation process, reporting errors when problems occur. The function of measuring, sampling, recording the operation process also works well. The control function controls the operation of the whole system, controlling the operation of each device separately. The control process error (for speed and gear ratio) is as low as 1%. Fairly finished user interface, the interface has friendly interface, easy to use, operate and monitor the system. The interface is not only for general users but also aimed at students who need to study more about the system, there is an open space to customize, do some experiments for teaching and learning purposes.

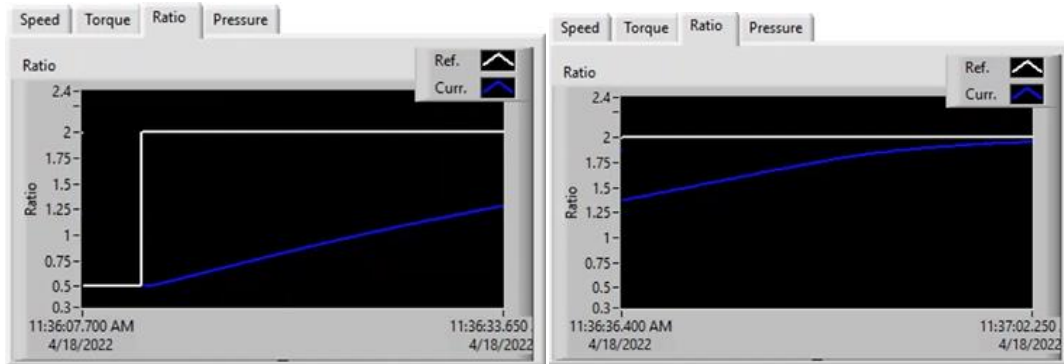


Figure 5.2: Characteristic when change ratio from 0.5 to 2.2

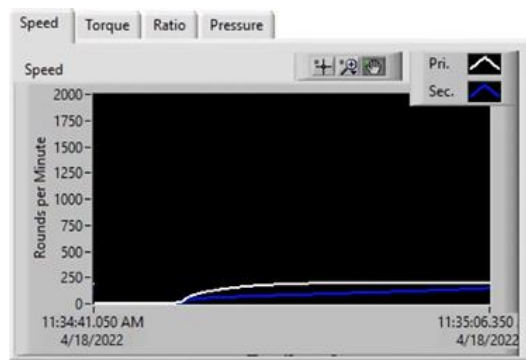


Figure 5.3: Speed's characteristic

5.1 Further development

Based on what has been achieved, there are some future developments for the topic:

- Added feature to simulate the road load.
- Change the gear ratio continuously or according to an available diagram to review performance, system response.
- Simulate belt slip.
- In addition to the feature of loading parameters for each set of belts, now only one set of belts is used, so there is only 01 set of parameters, add this feature if more belt sets are available.

Acknowledgement

This paper is the result of effort and research at the Bosch Automotive Lab. On campus, which is the collaboration between Ho Chi Minh City University of Technology (HCMUT) and Bosch Vietnam CO., LTD. This research is funded by VNU-HCM under grant number C2021-20-11. The authors acknowledge the support of time and facilities from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this paper.

References

- Bosch Mobility channels*. (2022). (BOSCH) Retrieved 2022, from <https://www.bosch-mobility-solutions.com/en/solutions/transmission-technology/dh-cvt-transmission/>
- Building a PID Controller with Your Computer* (2021). [Motion Picture].
- EN | Bosch Continuously Variable Transmission* (2013). [Motion Picture]. Retrieved 2013, from <https://www.youtube.com/watch?v=dp8IHBDFhpo>
- EN | Bosch Pushbelt - Efficiency on the Road* (2013). [Motion Picture].
- Enang, W. P. (2015). *ResearchGate*. Retrieved from https://www.researchgate.net/figure/Engine-fuel-consumption-map_fig3_273760159
- F. Cottet, E. G. (2005). *Systèmes temps réel de contrôle-commande, Conception et Implémentation*, Dunod, 561.
- Instruments, N. (1998). LabVIEW™ 5 Software Reference and User Manual.
- Jha, S., & R., A. G. (2015). Continuously Variable Transmission Control Strategy Review. 1-5.
- NGO Khanh Hieu, G. E. (2009). DARTSVIEW, A TOOLKIT FOR DARTS IN LABVIEW. *SCIENCE & TECHNOLOGY DEVELOPMENT*.