

Development of a (U-Zr)N- Fuel Synthesis Control and Data Acquisition Program in LabVIEW

Gustaf Holst
Ludvig Thorsen



Bachelor Thesis in Reactor Physics
Stockholm, Sweden, 2013

This page intentionally left blank.

ABSTRACT

In this thesis, software for control and data acquisition was developed. The main purpose of the software is to provide overview and control of uranium and zirconium nitride syntheses. This includes connecting multiple gas flow controllers and a furnace to the software via a DAQ-module.

The developing platform was LabVIEW, a graphical programming language, which offers both great challenges and great advantages. The advantages of easily configurable programs need to be weighed against the difficulty to organize and maintain.

The resulting software lives up to most of the requirements, however the control engineering of the furnace was much more challenging than first expected. There are still many improvements that can be made but the end product is fully functional, and provides the ability to simultaneously monitor and control synthesis reactions with multiple gases and reaction environments.

This page intentionally left blank.

SAMMANFATTNING

I denna avhandling, har programvara för styrning och datainsamling utvecklats. Det huvudsakliga syftet med programmet är att ge överblick och kontroll av uran- och zirkoniumsynteser. Detta inkluderar anslutning mellan ett flertal styrenheter för gasflöde, en ugn och programvaran via en DAQ-modul.

Plattformen var LabVIEW, ett grafiskt programmeringsspråk, som erbjuder både stora utmaningar och stora fördelar. Fördelarna med lätt konfigurerbara program måste vägas mot svårigheten att organisera och underhålla.

Den resulterande programvaran lever upp till de flesta kraven, men reglerteknik av ugnen var mycket mer utmanande än först väntat. Det finns fortfarande många förbättringar som kan göras, men slutprodukten är fullt fungerande, och ger möjlighet att samtidigt övervaka och kontrollera syntesreaktioner med flera gaser och reaktionsmiljöer.

This page intentionally left blank.

FOREWORD

We would like to thank reactor physics department and in particular Waclaw Gudowski for making this thesis possible in the first place. Also thanks to Mikael Jolkkonen for providing an interesting and educational project. Last but not least we would like to thank Kyle Johnson for all his guidance and support, it has been invaluable to us.

Ludvig & Gustaf

This page intentionally left blank.

TABLE OF CONTENTS

ABSTRACT	I
SAMMANFATTNING	III
FOREWORD	V
TABLE OF CONTENTS	VII
1 INTRODUCTION	1
1.1 BACKGROUND.....	1
1.2 PURPOSE.....	1
1.3 LIMITATIONS.....	1
2 NITRIDE SYNTHESIS	3
2.1 NITRIDE FUELS.....	3
2.2 THE SYNTHESIS.....	3
3 HARDWARE	5
3.1 THE FURNACE.....	5
3.2 THE FLOW CONTROLLERS.....	5
3.3 THERMOCOUPLES.....	6
3.4 THE CDAQ.....	6
4 THE SETUP	7
5 THE PROCESS CONTROL AND ACQUISITION SOFTWARE	9
5.1 PROBLEM DESCRIPTION.....	9
5.2 DEVELOPMENT ENVIRONMENT.....	9
5.3 DEVELOPMENT.....	9
5.4 RESULT.....	11
5.5 DISCUSSION.....	11
6 DISCUSSION	13
6.1 DISCUSSION.....	13
7 BIBLIOGRAPHY	15
APPENDIX A: USER MANUAL	17

1 INTRODUCTION

1.1 Background

Nuclear power generation facilities around the world produce around 200 000 m³ of LILW¹ and 10 000 m³ of HLW² each year worldwide [1]. The only option of managing this waste today is to store it for magnitudes of 10⁵ years until radioactivity has subsided to natural levels.

A proposed solution is to transmute and burn the waste in a fast neutron spectrum in Generation IV nuclear reactors. In order to do this, uranium is transformed into a nitride instead of an oxide, as is done in conventional reactors. This results in a harder neutron spectrum, making it possible to fission isotopes such as ²³⁸U, ²⁴⁰Pu, ²⁴¹Am etc. The waste from these reactors will be dangerous for magnitudes of 10³ years instead [3].

GENIUS [2] is a collaboration between KTH, Chalmers and Uppsala University aiming at developing technology necessary for safe and economic deployment of Generation IV nuclear reactors. This thesis is a part of that collaboration.

1.2 Purpose

The purpose of this thesis is to explain the different stages in the development of a LabVIEW control and data acquisition program. This includes controlling and acquiring data from multiple gas flow controllers and controlling and measuring the temperature of a furnace. It also includes presenting the data in an accessible way that can easily be presented at a later stage. The main purpose of the program is to monitor and control manufacturing of UN and ZrN powders. These powders are one part in the production line of nitride fuel pellet manufacturing for Generation IV nuclear reactors [4].

1.3 Limitations

Developing this kind of control and data acquisition software gives huge potential for making near to endless different kinds of extra functions such as various alerts, monitors, live calculations and integration of external hardware. Since this project is meant to span over one term at 20 hours per week, however, certain limitations need to be set.

The thesis is hence limited to making software for control and data acquisition of three flow controllers and one oven. The data will be collected, graphed and logged. The user will have the ability to set up ramps and gas flows prior to the synthesis run.

Consequently, the thesis will not delve deeply into the field of control engineering.

¹ Low and Intermediate Level Waste

² High Level Waste

2 NITRIDE SYNTHESIS

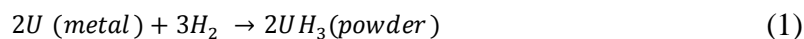
2.1 Nitride fuels

Uranium nitride is chosen for two reasons: the first one is that the thermal conductivity is very high, i.e. the difference between the inner and outer fuel temperature of the pellet is very small. This allows for higher operating temperatures. The second reason is that the neutron economy is much better. Uranium nitride is a far less effective moderator than UO_2 , therefore a fast neutron spectrum can be maintained. More importantly it gives a higher margin for fuel melt [5].

2.2 The synthesis

This section will briefly explain to the reader the theory behind the part of the manufacturing process that the software controls. The goal is to create uranium nitride powder. The same process can be used to create zirconium nitride powder.

A uranium metal sample is placed in the furnace and by the following chain of reactions, it is transformed to a nitride powder [4].



In reaction (1), called *Hydriding*, the uranium absorbs hydrogen in an argon atmosphere at a temperature of 300 °C. Since the density of uranium hydride is much lower than for uranium metal, the UH_3 expands and cracks which leads to the formation of UH_3 -powder.

The argon is necessary to provide an inert atmosphere and to prevent backflow when H_2 is absorbed, and to prevent O_2 ingress. To further prevent H_2O or O_2 ingress, all gases are first put through a heater that purifies the gas. If any oxygen would slip into the system it would, very exothermically, react with the uranium. Therefore keeping a high purity of the gas is paramount.



In reaction (2), called *Nitriding*, the hydrogen is driven from the system, and replaced by nitrogen, at a temperature of ca. 300°C. At this stage all the uranium is in the form of powder and hence is much more reactive.



The last reaction (3) is called *DeNitriding* and here U_2N_3 forms into the UN at a temperature of 1150 °C. Again the argon atmosphere is present for the same reasons as in reaction (1).

The last stage is to compress the powder into a pellet which can then be analyzed as part of its development and qualification as a potential nuclear fuel.

3 HARDWARE

3.1 The furnace

The furnace integrated into the system has a range up to 1200 °C. A quartz, alternatively alumina, tube containing the actual reaction is mounted inside. The furnace is powered by a thyristor giving 50A and 48V which gives a total output power of 2.4 kW. This thyristor is controlled by a voltage ranging between 0-5 v giving a percentage of output power between 0-100%. The voltage is controlled by the DAQ-module connected to the computer.

There is no control system integrated into any part of the oven. This is done by PID-controllers integrated into Labview. PID stands for Proportional Integral Derivative and is a big subject in control engineering [6].

- The proportional term simply takes the difference between the set value and the actual current value and multiplies it with a constant. This gives stronger or weaker response to the error depending on the magnitude of the set constant.
- The integral term adds all the errors over time. This term is meant to take into account the steady-state error.
- The derivative term responds to the rate of change of the error. This term is meant to flatten out the response curve and minimize overshoot.

When the furnace is properly insulated the heating rate is much higher than the cooling rate and both vary depending on the furnace temperature. This makes it very hard to find proper constants for the PID-control over all temperatures. Fortunately, the thermal inertia of the system means that PID parameters need not to be so accurate.

3.2 The flow controllers

There are three flow controllers that must be integrated with the software. Two of these controllers are responsible for both monitoring and controlling the gas while the last one will only monitor the gas. The communication with the DAQ device is analog, which means only a voltage difference is read. It ranges from 0-5 volts and to help the user interpret these values, equations from the manufacturers are supplied. Some parameters needed in these equations, such as the pressure, and gas content in the system are up to the user to provide. From these equations the user can discern what gas flow is currently running through the system. This also applies when inputting the desired gas flow in the system; the same equations are used but now the flow is known rather than the voltage.

These flow controllers are calibrated for a specific gas and the equations are not guaranteed to work correctly if a different gas is running through the controllers. To solve this problem the manufacturers have a calibration homepage where a calibration factor can be calculated which can be applied to the equation in order to find a correct value for the flow.

3.3 Thermocouples

A thermocouple is used to measure the current temperature in the furnace. The physics behind a thermocouple are governed by a potential difference being created between the hot and cold ends of a metal wire being exposed to different temperatures [7]. Different materials give different potentials and by twinning two materials against each other in an insulated way, a potential difference can be measured which corresponds to the temperature they are being exposed to.

The current thermocouple being used is a K type. A K type thermocouple is the most common one and is made by two conductors; one being an alloy of nickel and chromium, Chromel, and the other an alloy of nickel and aluminum, Alumel. It is best used in the range from -200 Celsius to 1350 Celsius. For this project it could put some strain on it because normal operations are not far from its top temperature. A better solution would be to use a platinum-based thermocouple but they're more expensive.

3.4 The cDAQ

A National Instruments cDAQ 9174 is used to handle the signal conversion between the analog and digital signals used to communicate between the software and the different hardware components. It is a modular device, i.e. it allows for different kinds of input and output setups. The cDAQ is connected to the computer by USB.

In this setup three modules are used:

- **NI 9201** – Receives analog signals in the range ± 10 V.
- **NI 9211** – Receives analog signals in the range ± 80 mV, used for thermocouples.
- **NI 9263** – Transmits analog signals in the range ± 10 V.

The cDAQ and LabVIEW handles all the digital to analog and analog to digital conversions. When called in LabVIEW the cDAQ returns all values in relevant units, e.g. the NI 9211 returns a temperature, not a voltage.

4 THE SETUP

This section explains how the hardware is setup and connected.

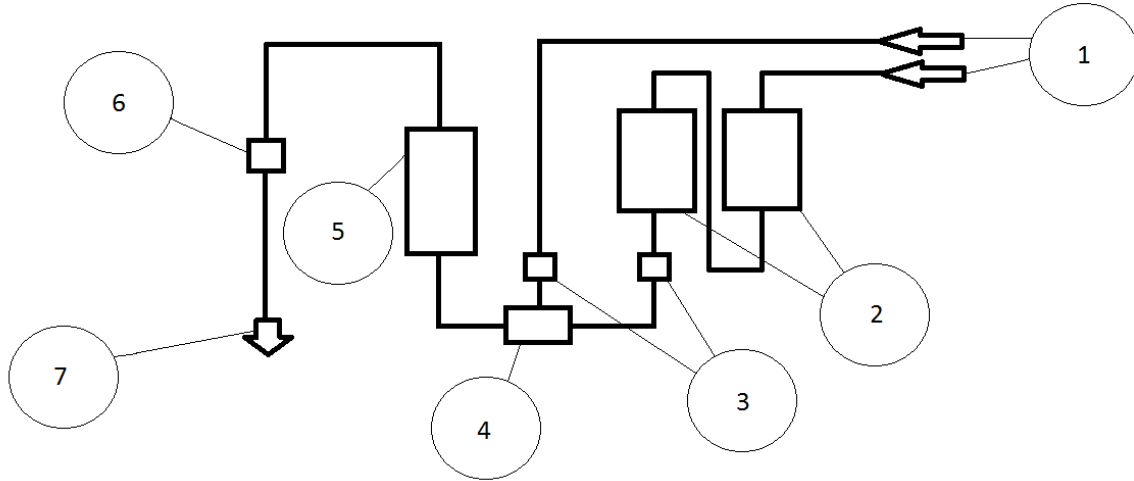


Figure 1 – The hardware setup.

Figure 1 is a simplified overview of the hardware setup. The components are described briefly below.

1. Gas inlet. The upper one for hydrogen and the lower one for argon or nitrogen.
2. Heated absorbers that remove unwanted molecules from the gas, e.g. oxygen.
3. Flow controllers. These both monitor and control the flow.
4. Gas mixer. If two gases are used they will mix here.
5. Furnace. This is where the (U-Zr)-pellets react with the gases.
6. Flow monitor. This only monitors the flow.
7. Gas outlet. This is where the gas exits the system.

5 THE PROCESS CONTROL AND ACQUISITION SOFTWARE

In this chapter the program for the system will be explained in more depth.

5.1 Problem description

The software must be able to control and monitor the UN and ZrN powder production processes, which could, for example, involve controlling the gas flow and measuring the furnace temperature. The software must be able to read instructions from a tabular input where the user specifies the ramps to be run. An example input is shown in the table below.

Table 1 - Example program input.

Gas flow 1 (l/s)	Gas flow 2 (l/s)	Start temp (C)	End temp (C)	Heating rate (C/s)	Time (s)
Argon 0.4	Hydrogen 0.2	20	100	0.66	120
Argon 0.2	-	100	100	0	100

When the software is running it needs to monitor the process values continuously and watch for any extra ordinary events. If the unexpected happens it needs to inform the user or, in the worst case, abort the program. At the end of a run a log file is generated where all the acquired data is written to the hard drive. It is important that, even if the run was not finished properly or aborted, a log file is written down so the user can troubleshoot effectively. The program also needs to support loading a log file to display valuable information from a previous run.

5.2 Development Environment

The program used for the development of the software is LabVIEW. LabVIEW is a program that is easy to work with and is widely used. A lot of the functions that are needed to solve these problems are already made available and it has support for communication with a lot of devices. This made it easy to integrate both the gas control system and furnace. Even more advanced problems such as controlling an oven could be performed by LabVIEW. This would normally require a deep understanding of control engineering but it was possible to simply follow the instructions supplied in LabVIEW.

5.3 Development

Similar software had already been developed by a former master student [4]. That software was on a computer that had crashed. More information about the problem and how the previous developer had solved it seemed interesting so some effort was put into retrieving the old software. The previous versions were not written in LabVIEW but instead in DASyLab. It shares similarities but a decision was made that it was not worth the time to learn DASyLab for a possible head start in the development of the new software.

Because manufacturing had been run before, the flow controllers and the furnace was already setup. Wires from them were connected to the old AD/DA device hooked up to the old computer rig. Since the old computer was no longer functional the wires needed to be reorganized. Fortunately the AD/DA device, which was heavy and bulky, were to be replaced by a modern AD/DA device that could be connected to a much more agile laptop.

Documentation on how the wiring was done to the old AD/DA device was available and a rewiring to the new AD/DA could be done. To get acquainted with LabVIEW and its interactions with the hardware, several smaller test programs were made. By doing this, experience with both the hardware and LabVIEW was obtained which made it a lot easier to plan and organize the new system effectively.

Wiring the flow controllers to the computer was a straight forward process and with the help of analog instruments a double-check that the correct gas flows received and sent were made. The oven was far more complicated than first thought. It was remarkably easy to get a temperature read in the oven with the thermocouple but to set a temperature, and in particular a temperature ramp, proved more difficult. The difference between sending a voltage to the oven instead of a flow controller is that for the controller there is a linear relationship between voltage and gas flow, while the oven is operated by power. As an example, if the oven is going full power at 20 degrees Celsius it would rapidly increase in temperature but if full power is set when it is at 1200 degrees Celsius it may be barely enough to keep the oven at that temperature.

Although the oven will asymptotically reach a certain temperature for each power, the important factor for the oven is the heating rate, Celsius per minute. If a linear relationship were used it would not keep up with the heating rate. This is a well-known problem and there is an entire field devoted to this, namely control engineering. Unfortunately one of the greater difficulties in this field is to control an oven, because it has inertia. If power ceases to be supplied to it, it may still increase in temperature for some time before cooling. After some research in this area and help from the local lab engineer, it was determined that a PID controller would solve this. A basic PID controller was already supplied in LabVIEW and the only thing the controller needed was the PID parameters. A PID controller from Turkey was ordered to calculate these parameters for the oven. Unfortunately these parameters were deemed to be too bad, but acted as a base for a trial and error approach. In the end acceptable parameters were found.

In order for LabVIEW to ensure that each loop runs smoothly on the computer, it automatically lowers the loop per second count. This posed a great problem when implementing the code in the lab, since the synthesis laptop did not have the same computational power as the computers the software was developed on. Many time dependent functions were based on loop time which made the values come out completely wrong. In order to fix this all functions were reformatted into true time dependence instead, i.e. calculations were based on the computer clock instead of the assumed loop time.

5.4 Result

The program that has been delivered has support for most of the functionality initially requested and a lot of extra functions that were requested during development. It has been an iterative process with a lot of back and forth feedback from the reactor physics institution.

In its current state the furnace control is very precise and the temperature differs only a few degrees Celsius from the set temperature, but if the program orders the furnace to stay at a fixed temperature, it will instead fluctuate around it. This implies that it is preferred that the furnace be on all the time and that the output voltage is more delicately calculated, rather than switching completely on/off. This behavior is controlled in the program by three PID parameters and it is something the user can test out over time, so it will work much better in the future when even more accurate PID parameters are determined.

A missing feature in the current program is that no security action is made automatically. For example, if the user is running a program with a heating rate that is higher than what the quartz tube supports, the program won't warn the user. An indicator with the current heating rate is displayed in the program though, so if the user knows the system limitations an action can be taken accordingly.

The data to be logged in a run is stored in the RAM and is first written down to the hard drive after a run is done. This is fine as long as the program isn't interrupted prematurely. But if the computer crashes, the data from the run will be lost. If, instead, the data were to be written to a log file right away, and appended continuously to it, the data would be saved up to the crash.

5.5 Discussion

A big difference when developing a program in LabVIEW, compared to conventional programming languages, is that it is a graphical programming language. It has its benefits, such as the ability to quickly create a new working project, and start to read from hardware components right away. The disadvantages, however, are that the framework for how to develop a program is quite fixed. The developer does not have much freedom when choosing different ways to solve problems. Managing big projects in LabVIEW is also hard because splitting it up into reasonable modules can be tricky. This makes it tough for a new developer to understand an old project.

It was probably a good thing that the difference between the computational power of the laptop in the lab and the computer the program was developed on was so big. Otherwise it might not ever have been found out that LabVIEW changes the loop per second count depending on computer. This could have given rise to small errors that would have been hard to find the source of.

6 DISCUSSION

6.1 Discussion

Over all, the work has been pleasant and very educational. Many areas of engineering have been visited, from electric and computer engineering to nuclear engineering. The most fulfilling part is that the software that was created will actually be used and is valuable to other people. There are many things, however, that could have been done differently and many improvements can still be made.

Controlling the furnace turned out to be much harder than previously thought. A deep understanding in the field of control engineering is necessary in order to solve the problem effectively. The design that is now implemented simply puts the furnace on 100% power if the temperature is under the set temperature and turns it off completely if it is above. This is not optimal. It was hoped for a controller that could put out a voltage that would correspond to the desired temperature. Although it would be possible to find a relationship between voltage and asymptotical temperature i.e. the temperature that the furnace will eventually stabilize at, the wide operating temperature makes this impossible without taking days to reach it, or exceeding the heating rate limits of the quartz tube.

Another problem is that there is a lot of dead volume in the system, so after changing gas it will take some time before all of the previous gas is flushed out. This is very problematic because the accuracy of the flow controllers is very dependent on gas type. Adding to the problem is that reactions happen inside the furnace e.g. ammoniac is produced, so even less is known about the gas composition in the out flow.

One way of solving this is to connect a mass spectrometer close to the flow controllers and measure the actual gas composition. A mass spectrometer was available but it was installed too late to integrate into the system. This would, however, only partly solve the problem since most of the dead volume lies before the first flow controllers, and hence the gas composition would have to be measured at two points to get an accurate reading from all flow controllers.

There are some lesser errors as well. When calculating the gas calibration constants the pressure difference has been ignored but it's in the magnitude of 1%. The flow controller has a measurement error of about 1% as well.

7 BIBLIOGRAPHY

- [1] International Atomic Energy Agency "Managing Radioactive Waste"
<http://www.iaea.org/Publications/Factsheets/English/manradwa.html>
- [2] "GENIUS" <http://genius.kth.se/>
- [3] J. Magili, V. Berthou, D. Haas, J. Galy, R. Schenkel, JW Wiese, G. Heusener, J. Tommasi, and G. Youinou. "Impact limits of partitioning and transmutation scenarios on the radiotoxicity of actinides in radioactive waste", 2003
- [4] T. Hollmer, "Manufacturing Methods for (U-Zr)N-Fuels" Master's Thesis, Technische Universität München, 16 May, 2011
- [5] K. Johnson, "Formation of Solid-Solution in Mixed Nitride Fuels" Master's Thesis, Kungliga Tekniska Högskolan, Spring 2012
- [6] Bennett, Stuart, "A history of control engineering, 1930-55", 1993
ISBN 987-0-86341-299-8
- [7] ASTM Committee E20 on Temperature Measurement, "Manual on the Use of Thermocouples in Temperature Measurement (4th Edition)", 1993

APPENDIX A: USER MANUAL
