TEMPERATURE PREDICTION AT FAST USING LSTMS

Silvia Peiro
SIST Internship
Jinhao Ruan

August 11, 2022

Abstract

The temperature of the water entering the electron gun at FAST needs to be regulated within specific parameters for adequate phase stability. Due to different characteristics of the system, it takes too long for the phase stability to be reached. This motivated the creation of a model to predict the oscillations in that temperature, so that operators would be able to know what settings would achieve the phase stability faster.

1 Background

The electron gun at the Fermilab Accelerator Science and Technology facility (FAST) is a RF photoinjector powered by a klystron, a tube in which electrons are produced by electric fields. For adequate phase stability, the temperature of the water entering the gun should be regulated to within 0.02 ºC ($Edelen\ etal.$,2016). The variables that can be controlled to accomplish this phase stability are the flow control valve setting and the heater power setting.

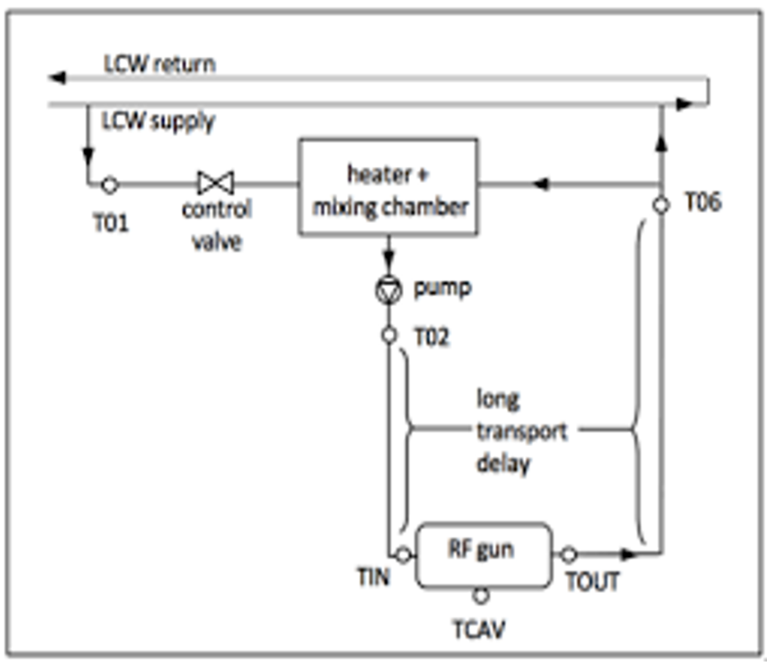

Figure 1: : Layout of the gun water system. T01, T02, TIN, TCAV, TOUT, and T06 are temperature sensors

The temperature takes a lot of time to stabilize for different causes:

-Large thermal and transport time constants make the temperature going out of the heating chamber different that the one that arrives to the cavity.($Edelen\ etal.$,2016).

-A feedback loop makes part of the water exiting the gun to circulate back to the heating chamber and affect the overall temperature of the water, which causes oscillations in the temperature of the water entering the gun.($Edelen\ etal.$,2016).

-There are fluctuations in the low conductivity water (LCW) supply temperature. While it is regulated to within ± 0.5 °C, larger spikes do occur, especially during operation of other large heat sources in the wider system at FAST. $(Edelen\ et al.$,2016$)$.

-The pipes through which the water flows are not insulated and pass through several different areas of the building (close to a fan or nearby equipment that releases heat) which results in a change of the temperature.$(Edelen\ et al.$,2016$)$.

-Due to the TCAV sensor location and the cavity geometry, the temperature recorded there will be higher than the real bulk cavity temperature under RF power.$(Edelen\ et al.$,2016$)$.

For these reasons, a predictive model was developed to predict the temperature of the cavity by getting the flow control valve readings and the heater power readings as inputs. This model would allow the operator to know what settings would cause less oscillation or more oscillation in the temperature and achieve the phase stability faster.

2 Development of the Model: LSTMs

The project was developed in pytorch, using LSTMs, long short-term memory networks, a type of recurrent neural network. A NN consists of a collection of functions with weighted connections between them. These weighted connections can be adjusted or trained until a desired output behavior is achieved, typically through an automated optimization procedure. A recurrent neural network, specifically, recognizes data's sequential characteristics and uses patterns to predict the next scenario. The control process of an LSTM resembles that of a recurrent neural network. It processes data passing on information as it propagates forward, but with different operations within the LSTM's cells. These operations are used to allow the LSTM to keep, forget or update information.$(Phi$,2018$)$.

The core concept of LSTMs are the cell state, and its gates. The cell state transfers information along the sequence chain. It works as the "memory" of the network. Information is added or removed to the cell state via gates as it advances the sequence$(Phi$,2018$)$. The gates decide which information is allowed on the cell state and can learn what information is relevant to keep or forget during training.

There are two different activations that regulate the information in each gate of the LSTM: the tanh and the sigmoid.$(Phi$,2018$)$.

The tanh function squishes values to always be between -1 and 1:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{1}$$

This prevents the vector values from reaching very high numbers in their transformations (which could cause others to be insignificant).$(Zarzycki$ &Ł$awrynczuk$,2021$)$.

The sigmoid activation is similar to the tanh, with the difference that the values are squished between 0 and 1:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{2}$$

That is helpful to update or forget data because any number multiplied by 0 is 0, causing values to disappears or be "forgotten." $(Zarzycki$ &Ł$awrynczuk$,2021$)$.On the other hand, any number multiplied by 1 is the same value, therefore, that value stays the same or is "kept." The network can learn which data is not important and forget it or which data is important and keep it.$(Zarzycki$ &Ł$awrynczuk$,2021$)$.

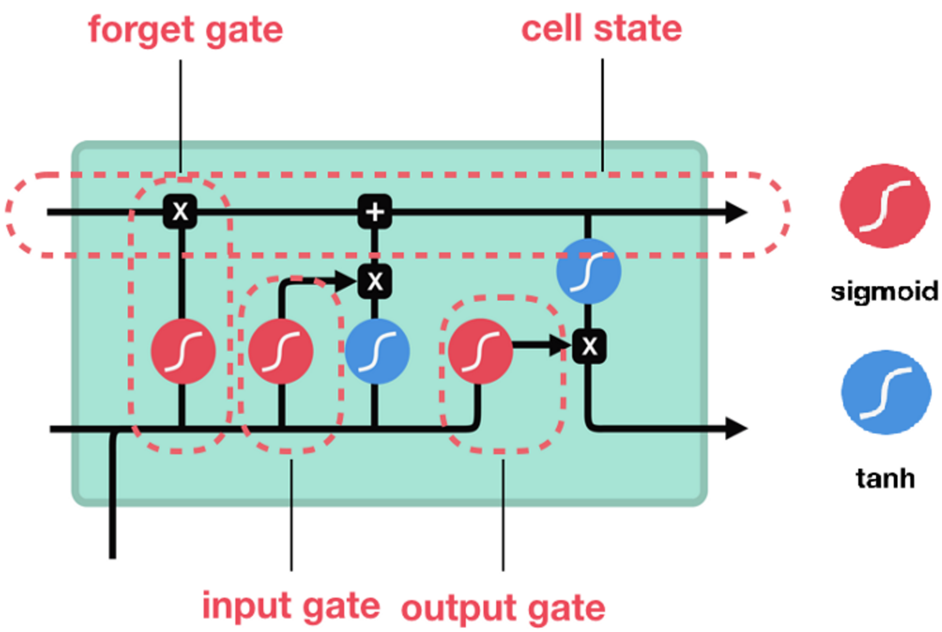

Figure 2: : Layer of an LSTM with its indicated gates

The first gate is the forget gate. This gate decides what information should be kept or eliminated. Information from the previous hidden state and from the current input is passed through the sigmoid function. The values come out between 0 and 1. If they are closer to 0 they will be forgotten while if they are closer to 1, they will be kept.(Phi,2018).

In order to update the cell state, there is the input gate. The previous hidden state and current input are passed through a sigmoid function, which decides what values will be updated.(Phi,2018). The hidden state and current input will also be passed into the tanh function to squish values between -1 and 1 to help regulate the network. Then, the tanh output will be multiplied with the sigmoid output and this last one will decide which information is important to keep from the first.(Phi,2018).

The outputs from the forget and input gates are used to update the cell state. The cell state gets pointwise multiplied by the forget vector, which allows to drop values in the cell state if it is multiplied by values near 0. (Phi,2018). Then, the product of this operation is used to do a pointwise addition with the output from the input gate, which updates the cell state to new relevant value and gives the new cell state.(Phi,2018).

3 Analysis & Results

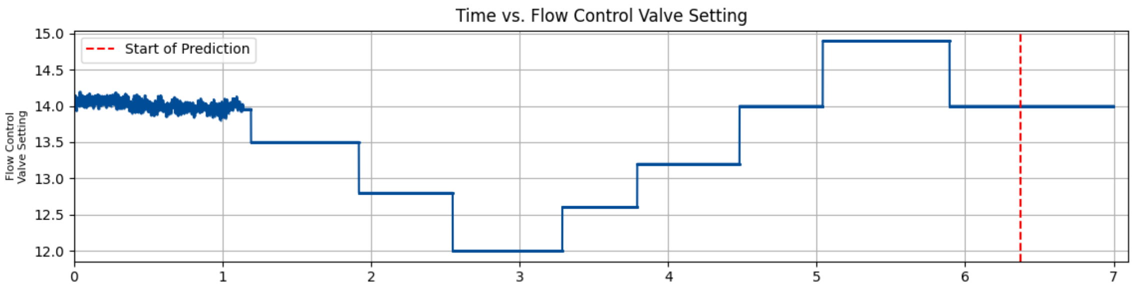

Figure 3: Flow control valve setting graphed over time (h). Dataset 1, prediction at 91% of data.

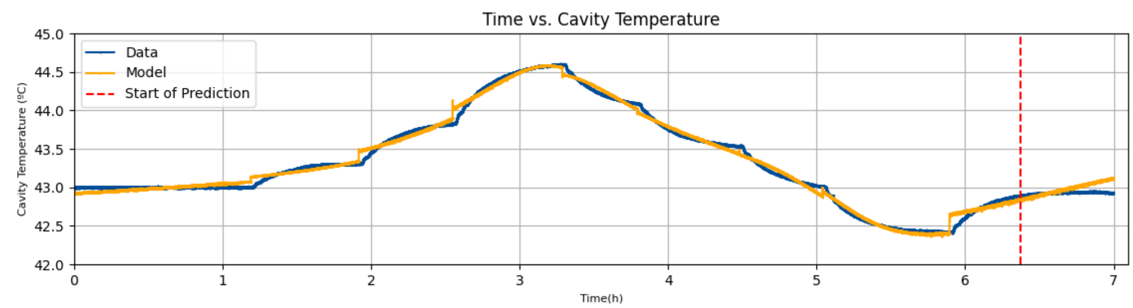

Figure 4: Temperature (ºC) graphed over time (h). Dataset 1, prediction at 91% of data.

Figures 3 and 4 show the evolution of the control valve setting and the temperature of the cavity over time.The model is shown in yellow, while the data is shown in blue. The red dashed line shows the point in which the model stopped being trained and started predicting, which happened at 91% of the total data. The relationship between the two is inversely proportional, as the valve setting increases the temperature decreases and vice-versa, which the model was able to follow, with an RMSE of 0.065.

$$RMSE = \sqrt{\frac{\Sigma_{i=1}^{N}(Predicted_i - Experimental_i)^2}{N}}, \tag{3}$$

where N is the number of data points.

However, the goal of this model was to predict when the phase stability would occur, the "flatness" in the graph, shown by the data at around 6.3 hours. The model does not predict this phase stability and registers and increase in temperature.

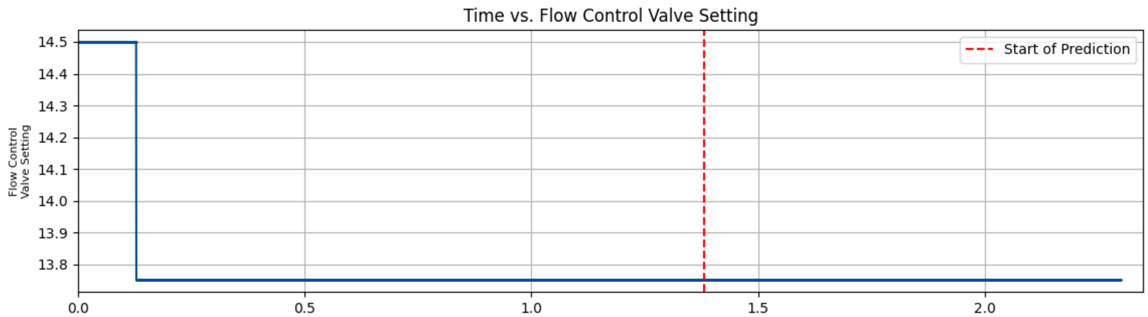

Figure 5: Flow control valve setting graphed over time (h). Dataset 2.

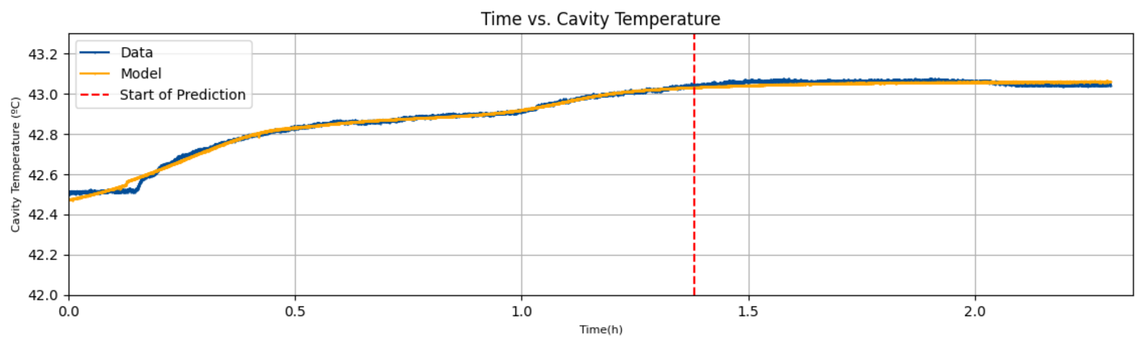

Figure 6: Temperature of the cavity (ºC) graphed over time (h). Dataset 2.

Figures 5 and 6 also show the evolution of the control valve setting and the temperature of the cavity over time, but with just one change in the valve setting. The model starts predicting at 60% of the data, achieving an RMSE of 0.057. In this case, the model was able to predict the phase stability, unlike with the first set of data. This might indicate the need for more training episodes with more varying data, so that the model can predict the behavior more accurately.

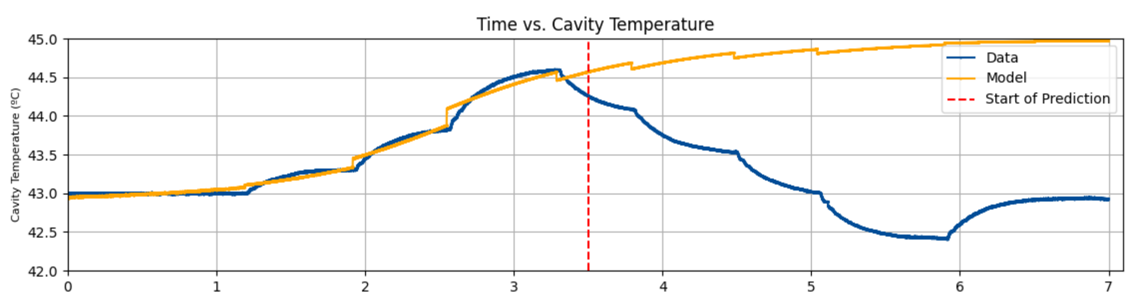

Figure 7: Temperature of the cavity (ºC) graphed over time (h). Dataset 1, prediction at 50% of data.

Using the first set of data, the model was prompted to predict the temperature of the cavity after an increase in the flow control valve setting. Because of their inversely proportional relationship, it was expected for the model to show a decrease in the temperature of the cavity. However, the model did not predict so.

It was discovered, then, that the model will not be able to predict trends that it has not ”seen” before, trends that have not been included in its training. Therefore, if an operator wanted to make some prediction about some specific behavior, they would need to show the model examples of that behavior for it to make a prediction.

4 Conclusion

The model still needs some improvement for it to predict more varied data and be more accurate. In order to make this happen, it would be interesting to introduce TOUT or T06 readings as inputs, which would account for the feedback loop and get a more precise prediction of the temperature of the cavity. This model was developed without making changes in the power setting. In the future, it could be interesting to introduce this changes in the model for it to have more range of prediction. After some improvements have been achieved, the next step would be to use the model to predict what adjustments at the inputs should be made to get the temperature to settle to a value faster and achieve the phase stability in lesser time.

5 Acknowledgments

I would like to recognize Jinhao Ruan, Auralee Edelen and Dean Edstrom for their guidance during the project. Michelle Ibrahim and Andres Quintero also deserve special mention for their advice and experience shared. Finally, I would like to thank the SIST program for the opportunity of this internship.

References

Edelen, A. L., Biedron, S. G., Chase, B. E., Edstrom, D., Milton, S. V., & Stabile, P. (2016). Neural networks for modeling and control of particle accelerators. Retrieved from `https://doi.org/10.1109/tns.2016.2543203` doi: 10.1109/tns.2016.2543203

Phi, M. (2018). *Illustrated guide to lstm's and gru's: A step by step explanation.* https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.

Zarzycki, K., & Ławrynczuk, M. (2021). Lstm and gru neural networks as models of dynamical processes used in predictive control: A comparison of models developed for two chemical reactors. Retrieved from `https://doi.org/10.3390/s21165625` doi: 10.3390/s21165625