**24.536 Reactor Experiments and 407.403 Advanced Nuclear Lab**

**HW #3: Reactor Operations Demo and Point Kinetics Simulations with Matlab**

## Introduction

The goals of this lesson were to review several key concepts from our last class on Reactor Kinetics, to discuss how to use Matlab to simulate some typical reactor transients, to illustrate the effect of inherent feedbacks on reactor behavior, and to observe and discuss a sequence of real operational transients within the UMLRR. Our simple analytical treatment along with the numerical simulation capability within Matlab allowed us to get a good understanding of reactor dynamics, and the actual live operational transients observed within the UMLRR were designed to enhance your understanding of these theoretical concepts within a real reactor environment.

Upon completion of this lesson, the student should have a solid foundation in the reactor kinetics and dynamics area, and a good understanding of several typical operational transients that can occur in real reactor cores. HW#3 emphasizes these topics, with two separate tasks, as follows:

1. Summarize the UMLRR Reactor Operations Demo
2. Perform and analyze some typical reactivity transients using the base Matlab simulation codes discussed in class.

The specific tasks and deliverables for each of these topic areas are described below:

## Task 0: Review/Study the Lecture Notes and the In-Class Matlab Simulations

Before starting the main tasks listed below, you should be sure that you have a good understanding of the main topics under study and the Matlab codes used in the in-class simulation demos. In particular, for this HW, you should carefully review the following document and Matlab routines:

1. J. R. White, "Solution of the Point Kinetics Equations," part of a series of Lecture Notes for the Nuclear Engineering Program at UMass-Lowell.
2. ***pksim_test.m, kinetics_data.m***, and ***pkeqns_nofdbk.m*** Matlab files/functions (contained in the ***HW3_mfiles.zip*** file in the Dropbox share folder for this class).

## Task 1: Summarize the UMLRR Reactor Operations Demo

This task wants you to create a series of plots that summarize the Reactor Operations Demo performed during class and to use these plots to help describe, explain and, in some cases, quantify the transients that were observed. Since the remote download capability is not currently working within the UMLRR_Online application, the history file from the reactor demo will be made available within the Dropbox share folder shortly after completion of the lab session. You should use the **umlrr_data** GUI to read this history file and generate the plots needed to support your discussion.

To guide your work on this task, you should perform the following calculations, answer the following questions, and/or describe your observations as requested for each of the seven (7) transients performed as part of the overall demo:

**Demo #1: Positive Reactivity Insertion** -- Is the observed power profile vs. time consistent with what you would expect for $\rho > 0$ and $k > 1$? Record the change in the regulating blade location that caused the power transient. Also estimate the time it takes for the reactor power to increase by a factor of two. With this latter result, compute the asymptotic reactor period. Now, with a known reactor period, $\tau$, estimate the amount of reactivity that was inserted by the RegBlade to initiate this transient.

**Hint:** To estimate $\rho$ you can use either the ***kinetics_gui*** program to plot $\tau$ vs. $\rho$ and obtain the desired result from the plot, or you can numerically evaluate of the reactivity equation with the UMLRR-specific kinetics parameters from the ***kinetics_data*** Matlab function. Note that writing a Matlab code to do the numerical evaluation is formally requested in Problem #1 of Task #2 (see below), so you might want to do that problem first...

**Demo #2: Negative Reactivity Insertion** -- Perform the same discussion and analyses as above for this situation where $\rho < 0$ and $k < 1$.

**Demo #3: Illustration of Auto Control** -- What does it mean when the regulating blade is in Auto Mode? Here we inserted Blade #2 a small amount into the reactor. Explain what happened to the power profile and the regulating blade location versus time. Is this behavior consistent with expectations? When Blade #2 was moved back to its original location, was the subsequent RegBlade movement as you expected? Explain this portion of the Demo thoroughly…

**Demo #4: Pump-Off Transient** -- Describe the key phenomena observed when the primary coolant pump was turned OFF with the regulating blade in Auto Mode. Be sure to identify and explain the sequence of events and the "cause and effect" relationships that were observed!

**Demo #5: Pump-On Transient (cold water insertion)** -- Describe the key phenomena observed when the primary coolant pump was turned ON with the RegBlade in Auto Mode. Again, be sure to identify and explain the sequence of events and the "cause and effect" relationships that were observed for this transient.

**Demo #6: Pump-Off Transient when in Manual Mode** -- Describe the key phenomena observed when the primary coolant pump was turned OFF with the regulating blade in Manual Mode. What is the key difference between this case and Demo #4? From this demo, explain why it is essential to have a negative temperature coefficient in an operating reactor.

**Demo #7: Pump-On Transient when in Manual Mode** -- Describe the key phenomena observed when the primary coolant pump was turned ON with the RegBlade in Manual Mode. What is the key difference between this case and Demo #5? For this case, it should be clear that "negative feedback can sometimes insert positive reactivity" -- explain this statement as part of your overall discussion.

## Task 2: Perform and analyze some typical reactivity transients using the base Matlab simulation codes discussed in class.

Perform each of the following simulations and discuss the results, as appropriate.

**Problem 1:** As you saw in the last HW and in the Demo #1 and #2 questions above, we often need to relate a measured reactor period, $\tau$, to the reactivity change, $\rho$, that caused the transient. Similarly, converting a known $\rho$ into an expected $\tau$ is also frequently required. Instead of

reading from a rough plot of $\tau$ vs. $\rho$ (as obtained from the **kinetics_gui**, for example), we can also do this numerically by "simply" evaluating the reactivity equation for the desired result,

$$\rho = \Lambda\omega + \sum_i \frac{\beta_i \omega}{\omega + \lambda_i} \tag{1}$$

where, of course, $\tau = 1/|\omega_1|$ is the reactor period.

In fact, there are two different cases of interest here, as follows:

**Case 1 -- Given $\tau$, calculate $\rho$:** This simply involves an **explicit evaluation** of eqn. (1) -- that is, simply plug $\omega = \omega_1 = 1/\tau$ into eqn. (1) and solve for $\rho$.

**Case 2 -- Given $\rho$, calculate $\tau$:** This case is not so simple since it requires finding the most positive root of an **implicit form** of eqn. (1) -- that is, find $\omega$ such that $F(\omega) = 0$, where

$$F(\omega) = \Lambda\omega + \sum_i \frac{\beta_i \omega}{\omega + \lambda_i} - \rho = 0 \tag{2}$$

This represents a classical root finding problem which, in Matlab, can be accomplished with the built-in ***fzero*** routine (if you are not familiar with this routine, type **help *fzero*** to get an overview and an example of its use).

Well, with the above background, your task for this problem is write two separate Matlab routines to address the above two cases. Use the ***kinetics_data.m*** function to get the required kinetics data for the UMLRR, and be sure to show a formal test of both your programs (make sure they work for both positive and negative values of $\tau$ and $\rho$). Note also that, from a reactor operator's viewpoint, for Case 1 you should request the doubling time (or halving time) from the user, convert this to reactor period, and then compute $\rho$ from eqn. (1). Similarly, for Case 2, be sure to output the doubling time as well as the reactor period for the given input step reactivity change. Finally, for consistency, ask the user for reactivity in units of %$\Delta$k/k for the Case 2 code, but always edit the input and results in both %$\Delta$k/k and dollars of reactivity (for ease of use). Also write both codes so that that are easy to understand and use…

**Problem 2:** In this problem we want to compare our simple analytical solution for a step change in reactivity that was used in HW#2 with a Matlab simulation that numerically solves the Generation Time formulation of Point Kinetics (i.e., as done in the ***pksim_test.m*** and ***pkeqns_nofdk.m*** routines). In particular, using $\rho$ = +0.05 %$\Delta$k/k for a positive reactivity situation and $\rho$ = -0.25 %$\Delta$k/k for a negative reactivity event, simulate and compare the analytical equations that we developed previously with the numerical solution for these step reactivity transients. For ease of comparison, plot the analytical P(t) profiles on the same axes with your numerical simulations. Also, for consistency, use $P_o$ = 5 kW as your starting critical power level before the step change in $\rho$. Discuss/explain any observed differences between the analytical and numerical simulations. Which P(t) profile is expected to be more accurate (analytical or numerical) relative to the behavior that would be observed in a real system? Explain…

**Problem 3:** Make a copy of the ***pkeqns_nofdbk.m*** Matlab function file and call it ***pkeqns_pfdbk.m***, where the ***pfdbk*** portion of the name stands for "power feedback". Now, modify the new file to include a feedback reactivity that is given by $\rho_f = \alpha_p(P(t) - P_o)$, where $\alpha_p = -7.8\times10^{-5}$ %Δk/k per kW (or $\alpha_p = -7.8\times10^{-10}$ Δk/k per W) is the power coefficient of reactivity and $P_o$ is the initial steady-state power level at critical prior to any perturbations to the system. Note that this task was demonstrated in class, so all you need to do is to repeat this procedure and make sure that everything is working properly.

Using $\rho = +0.05$ %Δk/k, simulate a step positive change in ρ with the reactivity feedback now included. Let $P_o = 5$ kW and choose a final simulation time that is long enough such that a new equilibrium power level is reached. Plot both P(t) and $\rho_{tot}(t) = \rho_{ext}(t) + \rho_f(t)$, and explain the observed behavior. Does the simulated P(t) behavior make sense? Is the new equilibrium power level reasonable (check this with a simple hand calculation)? Does the plot of total reactivity behave as expected? How long does it take to reach the new equilibrium (approximately)? Explain all your observations here and how these differ from the feedback-free positive reactivity transient simulated in Problem #2…

**Problem 4:** This scenario simply wants you to use the unmodified ***pkeqns_nofdbk.m*** function to simulate the behavior of the reactor with a sinusoidal ρ(t) as input. In particular, the input reactivity should approximate a sinusoid with a period of 2 seconds and a magnitude of 50 cents, or $\rho_{ex}(t) = 0.5\beta_{eff}\sin(\pi t)$

Simulate this transient with $P_o = 5$ kW and explain the observed P(t) profile for this case. In particular, explain why P(t) has an upward trend and why it is not a pure sinusoidal profile.

**Hint:** Look at the τ vs. ρ plot from the ***kinetics_gui*** program to "see" the behavior of positive vs. negative reactivity changes…

**Problem 5:** This last reactor simulation wants you to use the unmodified ***pkeqns_nofdbk.m*** function to simulate the behavior of a subcritical system (which, by definition, has no feedback mechanisms since the power level is negligibly low). Assume that the reactor is initially 3 dollars subcritical (i.e. $\rho_o = -3\beta_{eff}$ Δk/k) and that the source level is $1.3\times10^7$ n/sec (as given in the ***kinetics_data.m*** file). Now, to initiate a transient, insert a step change of +1 dollar of reactivity (i.e. $\rho_{ext} = +\beta_{eff}$ Δk/k). Run the simulation for several minutes -- until a new equilibrium is reached. Explain your observations…

**Hint:** Be careful to properly identify and set the proper initial conditions for this situation -- this is the key difference in doing this simulation!!! Also, since the initial power level is very low (as determined by the source strength, Q, and level of subcriticality, $\rho_o$), you may want to plot $P(t)/P_o$ as the desired reactor response here.

**Documentation and Submission of HWs**

In general, I expect a professional, well-written, semi-formal report for each HW assignment in this course. Please refer to HW#1 regarding the format for each HW assignment in this course -- **they should all be done and submitted in a similar fashion!!!**

For this HW, you will need to include the Matlab codes used to do the simulations for Task #2 -- so be sure not to forget these as part of the overall HW package. As in previous HWs, please integrate the plots requested as part of Task #1 directly within the Word file that discusses the UMLRR Operations Demo -- make this a professional treatment. For the Task #2 numerical simulations, do each problem separately, directly embed the resultant Matlab plots into the Word file, and thoroughly explain the observed behavior and requested comparisons. As done previously, please put everything together, including all your Matlab m-files, in a single zip file -- **only one zip file per HW please** -- and **email this to me before 4 pm (UML time) on the Sunday** before our next class. Note that this HW is a bit of work, so get started early and don't be bashful about asking questions, as needed…

Good luck…