Proceedings of ICONE-23
23rd International Conference on Nuclear Engineering
May 17-21, 2015, Chiba, Japan

ICONE23–1934

A FORMULATION OF ROD BASED NONLINEAR MODEL PREDICTIVE CONTROL OF NUCLEAR REACTION WITH TEMPERATURE EFFECTS AND XENON POISONING

Timothy A.V. Teatro[1], **Phillip McNelles**[2], **J. Mikael Eklund**[1]

[1] Department *of* Electrical, Computer *and* Software Engineering
[2] Faculty *of* Energy Systems *and* Nuclear Science
University *of* Ontario Institute *of* Technology
Oshawa, Ontario, CANADA

Keywords: Reactor Control; Model Predictive Control; Reactor Modeling.

ABSTRACT

This paper presents a formulation of a nonlinear model predictive controller for steering of a nuclear reaction in a pressurized water reactor via control rod displacement. The predictive model, based on point kinetic equations, includes an averaged delayed neutron group, thermal dependence on reactivity, heat exchange between temperature and moderator and Xenon poisoning. From the model, a finite horizon optimal control problem is cast in the form of a control Hamiltonian minimization. A basic gradient descent scheme is derived to solve the optimal control problem. The predictive model and gradient descent are used to form the basis from which the model predictive controller is formulated.

1 INTRODUCTION

In this work we detail the formulation of a nonlinear model predictive controller for a pressurized water reactor. This control method was chosen because of its power to solve nonlinear optimal control problems with soft and hard constraints. Our choice of reactor was a matter of convenience. The authors Gábor, Fazekas, Szederkényi and Hangos provide a state space model in Gábor et al. (2011) that is tailored to this sort of control study. Our intention in future work is to refine the model, extend it to CANDU reactors and perform extensive verification and stability analysis.

Model predictive control (MPC) was originally developed for process control (Richalet et al., 1978; Qin and Badgwell, 2003). It is an optimal control method that involves using a mathematical model of the plant to forecast the result of applying a set of control actions for a finite period, or *horizon*, of time. The model is used to find a control plan which optimizes a function expressing a measure of cost (or payoff, depending on your disposition). The state trajectory is formed online by rapidly generating these optimal finite horizon plans and executing a subset from the beginning of each. Each forecast is seeded with the most current reckoning of state, providing the feedback for closed loop control. So, as the plant traverses its state space, the controller is constantly planning the next action based on predicted behaviour.

The method can be applied to nonlinear models, and is then specifically called Nonlinear Model Predictive Control (NMPC) (Findeisen et al., 2003; Mayne et al., 2000).

The notion of optimal behaviour is reflected in the cost function. For example, in a hypothetical process where we wish the temperature of some component T_A to remain below a value $T_{A,max}$, our function may be a *cost function* with a term accumulating value as T_A approaches $T_{A,max}$ from below. Terms are added to the cost function to penalize undesired behaviour by increasing the cost function . Other behaviours can be incentivized by making negative contributions to the cost.

In the next section, §2, the predictive model is formulated into a function to predict the resulting change in state from a control action. In §3, a gradient descent scheme is designed to minimize a cost measure based on the predictive model. An NMPC control algorithm is defined based on that optimization scheme.

2 MODEL DETAILS

For the present research we adopt a state space model that was developed in stages between 2008 and 2011 (Fazekas et al., 2007; Gábor and Fazekas, 2009; Gábor et al., 2009, 2011). The latest paper expounding the particular realization of the model was published in 2011 by Gábor, Fazekas, Szederkényi and Hangos. The model is derived from fundamental principles and include temperature effects and Xenon poisoning. Model parameters given in the Gábor paper (see Table 1) are based on system identification performed on data from a VVER-440 Model V213 pressurized water reactor at the Paks Nuclear Power Plant in Hungary.

This section outlines key aspects of the model for the reader's convenience.

The key modelling assumptions from (Gábor et al., 2011) are:

1. The reactor is a time-dependent, spatially homogeneous, lumped parameter system with a single group.
2. The dynamics model of the reactor is derived from point kinetic equations.
3. Delayed neutron emitting precursors are averaged into a single group.
4. The fuel, moderator and control rods comprise the entire system.
5. The dependence of reactivity on rod position is quadratic.
6. The dependence of reactivity on temperature is linear.
7. The dependence of reactivity on Xenon concentration is linear.
8. Boron concentration and reactivity coefficients are constant.
9. The moderator flow rate is constant.
10. The reactor is thermally closed.

The model is suited for time-scales where fuel composition changes are negligible and the delayed neutron fraction, β, and mean neutron generation time, Λ, are constant. Online model adaptation over a fuel cycle is a matter for later consideration.

The model consists of six state equations describing neutron number density n_{N}, the number density of delayed neutron emitting nuclei n_{C}, the temperature of the fuel τ_{f}, the temperature of the moderator τ_{m}, and the number densities of Iodine and Xenon nuclei scaled inversely by the macroscopic fission cross-section: $\nu_{\mathrm{I}} = n_{\mathrm{I}}/\Sigma_{\mathrm{f}}$ and $\nu_{\mathrm{X}} = n_{\mathrm{Xi}}/\Sigma_{\mathrm{f}}$ respectively:

$$\frac{\mathrm{d}n_{\mathrm{N}}}{\mathrm{d}t} = \frac{\beta\left(\rho(t)-1\right)}{\Lambda}\, n_{\mathrm{N}}(t) + \frac{\beta}{\Lambda}\, n_{\mathrm{C}}(t), \tag{1a}$$

$$\frac{\mathrm{d}n_{\mathrm{C}}}{\mathrm{d}t} = \lambda_{\mathrm{C}}\left(n_{\mathrm{N}}(t) - n_{\mathrm{C}}(t)\right), \tag{1b}$$

$$\frac{\mathrm{d}\tau_{\mathrm{f}}}{\mathrm{d}t} = -A_1\left(\tau_{\mathrm{f}}(t) - \tau_{\mathrm{m}}(t)\right) + A_1\left(\tau_{\mathrm{f}}^{\mathrm{o}} - \tau_{\mathrm{m}}^{\mathrm{o}}\right)\frac{n_{\mathrm{N}}(t)}{n_{\mathrm{N}}^{\mathrm{o}}}, \tag{1c}$$

$$\frac{\mathrm{d}\tau_{\mathrm{m}}}{\mathrm{d}t} = A_3\left(\tau_{\mathrm{f}}(t) - \tau_{\mathrm{m}}(t)\right) - A_3\frac{\tau_{\mathrm{f}}^{\mathrm{o}} - \tau_{\mathrm{in}}^{\mathrm{o}}}{\tau_{\mathrm{m}}^{\mathrm{o}} - \tau_{\mathrm{in}}^{\mathrm{o}}}\left(\tau_{\mathrm{m}}(t) - \tau_{\mathrm{in}}\right), \tag{1d}$$

$$\frac{\mathrm{d}\nu_{\mathrm{I}}}{\mathrm{d}t} = \gamma_{\mathrm{I}}\frac{n_{\mathrm{N}}(t)}{n_{\mathrm{N}}^{\mathrm{o}}}\,\phi_0 - \lambda_{\mathrm{I}}\,\nu_{\mathrm{I}}(t), \tag{1e}$$

$$\frac{\mathrm{d}\nu_{\mathrm{X}}}{\mathrm{d}t} = \gamma_{\mathrm{Xi}}\frac{n_{\mathrm{N}}(t)}{n_{\mathrm{N}}^{\mathrm{o}}}\,\phi_0 + \lambda_{\mathrm{I}}\,\nu_{\mathrm{I}}(t) - \lambda_{\mathrm{Xi}}\nu_{\mathrm{X}}(t) - \frac{n_{\mathrm{N}}(t)}{n_{\mathrm{N}}^{\mathrm{o}}}\,\phi_0\,\sigma_{\mathrm{X}}\,\nu_{\mathrm{X}}(t). \tag{1f}$$

The 'o'-shaped superscripts indicate nominal values in full operation.

The neutron density state equation, (1a), comprises the emissions of the primary fission process and the decay of delayed neutron precursor. The reactivity,

$$\rho = \underbrace{\alpha_{\mathrm{f}}\left(\tau_{\mathrm{f}} - \tau_{\mathrm{f}}^{\mathrm{o}}\right) + \alpha_{\mathrm{m}}\left(\tau_{\mathrm{m}} - \tau_{\mathrm{m}}^{\mathrm{o}}\right)}_{\text{Temperature Dependence}} + \underbrace{p_2 z^2 + p_1 z + p_0}_{\text{Rod Dependence}} - \underbrace{\frac{\sigma_{\mathrm{X}}}{\beta}\left(\nu_{\mathrm{X}}[k] - \nu_{\mathrm{X}}^{\mathrm{o}}\right)}_{\text{Xenon Poisoning}}, \tag{2}$$

contains the dependences on the moderator and fuel temperatures, control rod displacement, z, and the effects of Xenon poisoning.

Equations (1a) and (1b) are the familiar point-kinetic equations from introductory literature (Lamarsh and Baratta, 2001, for example). The dependence of the reactivity on temperature and poison related quantities induces the extension of state space to include those quantities.

Equations (1c) and (1d) express the effects of Newton's law of cooling for heat exchange between the fuel and moderator in proportion to the difference in their temperatures. The parameters A_1 and A_3 arise from combining physical properties of the reactor, such as the heat capacities of the fuel and moderator and the transfer surface area. The values of A_1 and A_3 were estimated by model identification in the Paks plant in Gábor et al. (2011). The inlet temperature, τ_{in}, and outlet temperature, τ_{out}, of the moderator are distinguished; and the nominal moderator temperature is therefore $\tau_{\mathrm{m}}^{\mathrm{o}} = (\tau_{\mathrm{in}}^{\mathrm{o}} - \tau_{\mathrm{out}}^{\mathrm{o}})/2$.

Though we neglect any direct interaction between Iodine and the neutron density in (1a), we consider Iodine generation as a fission product because Iodine-135 β-decays to Xenon-135 which poisons the nuclear reaction. This relationship is expressed in (1f).

To be employed as a computational scheme, and to limit the dimensionality of the optimization, we discretize this model.

2.1 The Discrete State Space Model

For the discrete model, the rod displacement is added to the six-dimensions of the continuous model (1a)–(1f). This addition is due to the fact that we shall be controlling the rod speed $\dot{z}$, which will modulate the rod displacement. The effect of rod displacement will in turn cascade through the state equations by way of coupling terms. But in order translate the effect of rod speed into the state space, it must be directly referenced in some term of the state equations.

The state $\boldsymbol{x}[k] \coloneqq \boldsymbol{x}(kT) \in \mathbf{R}^7$ at discrete time step k is arranged as

$$\boldsymbol{x}[k] = \Big[z[k],\ n_{\mathrm{N}}[k],\ n_{\mathrm{C}}[k],\ \tau_{\mathrm{f}}[k],\ \tau_{\mathrm{m}}[k],\ \nu_{\mathrm{I}}[k],\ \nu_{\mathrm{X}}[k]\Big]^{\mathsf{T}}. \tag{3}$$

As a shorthand, the trajectory of states over a forecast horizon of N steps will may be referred to as $\boldsymbol{x}$. That is, $\boldsymbol{x} = \{\boldsymbol{x}[k] : k = 0, \ldots, N-1\}$.

The predictive model at the root of the NMPC controller will be based on an Euler integration of the equations of state (1a)–(1f)

and the rod variable:

$$\begin{bmatrix} z[k+1] \\ n_{\mathrm{N}}[k+1] \\ n_{\mathrm{C}}[k+1] \\ \vdots \\ \nu_{\mathrm{X}}[k+1] \end{bmatrix} = \begin{bmatrix} z[k] \\ n_{\mathrm{N}}[k] \\ n_{\mathrm{C}}[k] \\ \vdots \\ \nu_{\mathrm{X}}[k] \end{bmatrix} + T \begin{bmatrix} \boldsymbol{u}(t) \\ \mathrm{d}n_{\mathrm{N}}/\mathrm{d}t \\ \mathrm{d}n_{\mathrm{C}}/\mathrm{d}t \\ \vdots \\ \mathrm{d}\nu_{\mathrm{X}}/\mathrm{d}t \end{bmatrix}_{t=kT} \tag{4}$$

with discrete sampling interval T, and $\boldsymbol{u}[k] = \frac{\partial z}{\partial t}$. This yields the state evolution equations:

$$z[k+1] = z[k] + \dot{z}[k]T, \tag{5a}$$

$$n_{\mathrm{N}}[k+1] = n_{\mathrm{N}}[k] + \Bigg(\frac{\beta\left(\rho[k]-1\right)}{\Lambda} n_{\mathrm{N}}[k] + \frac{\beta}{\Lambda} n_{\mathrm{C}}[k] \Bigg) T, \tag{5b}$$

$$n_{\mathrm{C}}[k+1] = n_{\mathrm{C}}[k] + \Big(\lambda_{\mathrm{C}} \left(n_{\mathrm{N}}[k] - n_{\mathrm{C}}[k] \right) \Big) T, \tag{5c}$$

$$\tau_{\mathrm{f}}[k+1] = \tau_{\mathrm{f}}[k] + \Bigg(- A_1 \left(\tau_{\mathrm{f}}[k] - \tau_{\mathrm{m}}[k] \right) + A_1 \left(\tau_{\mathrm{f}}^{\mathrm{o}} - \tau_{\mathrm{m}}^{\mathrm{o}} \right) \frac{n_{\mathrm{N}}[k]}{n_{\mathrm{N}}^{\mathrm{o}}} \Bigg) T, \tag{5d}$$

$$\tau_{\mathrm{m}}[k+1] = \tau_{\mathrm{m}}[k] + \Bigg(A_3 \left(\tau_{\mathrm{f}}[k] - \tau_{\mathrm{m}}[k] \right) - A_3 \frac{\tau_{\mathrm{f}}^{\mathrm{o}} - T_{\mathrm{in}0}}{\tau_{\mathrm{m}}^{\mathrm{o}} - \tau_{\mathrm{in}}^{\mathrm{o}}} \left(\tau_{\mathrm{m}}[k] - \tau_{\mathrm{in}}[k] \right) \Bigg) T, \tag{5e}$$

$$\nu_{\mathrm{I}}[k+1] = \nu_{\mathrm{I}}[k] + \left(\gamma_{\mathrm{I}} \frac{n_{\mathrm{N}}[k]}{n_{\mathrm{N}}^{\mathrm{o}}} \phi_0 - \lambda_{\mathrm{I}} \nu_{\mathrm{I}}[k] \right) T, \tag{5f}$$

$$\nu_{\mathrm{X}}[k+1] = \nu_{\mathrm{X}}[k] + \Bigg(\gamma_{\mathrm{Xi}} \frac{n_{\mathrm{N}}[k]}{n_{\mathrm{N}}^{\mathrm{o}}} \phi_0 + \lambda_{\mathrm{I}} \nu_{\mathrm{I}} - \lambda_{\mathrm{Xi}} \nu_{\mathrm{X}}[k] - \frac{n_{\mathrm{N}}[k]}{n_{\mathrm{N}}^{\mathrm{o}}} \phi_0 \sigma_{\mathrm{X}} \nu_{\mathrm{X}}[k] \Bigg) T, \tag{5g}$$

For the sake of notational convenience and convention, we conclude this section with the definition of the vector valued function function $\boldsymbol{f} : \mathbf{R}^7 \times \mathbf{R} \to \mathbf{R}^7$ which maps a given state-control pair to the subsequent state:

$$\boldsymbol{f}\left(\boldsymbol{x}[k], \boldsymbol{u}[k]\right) \coloneqq \boldsymbol{x}[k+1] = \begin{bmatrix} (5a) \\ (5b) \\ \vdots \\ (5g) \end{bmatrix} \tag{6}$$

In this case, the control vector $\boldsymbol{u}[k]$ is just the scalar $\dot{z}[k]$. However, its will still appear in boldface, since we wish to preserve the notation $\boldsymbol{u} = \{ z[k] : k = 0, \ldots, N-1 \}$.

3 CONTROLLER FORMULATION

The NMPC method is a sliding window optimal control algorithm. Rather than finding the optimal path from start to finish, (which is not generally feasible), we set about a series of small optimal control problems for a finite time in the plant's future. As the system moves through its state space, the controller is constantly looking ahead to find the optimal path to the edge of the current time horizon.

The ability to look ahead requires a model—which we formulated in §2.1. Next, we will need to structure the optimization of a cost function based on that model. Then finally, we will define the NMPC algorithm which organizes the steps of prediction, optimization and execution which form the subsequent state-space trajectory.

The optimal control problem at hand is to find the set of N control actions, $\boldsymbol{u}$, which minimize a cost functional J_N for the time horizon TN, subject to the constraints imposed by our state evolution equations. In formal terms, we wish to find the optimal set $\boldsymbol{u}^*$ such that

$$\boldsymbol{u}^* \coloneqq \underset{\{u[1], u[2], \ldots, u[N]\}}{\arg\min} J_N(\boldsymbol{x}, \boldsymbol{u}), \tag{7}$$

subject to the discrete state trajectory model, Eq. (6).

The discrete cost function is quadratic:

$$\begin{aligned} J_N(\boldsymbol{x}, \boldsymbol{u}) &\coloneqq \sum_{k=0}^{N-1} \left(\left\| \tilde{\boldsymbol{x}}[k] \right\|_{\boldsymbol{Q}}^2 + R \left(\boldsymbol{u}[k] \right)^2 \right) \\ &= \sum_{k=0}^{N-1} L(\boldsymbol{x}[k], \boldsymbol{u}[k]). \end{aligned} \tag{8}$$

where $\|\boldsymbol{x}\|_{\boldsymbol{P}}$ denotes the 2-norm weighted by a positive definite matrix $\boldsymbol{P}$: $\|\boldsymbol{x}\|_{\boldsymbol{P}}^2 \coloneqq \boldsymbol{x}^{\mathrm{T}} \boldsymbol{P} \boldsymbol{x}$. The tracking error, $\tilde{\boldsymbol{x}}[k] = \boldsymbol{x}^{\mathrm{ref}}[k] - \boldsymbol{x}[k]$, is the difference between the forecast path and a reference. This cost function penalizes tracking error and control effort. The tracking terms serve as quadratic attractors to each point in the horizon. The penalty on control effort precludes solutions which drive the control rods with infinite speeds. The function label L is given to the summand in (8), which is conventionally called the *running cost*. The factors $\boldsymbol{Q}$ and R are weighting parameters which are used to tune the controller.

Following the formulation of similar NMPC controllers (Teatro et al., 2014; Fahimi, 2007; Sutton and Bitmead, 2000; Eklund et al., 2012), we solve the minimization problem, (7), by solving the Euler-Lagrange differential equations, using a variational expansion to construct a gradient descent scheme. The cost measure in (8) is augmented with a set of Lagrange multipliers $\{\boldsymbol{p}[k] : k = 1, \ldots, N\}$ enforcing (6) as constraints:

$$J_\Lambda = \sum_{k=1}^{N-1} \Big(L(\boldsymbol{x}[k], \boldsymbol{u}[k]) + \boldsymbol{p}[k+1]^{\mathrm{T}} \left(\boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{u}[k]) - \boldsymbol{x}[k+1] \right) \Big). \tag{9}$$

The state trajectory is computed from (6), so will always be the case that $\boldsymbol{x}[k+1] - \boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{u}[k]) = 0$. We may therefore freely

choose the Lagrange multipliers without affecting the balance of (9).

In following with Pontryagin's formalism (Kirk, 2004, Ch. 5), we define the control Hamiltonian as

$$\mathscr{H}[k] = L(\boldsymbol{x}[k], \boldsymbol{u}[k]) + \boldsymbol{p}[k+1]^{\mathsf{T}}\, \boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{u}[k]). \tag{10}$$

Rewritten in terms of (10), the augmented cost measure, (9), becomes

$$J_{\Lambda} = \mathscr{H}_0[0] + \boldsymbol{p}_N^{\mathsf{T}} \boldsymbol{x}[N] + \sum_{k=1}^{N-1} \left(\mathscr{H}[k] - \boldsymbol{p}[k]^{\mathsf{T}}\, \boldsymbol{x}[k] \right). \tag{11}$$

In order to minimize J_{Λ}, we compute the differential change in J_{Λ} with respect to its dependants

$$\begin{aligned} \mathrm{d}J_{\Lambda} = -\boldsymbol{p}_N^{\mathsf{T}}\, \mathrm{d}\boldsymbol{x}_N + \frac{\partial \mathscr{H}_0[0]}{\partial \boldsymbol{x}[0]}\, \mathrm{d}\boldsymbol{x}[0] + \frac{\partial \mathscr{H}_0[0]}{\partial \boldsymbol{u}[0]}\, \mathrm{d}\boldsymbol{u}[0] \\ + \sum_{k=1}^{N-1} \left[\left(\frac{\partial \mathscr{H}[k]}{\partial \boldsymbol{x}[k]} - \boldsymbol{p}[k]^{\mathsf{T}} \right) \mathrm{d}\boldsymbol{x}[k] + \frac{\partial \mathscr{H}[k]}{\partial \boldsymbol{u}[k]}\, \mathrm{d}\boldsymbol{u}[k] \right]. \end{aligned} \tag{12}$$

Now in a position to make choices regarding our Lagrange multipliers so as to simplify (12) as much as possible, we choose

$$\boldsymbol{p}_N^{\mathsf{T}} = \boldsymbol{0} \quad \text{and} \quad \boldsymbol{p}[k]^{\mathsf{T}} = \frac{\partial \mathscr{H}[k]}{\partial \boldsymbol{x}[k]} + \frac{\partial \mathscr{H}[k]}{\partial \tilde{\boldsymbol{x}}[k]}\, \frac{\partial \tilde{\boldsymbol{x}}[k]}{\partial \boldsymbol{x}[k]} \tag{13}$$

or more explicitly

$$\boldsymbol{p}[k]^{\mathsf{T}} = \tilde{\boldsymbol{x}}[k]^{\mathsf{T}}\, \boldsymbol{Q}\, \frac{\mathrm{d}\tilde{\boldsymbol{x}}[k]}{\mathrm{d}\boldsymbol{x}[k]} + \boldsymbol{p}[k+1]^{\mathsf{T}}\, \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}[k]}\,. \tag{14}$$

The constant factors from the differentiation are absorbed into the weighting parameters. The choice of Lagrange multipliers consequently reduces (12) to

$$\mathrm{d}J_{\Lambda} = \sum_{k=1}^{N-1} \frac{\partial \mathscr{H}[k]}{\partial \boldsymbol{u}[k]}\, \mathrm{d}\boldsymbol{u}[k] + \boldsymbol{p}_0^{\mathsf{T}}\, \mathrm{d}\boldsymbol{x}[0]. \tag{15}$$

The right-most term of the above equation is zero, since $\boldsymbol{x}[0]$ is constant and $\mathrm{d}\boldsymbol{x}[0]$ is therefore also zero. The final effort is in evaluating $\partial\mathscr{H}[k]/\partial\boldsymbol{u}[k]$ by expanding and differentiating (10), to reveal that

$$\frac{\partial \mathscr{H}[k]}{\partial \boldsymbol{u}[k]} = R u[k] + \boldsymbol{p}[k+1]^{\mathsf{T}}\, \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}[k]}\,, \tag{16}$$

giving us the tools to compute the gradient elements of $\mathscr{H}[k]$. That gradient forms the basis of our gradient descent which iteratively minimizes the cost function with respect to each element of $\boldsymbol{u}$.

Quite some effort goes into unpacking the Jacobian tensor expressed in the derivatives $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}[k]}$ and $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}[k]}$. It is nonetheless instructive to observe the coupling in the system through this structure.

The gradient of $\boldsymbol{f}$ with respect to state, in numerator layout, is

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}[k]} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \partial_z n_{\mathrm{N}} & \multicolumn{2}{c}{\multirow{2}{*}{$\boldsymbol{B}_1$}} & \alpha_{\mathrm{f}} & \alpha_{\mathrm{m}} & 0 & 0 \\ 0 & & & 0 & 0 & 0 & 0 \\ 0 & \partial_{n_{\mathrm{N}}} \tau_{\mathrm{f}} & 0 & \multicolumn{2}{c}{\multirow{2}{*}{$\boldsymbol{B}_2$}} & 0 & 0 \\ 0 & 0 & 0 & & & 0 & 0 \\ 0 & \partial_{n_{\mathrm{N}}} \nu_{\mathrm{I}} & 0 & 0 & 0 & \multicolumn{2}{c}{\multirow{2}{*}{$\boldsymbol{B}_3$}} \\ 0 & \partial_{n_{\mathrm{N}}} \nu_{\mathrm{X}} & 0 & 0 & 0 & & \end{bmatrix}, \tag{17}$$

where

$$\boldsymbol{B}_1 = \begin{bmatrix} \dfrac{\beta(\rho-1)}{\Lambda} & \dfrac{\beta}{\Lambda} \\ \lambda_{\mathrm{C}} & -\lambda_{\mathrm{C}} \end{bmatrix},$$

$$\boldsymbol{B}_2 = \begin{bmatrix} -A_1 & A_1 \\ A_3 & -A_3 \left(1 + A_3 \dfrac{\tau_{\mathrm{f}}^{\mathrm{o}} - T_{\mathrm{in0}}}{\tau_{\mathrm{m}}^{\mathrm{o}} - \tau_{\mathrm{in}}^{\mathrm{o}}} \right) \end{bmatrix},$$

$$\boldsymbol{B}_3 = \begin{bmatrix} -\lambda_{\mathrm{I}} & 0 \\ \lambda_{\mathrm{I}} & -\lambda_{\mathrm{Xi}} - \dfrac{\sigma_{\mathrm{X}} \phi_0 n_{\mathrm{N}}[k]}{n_{\mathrm{N}}^{\mathrm{o}}} \end{bmatrix},$$

and

$$\begin{aligned} \partial_{n_{\mathrm{N}}} \tau_{\mathrm{f}} &= \frac{\partial \tau_{\mathrm{f}}}{\partial n_{\mathrm{N}}[k]} = A_1 \frac{\tau_{\mathrm{f}}^{\mathrm{o}} - \tau_{\mathrm{m}}^{\mathrm{o}}}{n_{\mathrm{N}}^{\mathrm{o}}}, \\ \partial_{n_{\mathrm{N}}} \nu_{\mathrm{I}} &= \frac{\partial \nu_{\mathrm{I}}}{\partial n_{\mathrm{N}}[k]} = \frac{\gamma_{\mathrm{I}} \phi_0}{n_{\mathrm{N}}^{\mathrm{o}}}, \\ \partial_{n_{\mathrm{N}}} \nu_{\mathrm{X}} &= \frac{\partial \nu_{\mathrm{X}}}{\partial n_{\mathrm{N}}[k]} = \frac{\gamma_{\mathrm{Xi}} \phi_0 - \sigma_{\mathrm{X}} \phi_0 \nu_{\mathrm{X}}[k]}{n_{\mathrm{N}}[0]}, \\ \partial_z n_{\mathrm{N}} &= \frac{\partial n_{\mathrm{N}}}{\partial z[k]} = p_2 z[k] + p_1. \end{aligned}$$

The block $\boldsymbol{B}_2$ contains the thermal coupling between the moderator and fuel. The block $\boldsymbol{B}_3$ contains the coupling between Iodine and Xenon concentrations. The neutron and delayed neutron precursor are coupled together in $\boldsymbol{B}_1$. But inspection of the second column of (17) illustrates how changes in n_{N} purvey every state equation except τ_{m}, which is not directly tied to the nuclear reaction. This is a key point, since n_{N} is the only state variable influenced by the rod displacement. In turn, the rod displacement is the only state variable influenced by our control variable:

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^{\mathsf{T}}.$$

This illustrates the path of influence of the control action.

A basic gradient descent algorithm implementing this formalism is listed in Algorithm 1. Starting from some initial condition $\boldsymbol{x}[0]$ and an initial guess for the optimal control history $\boldsymbol{u}$, one can use the derived formalism to iteratively improve $\boldsymbol{u}$ to approach $\boldsymbol{u}^*$. The relationships developed in this section are used to find the gradient elements $\partial\mathscr{H}[k]/\partial\boldsymbol{u}[k]$, which are then used to step $\boldsymbol{u}$ towards $\boldsymbol{u}^*$. Convergence rate and stability are balanced (in part) by the step factor Δ. The iteration loop is broken when the Euclidean norm of the vector formed by the gradient elements is sufficiently small. This algorithm is far from the best approach, but it may be the most straightforward for the purposes of demonstration. Compared to more sophisticated methods, it sacrifices elegance and performance for intuition as it illustrates the basic relationships intrinsic to any gradient descent scheme.

An NMPC algorithm based on Algorithm 1 is listed in Algorithm 2. The optimization is carried out for a segment of the path to the target that is N steps long. A subset of that optimal path, the first C steps are executed, and the optimization is carried out

Algorithm 1 A general algorithm for iteratively optimizing a control history against a cost function.

$\boldsymbol{u} \leftarrow$ initial guess
$\boldsymbol{x}_1 \leftarrow$ starting position
while norm$\left([\partial\mathscr{H}_1/\partial\boldsymbol{u}_1, \cdots, \partial\mathscr{H}_N/\partial\boldsymbol{u}_N]\right) > tolerance$ **do**
  for $k = 1, \ldots, (N-1)$ **do**
    $\boldsymbol{x}[k+1] \leftarrow \boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{u}[k])$
  end for
  Compute $\boldsymbol{p}_N$ from Eq. (13)
  for $k = (N-1), \ldots, 1$ **do**
    Compute $\boldsymbol{p}[k]$ from Eq. (14)
  end for
  for $k = 1, \ldots, (N-1)$ **do**
    Compute gradient element $\partial\mathscr{H}[k]/\partial\boldsymbol{u}[k]$ from Eq. (16)
  end for
  for k=1,..., N **do**
    $\boldsymbol{u} \leftarrow \boldsymbol{u} - \Delta\left(\partial\mathscr{H}[k]/\partial\boldsymbol{u}[k]\right)$
  end for
end while

Algorithm 2 A general statement of the Model Predictive Control algorithm

$\boldsymbol{u} \leftarrow$ initial guess
$\boldsymbol{x}_1 \leftarrow$ starting position
while not sufficiently close to target **do**
  Get $\boldsymbol{u}^*$ and $\boldsymbol{x}^*$ over N using Algorithm 1
  Execute the first C steps of $\boldsymbol{u}^*$
  for $k = C, \ldots, N$ **do**
    $\boldsymbol{x}[k-C] \leftarrow \boldsymbol{x}[k]^*$
    $\boldsymbol{u}[k-C] \leftarrow \boldsymbol{u}[k]^*$
  end for
end while

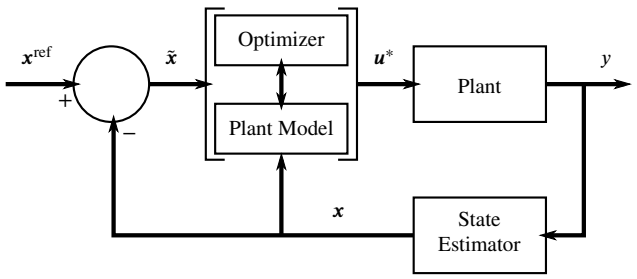

Figure 1: A block diagram for a system with MPC controller.

again for the next set of N steps. This process is repeated until the target is reached. The integer C is the *control horizon*.

The general configuration of an NMPC controller is illustrated in the block diagram in Figure 1. In that figure, the target state is passed into the NMPC block which consists of the optimizer and system model. These components exchange information until $\boldsymbol{u}^*$ is found. A subset of $\boldsymbol{u}^*$ is executed by the plant and the resulting state of the plant and other output information is available in y. Whatever routines and devices are needed to convert y into the state vector $\boldsymbol{x}$ is encompassed in the *state estimator*. The state estimator feeds the state back to compute the error signal and to seed the NMPC block.

The values of key constants and parameters are tabulated in Table 1 near the end of this document.

4 CURRENT AND FUTURE WORK

We currently have an implementation of the control system written in Python with Numpy (Millman and Aivazis, 2011; van der Walt et al., 2011) which we are tuning and testing. The key milestones ahead are extension to CANDU reactors and validation.

Since our group has no access to VVER reactors for validation, we will immediately begin adaptation to CANDU reactors. We will perform a system identification to compute model parameters such as A_1/A_3 and the rod parameters. We may also extend the reactivity model in order to be sure that it is accurate across the full operating range of rod displacement.

Our facility is home to an advanced CANDU simulator (U. of Ontario Inst. of Tech., Nuclear Sim. Lab.) which we intend to use for model validation. An analysis of the stability of the controller will be integral to this portion of the research.

The present model works for periods on the order of a few hours, but is invalid over periods where fuel composition changes significantly. A future branch of research will involve online model adaptation so that the controller is viable through the entire fuel cycle.

In the final states of the research, we plan to implement the validated controller on a Field Programmable Gate Array (FPGA) chip. This will accompany a detailed fault analysis of the technology.

BIBLIOGRAPHY

Eklund, J. M., Sprinkle, J., and Sastry, S. Symmetric and Asymmetric Pursuit Evasion Games With Model Predictive Control. *IEEE Transactions on Control Systems Technology*, In Press, 2012.

Fahimi, F. Non-linear model predictive formation control for groups of autonomous surface vessels. *International Journal of Control*, 80(8):1248–1259, August 2007. ISSN 0020-7179. doi: 10.1080/00207170701280911. URL `http://resolver.scholarsportal.info.uproxy.library.dc-uoit.ca/resolve/00207179/v80i0008/1248_nmpfcfgoasv` `http://www.tandfonline.com/doi/abs/10.1080/00207170701280911`.

Fazekas, C., Szederkényi, G., and Hangos, K. A simple dynamic model of the primary circuit in VVER plants for controller design purposes. *Nuclear Engineering and Design*, 237(10): 1071–1087, May 2007. ISSN 00295493. doi: 10.1016/j.nucengdes.2006.12.002. URL `http://www.sciencedirect.com/science/article/pii/S0029549306006789`.

Findeisen, R., Imsland, L., Allgower, F., and Foss, B. A. State and Output Feedback Nonlinear Model Predictive Control:

Table 1: List of symbols and parameters. Listed values are taken from Gábor et al. (2011), and are for the convenience of those inclined to test the presented framework.

Symbol	Units	Value (if applicable)	Description
$\boldsymbol{x}$	[mixed]	–	State vector.
$\boldsymbol{u}$	[mixed]	–	Control vector.
n_{N}	cm^{-3}	–	Neutron concentration $\phi_T = n_{\mathrm{N}}\phi_0 \,/\, n_{\mathrm{N}}^{\mathrm{o}}$.
n_{C}	cm^{-3}	–	Concentration of the delayed neutron precursor.
τ_{f}	°C	–	Fuel temperature.
τ_{m}	°C	–	Moderator temperature.
ν_{I}	cm	–	Scaled Iodine concentration: $n_{\mathrm{I}}/\Sigma_{\mathrm{f}}$.
ν_{X}	cm	–	Scaled Xenon concentration: $n_{\mathrm{Xi}}/\Sigma_{\mathrm{f}}$.
ϕ_0	$\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$	1.2×10^{13}	Nominal operating flux.
$n_{\mathrm{N}}^{\mathrm{o}}$	cm^{-3}	5.025×10^{7}	Nominal neutron concentration.
β	–	0.0065	Delayed neutron fraction.
Λ	s	2.18×10^{-5}	Avg. generation time.
α_{f}	$°\mathrm{C}^{-1}$	5.362×10^{-3}	Fuel temperature feedback coefficient.
α_{m}	$°\mathrm{C}^{-1}$	-2.075×10^{-2}	Moderator temperature feedback coefficient.
σ_{X}	cm^2	2.805×10^{-18}	Microscopic absorption cross-section.
λ_{I}	s^{-1}	2.849×10^{-5}	Decay constant of Iodine.
λ_{Xi}	s^{-1}	2.150×10^{-5}	Decay constant of Xenon.
λ_{C}	s^{-1}	7.728×10^{-2}	Decay constant of delayed neutron emitting nuclei.
γ_{I}	–	6.39×10^{-2}	Iodine yield.
γ_{Xi}	–	2.150×10^{-5}	Xenon yield.
$\tau_{\mathrm{f}}^{\mathrm{o}}$	°C	6.19×10^{2}	Nominal fuel temperature.
$\tau_{\mathrm{in}}^{\mathrm{o}}$	°C	220	Nominal inlet water temperature.
$\tau_{\mathrm{m}}^{\mathrm{o}}$	°C	280	Nominal average moderator temperature.
p_0, p_1, p_2	$\mathrm{m}^{-2}, \mathrm{m}^{-1}, -$	0.0401, −0.440, −0.996	Rod dependence parameters on reactivity.
A_1, A_3	s^{-1}	0.1056, 0.8757	Heat transfer coefficients.
$\boldsymbol{Q}$	–	–	Positive-definite weighting matrix for error tracking cost term.
$\boldsymbol{R}$	–	–	Scalar weight of control effort cost term.

An Overview. *European Journal of Control*, 9(2-3):190–206, January 2003. ISSN 09473580. doi: 10.3166/ejc.9.190-206. URL http://www.sciencedirect.com/science/article/pii/S0947358003702751.

Gábor, A. and Fazekas, C. Modeling and identification of a nuclear reactor with temperature effects and xenon poisoning. *IECON*, pages 1639–1644, 2009. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5414753.

Gábor, A., Fazekas, C., Szederkényi, G., and Hangos, K. M. Modeling and identification of a nuclear reactor with temperature effects and Xenon poisoning. In *2009 35th Annual Conference of IEEE Industrial Electronics*, pages 1639–1644. IEEE, November 2009. ISBN 978-1-4244-4648-3. doi: 10.1109/IECON.2009.5414753. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5414753.

Gábor, A., Fazekas, C., Szederkényi, G., and Hangos, K. M. Modeling and Identification of a Nuclear Reactor with Temperature Effects and Xenon Poisoning. *European Journal of Control*, 17(1):104–115, January 2011. ISSN 09473580. doi: 10.3166/ejc.17.104-115. URL http://www.sciencedirect.com/science/article/pii/S0947358011705726.

Kirk, D. E. *Optimal Control Theory: An Introduction*. Dover Publications, 2004. ISBN 0486434842.

Lamarsh, J. R. and Baratta, A. J. *Introduction to Nuclear Engineering*. Prentice Hall, 2001. ISBN 0201824981. URL http://books.google.com/books?id=xSNnQgAACAAJ&pgis=1.

Mayne, D., Rawlings, J., Rao, C., and Scokaert, P. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000. URL http://www.sciencedirect.com/science/article/pii/S0005109899002149.

Millman, K. J. and Aivazis, M. Python for Scientists and Engineers. *Computing in Science & Engineering*, 13(2):9–12, March 2011. ISSN 1521-9615. doi: 10.1109/MCSE.2011.36. URL http://scitation.aip.org/content/aip/journal/cise/13/2/10.1109/MCSE.2011.36.

Qin, S. and Badgwell, T. A. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11(7):733–764, July 2003. ISSN 09670661. doi: 10.1016/S0967-0661(02)00186-7. URL http://www.sciencedirect.com/science/arti-

cle/pii/S0967066102001867
http://linkinghub.elsevier.com/retrieve/pii/S0967066102001867.

Richalet, J., Rault, A., Testud, J., and Papon, J. Model predictive heuristic control. *Automatica*, 14(5):413–428, September 1978. ISSN 00051098. doi: 10.1016/0005-1098(78)90001-8. URL http://linkinghub.elsevier.com/retrieve/pii/0005109878900018.

Sutton, G. J. and Bitmead, R. R. Performance and Computational Implementation of Nonlinear Model Predictive Control on a Submarine. In Allgöwer, F., Zheng, A., and Byrnes, C. I., editors, *Nonlinear Model Predictive Control*, volume 26 of *Progress in Systems and Control Theory*, pages 461–472. Birkhäuser Basel, 2000. ISBN 978-3-0348-8407-5. doi: 10.1007/978-3-0348-8407-5_27. URL http://dx.doi.org/10.1007/978-3-0348-8407-5_27.

Teatro, T. A. V., Eklund, J. M., and Milman, R. Nonlinear Model Predictive Control for Omnidirectional Robot Motion Planning and Tracking with Avoidance of Moving Obstacles. *Canadian Journal of Electrical and Computer Engineering*, 37(3), 2014. doi: 10.1109/CJECE.2014.2328973.

University of Ontario Institute of Technology. Nuclear Simulation Laboratory. URL http://nuclear.uoit.ca/about/learning-environment/nuclear-simulation-laboratory.php.

van der Walt, S., Colbert, S. C., and Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering*, 13(2):22–30, March 2011. ISSN 1521-9615. doi: 10.1109/MCSE.2011.37. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5725236.