Tracking Action Potentials of Nonlinear Excitable Cells Using Model Predictive Control

Md. Ariful Islam, Abhishek Murthy, Tushar Deshpande, Scott D. Stoller and Scott A. Smolka

Department of Computer Science
Stony Brook University
Stony Brook, New York 11794
Email: {mdaislam,amurthy,tushard, stoller,sas}@cs.sunysb.edu

Ezio Bartocci and Radu Grosu

Department of Computer Engineering
Vienna University of Technology
Vienna, Austria
Email: {ezio.bartocci,radu.grosu} @tuwien.ac.at

***Abstract*—We present explicit and online Model Predictive Controllers (MPCs) for an excitable cell simulator based on the nonlinear FitzHugh-Nagumo model. Despite the plant's nonlinearity, we are able to formulate the model predictive control problem as an instance of quadratic programming, using a PieceWise Affine (PWA) abstraction of the plant. The speed-versus-accuracy tradeoff for the explicit and online versions is analyzed on various reference trajectories. Our MPC-based approach, enabled by the PWA abstraction, presents a framework for designing automated *in silico* biomedical control strategies for excitable cells, such as cardiac myocytes and neurons.**

Keywords–Biocomputing; Model Predictive Control; Excitable Cells.

I. Introduction

Excitable cells, like neurons and cardiac myocytes, are building blocks of mammalian organ systems like the nervous and the cardiovascular systems. They exhibit characteristic cyclical responses to electrical stimuli, which could be provided externally or by neighboring cells via diffusion. The response is observed in terms of the change in their transmembrane potential in time and is called the Action Potential (AP). The cells are arranged contiguously to form the corresponding tissue. The periodic electrical excitation and diffusion at the cell-level leads to emergent patterns of electrical-wave propagation at the tissue-level [1]. Anomalous patterns at the tissue-level are associated with potentially fatal disorders like epilepsy and cardiac arrhythmias. For example, *reentry*, which corresponds to spiral waves in the cardiac tissue, is a precursor of Atrial Fibrillation (AFib) [2].

Controlling the cell-level response is critical for countering abnormal patterns at the tissue level. Excitable cells have the following distinguishing features that pose challenges to designing effective control strategies.

- **Nonlinearity**: The state space models for neurons and cardiac myocytes have highly nonlinear vector fields, which leads to multiple time scales.
- **Noise**: An actuator controlling a biological entity receives noisy readings corresponding to the state of the plant. Thus, robustness is critical while designing a control law.
- **Dimensionality**: Excitable cells are large dynamical systems and many state variables could be non-observable.

Model predictive control, a widely used process-control strategy, is well suited for biomedical applications involving excitable cells. It involves solving a finite horizon open-loop optimal control problem subject to the dynamics of the *plant*, which is the system to be controlled. Based on the measurements obtained at time T, the controller predicts the dynamic behavior of the system over a prediction horizon (T_p) and optimizes the control input over a control horizon ($Tc < T_p$) such that a predetermined open-loop performance objective function is minimized [3]. The objective function usually measures the plant's divergence from a prescribed reference trajectory, and thus is minimized by the controller. Disturbances and model mismatch constrain the controller's performance. The optimization can either be performed online (for accuracy) or can be done offline (for speed).

We present implementations of both the online and offline strategies for model predictive control of a neuron. Specifically, MPCs for a nonlinear model-based simulator of an neuron are presented. Fast and accurate model predictive control of excitable cells can be used for in-silico testing of biomedical control strategies, where a control law is designed and tested in software before fabrication. The biological entity being controlled is modeled using a simulator and the control law is tested on it, in software. Authors in [4] and [5] present novel strategies controlling and managing anomalous behaviors of neurons (epilepsy) and cardiac myocytes (ventricular tachycardia) respectively.

Model predictive control of plants with nonlinear dynamics, such as neurons, has garnered interest in the community, due to its unique challenges and wide-ranging applicability [3], [6]. The nonlinearity of the neuron dynamics results in an instance of nonlinear optimization to be solved during MPC. In general, nonlinear optimization is NP hard [7] and

thus the implementation of online MPC is computationally expensive. Explicit MPC for nonlinear systems was proposed in [8], and has limited tool support. To circumvent these issues, *we adopt an approximately equivalent PieceWise Affine (PWA) abstraction of the nonlinear neuron model for both the online and explicit MPCs.* In [9], the Mixed Logical Dynamical (MLD) formalism was introduced for modeling systems whose state variables evolve continuously in time subject to logical constraints. The MPC problem for MLD systems was shown to be an instance of Mixed-Integer Quadratic Programming (MIQP). Later in [10], it was shown that PWA systems are equivalent to MLD systems. Thus, converting the nonlinear neuron model to an approximately equivalent PWA form transforms the corresponding MPC problem from an instance of nonlinear optimization to one of MIQP. Also, the PWA abstraction enables the design of explicit MPC by using the Multi-Parametric Toolbox (MPT) [11] in MATLAB.

Next, we outline the architecture of the controllers, see Fig. 1. The plant simulates an excitable cell and outputs the AP corresponding to the input stimuli provided by the MPC. The MPC's goal is to compute optimal inputs such that the plant tracks, in discrete time, a reference trajectory that consists of a nominal sequence of APs.

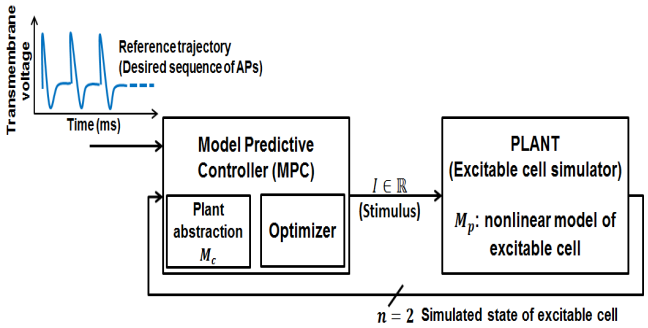


Figure 1. Architecture for tracking action potentials of nonlinear excitable cells using MPCs.

1) The plant uses a nonlinear model M_p of an excitable cell for its simulation. We use the FitzHugh-Nagumo (FHN) model [12], described in Section II, as M_p.
2) The plant outputs the n-dimensional state of M_p as the state of the underlying excitable cell. For the FHN model $n = 2$, and one of the state variables is the dimensionless transmembrane potential, which tracks the reference trajectory.
3) The MPC uses a PWA abstraction, M_C, of the plant model, to predict the behavior of the cell under simulation. We use a modified version of the hybrid model proposed in [13], henceforth referred to as the Dumas-Rondepierre (DR) model, as M_c. The PWA abstraction is used to cast the MPC's optimization problem as an instance of MIQP.
4) MPC is also equipped with an optimizer to compute the optimal stimulus input I, such that the observed state of the plant tracks a pre-defined reference trajectory.

The following simplifying assumptions are made in our implementation, and justified in the appropriate sections of the paper:

1) The plant's state is completely visible to the MPC. Ideally, only the membrane potential is measurable and the internal state is hidden.
2) No exogenous inputs (noise) are considered in the current implementation.
3) The only mismatch between the plant and its model, M_p, used by the MPC is due to the PWA abstraction.

We summarize our contributions below before outlining the remaining sections.

1) MPCs for a nonlinear model-based neuron have been designed by using a PWA abstraction. The resulting MIQP optimization instance is solved using both online and explicit approaches.
2) The PWA abstraction is used to enable the design of explicit MPC in MPT. The toolbox has been extended to track moving reference trajectories by augmenting the state vectors and thus making the penalty matrices time-varying.
3) The online and explicit approaches to the PWA abstraction-based MPC are compared using several test cases to analyze the tradeoff between accuracy and speed.

The remainder of the paper is organized as follows. The next section introduces the two models M_p and M_c in detail. We formulate the MPC problem for the FHN model-based plant in Section III. The implementation details follow in Section IV. Then, we compare and contrast the online and explicit strategies in Section V. We discuss related work in Section VI before concluding with directions for future work in Section VII.

II. Physiological Background

As mentioned in the previous section, excitable cells are characterized by their response to an external electric current, called the stimulus. Nonlinear Differential Equation Models (DEMs) capture the behavior of excitable cells in terms of the change in the transmembrane potential in time, as the cell oscillates between depolarization and repolarization in response to the stimulus.

The FHN model [12] is a two-dimensional system of differential equations, representing the dynamics of a neuron:

$$\dot{v} = v(1-v)(v-a) - w + I(t), \tag{1a}$$
$$\dot{w} = bv - cw, \tag{1b}$$

where v is the dimensionless transmembrane potential, w is a dimensionless recovery variable, I is the magnitude of the stimulus current and the parameters a, b and c are given in Table I.

TABLE I. Parameters of the FHN model (M_p) used by the plant to simulate an excitable cell.

Parameter	a	b	c
Value	0.20	0.05	0.01

The MPC uses a Modified DR (MDR) model [13], a PWA version of the FHN model, to predict the plant's behavior (simulation of the excitable cell). The cubic term in (1a) is linearized to obtain the PWA dynamics (2):

$$\dot{v} = \tilde{p}(v) - w + I(t), \tag{2}$$

where

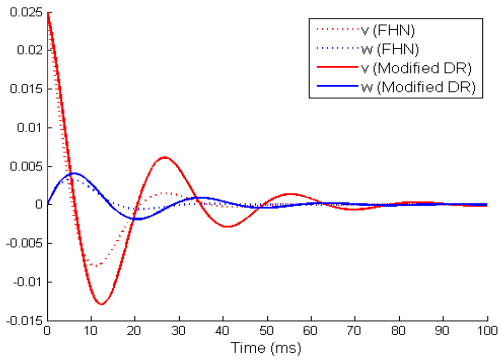


Figure 2. Simulations of the FHN and the MDR model. Maximum absolute L1 error for v = 0.0056 and w = 0.0013.

$$\tilde{p}(v) = \begin{cases} \frac{p(v_-)}{v_-} v & v < v_- \\ \left[\frac{p(v_+)-p(v_-)}{v_+-v_-}\right] v + \left[p(v_+) - \frac{p(v_+)-p(v_-)}{v_+-v_-} v_+\right] & v_- \le v \le v_+ \\ \frac{p(v_+)}{1-v_+}(1-v) & v > v_+. \end{cases} \tag{3}$$

The constants v_+ and v_- are given by

$$v_- = \frac{a + (1 - \sqrt{a^2 - a + 1})}{3}, \text{and} \tag{4a}$$

$$v_+ = \frac{a + (1 + \sqrt{a^2 - a + 1})}{3}. \tag{4b}$$

The function $p(.)$ is given by

$$p(v_-) = v_-(1 - v_-)(v_- - a), \text{and} \tag{5a}$$

$$p(v_+) = v_+(1 - v_+)(v_+ - a). \tag{5b}$$

The MDR model model represents M_c used by the controller to predict the plant's behavior and compute optimal stimuli values. It can be viewed as a hybrid model consisting of three modes and can be written in the following format:

$$\dot{\mathbf{x}} = A_i\mathbf{x} + B_i\mathbf{u} + \mathbf{f_i} \tag{6}$$

$$\mathbf{y} = C_i\mathbf{x} + D_i\mathbf{u} + \mathbf{g_i} \tag{7}$$

where

- i = index of the mode (piece),
- $\mathbf{x(t)} = \mathbb{R}^n$ state vector,
- $\mathbf{u(t)} = \mathbb{R}^m$ input vector,
- $\mathbf{y(t)} = \mathbb{R}^p$ output vector,
- $A_i = n \times n$ the *Dynamics matrix* for mode i,
- $B_i = n \times m$ the Input matrix for mode i,
- $C_i = p \times n$ the Output matrix for mode i,
- $D_i = p \times m$ the Feed-through matrix,
- f_i = non-homogeneity in dynamics - real vector of size $n \times 1$ and
- g_i = real vector of size $p \times 1$.

In the MDR model, we have:

1) Three modes, i.e., $1 \le i \le 3$.
2) Two states v and w, i.e., $n = 2$.
3) One input, Stimulus I, i.e., $m = 1$.
4) Two outputs v and w produced by plant, i.e., $p = 2$ (all the states are observable).

The MPC works in discrete time. MATLAB's *c2dm* function was used, with a sampling rate of 0.01 sec/sample and the *zero-order hold (zoh)* method, to generate the discrete time version of the MDR model. We obtain the following matrices based on the parameters in Table I. The superscript "d" denotes the corresponding matrix in the discrete time version.

1) Mode 1, i = 1
   a) Invariant: $v < 0.0945$.
   b) $A_1 = \begin{bmatrix} -0.0955 & -1 \\ 0.05 & -0.01 \end{bmatrix}$, $A_1^d = \begin{bmatrix} -0.9990 & -0.01 \\ 0.0005 & 0.9999 \end{bmatrix}$.
   c) $B_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $B_1^d = \begin{bmatrix} 0.01 \\ 0 \end{bmatrix}$.
   d) $f_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $f_1^d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.
   e) $C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $C_1^d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.
   f) $D_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $D_1^d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.
   g) $g_1 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $g_1^d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.
2) Mode 2, i = 2
   a) Invariant = $0.0945 \le v \le 0.7055$.
   b) $A_2 = \begin{bmatrix} 0.1867 & -1 \\ 0.05 & -0.01 \end{bmatrix}$, $A_2^d = \begin{bmatrix} 1.0019 & -0.01 \\ 0.0005 & 0.9999 \end{bmatrix}$.
   c) $B_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $B_2^d = \begin{bmatrix} 0.01 \\ 0 \end{bmatrix}$.
   d) $f_2 = \begin{bmatrix} -0.0267 \\ 0 \end{bmatrix}$, $f_2^d = \begin{bmatrix} -0.0267 \\ 0 \end{bmatrix}$.
   e) $C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $C_2^d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.
   f) $D_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $D_2^d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.
   g) $g_2 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $g_2^d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.
3) Mode 3, i = 3
   a) Invariant = $v > 0.7055$.
   b) $A_3 = \begin{bmatrix} -0.3566 & -1 \\ 0.05 & -0.01 \end{bmatrix}$, $A_3^d = \begin{bmatrix} 0.9964 & -0.01 \\ 0.0005 & 0.9999 \end{bmatrix}$.
   c) $B_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $B_3^d = \begin{bmatrix} 0.01 \\ 0 \end{bmatrix}$.
   d) $f_3 = \begin{bmatrix} 0.3566 \\ 0 \end{bmatrix}$, $f_3^d = \begin{bmatrix} 0.3566 \\ 0 \end{bmatrix}$.

e) $C_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $C_3^d = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

f) $D_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $D_3^d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

g) $g_3 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $g_3^d = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

Fig. 2 compares the FHN model and the MDR model in continuous time. A stimulus consisting of a spike of height 2.5 at the first time step was used to excite the model. The simulation was performed in MATLAB using the Euler method using a time step of 0.01 ms till 100 ms. Initial conditions were $v = 0$ and $w = 0$.

III. MPC Problem Formulation

Based on the $n \times 1$ state measurement (assuming that the complete state of the plant is observable: a state estimator would be needed in case of partial observability.) $\mathbf{x}(t)$ obtained at time t, the MPC predicts the dynamic behavior of the system and optimizes the control inputs such that the objective function in (8) is minimized:

$$\begin{aligned}
&\underset{U=[\mathbf{u}(t),\ldots,\mathbf{u}(t+N-1)]}{\text{minimize}} J(U, \mathbf{x}(t)) = \\
&\sum_{k=1}^{N} [(\mathbf{x}(t+k) - \mathbf{x}_{ref}(t+k))' \lambda^k . Q(\mathbf{x}(t+k) - \mathbf{x}_{ref}(t+k)) \\
&+ (\Delta \mathbf{u}(t+k-1))' R(\Delta\ \mathbf{u}(t+k-1))] \\
&\text{subject to:} \\
&\mathbf{x}(t+k+1) = A_i^d \mathbf{x}(t+k) + B_i^d \mathbf{u}(t+k) + \mathbf{f}_i^d, \\
&\mathbf{y}(t+k) = C_i^d \mathbf{x}(t+k) + D_i^d \mathbf{u} + \mathbf{g}_i, \\
&\text{where} \\
&\Delta \mathbf{u}(t+k-1) = \mathbf{u}(t+k-1) - \mathbf{u}(t+k-2) \\
&0 \leq k \leq N-1 \text{ and } 1 \leq i \leq 3.
\end{aligned} \tag{8}$$

Optimization is performed over a finite horizon of length N. Q is an $n \times n$ identity matrix and $0 < \lambda \leq 1$ is a parameter that assigns exponentially receding weights to the predicted deviations, $(\mathbf{x}(t+k) - \mathbf{x}_{ref}(t+k))$, over the horizon. Thus, the scheme is also called receding horizon control. R is a positive definite matrix that determines the penalty on differences between consecutive inputs.

The optimization problem is solved at time t and the inputs are calculated for the next N time steps. Only the next input is passed on to the plant, before repeating the MPC process.

IV. Implementation of Model Predictive Controllers for the FHN model-based Plant

Explicit and online MPCs were implemented for the FHN model-based plant using the MDR model for prediction purposes. The receding horizon parameter λ was fixed at 0.8 and $R = \begin{bmatrix} 10^{-3} \end{bmatrix}$ was the input penalty matrix. Next, we describe the implementation aspects of the online and explicit MPCs.

A. Online MPC

Online MPC involves solving (8) at every time step in runtime. The constrained nonlinear optimizer *fmincon* [14] was used to implement online MPC in MATLAB. At each time step t, the current state $\mathbf{x}(t)$ of the plant and the reference input $[\mathbf{x}_{ref}(t+1), ..., \mathbf{x}_{ref}(t+N)]$ over the finite horizon N were provided to the controller which then computed the optimal input for the FHN plant. An interior point algorithm was used for optimization. The FHN plant was then simulated using Euler method for one time step by applying the optimal input. This process was repeated for the whole simulation duration.

B. Explicit MPC

In explicit MPC, the optimization problem of (8) is cast as an instance of multi-parametric quadratic programming (mpQP) and solutions are computed offline for possibly overlapping polyhedral partitions, also known as *coverings*, of the state space. As shown in Fig. 3, the result of this one-time computation is a table of control laws corresponding to the partitions. At runtime, the current state sample is tested for membership in the list of partitions. The state may lie in more than one region due to possible overlap. In this case, the control law resulting in the most optimal value of the objective function is applied. The process is then repeated for the next state sample.

We implemented an explicit MPC for the FHN model-based neuron simulator using the MDR model as its PWA abstraction. The Multi Parametric Toolbox (MPT) [11] was used to implement the MDR-model based explicit MPC in MATLAB. The current implementation of MPT supports time-varying reference trajectories, but it considers constant reference at every time step of prediction horizon. We extended the tool to overcome this limitation. In the remaining sections, we elaborate on these modifications.

The key idea for incorporating time-varying reference trajectories is as follows. The reference trajectory over the prediction horizon, $\mathbf{x}_{ref}$ is considered to be to be a sequence of unknown variables. Then these unknown variables are used to augment the state vector $\mathbf{x}$. Then, the dynamics of the augmented system is reformulated in a Δu-form, as the input necessary to keep the states at the reference are also not generally known. In this formulation, the input at time k is $\Delta u(k)$, where $u(k-1)$ is an additional state in the dynamical model. So, the original system input can be obtained as $u(k) = u(k-1) + \Delta u(k)$. The state update equation is then given by (9).

$$\begin{aligned}
\begin{pmatrix} x(k+1) \\ u(k) \\ x_{ref}(k+1) \end{pmatrix} &= \begin{pmatrix} A_i & B_i & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} x(k) \\ u(k-1) \\ x_{ref}(k) \end{pmatrix} \\
&+ \begin{pmatrix} A_i \\ B_i \\ 0 \end{pmatrix} \Delta u(k),
\end{aligned} \tag{9}$$

As the state vector is augmented with new state variables, the penalty matrix Q needs to be augmented, too. The newly augmented penalty matrix is given by

$$\begin{pmatrix} Q & 0 & -Q \\ 0 & 0 & 0 \\ -Q & 0 & Q \end{pmatrix}$$

In our modification scheme, we consider time-varying reference at all steps of the prediction horizon. We augment x with the reference state vector for all steps of the horizon. The

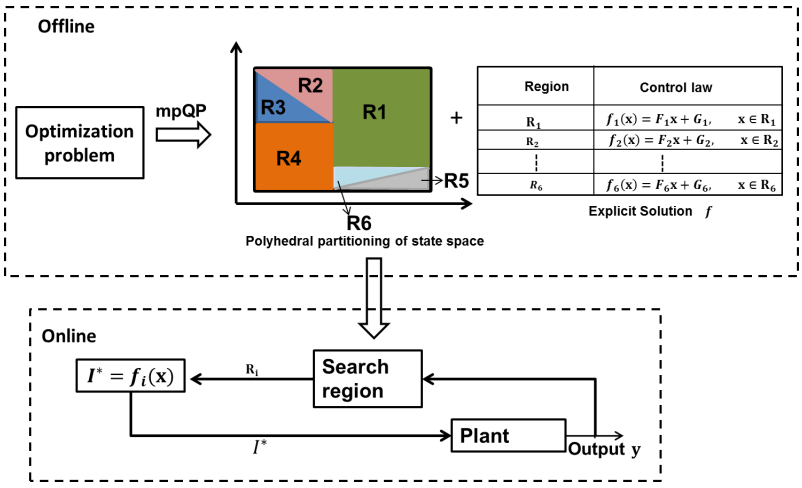


Figure 3. Workflow of Explicit MPC.

newly modified state update equation is given by (10)

$$\begin{pmatrix} x(k+1) \\ u(k) \\ x^1_{ref}(k+1) \\ \vdots \\ x^N_{ref}(k+1) \end{pmatrix} = \begin{pmatrix} A_i & B_i & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} x(k) \\ u(k-1) \\ x^1_{ref}(k) \\ \vdots \\ x^N_{ref}(k) \end{pmatrix} + \begin{pmatrix} A_i \\ B_i \\ 0 \end{pmatrix} \Delta u(k), \tag{10}$$

where N is the number of steps in the horizon and $x^j_{ref}(k)$ is the reference vector in j^{th} time step of the horizon.

Due to the receding horizon principle, the penalty matrix will be different for each step of prediction horizon. Current implementation of MPT is amenable to add varying penalty matrix. The general form of the penalty matrix is given by

$$\begin{pmatrix} Q(k) & 0 & -Q(k) & 0 \\ 0 & 0 & 0 & 0 \\ -Q(k) & 0 & Q(k) & 0 \end{pmatrix},$$

where $Q(k) = \lambda^k Q$ for $k-th$ step of the prediction horizon.

The positions of $-Q(k)$ in the first row and $Q(k)$ in the third row are adjusted based on k.

V. Results

We conducted two sets of experiments on the online and the explicit MPCs to compare and contrast them.

Experiment Set 1: *Speed vs. Accuracy Tradeoff for Different Reference Trajectories*

The online and the explicit MPCs were tested against the same reference trajectory to study the speed vs. accuracy tradeoff. Two reference trajectories $[v^i_r(t), w^i_r(t)]$, $i = 1, 2$ were generated by simulating the FHN model. The protocols used to generate them were as follows.

S1 Protocol for generating $[v^1_r(t), w^1_r(t)]$

1) Initial conditions: $v = 0$, $w = 0$ (rest conditions).
2) Time step used in the simulation: 0.1 ms.
3) Total duration of simulation: 240 ms (2400 time steps).
4) Stimuli pattern: One time-step-long (0.1ms) supra-threshold stimulus pulses of intensity (height) 1.5 were applied every 80 ms, to produce three APs in the simulation. Thus, the pacing frequency was 12.5 Hz.

S2 Protocol for generating $[v^2_r(t), w^2_r(t)]$

1) Initial conditions: $v = 0$, $w = 0$ (rest conditions).
2) Time step used in the simulation: 0.1 ms.
3) Total duration of simulation: 240 ms (2400 time steps).
4) Stimuli pattern: 10-steps-long (1 ms) supra-threshold stimulus pulses of intensity (height) 1.5 were applied every 80 ms, to produce three APs in the simulation. Thus, the pacing frequency was again 12.5 Hz.

The simulation was carried out using the Euler's method of numerical integration in MATLAB. Both the MPCs were tested against the S1 and S2 reference trajectories using a 3-step lookahead horizon. Their performance was compared using the following two metrics:

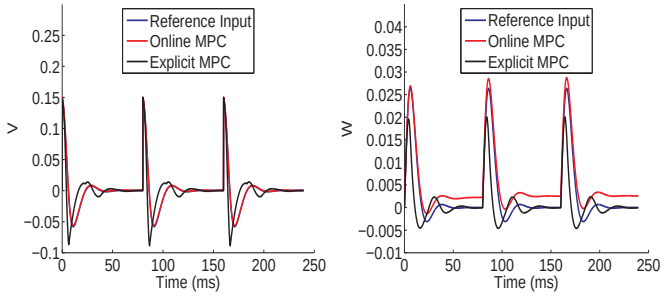
1) **Accuracy of the plant's operation** (μ^v_{l2}, μ^w_{l2}): measured using the mean L2 error between the reference trajectory and the output of the simulation carried out by the plant.
2) **Timeliness constraint on the MPC**: dictates that the working of the MPC must be fast enough to cope with the plant's operation. The degree to which the two MPCs met this constraint was measured as follows. The time taken by the Euler method-based simulation for producing the reference trajectory was noted, say t_1 secs. The plant + MPC combination was run on a single thread in a lock-step fashion, i.e., the plant was halted till the MPC finished its computation and provided the stimuli value for the next time step. The total time taken for tracking the reference trajectory was noted, say t_2 secs. Then, $t_{12} = (t_2 - t_1)$ provided an estimate of the time taken by the MPC to compute the stimuli. Ideally, $t_{12} < t1$, which ensures that the MPC's computation runs faster than the rate at which the plant evolves (simulates the FHN model).

Table II provides performance metrics for the two MPCs. Fig. 4 and Fig. 5 plot the evolution of v and w for protocols

S1 and S2, respectively.

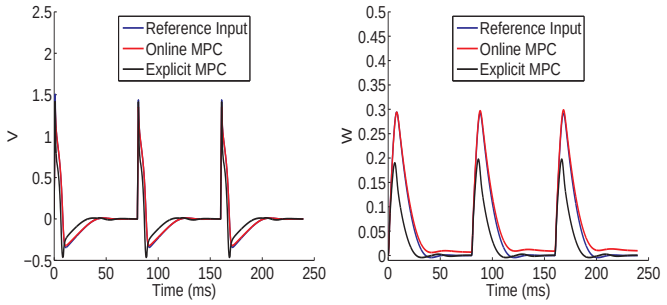
TABLE II. PERFORMANCE METRICS FOR ASSESSING THE SPEED VS. ACCURACY TRADEOFF ACROSS DIFFERENT REFERENCE TRAJECTORIES.

Protocol	Controller	μ_{l2}^{v}	μ_{l2}^{w}	$t_1(s)$	$t_2(s)$
S1	**Online MPC**	2.8×10^{-5}	4.9×10^{-5}	0.023	136.8
	Explicit MPC	4.3×10^{-4}	1.1×10^{-4}	0.023	88.8
S2	**Online MPC**	1.8×10^{-4}	2.2×10^{-4}	0.023	147.3
	Explicit MPC	2.7×10^{-2}	1.1×10^{-2}	0.023	88.3



(a) Evolution of v. (b) Evolution of w.

Figure 4. Performance of the online and explicit MPCs on spike-shaped stimuli produced by the S1 protocol.



(a) Evolution of v. (b) Evolution of w.

Figure 5. Performance of the online and explicit MPCs on rectangular pulse-shaped stimuli produced by the S2 protocol.

Discussion

Following inferences can be made from the results shown in the preceding paragraphs:

1) *The accuracy of the fmincon-based online MPC is better than the MPT-based explicit MPC.* This can be attributed to an accurate solution to the optimization problem, found by fmincon at run time, for the specific current state of the plant. The explicit MPC on the other hand, partitions the state space and finds a common control law for the whole partition. Accuracy is lost in this process.
2) *The fmincon-based online MPC is much slower than the MPT-based explicit MPC.* FMINCON exhaustively explores the whole state space at every time step of the plant's operation. This leads to its slow operation. The most time consuming step for the explicit MPC is searching for the partition corresponding to the current state, and this is achieved much faster than the online MPC's operation.

Experiment Set 2: *Effect of Horizon Length on Explicit MPC* Explicit MPC is enabled for time-varying reference trajectories by augmenting the state vectors and reformulating the dynamics. State space augmentation leads to an exponential increase in the number of polyhedral partitions. Table III compares the build-time and the number of partitions for different horizon lengths. Increasing the horizon length N is

TABLE III. EFFECT OF N ON EXPLICIT MPC DESIGN IN MPT. HORIZON LENGTH OF 0, WHICH CORRESPONDS TO 0-STEP LOOKAHEAD, IS SPECIFIED FOR COMPARISON PURPOSES.

Horizon length	Build-time in MPT (secs.)	Number of partitions
0	0.45	3
1	2.78	30
2	51.36	277
3	576.41	2581

expected to improve the predictive accuracy of MPC. In the case of MPT-based design of explicit MPC, we observed that accuracy did not improve considerably on changing the horizon length from 2-step to 3-step lookahead. For both cases, the mean L2 errors for v and w were recorded to be around 0.027 and 0.001 respectively (for S2-type stimuli).

Having a smaller horizon leads to significant reduction in the search space, while searching for the partition corresponding to a given state sample. This reduction is critical as the search operation is performed at every time step during operation. In our case, t_2 reduced from 88.3 secs. to 11.2 secs when the horizon length was changed from 3 to 2 for the S2 protocol.

VI. RELATED WORK

MPC has been widely used in many domains like the chemical, food-processing, automotive, and aerospace industries. An exhaustive survey of both the theoretical and the practical aspects can be found in [15]. Explicit MPC, which is relatively new, has been surveyed in [16]. Recently MPC has found interesting biological and biomedical applications. In [17], [18], a platform for in silico realtime closed-loop control of gene expression in yeast has been proposed. It uses MPC to perturb inducible promoters in a systematic way to gain insights about gene expression. MPC has been successfully applied to devise therapeutic strategies in [19], [20], [21], [22]. In [23] MPC is used for functional electrical stimulation to estimate stimulation patterns for muscles that have been paralyzed due to spinal cord injury. Controllers for tracking neuron APs are designed in [24], [25] and thus are closest to our work. We compare and contrast each of them with our MPC-based approach below.

In [24], an adaptive input-output feedback linearization controller is presented to track a nominal AP using the FHN model. In contrast, MPC is a feed-forward control technique. Its predictive capability allows the controller to quickly adapt to a model mismatch caused due to the degradation/aging of excitable cells. Parameter estimation and tuning the model are the only steps involved in adapting to the changes, whereas a feedback controller needs complete redesign. Also, explicit MPC can track arbitrary fast-changing APs whereas the controller in [24] is designed for a nominal AP.

In [25], the authors present a sophisticated controller for a neuron based on the Hodgkin-Huxley (HH) model [26]. The HH model is augmented with random variables to capture stochastic behavior and external disturbances. Membrane potential is treated as the only observable and a state-estimator is employed for the other hidden variables of the HH model. On the other hand, we focus on comparing the online and

explicit approaches to MPC in the case of a nonlinear plant being modeled using a PWA abstraction. The realistic setting of [25] is complementary to our work and provides directions for extending our scheme. Also, the FHN and the MDR models used in our work are order-reduced versions of the HH model.

VII. Conclusion and Future Work

Explicit and online MPCs were presented for tracking a reference sequence of APs using an FHN model-based neuron simulator. The MPCs employ a PWA abstraction of the nonlinear plant, thus enabling a QP formulation of the model predictive control optimization problem. The speed versus accuracy tradeoff was assessed using several test cases. The online approach provides excellent accuracy, but fails to satisfy the timeliness constraint. Offline MPC on the other hand, satisfies the timeliness constraint for a limited set of reference trajectories, but provides relatively lower accuracy than the online version.

We plan to pursue a combined approach that uses the best features of both the explicit and online MPCs to achieve high accuracy while satisfying the timeliness constraint. We also plan on adding noise to our implementation to test the robustness of the controller. Better QP solvers and search techniques will be explored to speed up the explicit MPC. The combined approach will then be examined for closed-loop stability and computational efficiency. In a realistic setting, the transmembrane potential is the only observable and state estimators would be added on the lines of [25]. We would like also to investigate real implementations of the proposed controllers using field-programmable gate arrays or other embedded microcontrollers. Finally, the MPC-based approach would be applied to complex excitable cells such as cardiac myocytes. In particular, we aim to investigate the use of piecewise multi-affine approximations of [27] to design explicit and online MPCs for cardiac cells.

Acknowledgments: We would like to thank the anonymous reviewers for their valuable comments. Research supported in part by NSF grants CCF-0926190 and CCF-1018459, and by AFOSR grant FA0550-09-1-0481.

References

[1] A. T. Winfree, "Heart muscle as a reaction - diffusion medium: The roles of electric potential diffusion, activation front curvature, and anisotropy," International Journal of Bifurcation and Chaos, vol. 7, no. 3, March 1997, pp. 487–526.

[2] A. Kléber, "The fibrillating atrial myocardium. What can the detection of wave breaks tell us?" Journal of cardiovascular research, vol. 48, no. 2, August 2000, pp. 181–184.

[3] R. Findeisen and F. Allgöwer, "An introduction to nonlinear model predictive control," in Proceedings of the 21st Benelux Meeting on Systems and Control, Veidhoven, 2002, pp. 1–23.

[4] K. N. Fountas et al., "Implantation of a closed-loop stimulation in the management of medically refractory focal epilepsy, a technical note," Stereotactic and Functional Neurosurgery, vol. 83, no. 4, 2005, pp. 153–158.

[5] S. Luther et al., "Low-energy control of electrical turbulence in the heart," Nature, vol. 475, 2011, pp. 235–239.

[6] S. J. Qin and T. A. Badgwell, "An overview of nonlinear model predictive control applications," in Nonlinear Predictive Control. Verlag, 2000, pp. 369–392.

[7] S. O. Krumke, "Nonlinear optimization," Lecture Notes, 2004.

[8] S. Summers, D. M. Raimondo, C. N. Jones, J. Lygeros, and M. Morari, "Fast explicit nonlinear model predictive control via multiresolution function approximation with guaranteed stability," in 8th IFAC Symposium on Nonlinear Control Systems, 2010, pp. 533–538.

[9] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," Automatica, vol. 35, no. 3, 1999, pp. 407 – 427.

[10] W. Heemels, B. D. Schutter, and A. Bemporad, "Equivalence of hybrid dynamical models," Automatica, vol. 37, no. 7, 2001, pp. 1085 – 1091.

[11] "Multi-Parametric Toolbox (MPT)," 2004, URL: http://control.ee.ethz.ch/~mpt [accessed: 2014-03-25].

[12] E. M. Izhikevich, Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting. The MIT Press, 2007.

[13] J. G. Dumas and A. Rondepierre, "Modeling the electrical activity of a neuron by a continuous and piecewise affine hybrid system," in Proceedings of the 6th international conference on Hybrid systems: computation and control. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 156–171.

[14] "MATLAB Optimization Toolbox," 2014, URL: http://www.mathworks.com/help/toolbox/optim [accessed: 2014-03-25].

[15] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: theory and practice a survey," Automatica, vol. 25, no. 3, 1989, pp. 335–348.

[16] A. Alessio and A. Bemporad, "A survey on explicit model predictive control," Nonlinear Model Predictive Control, Lecture Notes in Control and Information Sciences, vol. 384, 2009, pp. 345–369.

[17] J. Uhlendorf et al., "Long-term model predictive control of gene expression at the population and single-cell levels," Proceedings of the National Academy of Sciences, vol. 109, no. 35, 2012, pp. 14 271–14 276.

[18] J. Uhlendorf, P. Hersen, and G. Batt, "Towards real-time control of gene expression: *in silico* analysis," in Proceedings of the IFAC World Congress, vol. 18, 2011, pp. 14 844–14 850.

[19] H. Chang, A. Astolfi, and H. Shim, "A control theoretic approach to venom immunotherapy with state jumps," in Proceedings of the International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2010, pp. 742–745.

[20] T. Chen, N. F. Kirkby, and R. Jena, "Optimal dosing of cancer chemotherapy using model predictive control and moving horizon state/parameter estimation," Computer Methods and Programs in Biomedicine, vol. 108, no. 3, 2012, pp. 973 – 983.

[21] S. L. Noble, E. Sherer, R. E. Hannemann, D. Ramkrishna, T. Vik, and A. E. Rundell, "Using adaptive model predictive control to customize maintenance therapy chemotherapeutic dosing for childhood acute lymphoblastic leukemia," Journal of Theoretical Biology, vol. 264, no. 3, 2010, pp. 990 – 1002.

[22] Jeffry A. Florian Jr., J. L. Eiseman, and R. S. Parker, "Nonlinear model predictive control for dosing daily anticancer agents using a novel saturating-rate cell-cycle model," Computers in Biology and Medicine, vol. 38, no. 3, 2008, pp. 339 – 347.

[23] S. Mohammed, P. Poignet, P. Fraisse, and D. Guiraud, "Toward lower limbs movement restoration with input-output feedback linearization and model predictive control through functional electrical stimulation," Control Engineering Practice, vol. 20, no. 2, 2012, pp. 182 – 195.

[24] R. Naderi, M. J. Yazdanpanah, A. Azemi, and B. Roaia, "Tracking normal action potential based on the FHN model using adaptive feedback linearization technique," in Proceedings of the IEEE International Conference on Control Applications (CCA), 2010, pp. 1458–1463.

[25] B.S. Chen and C.W. Li, "Robust observer-based tracking control of hodgkin-huxley neuron systems under environmental disturbances," Neural computation, vol. 22, no. 12, 2010, pp. 3143–3178.

[26] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," Journal of Physiology, vol. 117, 1952, pp. 500–544.

[27] R. Grosu et al., "From cardiac cells to genetic regulatory networks," in Proceedings of Computer Aided Verification, ser. Lecture Notes in Computer Science, vol. 6806. Springer Berlin/Heidelberg, 2011, pp. 396–411.