Predicting and Evaluating the Effect of Bivalirudin in Cardiac Surgical Patients

Qi Zhao, Thomas Edrich, *Member, IEEE*, and Ioannis Ch. Paschalidis*, *Senior Member, IEEE*

***Abstract*—Bivalirudin, used in patients with heparin-induced thrombocytopenia, is a direct thrombin inhibitor. Since it is a rarely used drug, clinical experience with its dosing is sparse. We develop two approaches to predict the *Partial Thromboplastin Time (PTT)* based on bivalirudin infusion rates. The first approach is model free and utilizes regularized regression. It is flexible enough to be used as predictors bivalirudin infusion rates measured over several time instances before the time at which a PTT prediction is sought. The second approach is model based and proposes a specific model for obtaining PTT which uses a shorter history of the past measurements. We learn population-wide model parameters by solving a nonlinear optimization problem. We also devise an adaptive algorithm based on the extended Kalman filter that can adapt model parameters to individual patients. The latter adaptive model emerges as the most promising as it yields reduced mean error compared to the model-free approach. The model accuracy we demonstrate on actual patient measurements is sufficient to be useful in guiding the optimal therapy.**

***Index Terms*—Bivalirudin, extended Kalman filter (EKF), nonlinear optimization, pharmacokinetics, regression.**

I. Introduction

BIVALIRUDIN antagonizes the effect of thrombin in the blood clotting cascade, thereby preventing complications from blood clotting. It is currently FDA approved for short-term anticoagulation of patients undergoing cardiac catheterization to prevent complications due to undesired blood clots [1]–[3]. Bivalirudin is infused as a "blood thinner" in patients who have or are suspected of having blood clots or risk of blood clotting and who have a contraindication to heparin. It is infused continuously, and is eliminated via the kidneys and by plasma protease-metabolism. It affects the coagulation parameters *Partial Thromboplastin Time (PTT)* and the *International Normalized Ratio (INR)* in a dose-dependent fashion. Both measure the ability of the blood to clot but while PTT is measured in seconds, INR is a dimensionless number. PTT, in particular, is measured in the lab by exposing the blood to a thrombogenic substance and then counting the seconds until a blood clot is formed.

As a rarely used drug, bivalirudin is used more frequently in the *Intensive Care Unit (ICU)* but the residents adjusting the infusion rate may have little experience, resulting in overdosing or underdosing. Adequate anticoagulation is necessary to avoid the risk of clot formation, but overshooting increases the risk of bleeding. There is considerable inter- and intraindividual variability in the response to bivalirudin; it is challenging to titrate the drug. Currently, only empirical titration of bivalirudin based on clinical experience or a simple nomogram is used to achieve desired anticoagulation [4]. For this reason, a mathematical model that predicts the PTT based on the past infusion rates of bivalirudin following dose adjustment would be extremely useful in guiding the optimal therapy.

In earlier works [5] and [6], we have built a simple one-state linear system model to describe the effect of bivalirudin in patients. The models were designed using MATLAB/simulink (MathWorks, Natick, MA, USA) and default parameter identification procedures. Motivated by this work, in this paper, we develop two new methods to predict the PTT values based not only on past bivalirudin infusion rates but also on a host of patient-specific physiological variables that characterize coagulation, renal, and liver function. The results we obtain substantially improve accuracy compared with our earlier work.

Our first method is model free, in the sense that a specific model does not need to be constructed in advance, and leverages regularized and kernelized regression. With uniform sampled data, standard time-series analysis methods (e.g., ARX, ARMAX models [7]) could have been a viable alternative. In our problem, however, we encounter highly nonuniform sampled data which challenge the standard methods, hence, our use of regularized regression. Our method is purely data driven and requires no explicit model to explain how bivalirudin affects PTT. It is flexible enough to use several samples of bivalirudin infusion rates from the immediate past in order to predict the current PTT values. Since we use a rich set of predictors, we devise a regularization approach that can eliminate unnecessary predictors and regress on a reduced predictor set so as to avoid overfitting.

Our second method develops a more complex explicit dynamic state-space system model than the one developed in [5] and [6]. This new model takes into account the elimination of

Manuscript received March 12, 2013; accepted August 29, 2013. Date of publication September 4, 2013; date of current version January 16, 2014. This work was supported in part by the NIH/NIGMS under Grant R01-GM093147, in part by the NSF under Grants EFRI-0735974, CNS-1239021, and IIS-1237022, in part by the ARO under Grants W911NF-11-1-0227 and W911NF-12-1-0390, in part by the ONR under Grant N00014-10-1-0952, and by the STAR (Surgical ICU Translational Research) Center at Brigham and Women's Hospital. T. Edrich and I. C. Paschalidis are equal contributors to this work. *Asterisk indicates corresponding author*.

Q. Zhao is with the Division of Systems Engineering, Boston University, Boston, MA 02215 USA (e-mail: zhaoqi@bu.edu).

T. Edrich is with the Department of Anesthesia, Perioperative and Pain medicine, Brigham and Women's Hospital, Boston, MA 02215 USA (e-mail: tedrich@partners.org).

*I. C. Paschalidis is with the Department of Electrical and Computer Engineering, Division of Systems Engineering, and Center for Information and Systems Engineering, Boston University, Boston, MA 02215 USA (e-mail: yannisp@bu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBME.2013.2280636

bivalirudin by the kidney and liver. We identify the model parameters by formulating a nonlinear optimization problem that minimizes the ℓ_2 norm of prediction error over a training set of measurements. As we mentioned before, we only have highly nonuniform sampled real data. Furthermore, the dosage of Bivalirudin given to patients should be carefully titrated to ensure patient safety. As a result, we cannot observe PTT values in response to arbitrary dosage. This suggests that canonical state-space system identification techniques (e.g., adaptive system identification [8], subspace state-space system identification [9]) are not applicable.

The nonlinear optimization problem we formulate is solved by leveraging quasi-Newton methods. The dynamic system model we obtain performs only somewhat worse than the model free approach, even though it uses a shorter history of past measurements. Building on this model, we develop an adaptive online algorithm based on the extended Kalman filter (EKF) that can adapt the model parameters to individual patients. The algorithm starts from population-wide optimal parameters and, as it observes the inputs and outputs, modifies the model parameter values to better fit an individual patient. This adaptive model outperforms the model-free method in terms of average prediction error, despite using a shorter history of the past measurements. This confirms the empirically known considerable individual variability in the response to bivalirudin. Further, with the adaptive model, the per-patient error variability is substantially reduced compared to the population-wide dynamic system model and it is on par with the corresponding error variance of the model-free method.

The approaches we put forth in this paper are general and can be applicable to a host of related problems. As ICUs and hospital wards accelerate the digitization of patient records, tremendous opportunities arise for automated and mathematically rigorous patient monitoring and medication dosing. It is in such a framework that the methods we develop can become useful.

The remainder of this paper is organized as follows. Section II presents the model-free regularized regression method. Section III presents the dynamic system model. Section IV develops the EKF to achieve model parameter adaptation. Finally, the concluding remarks appear in Section V.

Notation: We use bold letters to denote vectors and matrices; typically, vectors are denoted by lower case letters and matrices by upper case letters. The vectors are assumed to be column vectors unless explicitly stated otherwise. For economy of space, we write $\mathbf{x} = (x_1, \ldots, x_n)$ for the n-dimensional column vector $\mathbf{x} \in \mathbb{R}^n$. Prime denotes transpose, $\|\cdot\|$ denotes the Euclidean norm, $\mathbf{0}$ denotes a vector or matrix with all components set to zero, and $\mathbf{I}$ is the identity matrix.

II. Regularized Regression

In this section, we present our first—"black box"—approach. The idea is to use regression with appropriate function regularizers that can help avoid overfitting. We start with some preliminary material on regularization and then describe the features we use and the results we obtained.

A. Preliminaries

For completeness and to establish our notation, we review the standard material on regularized regression from [10] and [11]. Let $\mathbf{x} \in \mathbb{R}^p$ denote the vector of features or predictors and $y \in \mathbb{R}$ the output or response. Suppose we are given a training set $(\mathbf{x}_i, y_i), i = 1, \ldots, N$, and we are interested in selecting a function $f : \mathbb{R}^p \to \mathbb{R}$ from some space $\mathscr{H}$ so that it approximately holds $y_i \approx f(\mathbf{x}_i)$ over the training set. To that end, we can adopt a loss function $L(y, f(\mathbf{x}))$ that quantifies the quality of the fitting and solve the optimization problem

$$\min_{f \in \mathscr{H}} \left\{ \sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i)) + \lambda G(f) \right\} \tag{1}$$

where $G(f)$ is a regularization term that penalizes functions from $\mathscr{H}$ which are not smooth enough, and λ is a parameter that determines the contribution of the penalty term. The purpose of the regularization term is to induce smoother functions, and thus help avoid overfitting. The parameter λ can be selected by cross validation as we will explain later.

An important special case concerns functions that live in a so called *reproducing kernel Hilbert space* $\mathscr{H}_K$. For some (positive semidefinite) *kernel* function $K(\mathbf{x}, \mathbf{z}), \mathbf{x}, \mathbf{z} \in \mathbb{R}^p$, written as

$$K(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{\infty} \gamma_j \phi_j(\mathbf{x}) \phi_j(\mathbf{z})$$

where $\phi_j(\cdot)$ are basis functions (e.g., polynomial, Gaussian, cosine, etc.) and the scalars $\gamma_j \geq 0$ satisfy $\sum_{j=1}^{\infty} \gamma_j^2 < \infty$, the functions $f \in \mathscr{H}_K$ can be written as

$$f(\mathbf{x}) = \sum_{j=1}^{\infty} c_j \phi_j(\mathbf{x}).$$

The regression problem (1) can now be written as

$$\min_{f \in \mathscr{H}_K} \left\{ \sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i)) + \lambda \|f\|^2_{\mathscr{H}_K} \right\} \tag{2}$$

where the regularizer is the norm in $\mathscr{H}_K$ defined as $\|f\|^2_{\mathscr{H}_K} = \sum_{j=1}^{\infty} c_j^2 / \gamma_j$. It can actually be shown that the solution to this problem has the form [11]

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \tag{3}$$

and, in light of this form, finding the parameters α_i amounts to solving a finite-dimensional optimization problem. If in fact one assumes a nice structure for the loss function, e.g., squared loss $L(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2$, then there are very efficient algorithms for finding the α_i. In the case of squared loss, all we need to do is solve a simple convex unconstrained quadratic optimization problem which admits a closed-form solution.

The choice of the kernel is important but very much dependent on the particular application one has to tackle. In our problem, a *polynomial kernel* performed better than alternatives. A polynomial kernel is of the form $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}'\mathbf{z} + 1)^d$, for $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$ and some integer d, which is the sum of all monomials

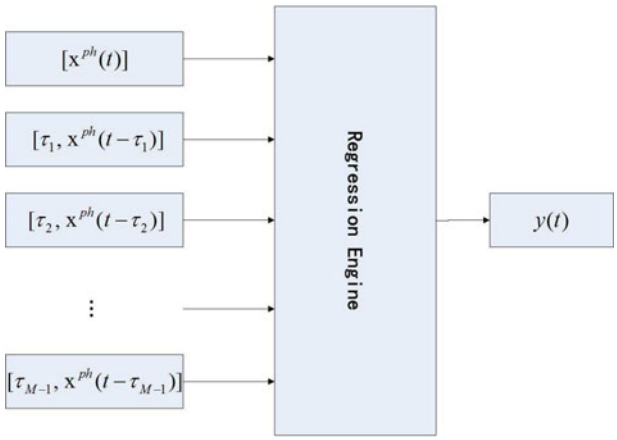


Fig. 1. Inputs (predictors) and the output (response) of the regression engine.

$\prod_{i=1}^{p} x_i^{d_i^x} z_i^{d_i^z}$ with $d_i^x, d_i^z = 0, 1, \ldots, d$ such that $\sum_{i=1}^{p} d_i^x \leq d$ and $\sum_{i=1}^{p} d_i^z \leq d$.

B. Predictors

In our problem, we have clinical data from 233 patients. The key quantity (response) we would like to predict is the PTT $y_i(t)$ of each patient i at each time t. As predictors, we include 11 key physiological variables sampled over M consecutive time instants $t > t - \tau_1 > \cdots > t - \tau_{M-1}$ where τ_m for $m = 1, \ldots, M-1$ denote the time lags between the consecutive measurements. These time lags (given in days) are included as the 12th predictor.

The 11 physiological variables are:

1) *Bival rate* (mg/kg/h)*:* the weight-based bivalirudin injection rate.
2) *GFR* (mL/min)*:* the glomerular filtration rate, reflecting the ability of the kidneys to eliminate bivalirudin. Decreased GFR would increase the serum level of bivalirudin and the PTT in an approximately linear fashion [12].
3) *PTT* (s)*:* last measured PTT value.
4) *INR* (unit-less)*:* the last measured INR value—a coagulation time that is distinct, but associated with the PTT. It increases as the serum level of bivalirudin increases.
5) *SGOT* (units/L)*:* the serum glutamic oxaloacetic transaminase, and
6) *SGPT* (units/L)*:* the serum glutamic pyruvic transaminase. Increased SGOT and SGPT suggest liver dysfunction and decreased production of clotting factors, thus, increasing the value of PTT. Since the liver produces clotting factors, liver dysfunction will increase the PTT.
7) *TBILI* (mg/dL)*:* total bilirubin, a "waste product" normally eliminated by the liver. In liver dysfunction, this is positively associated with a rising PTT.
8) *ALB* (g/L)*:* Albumin, which is reduced in the case of liver failure and is therefore associated with a rising PTT.
9) *PLT* (K/mcL)*:* Platelet count. Platelets help form blood clots with clotting factors from the liver. They are utilized when a clot is formed. A decreasing platelet count can indicate ongoing clotting with consumption of clotting factors, thus, elevating the PTT and INR.
10) *HCT* (%)*:* Hematocrit. HCT is the volume percentage of red blood cells in the blood. When patients loose blood during operations and other nonoperative bleeding, then fluids such as normal saline are provided to make up for the blood volume lost. This, however, lowers the HCT. At the same time, the added volume dilutes the clotting factors in the blood and causes PTT and INR to increase.
11) *FIB* (mg/dL)*:* Fibrinogen. This protein helps produce clots and its decreased concentration may indicate that clotting is occurring. It follows that clotting factors are being depleted which will cause elevated PTT and INR.

Let now $\mathbf{x}_i^{ph}(t) \in \mathbb{R}^{11}$ denote the vector containing the 11 physiological variables presented previously for patient i at time t. Note that $\mathbf{x}_i^{ph}(t)$ contains PTT and INR values at time $t - \tau_1$. To predict $y_i(t)$, we will use the predictor vector

$$\mathbf{x}_i(t) = (\mathbf{x}_i^{ph}(t), \tau_1, \mathbf{x}_i^{ph}(t-\tau_1), \ldots, \tau_{M-1}, \mathbf{x}_i^{ph}(t-\tau_{M-1})) \tag{4}$$

that is, M consecutive measurements of $\mathbf{x}_i^{ph}(t)$ and the corresponding time lags which form a $(12M-1)$-dimensional vector (see Fig. 1).

Our regression engine uses a polynomial kernel $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}'\mathbf{z} + 1)^d$ which can be written in terms of its (orthonornal) eigenfunctions, say as $K(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{J} \phi_j(\mathbf{x})\phi_j(\mathbf{z})$, where J is the number of eigenfunctions. For our polynomial kernel, it holds $J = \binom{12M-1+d}{d}$. The response function takes the form $f(\mathbf{x}) = \sum_{j=1}^{J} c_j \phi_j(\mathbf{x})$. We write $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_J(\mathbf{x}))$. We formulate a slightly different problem than (2) in that we use the so-called elastic net penalty [13]. More specifically, we minimize over the coefficients $\mathbf{c} = (c_1, \ldots, c_J)$ the expression

$$\sum_{i=1}^{N} \sum_{t_i} [y_i(t) - \mathbf{c}'\boldsymbol{\phi}(\mathbf{x}_i(t))]^2 + \lambda_1 \sum_{j=1}^{J} c_j^2 + \lambda_2 \sum_{j=1}^{J} |c_j| \tag{5}$$

where N henceforth represents the number of patients in the training set and the summation over t_i is over the time instances in which we have measurements for those patients. The difference with (2) is the addition of a penalty term equal to the ℓ_1 norm of $\mathbf{c}$. This is done to induce sparsity and eliminate features that are "redundant" and not helpful in predicting $y_i(t)$. In our experiments, we found that it was important to include such a sparsity-inducing penalty especially because we use multiple time instances from the past to predict the current PTT value. Notice that the objective function in (5) is convex, thus, this problem can be easily solved using the standard optimization techniques. The parameters λ_1 and λ_2 can be tuned using cross validation as we will describe next. We also note that before solving (5), we standardized the predictors to have zero mean and unit variance.

C. Results

We randomly split the data into three sets of equal size. The first two sets are used for training and the third set for testing. To select the optimal values for the parameters λ_1 and λ_2 in (5), we use leave-one-out sixfold cross validation applied to the

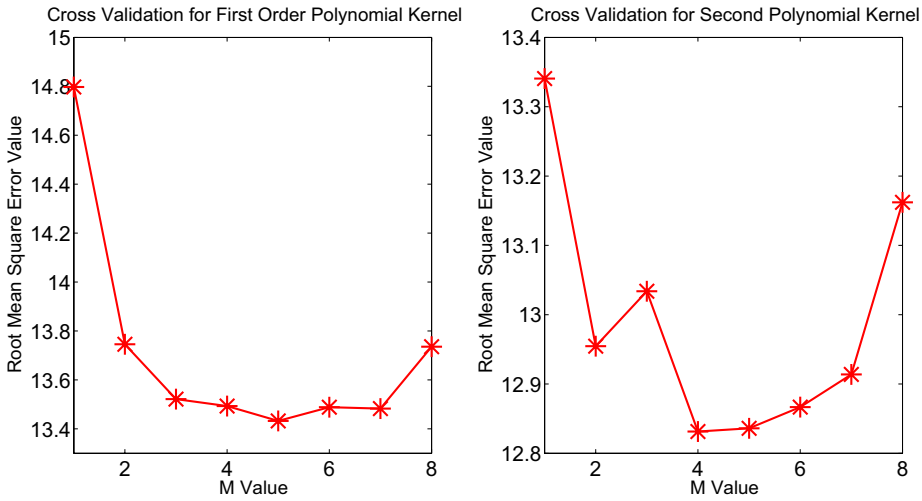


Fig. 2. Performance of the regularized polynomial regression for $d = 1$ and $d = 2$ for different numbers M of time series inputs in the test set.

training set. More specifically, we subdivide the training set into six equal parts. For a discretized collection of (λ_1, λ_2), we train [i.e., solve (5)] on the first five parts and use the sixth part to compute the prediction error [i.e., the first term in (5)]. This is done for each one-sixth of the data left out of the training set and used only for validation and we then compute the average of the corresponding six prediction errors. We use this average value to select the best (λ_1, λ_2). With values of (λ_1, λ_2) now fixed, we resolve (5) on the whole training set and determine which c_j's are (close to) zero. We eliminate the corresponding features and then solve once more a simple regression problem [i.e., a problem with just the first term in (5)] on the whole training set using only the features that were not eliminated. This yields a final vector $\mathbf{c}$ and a prediction

$$\hat{y}_i(t) = \mathbf{c}'\boldsymbol{\phi}(\mathbf{x}_i(t)) \tag{6}$$

for each patient i and time t. The reason for performing a simple regression with the noneliminated features is that (5) naturally biases coefficients c_j toward smaller values.

For performance evaluation, we use two performance metrics. The first is the *root mean square error (RMSE)*, which for patient i is defined as

$$\text{RMSE}_i = \sqrt{\frac{1}{T_i}\sum_{t=t_i^1}^{t_i^{T_i}} (\hat{y}_i(t) - y_i(t))^2} \tag{7}$$

where $t_i^1, \ldots, t_i^{T_i}$ are the time instants at which we make a PTT prediction for patient i.[1] We define RMSE for the whole population of patients as the average per patient RMSE, i.e., $\text{RMSE} = \frac{1}{N_t}\sum_{i=1}^{N_t} \text{RMSE}_i$, where N_t is the number of patients in the test set. We also define σ_{RMSE} to be the standard deviation of the RMSE_i values, which captures the variability of RMSE_i from RMSE.

To capture a notion of "relative" error, we also compute the *normalized root mean square error (NRMSE)* defined for each patient i as

$$\text{NRMSE}_i = \sqrt{\frac{1}{T_i}\sum_{t=t_i^1}^{t_i^{T_i}} [(\hat{y}_i(t) - y_i(t))/y_i(t)]^2}. \tag{8}$$

As with the RMSE, we define the population-wide NRMSE as the average of NRMSE_i over the patients and σ_{NRMSE} as the standard deviation of the NRMSE_i values.

TABLE I
PERFORMANCE OF THE REGULARIZED POLYNOMIAL REGRESSION (TEST SET)

	$d = 1$	$d = 2$
RMSE	11.54	18.69
σ_{RMSE}	4.04	5.37
NRMSE	21.44%	36.58%
σ_{NRMSE}	6.33%	9.82%

Starting with the prediction in (6), we explored several instances of polynomial kernels corresponding to various degrees $d\ (= 1, \ldots, 4)$ and also varied the length M which determines how far back in time we go to define the feature vector. We used again sixfold cross validation over the training set and plot the results in Fig. 2 for kernel parameters $d = 1$ and $d = 2$, respectively, as these were the ones that yielded the best results. From the plots, it follows that for each case there is an optimal value for M, which is $M = 5$ for $d = 1$ and $M = 4$ for $d = 2$. This makes intuitive sense. As we increase M, the performance initially improves since the predictors provide more information but after some value of M, the performance deteriorates due to overfitting.

Finally, and for these two kernels and corresponding optimal M values, we evaluate the performance of the prediction given by (6) on the test set, i.e., the one-third of the data isolated from the training set in the beginning. Table I reports the results. It follows that the linear kernel performs better. We should note that the quadratic kernel achieves slightly less RMSE than the linear kernel on the training set (cf., Fig. 2) but apparently it overfits and the performance is not as good in the test set.

In summary, a key advantage of the regression method we presented in this section is that it does not require any

[1] We note that since we use a history of $M - 1$ measurements for the prediction at time t, we only make predictions starting from the Mth available measurement for each patient.

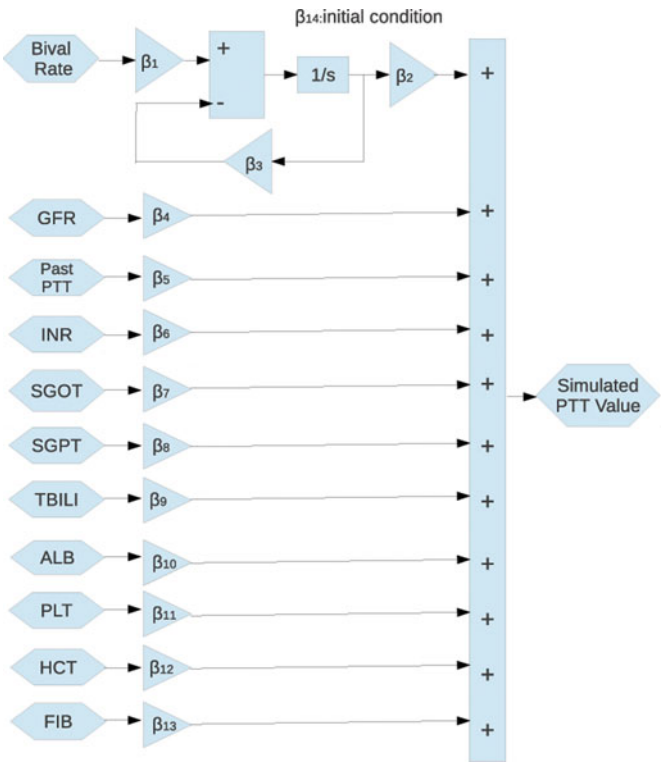


Fig. 3. Single-state linear model includes a gain, $1/\beta_3$, representing the elimination time constant of bivalirudin from the body. The constant β_2 provides for the translation from serum concentration to the site-effect (PTT).

understanding of the mechanism by which bivalirudin affects PTT values for patients. The next section explores whether a specific model can provide better performance.

III. Explicit Dynamic System Model

This section introduces a *multiple input single output* dynamic system model that attempts to explicitly account for the way bivalirudin affects PTT values in patients.

A. Model

The dynamic system model is shown in Fig. 3. It seeks to represent how bivalirudin acts in a single generic patient and uses as inputs the exact same physiological variables used as predictors in Section II. In the absence of established quantitative models to relate most of the input variables to PTT, a simple linear model was assumed. Moreover, most of these variables change slowly over time and were measured infrequently (e.g., once per day) in the data we analyzed. As a result, they were modeled without a dynamic component.

To make things more precise, in this dynamic system, there are 11 inputs which are denoted by $u_i(t), i = 1, \ldots, 11$, and correspond to the physiological variables of Section II. We note that for ease of notation, we will suppress the patient identifier in our generic model description. The input $u_1(t)$, in particular, denotes the bivalirudin infusion rate, and the remaining inputs correspond to the physiological variables 2–11 detailed earlier. These inputs capture the patient's indicators of the renal and liver function. There is only one output—the PTT value—which is denoted by $y(t)$. There is also a single-state variable denoted by $x(t)$. The model has 14 unknown parameters: 13 of which correspond to the various gains and are denoted by $\beta_i, i = 1, \ldots, 13$. The initial condition of the system is the 14th unknown parameter and is denoted by $x(0)$. We will refer to $\mathbf{z} = (\beta_1, \ldots, \beta_{13}, x(0))$ as the parameter vector. In order to be consistent with medical intuition and to avoid overfitting, upper and lower bounds will be introduced for some of the parameters.

Let $\mathbf{u}(t) = (u_1(t), \ldots, u_{11}(t))$. The system dynamics can be expressed as follows:

$$\begin{aligned} \dot{x}(t) &= \mathbf{A}x(t) + \mathbf{B}\mathbf{u}(t) \\ y(t) &= \mathbf{C}x(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \tag{9}$$

where $\mathbf{A} = -\beta_3, \mathbf{B} = [\beta_1 \quad 0 \quad \cdots \quad 0], \mathbf{C} = \beta_2$, and $\mathbf{D} = [0 \ \beta_4 \ \cdots \ \beta_{13}]$. Clearly, this is a *linear time invariant (LTI)* dynamic system. The challenge is that we only have the sampled input, $\mathbf{u}(t)$, and observation values, $y(t)$, at certain times t for each patient. It is therefore needed to translate the continuous-time system dynamics to discrete-time dynamics.

Using a standard conversion from continuous to discrete time dynamics in LTI systems (see, e.g., [14]) we can write

$$x(t+\tau) = e^{\mathbf{A}\tau}x(t) + \int_t^{t+\tau} e^{\mathbf{A}(t+\tau-s)}\mathbf{B}\mathbf{u}(s)ds \tag{10}$$

where in our case $e^{\tau\mathbf{A}} = e^{-\beta_3\tau}$. Assuming $\mathbf{u}(s) = \mathbf{u}(t)$ for $s \in [t, t+\tau]$ and after some algebra, we arrive at the following discrete-time dynamics:

$$\begin{aligned} x(t+\tau) &= e^{-\beta_3\tau}x(t) + \frac{\beta_1}{\beta_3}(1 - e^{-\beta_3\tau})u_1(t) \\ y(t) &= \beta_2 x(t) + \textstyle\sum_{i=4}^{13} \beta_i u_{i-2}(t). \end{aligned} \tag{11}$$

These equations characterize a discrete-time LTI system for which we have a history of the sampled input and output values. Next, we describe how the training set can be used to identify the unknown parameters, namely, the initial condition $x(0)$ and the parameters $\beta_i, i = 1, \ldots, 13$.

B. Parameter Identification

As we did in Section II, we randomly split our dataset of 233 postcardiac surgical ICU patients into a training set corresponding to $2/3$ of the total (155 patients) and a test set corresponding to one-third of the total (78 patients). We will use the former to identify the unknown system parameters and the latter to evaluate the performance of the resulting model.

Let us use a subscript j to denote the model primitives, i.e., the state $x_j(t)$, output $y_j(t)$, and inputs $\mathbf{u}_j(t)$ for each patient $j = 1, \ldots, N$, where N denotes the number of patients in the training set. To distinguish between measurements of $y_j(t)$ and predictions based on the system dynamics [cf. (11)], we use $y_j(t)$ for the former and $\hat{y}_j(t)$ for the latter. Suppose for each patient j, we have T_j measurements at times $t_j^1, \ldots, t_j^{T_j}$, where we adopt the convention $t_j^0 = 0$ for all j. Using the discrete-time system dynamics from (11), we formulate the following nonlinear optimization problem in order to identify the unknown

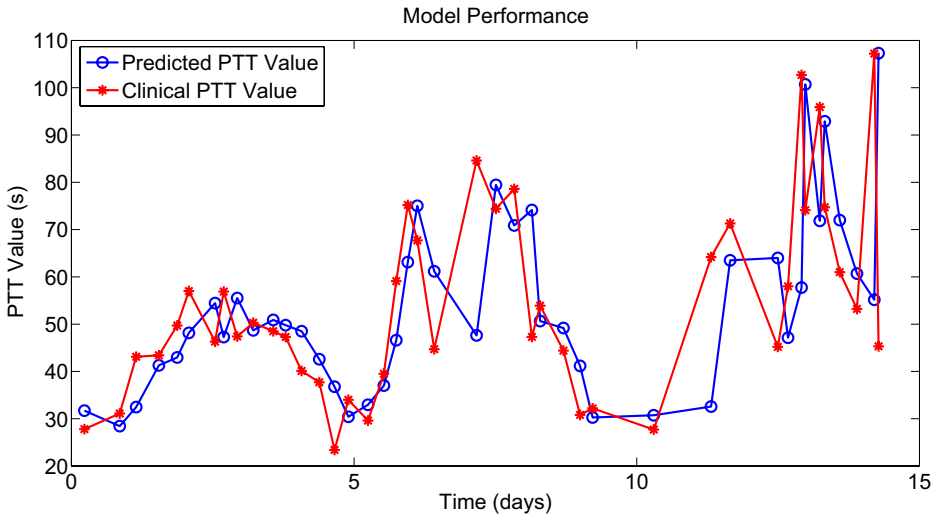


Fig. 4. Illustrating the performance of the dynamic system model of Section III for a particular patient. The blue "o" represent predicted PTT values from our model and the red "$*$" represent actual measured values.

system parameters:

$$\begin{aligned}
\min \sum_{j=1}^{N} \sum_{t=t_j^1}^{t_j^{T_j}} (\hat{y}_j(t) - y_j(t))^2 \\
\text{s.t. } x_j(t_j^n) = e^{-\beta_3 (t_j^n - t_j^{n-1})} x_j(t_j^{n-1}) \\
+ \frac{\beta_1}{\beta_3}(1 - e^{-\beta_3 (t_j^n - t_j^{n-1})}) u_{j,1}(t_j^{n-1}) \\
\forall j = 1, \ldots, N;\ n = 1, \ldots, T_j \\
\hat{y}_j(t_j^n) = \beta_2 x_j(t_j^n) + \sum_{i=4}^{13} \beta_i u_{j,(i-2)}(t_j^n) \\
\forall j = 1, \ldots, N;\ n = 1, \ldots, T_j \\
\beta_m \leq 0,\ m = 1, 2, 3, 5, 7, 8, 9, \\
\beta_n \geq 0,\ n = 4, 6, 10, 11, 12, 13 \qquad (12)
\end{aligned}$$

where the decision variables are $x(0)(= x(t_j^0)$ for all j and the parameters $\beta_i, i = 1, \ldots, 13$. One can easily substitute the equality constraints into the objective function and obtain a simpler nonlinear optimization problem with the only remaining constraints being bounds on some decision variables. Using counterexamples, it can be shown that the objective function obtained in this manner is not convex in the decision variables.

Although a lot of methods exist for nonlinear optimization problems, we used the *Broyden–Fletcher–Goldfarb–Shanno (BFGS) method* [15], which is considered the most effective general purpose quasi-Newton method. The quasi-Newton methods are gradient methods of the form

$$\mathbf{z}^{k+1} = \mathbf{z}^k + \alpha^k \mathbf{d}^k, \qquad \mathbf{d}^k = -\mathbf{D}^k \nabla f(\mathbf{z}^k)$$

where $f(\cdot)$ denotes the objective function, $\mathbf{z}^k$ the decision variables at the kth iteration of the method, α^k is the stepsize at the kth iteration, and $\mathbf{D}^k$ is a positive definite scaling matrix that scales the gradient at the kth iteration. Rather than determining $\mathbf{D}^k$ by computing a Hessian and inverting it, which is computationally expensive, the quasi-Newton methods recursively estimate the inverse of the Hessian by using successive iterates of $\mathbf{z}^k$ and $\nabla f(\mathbf{z}^k)$.

TABLE II
OPTIMAL PARAMETERS VALUES

index	name	value
1	k_dist	57.39
2	k_coag	59.74
3	1/T	60.00
4	kGFR	-2.65×10^{-7}
5	kINR	0.77
6	kPTT	3.13
7	kSGOT	4.66×10^{-7}
8	kSGPT	2.70×10^{-4}
9	kTBILI	0.21
10	kALB	-4.23×10^{-6}
11	kPLT	-7.59×10^{-8}
12	kHCT	-4.36×10^{-7}
13	kFIB	-4.16×10^{-8}
14	$x(0)$	0.61

TABLE III
PERFORMANCE OF THE DYNAMIC SYSTEM MODEL

RMSE	14.65
σ_{RMSE}	6.2
NRMSE	26.42%
σ_{NRMSE}	11.04%

We solved (12) on the training set using BFGS and obtained the optimal solution shown in Table II. To avoid getting stuck at shallow local minima, which is possible in the absence of convexity, we used a multistart approach, namely, we started BFGS from multiple randomly selected initial points and selected the best local minimum we obtained. Using the optimal parameters from Table II, we evaluated the performance of the prediction on the test set and obtained the results shown in Table III. It can be seen that the performance is somewhat worse than the one obtained with polynomial regression. This suggests that the simple model we devised in this section does a decent job of

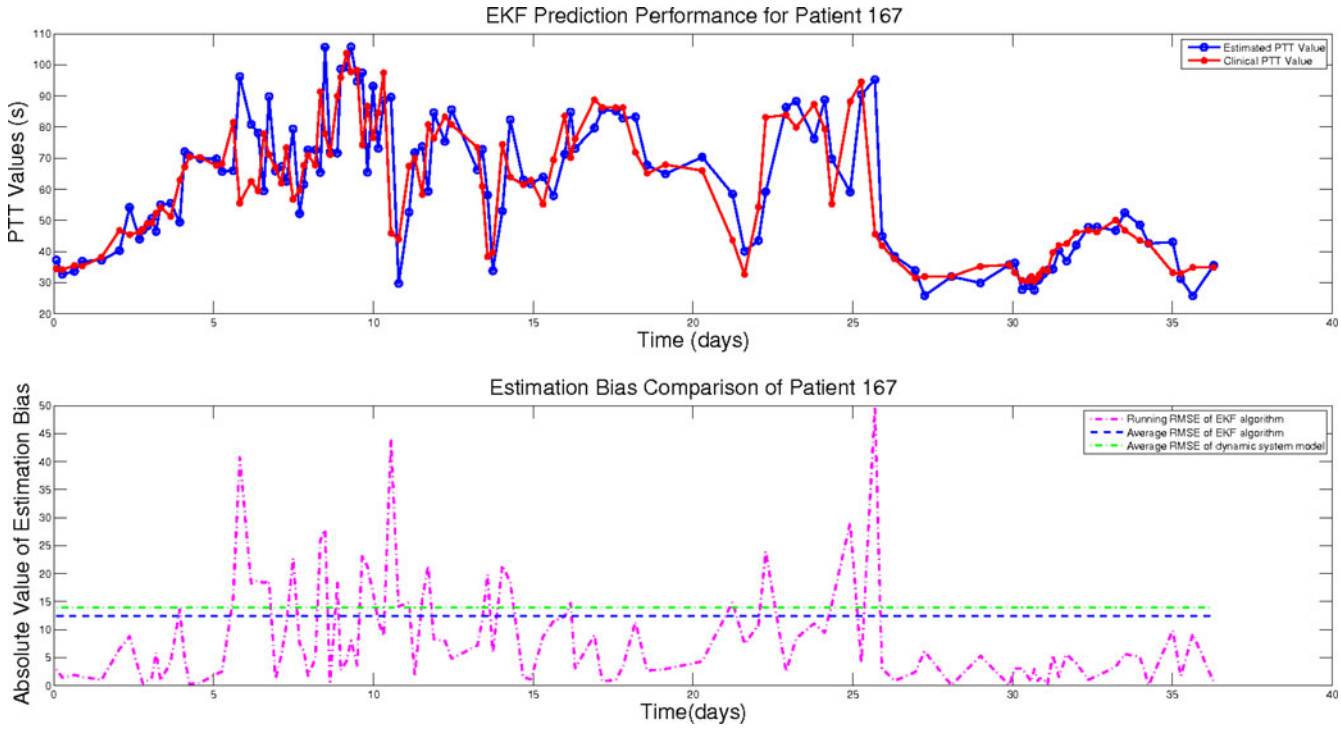


Fig. 5. Illustrating the performance of the EKF algorithm for a particular patient. The blue "o" represent predicted PTT values from our model and the red "$*$" represent actual measured values. The top figure plots estimated and measured PTT and the bottom figure plots the running RMSE at each step. Additionally, the blue-dot line marks the (time) average RMSE of the EKF for this patient while the green-dot line marks the (time) average RMSE of the population-wide dynamic system model applied to this patient.

capturing the key effect of bivalirudin. It uses just a single state and, as it can be inferred from (11), the prediction at time t depends on the inputs $\mathbf{u}(t)$ at t, and the state-input pair, $x(t-\tau)$ and $\mathbf{u}(t-\tau)$ at the previous time instant $t-\tau$. In contrast, the best result with polynomial regression was obtained with a linear kernel but using a history of inputs going back 4 consecutive time instants. One can of course devise a more complicated model that uses more memory, but, we elected not to pursue this direction in the interest of simplicity and taking into account that the potential difference in performance that is to be gained would not be big.

To further illustrate how well the predictor matches the measured values, we plot in Fig. 4 the predicted and actual PTT values over time for a particular randomly selected patient.

We note that the solution shown in Table II is obtained from a training set containing data from many patients and thus corresponds to optimal "population-wide" parameters. The signs of these parameters are consistent with medical intuition. It can be seen that some parameter values are quite small, yet, their presence improves the dynamic model performance; eliminating them will lead to worse performance.

In addition to providing a simple model that explicitly models the effect of bivalirudin, the work in this section has additional benefits. As we will see next, having an explicit model allows us to adapt model parameters to better fit each individual patient.

IV. Adaptive Model: EKF

In this section, we focus on an arbitrary individual patient and seek a method to adapt the parameters of the model we proposed in Section III in order to better fit this particular patient. To that end, we view the model parameters as the "states" of a system and the output $y(t)$ as a nonlinear function of that state. We devise a recursive method to estimate the state. Due to the nonlinearity of $y(t)$, we use the EKF (see e.g., [16]).

Let us denote the state of the system by $\mathbf{z} = (\beta_1, \ldots, \beta_{13}, x(0))$, which are exactly the model parameters we want to estimate. We assume we have measurements of the inputs $\mathbf{u}(t)$ and the PTT values $y(t)$ over many time instants. We will index these time instants by k, with $k=0$ corresponding to $t=0$ and $k=1,2,\ldots,T$ corresponding to the time instants $t^1,\ldots,t^T$ at which we have measurements. (Notice we use the same notation as in Section III but suppress the index j used

1) Initialization:
   a) Set $\hat{\mathbf{z}}_{0|0} = (\beta_1, \ldots, \beta_{13}, x(0))$ using the values obtained by solving the population-wide problem (12), and
   b) set $\mathbf{P}_{0|0} = \mathbf{I}$.
2) Predict:
   a) $\hat{\mathbf{z}}_{k+1|k} = \hat{\mathbf{z}}_{k|k}$;
   b) $\mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_{k|k}\mathbf{A}'$.
3) Update:
   a) $\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{C}'(\mathbf{C}\mathbf{P}_{k+1|k}\mathbf{C}' + \sigma^2)^{-1}$;
   b) $\hat{\mathbf{z}}_{k+1|k+1} = \hat{\mathbf{z}}_{k+1|k} + \mathbf{K}_{k+1}(y_{k+1} - h(\hat{\mathbf{z}}_{k+1|k}))$;
   c) $\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C})\mathbf{P}_{k+1|k}$.

Fig. 6. EKF algorithm for recursively estimating model parameters for an individual patient.

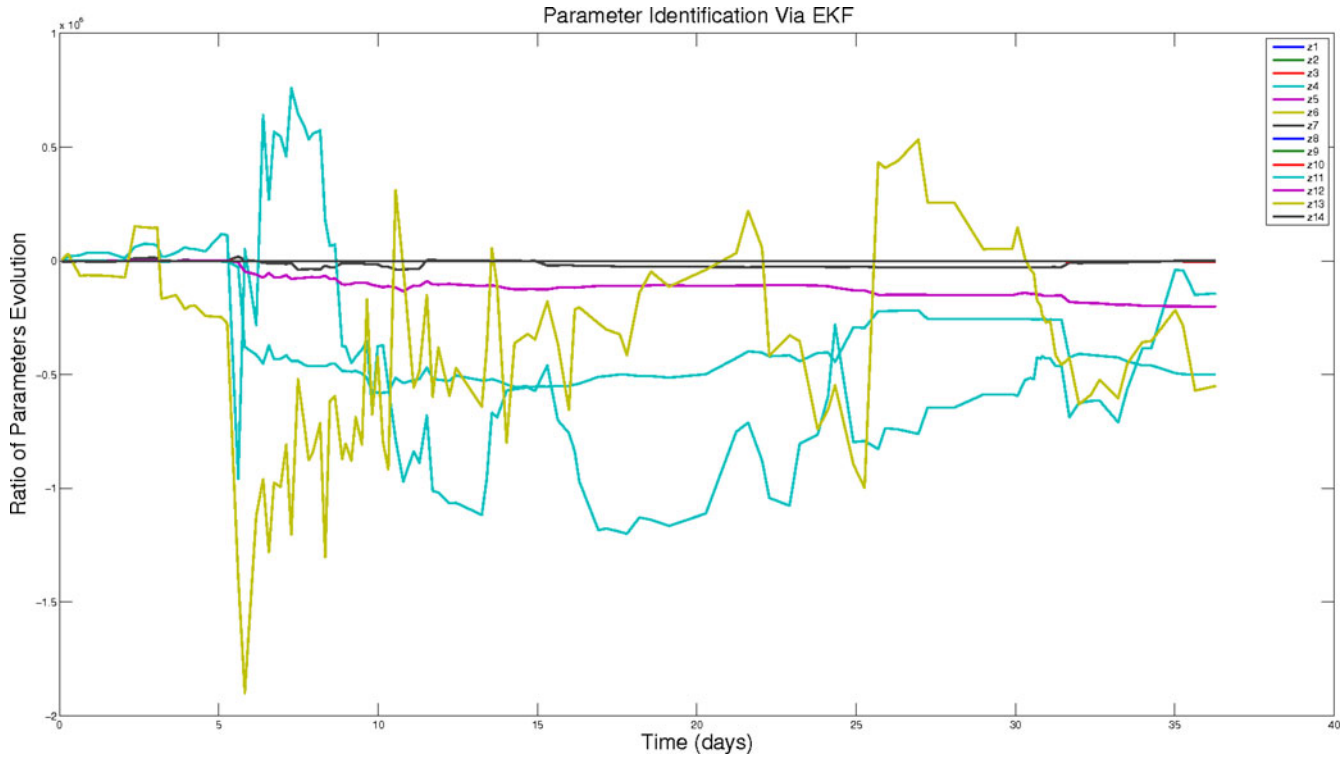


Fig. 7. Illustrating the evolution of model parameter values $\mathbf{z} = (\beta_1, \ldots, \beta_{13}, x(0))$ during the course of EKF algorithm.

there to identify a patient.) We view the state $\mathbf{z}$ has been invariant over time and not affected by noise, while the output y depends on $\mathbf{z}$ but is subject to noise due to both the measurement noise and model error. We can therefore write the (discrete) system dynamics as

$$\begin{aligned} \mathbf{z}_k &= \mathbf{z}_{k-1} \\ y_k &= h(\mathbf{z}_k) + \nu_k. \end{aligned} \tag{13}$$

In the above, $h(\cdot)$ is a known nonlinear function that expresses y_k as function of the parameter vector $\mathbf{z}_k$ and $\mathbf{u}_k, \mathbf{u}_{k-1}, \ldots, \mathbf{u}_0, x_0$ as specified by the dynamics in (11). The random variable ν_k represents the noise and we assume it is i.i.d. over time, zero mean Gaussian, with variance σ^2, that is $\nu_k \sim N(0, \sigma^2)$ for all k.

Let $\mathbf{A} = \mathbf{I}$, and $\mathbf{C} = \nabla h'(\mathbf{z}_k)$. The EKF algorithm is given in Fig. 6, where "hat" denotes the estimate, $\mathbf{P}$ the error covariance, and $\mathbf{K}$ the Kalman gain.

To demonstrate the effect of this algorithm, we randomly selected a patient who has adequate sample data and applied the EKF algorithm of Fig. 6 (using $\sigma^2 = 0.006$). The results are shown in Fig. 5. It is evident that after some initial steps, the algorithm "learns" better values for the model parameters than the ones in the population-wide model and produces better predictions for this particular patient. The model parameter values for the same patient during the course of the EKF algorithm are shown in Fig. 7. It can be seen that they do "adapt" over time from the initial population-wide values to values that are more appropriate for this patient.

To test the performance of the algorithm on a larger set of patients, we selected patients with enough samples; in particular, more than 60 data points. There are 19 out of 78 patients in our test data set with more than 60 data points. We applied the EKF (using again $\sigma^2 = 0.006$) with the optimal population-wide parameter values as our initial point. By doing so, the "warming process" of the EKF algorithm can be reduced significantly and we have a decent model even at the early steps; the latter being important for patient safety. In contrast, if we use an arbitrary initial point, the EKF takes at least 25 steps for the parameter values to stabilize. Moreover, the model error in these early steps becomes quite large which is unacceptable in an eventual use of our system in clinical practice.

TABLE IV
PERFORMANCE OF THE EKF ALGORITHM

RMSE	8.61
σ_{RMSE}	3.28
NRMSE	16.55%
σ_{NRMSE}	6.90%

Table IV reports average results from the EKF algorithm applied to the patients in the reduced test set of 19 patients described in the previous paragraph. The RMSE and NRMSE are computed on a per-patient basis and then averaged over these patients. By comparing Tables III and IV, it follows that the EKF algorithm significantly improves the performance. The NRMSE improves by 37% (9.87% in 26.42%) while the variance is reduced by 38%, which is significant. This confirms the significant individual patient variability in response to bivalirudin which has been only empirically observed.

Furthermore, the EKF algorithm leads to improved performance even in comparison to the model-free method of Section II which, as we have discussed, uses a longer history

of past measurements. In particular, NRMSE improves by 23% while the variance stays about the same.

V. Conclusion

We have developed two main approaches to predict the effect of bivalirudin in cardiac surgical patients. The first approach is *model free* and leverages regularized regression. We find that a linear kernel performs best and that the corresponding set of predictors uses a collection of physiological variables characterizing bivalirudin infusion rate, several coagulation indicators, and indicators of the renal and liver function sampled over a set of four time instances before the time at which a PTT prediction is sought. Namely, this model-free method uses a history of four prior feature vectors to make a PTT prediction.

Our second approach is *model based* and constructs a specific model that captures how bivalirudin affects PTT values. The model uses a shorter history of prior feature vectors than the model-free approach in order to arrive at a prediction. The model parameter identification is done by solving a nonlinear optimization problem over a training set. The model-based approach produces a somewhat worse performance than the model free one, which is understandable given the shorter history used.

The model-based approach, however, enables the development of an adaptive EKF algorithm that can adapt the model parameters to individual patients. This approach produces the best average performance and a variance which is almost identical to the model-free method. Compared to the population-wide optimal dynamic model, the NRMSE is reduced by 37% and the standard deviation of the per-patient NRMSE is reduced by 38%. This shows that patient-specific models have significant advantages over population-wide models.

The mathematical models and prediction approaches described in this paper may provide a better reference to guide the optimal therapy in cardiac patients in need of bivalirudin. In addition, such mathematical ideas and methods may be useful to test medication dosing strategies and may provide a mathematical mechanism for development and testing of nomograms.

References

[1] J. Bittl, B. Chaitman, F. Feit, W. Kimball, and E. Topol, "Bivalirudin versus heparin during coronary angioplasty for unstable or postinfarction angina: Final report reanalysis of the bivalirudin angioplasty study," *Amer. Heart J.*, vol. 142, no. 6, pp. 952–959, 2001.

[2] A. Lincoff, J. Bittl, R. Harrington, F. Feit, N. Kleiman, J. Jackman, I. Sarembock, D. Cohen, D. Spriggs, R. Ebrahimi, G. Keren, J. Carr, E. A. Cohen, A. Betriu, W. Desmet, D. J. Kereiakes, W. Rutsch, R. G. Wilcox, P. J. de Feyter, A. Vahanian, and E. J. Topol, "Bivalirudin and provisional glycoprotein IIb/IIIa blockade compared with heparin and planned glycoprotein IIb/IIIa blockade during percutaneous coronary intervention," *JAMA: J. Amer. Med. Assoc.*, vol. 289, no. 7, pp. 853–863, 2003.

[3] G. W. Stone, B. McLaurin, D. Cox, M. Bertrand, A. Lincoff, J. Moses, H. White, S. Pocock, J. Ware, F. Feit, A. Colombo, P. E. Aylward, A. R. Cequier, H. Darius, W. Desmet, R. Ebrahimi, M. Hamon, L. H. Rasmussen, H. J. Rupprecht, J. Hoekstra, R. Mehran, and E. M. Ohman, "Bivalirudin for patients with acute coronary syndromes," *New Engl. J. Med.*, vol. 355, no. 21, pp. 2203–2216, 2006.

[4] T. H. Kiser, A. M. Mann, T. C. Trujillo, and K. L. Hassell, "Evaluation of empiric versus nomogram-based direct thrombin inhibitor management in patients with suspected heparin-induced thrombocytopenia," *Amer. J. Hematol.*, vol. 86, no. 3, pp. 267–272, 2011.

[5] T. Edrich, G. Frendl, J. Rawn, and I. C. Paschalidis, "Modeling the effects of bivalirudin in cardiac surgical patients," in *Proc. IEEE 33rd Annu. Int. Conf. Eng. Med. Biol. Soc.*, Boston, MA, USA, Aug. 30–Sep. 3, 2011, pp. 120–123.

[6] T. Edrich, G. Frendl, and I. C. Paschalidis, "Tachyphylaxis in post-cardiac surgical patients receiving bivalirudin—A retrospective dynamic study using a PKPD model," *Crit. Care Med.*, vol. 39, no. 12, p. 198, Dec. 2011.

[7] E. J. Hannan and M. Deistler, *The Statistical Theory of Linear Systems*. New York, NY, USA: Wiley, 1988.

[8] P. Ioannou and B. Fidan, *Adaptive Control Tutorial*. Philadelphia, PA, USA: SIAM, 2006.

[9] L. Ljung, *System Identification: Theory for the User*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1999.

[10] T. Evgeniou, M. Pontil, and T. Poggio, "Regularization networks and support vector machines," *Adv. Comput. Math.*, vol. 13, no. 1, pp. 1–50, 2000.

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. ed. New York, NY, USA: Springer-Verlag, 2009.

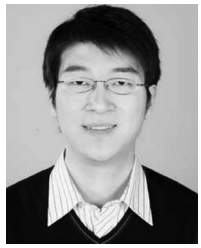
[12] T. M. Company. (2011). "Angiomax (bivalirudin) Prescribing Information," [Online]. Available: http://www.themedicinescompany.com/page/angiomax-us/1

[13] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *J. Roy. Statist. Soc.: Series B (Statist. Methodol.)*, vol. 67, no. 2, pp. 301–320, 2005.

[14] B. Friedland, *Control System Design: An Introduction to State-Space Methods*. New York, NY, USA: Dover, 2005.

[15] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA, USA: Athena Scientific, 1999.

[16] G. Welch and G. Bishop, "An introduction to the Kalman filter," Dept. of Computer Science, Univ. of North Carolina, Chapel Hill, NC, USA, Tech. Rep. TR 95-041, 2006.

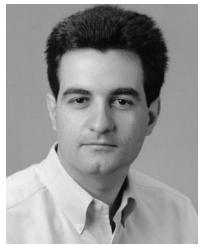


Qi Zhao received the B.S. degree in electrical engineering and automation from the University of Science and Technology, Beijing, China, in 2010, and the M.S. degree in electrical engineering from Boston University, MA, USA, in 2012, where he is currently working toward the Ph.D. degree in the Division of Systems Engineering.

His current research interests include systems and control, optimization, operations research, computational biology, and bioinformatics.

Thomas Edrich (M'11) received the B.S. degree in electrical engineering from the University of Colorado at Boulder, Boulder, CO, USA, in 1992, and the M.D. and Ph.D. degrees from the Ludwig-Maximilians University of Munich, Munich, Germany, in 1996 and 1999, respectively, where he specialized in anesthesia and critical care medicine.

He is currently an Assistant Professor of Anesthesia at the Brigham and Women's Hospital, Harvard Medical School, Boston, MA, USA. His research interests include PKPD modeling, medical control systems, as well as, critical care ultrasound applications.



Ioannis Ch. Paschalidis (M'96–SM'06) received the M.S. and Ph.D. degrees from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1993 and 1996, respectively, both in electrical engineering and computer science.

In September 1996, he joined Boston University, where he is currently a Professor and Distinguished Faculty Fellow with appointments in the Department of Electrical and Computer Engineering and the Division of Systems Engineering. He is a Co-Director of the Center for Information and Systems Engineering. He has held visiting appointments with MIT, and the Columbia University Business School. His current research interests include the fields of systems and control, networking, applied probability, optimization, operations research, computational biology, and medical informatics.