34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

# A Coupling Method for a Cardiovascular Simulation Model which includes the Kalman Filter

Yuki Hasegawa, Takao Shimayoshi, Akira Amano, Tetsuya Matsuda

*Abstract*— **Multi-scale models of the cardiovascular system provide new insight that was unavailable with in vivo and in vitro experiments. For the cardiovascular system, multi-scale simulations provide a valuable perspective in analyzing the interaction of three phenomenons occurring at different spatial scales: circulatory hemodynamics, ventricular structural dynamics, and myocardial excitation-contraction. In order to simulate these interactions, multiscale cardiovascular simulation systems couple models that simulate different phenomena. However, coupling methods require a significant amount of calculation, since a system of non-linear equations must be solved for each timestep. Therefore, we proposed a coupling method which decreases the amount of calculation by using the Kalman filter. In our method, the Kalman filter calculates approximations for the solution to the system of non-linear equations at each timestep. The approximations are then used as initial values for solving the system of non-linear equations. The proposed method decreases the number of iterations required by 94.0% compared to the conventional strong coupling method. When compared with a smoothing spline predictor, the proposed method required 49.4% fewer iterations.**

## I. Introduction

In silico experiments provide a third mode of analysis, when in vivo and in vitro experiments are not adequate for the investigation of complicated systems. In these models, individual models that describe a certain phenomenon in the cardiovascular system such as excitation-contraction of myocardial cells or the hemodynamics or the circulatory system are solved simultaneously by using coupling methods. Coupling models of different scales brings us closer to describing the complex cardiovascular system. However, coupling models of different scales introduces stability issues and requires extensive calculation [1], [2]. For this reason, many multi-scale models limit the number of models in its composition or simplify these models.

Previously, we proposed a coupling method that was stable and faster than conventional methods [1]. In this method, before a system of non-linear equations was solved at each timestep, a predictor was used to calculate an initial approximation for the root-finding algorithm. Because the approximation was close to the solution, the root-finding algorithm required only a few iterations to solve the equations. For the approximation, the errors contained in the solutions of past timesteps were first removed by the smoothing spline, and then from the newly acquired estimates an extrapolator calculated the initial approximation. This allowed for more stable and accurate predictions than methods that did not perform any error removal.

The study mentioned above showed that the proposed coupling method completed simulations faster than conventional methods. However, the study manually selected the smoothing parameter, which required more simulations, and did not compare the smoothing spline with other smoothing methods. In this paper, we introduce a new coupling method that uses the Kalman filter for prediction. With the introduction of this new prediction method, we will be able to compare its performance with the smoothing spline. Additionally, methods for automatic parameter selection are used for both the Kalman filter and the smoothing spline.

Y. Hasegawa and T. Matsuda are with Graduate School of Informatics, Kyoto Univerisity, Kyoto, 606-8501, Japan (phone:+81-75-753-3373; fax:+81-75-753-3375; e-mail:yhasegawa@sys.i.kyoto-u.ac.jp)

T. Shimayoshi is with ASTEM RI, Kyoto, 600-8813, Japan

A. Amano is with Department of Bioinformatics, Ritsumeikan University, Kyoto, 611-0031, Japan

## II. Cardiovascular Simulation Model

In this paper, we used the cardiovascular simulation model introduced in [1] and [2], which is composed of three parts: the myocardial excitation-contraction model, the left ventricular structural dynamic model, and the circulatory model. However, the equations to stabilize the myocardial excitation-contraction model, proposed in [2], was removed in the current study.

### A. Myocardial Excitation-Contraction Model

The Kyoto model was used to describe myocardial cell physiology and contraction dynamics[3]. This model uses ODEs to describe in detail the excitation-contraction coupling mechanism. Myocardial contraction force calculated by this model is dependent on sarcomere length. Therefore, sarcomere length $l$ is the input variable for the function to calculate contraction force: $f_b(l)$.

### B. Left Ventricular Structural Dynamic Model

The material properties and structural dynamics of the left ventricle were modeled by the left ventricular finite element model (LVFEM)[2]. The LVFEM calculates the equilibrium,

$$0 = H(L, F_b, P_{lv}, V_{lv}), \tag{1}$$

where $N$ is the number of elementsC$F_b$ and $L$ are N-dimensional vectors of contraction force and length, repectively, for each element. $P_{lv}$ and $V_{lv}$ are ventricular pressure and volume, respectively.

### C. Circulatory hemodynamics

The blood flow within the circulatory system is determined by the hemodynamic factors. The circulatory hemodynamics, such as arterial resistance and capacitance, pre-load and afterload, was modeled by the general 3-element windkessel model [1].

978-1-4577-1787-1/12/$26.00 ©2012 IEEE

In the model, change in left ventircular volume $V_{lv}$ depends on left ventricular pressure $P_{lv}$,

$$\frac{dV_{lv}}{dt} = Q_{in}(P_{lv}) - Q_{out}(P_{lv}), \tag{2}$$

where $Q_{in}$ and $Q_{out}$ is the blood flow from the vein into the left ventricle and blood flow from the left ventricle to the artery, respectively.

## III. Strong Coupling Method

The strong coupling method simultaneously solves the LVFEM, myocardial excitation-contraction model and circulatory model at each timestep. The three equations can be combined into a system of non-linear equations:

$$f(\mathbf{x}) = \mathbf{0} \quad \mathbf{x} \equiv [L_1, \cdots, L_N], \tag{3}$$

where $\mathbf{x}$ is a vector containing the lengths used to calculate myocardial contraction force $f_b$ for each element.

In practice an approximation of the solution to Eq. 3 is obtained by satisfying the inequality $f(\mathbf{x}) < \varepsilon$, where $\varepsilon$ is the tolerance.

In the conventional strong coupling method, the initial values used at the beginning of the root-finding algorithm for timestep $t_n$ is the solution obtained for the last timestep $\mathbf{x}^{(t_{n-1})}$.

## IV. Prediction Methods

The prediction method calculates an approximation to the solution at the beginning of each timestep by using values of past timesteps. These values contain error, because of the convergence tolerance for Eq. 3. Estimates of past values are obtained and used for the prediction to increase the accuracy of the approximation. If the approximation is close to the solution, then the algorithm will find the solution in fewer iterations.

In this paper, two prediction methods are compared. One of the prediction methods that uses the smoothing spline was proposed previously in [1]. The prediction method we are proposing in this paper uses the Kalman filter. Both of the methods will be explained in this section. Methods for automatic parameter selection for the smoothing spline and Kalman filter will be introduced, as well.

### A. Smoothing Spline

*1) Smoothing and prediction method:* The smoothing spline is a non-parametric method of removing noise from the values obtained in past timesteps. The estimations are obtained by finding the minimizer of the following function:

$$\sum_{i=n-m}^{n} (L^{(t_i)} - \hat{L}^{(t_i)})^2 + \lambda \int_{t_{n-m}}^{t_n} \hat{L}''^2 \, dx, \tag{4}$$

where $\lambda$ is the smoothing parameter, $m$ is the number of past timesteps to be estimated, and $\hat{L}$ is the estimated value.

To calculate an approximation to the solution from the estimated values, we used the third-order extrapolator:

$$\begin{aligned} L_0^{(t_n)} = {} & 4 \cdot \hat{L}^{(t_{n-1})} - 6 \cdot \hat{L}^{(t_{n-2})} \\ & + 4 \cdot \hat{L}^{(t_{n-3})} - \hat{L}^{(t_{n-4})}, \end{aligned} \tag{5}$$

where $L_0^{(t_n)}$ is the approximation of the solution for the current step.

*2) Generalized Cross Validation:* Generalized cross-validation (GCV) is used to select the smoothing parameter $\lambda$ in Eq. 4. GCV calculates a score for a value of $\lambda$ based on the mean square error of the estimated values. The minimizer of the GCV score is the optimal $\lambda$ for the given data set.

### B. Kalman Filter

*1) Dynamic system model and filtering method:* The Kalman filter makes predictions based on a predefined linear dynamic system model. The system model has the following form:

$$\hat{\mathbf{x}}^{(t_n)} = \mathbf{A}\hat{\mathbf{x}}^{(t_{n-1})} + \mathbf{w}, \tag{6}$$

where $\mathbf{A}$ is the state-transition matrix for state $\hat{\mathbf{x}}$ and $\mathbf{w}$ is process noise.

The relation between the true state that is described by the system model $\hat{\mathbf{x}}$ and observed state $\mathbf{z}$ is defined as

$$\mathbf{z}^{(t_n)} = \mathbf{H}\hat{\mathbf{x}}^{(t_n)} + \mathbf{v}, \tag{7}$$

where $\mathbf{H}$ is the observation matrix and $\mathbf{v}$ is the observation noise. Process noise $\mathbf{w}$ and observation noise $\mathbf{v}$ are assumed to be white:

$$\begin{aligned} p(\mathbf{w}) &\sim \mathcal{N}(0, \mathbf{Q}) \\ p(\mathbf{v}) &\sim \mathcal{N}(0, \mathbf{R}), \end{aligned} \tag{8}$$

where $\mathbf{Q}$ and $\mathbf{R}$ are the covariance matrices for the process and observation noise, respectively.

The Kalman filter calculates the a priori $\hat{\mathbf{x}}^-$ and the a posteriori $\mathbf{x}^+$ estimates based on the linear dynamic system model and observation model. The a priori estimate $\hat{\mathbf{x}}^-$ is obtained from

$$\hat{\mathbf{x}}^{-(t_n)} = \mathbf{A}\mathbf{x}^{+(t_{n-1})} \tag{9}$$

where $\mathbf{x}^{+(t_{n-1})}$ is the a posteriori estimate from the last timestep $(t_{n-1})$.

In the a posteriori step, the filter compares the estimated observed state $\mathbf{H}\hat{\mathbf{x}}^{-(t_n)}$ and the actual observed state $\mathbf{z}^{(t_n)}$,

$$\tilde{\mathbf{y}}^{(t_n)} = \mathbf{z}^{(t_n)} - \mathbf{H}\hat{\mathbf{x}}^{-(t_n)}, \tag{10}$$

where $\tilde{\mathbf{y}}^{(t_n)}$ is the innovation. The residual is multiplied with the Kalman gain $\mathbf{K}^{(t_n)}$ to correct the a prior estiamte $\hat{\mathbf{x}}^{-(t_n)}$,

$$\hat{\mathbf{x}}^{+(t_n)} = \hat{\mathbf{x}}^{-(t_n)} + \mathbf{K}^{(t_n)}\tilde{\mathbf{y}}^{(t_n)}, \tag{11}$$

where, $\hat{\mathbf{x}}^{+(t_n)}$ is the a posteri estimate.

The Kalman gain is calculated by the following equations:

$$\mathbf{P}^{-(t_n)} = \mathbf{A}\mathbf{P}^{+(t_{n-1})}\mathbf{A}^T + \mathbf{Q} \tag{12}$$

$$\mathbf{P}^{+(t_n)} = (I - \mathbf{K}^{(t_n)}\mathbf{H})\mathbf{P}^{-(t_n)} \tag{13}$$

$$\mathbf{K}^{(t_n)} = \frac{\mathbf{P}^{-(t_n)}\mathbf{H}^T}{\mathbf{H}\mathbf{P}^{-(t_n)}\mathbf{H}^T + \mathbf{R}}, \tag{14}$$

where, $\mathbf{P}^{-(t_n)}$ and $\mathbf{P}^{+(t_n)}$ are the covariances for the a prior and a posterior estimates.

For our simulation model, we defined the non-zero components of the transition matrix $\mathbf{A}$ and the state $\hat{\mathbf{x}}$ as

$$\begin{aligned}\hat{\mathbf{x}} &= [L_1, \dot{L}_1, \cdots, L_N, \dot{L}_N]^T \\ \mathbf{A}_{(i,i)} &= 1, \quad i = 1, \cdots, 2N \\ \mathbf{A}_{(2i,2i+1)} &= \Delta t, \quad i = 1, \cdots, N \end{aligned} \tag{15}$$

where $(x, y)$ represent the component in the $x$ th row and $y$ th column and $N$ is the number of elements in the LVFEM. The non-zero componenets of the observation matrix $\mathbf{H}$ were defined as

$$\mathbf{H}_{(i,2i+1)} = 1, \quad i = 0, 1 \cdots, N-1. \tag{16}$$

*2) Using the normalized autocorrelation for parameter selection:* The parameters for the Kalman filter are the covariances for the process and observation noise, $\sigma_w$ and $\sigma_v$, which are used to calculate the matrices $\mathbf{Q}$ and $\mathbf{R}$, respectively. The parameters are optimal if the sequence of innovation calculated in Eq. 10 is white. To test the whiteness of this sequence, the normalized autocorrelation of the innovation for each timestep is calculated. If the autocorrelations are within the 95% confidence limits, then the innovation sequence is white, and the parameters are optimal for the data set [4].

## V. Experiment

### A. Methods

In order to determine the parameters for the smoothing spline and Kalman filter, sample data were obtained by a simulation performed with the conventional method in advance. The smoothing parameter $\lambda$ and the number of past variables used for the estimation $m$ was determined by GCV. The parameters for the Kalman filter were determined by testing if the normalized autocorrelations fall within the 95% confidence limits. The parameters with the most number of autocorrelations falling within the confidence interval was calculated by solving an optimization problem. The parameters obtained by the above mentioned methods are listed in table 1.

A 4 element LVFEM was used for the experiments. Timestep length was set to 0.1 ms. Simulations were performed for 4000 timesteps, which is one cardiac cycle for the Kyoto model. At each timestep, the non-linear system of equations were solved by Broyden's method. The convergence criteria was set to $10^{-4}\mu m$. Simulations were executed on a cluster with 2 Quad-Core AMD Opteron ProcessorsD DynaBios[5], a simulation platform which includes simBio [6] to calculate the myocardial cell model and commercially available finite element method solver MSC Marc, is the software used for the simulations.

### B. Results

Table 2 shows the results from the experiment. Number of iterations, jacobian evaluations, and function evaluations are the respective values needed to solve Eq. 3 by Broyden's method summed over 4000 timesteps. Function evaluations refers to the evaluation of Eq. 3. The Kalman filter and smoothing spline method reduced iterations by 88.1% and 94.0%, respectively, compared with the conventional method. Wall clock time is the real time from the beginning of the simulation to completion. The Kalman filter and smoothing spline reduced wall clock time by, 56.2% and 59.4%, respectively, compared with the conventional method. RMSPE1 is the root mean square precentage error (RMSPE) between the initial value (or predicted value for the prediction methods) and the converged value. RMSPE2 is the RMSPE between the converged values from the conventional method and the prediciton methods.

TABLE 1
PARAMETERS

| Kalman Filter | | Smoothing Spline | |
|---|---|---|---|
| $\sigma_w$ | $\sigma_v$ | $\lambda$ | m |
| 100000 | 1050 | 20.2428 | 57 |

TABLE 2
RESULTS

| | Conventional | Kalman Filter | Smoothing Spline |
|---|---|---|---|
| #iterations | 2782 | 330 | 167 |
| #jacobian evaluations | 1790 | 263 | 126 |
| #function evaluations | 13942 | 5382 | 4671 |
| wall clock time (min) | 71.0 | 31.1 | 28.8 |
| rmspe1 (%) | 4.88E-3 | 5.52E-4 | 6.69E-4 |
| rmspe2 (%) | - | 1.26E-2 | 1.56E-2 |

## VI. Discussion

### A. RMSPE1 and the number of iterations

The proposed prediction methods were able to reduce the number of iterations needed to solve the non-linear equations. The precition methods also reduced the RMSPE between the initial value and the converged value (RMSPE1). The converged value used here is not the real solution. Therefore, RMSPE1 does not represent actual error. However, the correlation between RMSPE1 and the number of iterations was 0.99. The high correlation between RMSPE1 and the number of iterations suggests that the prediction methods reduced the number of iterations by predicting an accurate approximation to the solution.

### B. Relationship of wall clock time and the number of iterations

Although the reduction in the number of iterations was about ten-fold, the wall clock time was only reduced by about half. This is because wall clock time is related directly to the number of function evaluations and not the number of iterations. Atleast 1 function evaluation is required for each timestep, and 4 function evaluations are needed for each jacobian evaluation. The number of function evaluations can be described by the following equation:

$$U_{function} = T + N \cdot U_{jacobian} + U_{iterations}, \tag{17}$$

where $U_{function}, U_{jacobian}, U_{iterations}$ represent the number of function evaluations, jacobian evaluations, and iterations, respectively. $T$ is the number of total timesteps and $N$ is the number of elements of the LVFEM.

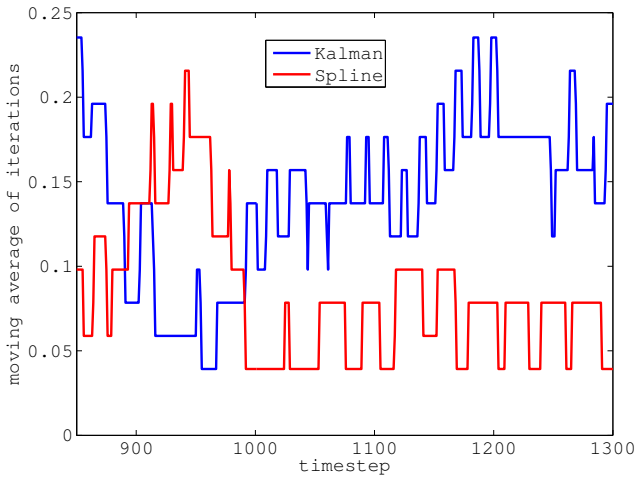


Fig. 1. Moving average of the iterations at each timestep

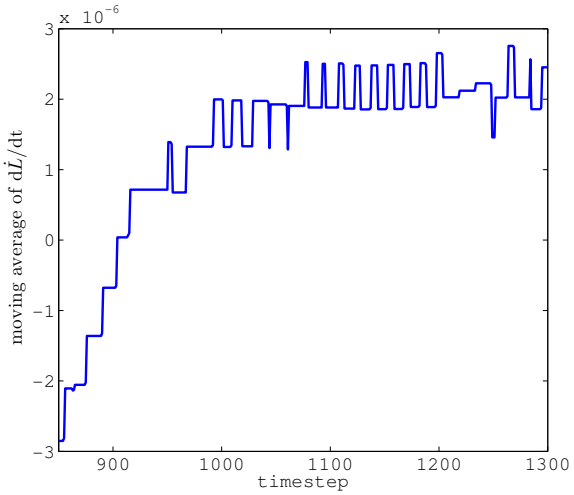


Fig. 2. Moving average of the difference of estimated velocity $\dot{L}$ from the Kalman filter

The correlation between the number of function evaluations and wall clock time was 9.93. Therefore, wall clock time is proportional to the number of function evaluations and is related to the number of iterations through Eq. 17.

### C. Performance difference between the Kalman filter and the smoothing spline

We will speculate on the difference in performance between the Kalman filter and the smoothing spline.

The Kalman filter uses a second-order predictor while a third-order extrapolator was combined with the smoothing spline. Additional experiments with first to fourth order extrapolators combined with the smoothing spline did not show any significant differences ($\pm 5$ deviations in iterations from the third-order experiment). Therefore, the difference in order between the prediction methods did not contribute to the difference in performance.

In order to analyze the difference in performances, we looked at the difference in the number of iterations for each timestep. We saw that the number of iterations differed significantly between timesteps 850 to 1300. The moving average (window frame 51 timesteps) of iterations during this interval is shown in fig. 1. Between timesteps 900 and 1000, the Kalman filter had fewer iterations than the smoothing spline. However, from timesteps 1000 to 1200 the number of iterations for the Kalman filter was higher than that of the smoothing spline.

Fig. 2 shows the moving averages (window frame 51 timesteps) for $d\dot{L}/dt$, calculated by the forward difference of the estimated velocity $\dot{L}$. Correlation between acceleration (fig. 2) and iterations (fig. 1) can be observed. This may be because the filter changes its estimated velocity only when the current prediction is incorrect. Therefore, the Kalman filter makes incorrect predictions while sarcomere length is accelerating. On the other hand, the smoothing spline uses best-fit and is independent of acceleration. This may be the reason for the smooothings spline performing better than the Kalman filter.

## VII. Conclusion

A new coupling method for a multi-scale cardiovascular simulation system was proposed. The proposed method used the Kalman filter to predict approximations to the solutions. The new proposed method did not perform better than the previously proposed method, which uses the smoothing spline.

A major problem with the Kalman filter is that it only corrects its estimated velocity when an error in prediction is made. Therefore, if cell contraction is undergoing acceleration, the Kalman filter is guaranteed to make errors in the prediction. Remedying this problem may increase the performance of the Kalman filter.

Also, we did not test the robustness of these methods and only performed the experiments under a single set of conditions. Changing the parameters of the models may affect the performance of the predictors. If experiments are done with different parameters, the predictors should function under those conditions, as well. Therefore, the robustness of these predictors needs to be investigated further.

## References

[1] Y. Hasegawa, T. Shimayoshi, A. Amano, and T. Matsuda, "A study on prediction methods for a cardiovascular strong-coupling simulation," pp. 137–140, 2011.

[2] A. Amano, Y. Takada, J. Lu, and et al., "An approximation model of myocardial crossbridge for weak coupling calculation of left ventricle model and circulation model," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, pp. 957–960, 2008.

[3] M. Kuzumoto, A. Takeuchi, H. Nakai, and et al., "Simulation analysis of intracellular $Na^+$ and $Cl^-$ homeostasis during $\beta$1-adrenergic stimulation of cardiac myocyte," *Prog. Biophys. Mol. Biol.y*, vol. 96, no. 1-3, pp. 171–186, 2008.

[4] R. Mehra, "On the identification of variances and adaptive kalman filtering," *Automatic Control, IEEE Transactions on*, vol. 15, no. 2, pp. 175–184, 1970.

[5] T. Shimayoshi, K. Hori, J. Lu, and et al., "Dynabios: A platform for cell/biodynamics simulators," *IEEJ T.*, vol. 127, no. 11, pp. 1928–1936, 2007.

[6] N. Sarai, S. Matsuoka, and A. Noma, "simbio: a java package for the development of detailed cell models," *Prog. Biophys. Mol. Biol.*, vol. 90, no. 1-3, pp. 360–377, 2006.