POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

HOMEWORK REPORT

Homework 1

SCIENTIFIC COMPUTING TOOLS FOR ADVANCED MATHEMATICAL MODELLING

Authors: FEDERICA BOTTA, SIMONE COLOMBARA AND MICHELE DI SABATO

Academic year: 2021-2022

---

1. Mathematical formulation of the problem

The framework of this project is the following: the normal diffusion of the differential potential in the heart is hindered by a re-entry of the signal, possibly caused by scar tissue in specific areas of the heart. This problem could cause an arrhythmia, which could be fatal. In order to restore the normal diffusion of the potential through the affected region, an Anti-Tachycardia Pacing (ATP) device is inserted, its purpose is to deliver an impulse to avoid the re-entry.
We are given three ECG signals observed for 450 ms of three different patients and a numerical solver to compute an approximation of an ECG. The goal of this project is to calibrate the Rogers-McCulloch model for the ionic current and the ATP strategy:

- ν_2: Rogers-McCulloch model parameter, which ranges in (0.0116, 0.0124)
- t_s: starting time of the shock, which ranges in (450 ms, 525 ms)
- Δt_s: duration of the shock. To maximize the duration of the battery it's best to keep this parameter below 10 milliseconds and possibly as small as possible.

Moreover, these parameters need to be patient-specific.

To calibrate ν_2, we have to compare the exact solutions with our numerical solutions for each patient, considering the problem without the intervention of the ATP device (until 450 ms). Instead, the "best" t_s and Δt_s are the ones that lead to an ECG equal to zero in the tracking window (600 ms, 800 ms).
The underlying model is the monodomain problem coupled with the Rogers-McCulloch ionic model which gives an estimation of the total ionic current (I_{ion}) and its gradient (g), being:

$$\begin{cases} I_{ion}(U_t, W_t) = \nu_0 U_t (1 - \frac{U_t}{v_{th}})(1 - \frac{U_t}{v_{pk}}) + \nu_1 U_t W_t, \\ g(U_t, W_t) = \nu_2 (\frac{U_t}{v_{pk}} - \nu_3 W_t). \end{cases} \tag{1}$$

Where U_t is the transmembrane potential and W_t is the unique gating variable. Note that the parameter ν_2 is used in this model.
We define the signals output during the time evolution (pseudo or numerical ECG) as $u(t)$ (without the presence of ATP device), having the exact signals of the three patients $u_{ex}(t)$ the problem for finding ν_2 becomes:

$$\text{find } \nu_2^* = \arg\min_{\nu_2} \|u(t) - u_{ex}(t)\|_{MSE} \quad \forall t \in (0, 450) \quad ms. \tag{2}$$

Instead, to find t_s and Δt_s we use the ν_2 value found from equation (2) for each subject and, this time, we also take into account the effect of the impulse from the ATP device. The mathematical formulation of the problem

is[1]:

$$\text{find } (t_s^*, \Delta t_s^*) := \underset{(t_s, \Delta t_s)}{\arg\min}(\|u(t) - 0\|_{MSE} + \lambda \Delta t_s) \qquad \forall t \in (600, 800) ms. \tag{3}$$

The first term aims at reaching a signal equal to 0 on the entire the tracking window and the term involving λ is used to penalize large impulse durations, which undermine the device's battery.

2. Methods

2.1. Algorithm for ν_2

We use a modified random search, based on adapting the standard deviation of the normal distribution, to solve equation (2):

Algorithm 1 Algorithm for ν_2 - Adaptive Random Search

for each patient **do**
  $\nu_{2,best} = 0.012$ ▷ Initially set ν_2 as the mid value
  $\texttt{MSE}_{best} = \textbf{MSE}(u_{num}(\nu_{2,best}), u_{ex}^{denoised})$
  while $\texttt{it} \leq \texttt{it}_{max}$ **do**
    if $\texttt{it} \leq 10$ **then**
      $\nu_{2,guess} = \nu_{2,best} + 0.004 \cdot \mathcal{N}(0,1)$ ▷ Burn-in iterations with fixed variance
    else
      $\nu_{2,guess} = \nu_{2,best} + 0.004 \, \frac{(\texttt{it}_{max} - it)}{\texttt{it}_{max} - 10} \cdot \mathcal{N}(0,1)$ ▷ Decreasing variance
    end if
    $\texttt{MSE}_{guess} = \textbf{MSE}(u_{num}(\nu_{2,guess}), u_{ex}^{denoised})$
    if $\texttt{MSE}_{guess} \leq \texttt{MSE}_{best}$ **then**
      $\texttt{MSE}_{best} = \texttt{MSE}_{guess}$
      $\nu_{2,best} = \nu_{2,guess}$
    end if
  end while
  Final estimation for the current patient is $\nu_{2,best}$ with $\texttt{MSE}_{best}$
end for

Remark: $u_{ex}^{denoised}(\nu)$ stands for the denoised exact solution and $u_{num}(\nu)$ for the numerical one.

Remark: For each iteration, the extraction of $\nu_{2,guess}$ is repeated until it falls in the range (0.0116, 0.0124), to avoid exceeding the bounds for this parameter.

Note that:

- It's a variation of the random search method because after some loops the variance is reduced according to the number of iterations;
- The MSE is computed using the difference between the numerical solution and the exact one, but the latter is denoised using a bandpass filter;
- We decided to set $\texttt{it}_{max} = 20$ in order to keep the computational load low.

2.2. Algorithm for Δt_s^* and t_s^*

Once the parameter ν_2 has been calibrated, to find the duration and timing of the shock, a simple random grid search carried out on the loss function defined in equation (3) is too expensive, since it requires knowing the numerical ECG for each patient and for each choice of $(t_s, \Delta t_s)$. Instead, we opted for a Bayesian approach, which consists in using a Bayesian Optimization algorithm, used to find the maximum of an objective function f whenever evaluating f itself is costly (financially or, as in our case, computationally). We relied on the articles

[1] we also tried using " $+\lambda(\Delta t_s)^2$ " instead of " $+\lambda \Delta t_s$ "

[1] and [3].
The core idea of Bayesian Optimization is that the objective function f is modeled a priori as a Gaussian Process, which is updated whenever new evaluations of f are available (thus obtaining the posterior distribution of f, which is proven to be still a Gaussian Process). Suppose that we already have collected n points in the dataset $\mathcal{D}_{1:n} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ on which we know the value of f: $f(\boldsymbol{x}_i) = f_i \ \forall i = 1, ..., n$. To select the $(n+1)$-th point, the algorithm chooses the one which is "most likely" to yield a higher value of f than the ones already at our disposal in our dataset $\mathcal{D}_{1:n}$. In the algorithm, the phrase "most likely" is quantified in the following way:

$$\boldsymbol{x}_{n+1} = \text{argmax}_{\boldsymbol{x}}\{u(\boldsymbol{x})\} \tag{4}$$

where u is called acquisition function. For our project, we tried using two kinds of acquisition functions: defining $f_n^* \overset{\text{def}}{=} \max_{i\in[1,n]} f(\boldsymbol{x}_i)$,

- Probability of Improvement: $u(\boldsymbol{x}) = \mathbb{P}(f(\boldsymbol{x}) \geq f_n^*) = \Phi\left(\frac{\mu(\boldsymbol{x}) - f_n^*}{\sigma(\boldsymbol{x})}\right)$
- Expected Improvement: $u(\boldsymbol{x}) = \mathbb{E}_n[\max\{0,\ f(\boldsymbol{x}) - f_n^*\}]$

The computation of both these functions is based on the assumption that f is modeled as a Gaussian Process. Finally, for our purpose we tried using the following objective functions and since our task is a minimization procedure, we set:

$$f(t_s, \Delta t_s) = -\big(\|u(t) - 0\|_{MSE} + \lambda \Delta t_s\big)$$
$$f(t_s, \Delta t_s) = -\big(\|u(t) - 0\|_{MSE} + \lambda (\Delta t_s)^2\big)$$

We report the most general form of a Bayesian Optimization algorithm:

Algorithm 2 Bayesian Optimization Algorithm

for n = 1, . . . , N **do**

Find the new point $\boldsymbol{x}_{n+1}$ by maximizing the acquisition function u: $\boldsymbol{x}_{n+1} = \text{argmax}_{\boldsymbol{x}\in A} u(\boldsymbol{x}|\mathcal{D}_{1:n})$

Evaluate the objective function f in the new point $\boldsymbol{x}_{n+1}$ and set $f_{n+1} := f(\boldsymbol{x}_{n+1})$

Augment the dataset: $\mathcal{D}_{1:n+1} = \{\mathcal{D}_{1:n}, (\boldsymbol{x}_{n+1}, f_{n+1})\}$

Compute the posterior mean $\mu(\cdot|\mathcal{D}_{1:n+1})$ and posterior variance $\sigma^2(\cdot|\mathcal{D}_{1:n+1})$

end for

Remark: A is the parameter space.
Remark: $\boldsymbol{x}$ is replaced by the couple $(t_s, \Delta t_s)$

2.3. Implementation

In the following list we report the most important functions in our repository, briefly explaining their content and describing some implementation details:

- `TF2D-nu-2.py`: a modified version of the numerical solver adapted so that it receives as input ν_2.
- `solver-ATP.py`: a modified version of the numerical solver adapted so that it receives as input the couple $(t_s, \Delta t_s)$.
- `nu2-calibration.py` : performs real ECG denoising, applies adaptive random search to tune ν_2 and saves the results in a dedicated directory.
- `find-best-savings.py`: returns the best values for parameter ν_2 for the three different patients.
- `BayesOpt-impulse-calibration.py` : computes 50 Bayesian Optimization steps, following 7 steps of pure random exploration. After finishing the exploration, the script saves the plot of the ECG for each patient and displays the impulse time and duration corresponding to the lowest value of the loss function. 50 iterations is an exaggerated number, as the algorithm consistently proposes a satisfactory couple in roughly 15 iterations.
- `results-analysis.py`: groups all the ν_2 parameters and their respective MSE obtained through Algorithm 1 and plots them for each patient.

The blue scripts are the ones that need to be run in order to obtain the main results. Also, we used some functions from the library [2].

Throughout the execution of the calibration, we noticed a recurring pattern, especially evident for the third patient: if we want to obtain a L^2 norm of the ECG in the tracking window small enough to annihilate the ECG signal, Δt_s ended up being quite high (especially for patient 3). Vice versa, if we tried to penalize a high Δt_s using a high value of λ, either the ECG did not reach zero at all or it reached zero much more slowly than the case with high Δt_s. The following list shows the final values for λ and the loss function used for the different patients (P1: Patient 1, P2: Patient 2, P3: Patient 3):

P1: L^2 norm in the tracking window $+ \lambda(\Delta t_s)^2$, $\lambda = 1$
P2: L^2 norm in the tracking window $+ \lambda(\Delta t_s)^2$, $\lambda = 1$
P3: L^2 norm in the tracking window $+ \lambda \Delta t_s$, $\lambda = 0.001$

We noticed that by restricting the bounds for Δt_s we were still able to find optimal results (i.e. the ECG was able to go to zero). Therefore, we restricted the optimization bounds for Δt_s to:

P1: $\Delta t_s \in (0, 2)\ ms$
P2: $\Delta t_s \in (0, 2)\ ms$
P3: $\Delta t_s \in (0, 8)\ ms$

Moreover, we noticed that for each patient the effective impulses were all characterized by similar ATP timing (see Figure 1). Thus, we calibrated again the impulse timing and duration for patient 3 (the one with the highest ATP duration), further limiting the optimization bounds for the timing to $(509, 513)\ ms$ and loss function, so, in the end, we set:

P3: L^2 norm in the tracking window $+ \lambda(\Delta t_s)^2$, $\lambda = 1$

Note that we increased lambda to 1 and squared Δt_s to penalize higher values of ATP duration.

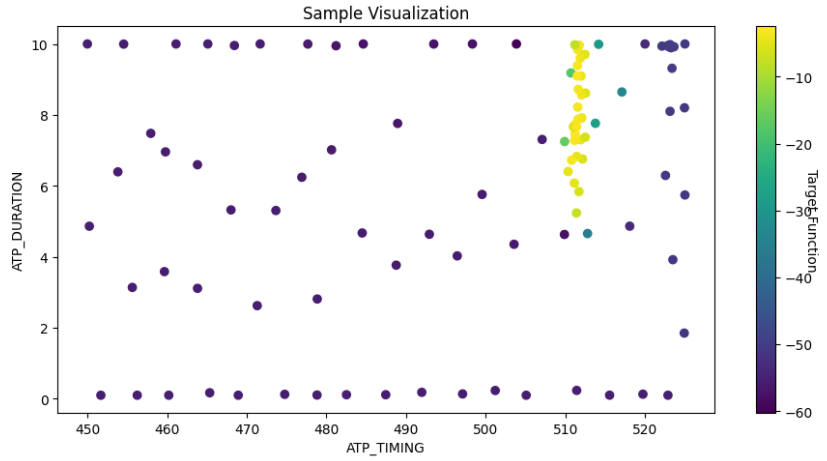

Figure 1: impulse timing vs impulse duration for patient 3 with 100 trials. The negative sign of the loss function is justified in Section 2.2.

3. Numerical results

3.1. Results for ν_2

Using the algorithm implemented in Section 2.1, we also implement a saving method for each run and each iteration, in order to be capable of finding the best result.
In the end, we obtain:

Table 1: Optimal ν_2 for each patient

Patient	ν_2	MSE
1	0.012131080952301447	0.020216118044778568
2	0.011852260526571526	0.01911400692573191
3	0.01174896977325422	0.021651723520038108

We are then compare the exact solution (with noise) and the numerical solution (with the optimal ν_2):

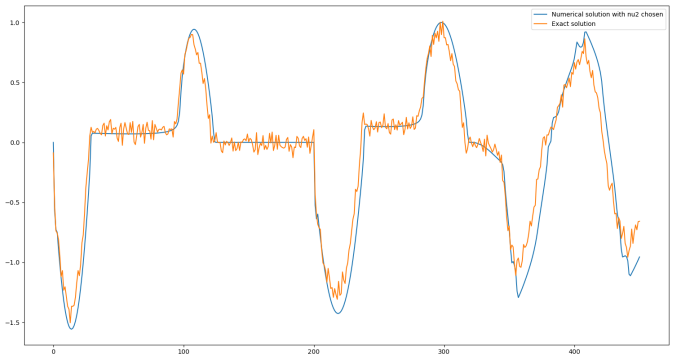

Figure 2: Comparison between numerical and exact solutions for the patient 1 with $\nu_2 = 0.012131080952301447$.

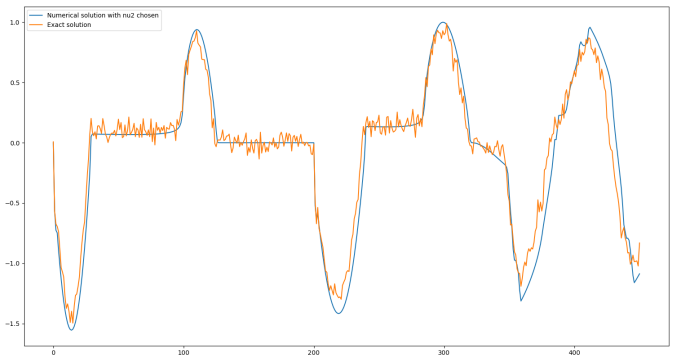

Figure 3: Comparison between numerical and exact solutions for the patient 2 with $\nu_2 = 0.011852260526571526$.

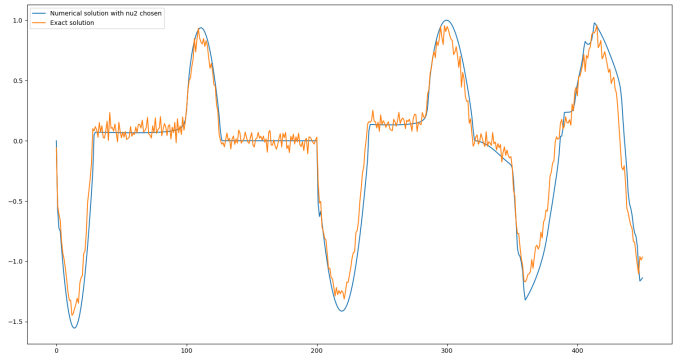

Figure 4: Comparison between numerical and exact solutions for the patient 3 with $\nu_2 = 0.011748969773254 22$.

By plotting the ν_2 parameter and the corresponding MSE reached, we obtain the following plot:

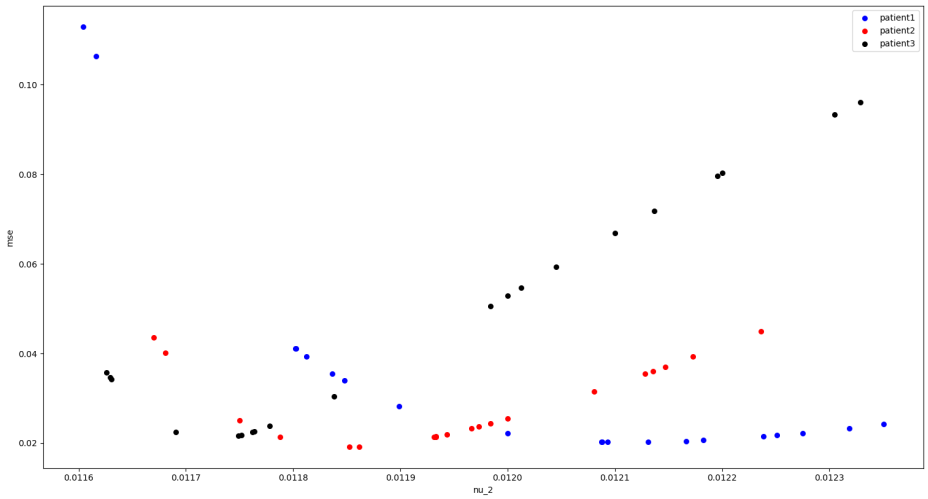

Figure 5: Comparison of different results

3.2. Results for t_s and Δt_s

The final values for ATP time and duration for the three patients are:

P1 : impulse delivered at time 484.707 ms with duration 2.697 ms
P2 : impulse delivered at time 501.435 ms with duration 2.804 ms
P3 : impulse delivered at time 511.579 ms with duration 5.038 ms

The following plots showcase the ability of our impulse strategy to reset the ECG.

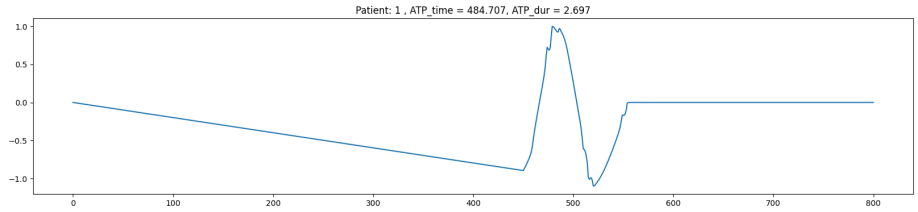

Figure 6: ECG for patient 1

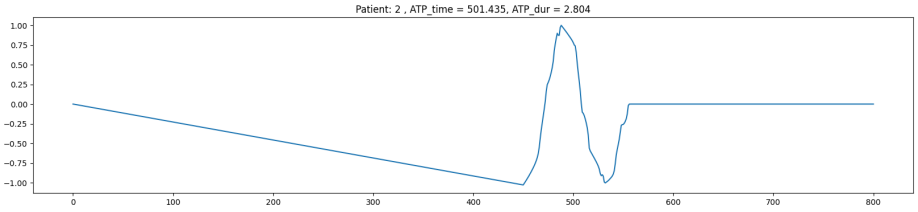

Figure 7: ECG for patient 2

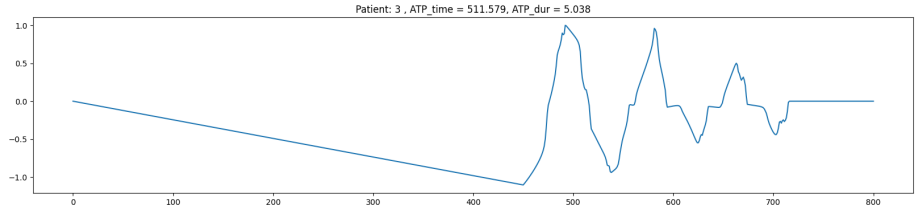

Figure 8: ECG for patient 3

Note that for efficiency we saved the initial values of the ECG for the first 450 milliseconds, hence the value plotted until this time instant in Figure 6, Figure 7 and Figure 8 can be ignored.

4. Bibliography

The documentation used to calibrate ν_2 is based on the content of the course, while the documentation for the Bayesian Optimization approach has been taken from [1], [2] and [3].

5. Conclusions

Concerning the problem of finding the optimal ATP timing and ATP duration, it is clear from Figure 1 that the Bayesian algorithm made lots of attempts in the region of the optimal values and wasted only a few iterations to search for different optimal areas. We are quite satisfied with these results because the computational cost of each iteration was very high and the algorithm effectively exploited the attempts, thus granting an accurate solution after a few iterations.
As previously explained, the added feature of a Bayesian Optimization algorithm with respect to a random search, is that it requires a lower number of iterations to propose a point that is close enough to the true argmin of the loss function. Throughout our project, we noticed that the algorithm was able to propose values of impulse timing and duration that would annihilate the ECG in the whole tracking window in just 15 iterations (roughly 30 minutes).
Further developments could consider joining the simultaneous estimation of all ν_2, t_s and Δt_s parameters in a unique algorithm, because an inaccurate ν_2 parameter could negatively influence the estimation of the other two variables, thus an optimized ν_2 could lead us to obtain an even better result.

References

[1] Brochu Eric. Bayesian optimization of expensive cost functions with application to active user modeling and hierarchical reinforcement learning. 2010.

[2] Nogueira Fernando. Bayesian optimization: Open source constrained global optimization tool for python. `https://github.com/fmfn/BayesianOptimization`.

[3] Frazier Peter. A tutorial on bayesian optimization. 2018.