Project 1
Predicting Cardiac Output from Arterial Blood Pressure

BME 580.431 Introduction to Computational Medicine I
Fall 2016
Drs. Raimond Winslow and Joseph Greenstein

Assigned Reading:
Sun et al (2009), The cardiac output from blood pressure algorithms trial. *Crit Care Med* 37(1): 72.
Sun et al (2005), Estimating cardiac output from arterial blood pressure waveforms: a critical evaluation using the MIMIC II database. *Comp Cardiol.* 32: 295.
Parlikar et al (2007), Model-based estimation of cardiac output and total peripheral resistance. *Comp Cardiol.* 34: 379.

The Circulatory System: An Introduction

The circulatory system forms a closed loop in which blood flows to carry oxygen from the lungs to the tissues throughout the body and to carry carbon dioxide back to the lungs (Fig. 1). The left side of the heart pumps oxygen-rich blood into the systemic arteries toward the capillaries. The now oxygen-depleted blood returns to the heart via the systemic veins and the right side of the heart pumps this blood into the pulmonary arteries to be distributed in the lungs. The oxygen-rich blood then returns to the left side of the heart via the pulmonary veins. Blood cells transit the full circuit in about one minute.

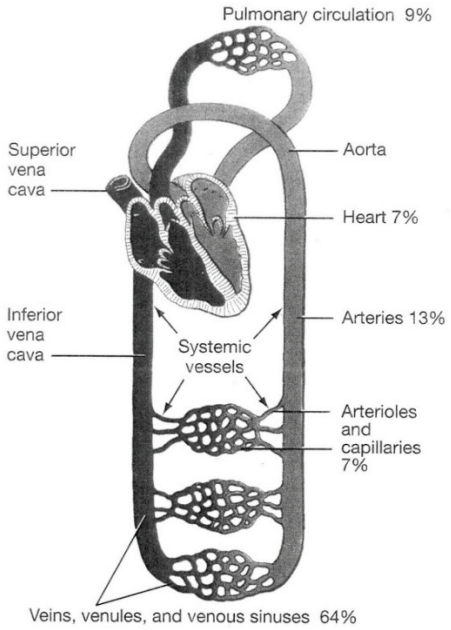

Figure 11.1 Schematic diagram of the circulatory system, showing the systemic and pulmonary circulations, the chambers of the heart, and the distribution of blood volume throughout the system. (Guyton and Hall, 1996, Fig. 14-1, p. 162.)

Blood pressure, *P*, is the force per unit area exerted on blood vessel walls, and varies both in time and distance along the circulatory system. A simple model of a blood vessel assumes it has a resistance, *R*, and that blood flow, *Q*, is linearly proportional to the pressure drop along the vessel such that $Q = \Delta P/R$. This is the equivalent to Ohm's law for an electrical circuit which relates current to voltage. In addition, blood vessels are elastic and will expand when blood pressure increases. The volume, *V*, of a vessel is linearly related the blood pressure such that $V = V_0 + CP$ where *C* is the compliance of the vessel and V_0 is the volume of the vessel at zero pressure. This is the equivalent to the capacitor equation in an electrical circuit which relates voltage and charge.

One of the earliest models of the heart and circulation is known as the Windkessel model which was formulated by physiologist Otto Frank near the turn of the 19th century. The name derives from the German word for 'bellows'. In a very simple version of this model we will consider the heart as a source of blood flow which passes through the arterial circulation modeled by a lumped single vessel. Figure 2 shows the equivalent electric circuit model in which a single resistor, *R*, represents the total peripheral resistance (TPR) and a single capacitor, *C*, represents the aggregate elastic properties of the systemic arteries.

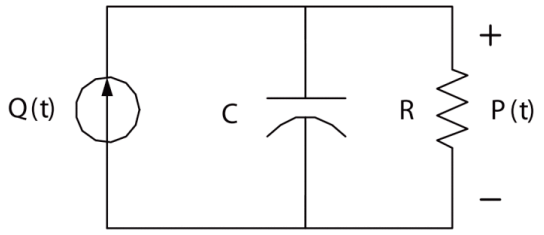

Figure 2. Windkessel model circuit representation (from Parlikar et al, 2007)

Using the equations for blood flow and vessel volume described above (assuming $V_0 = 0$), the differential equation for the Winkessel model which relates aortic arterial blood pressure (ABP) as a function of time, $P(t)$, to cardiac output (CO) (i.e. aortic blood flow), $Q(t)$, can be derived as

$$C\frac{dP(t)}{dt} + \frac{P(t)}{R} = Q(t)$$

If we average this over the cardiac cycle, for the n^{th} beat we obtain

$$C_n\frac{\Delta P_n}{T_n} + \frac{\bar{P}_n}{R_n} = \mathrm{CO}_n$$

where T_n is the period, ΔP_n is the change in ABP, and $\bar{P}_n$ is the average ABP over the cycle (see Parlikar et al for details). In steady-state, the change in ABP is proportional to the change in volume of the circulation, which is equal to the volume of blood ejected from the heart, or stroke volume (SV). A number of estimators based on this simple model are designed to

estimate SV and then CO can then be calculated when heart rate (HR) is known since CO = SV × HR. The equation above can be rewritten as

$$\mathrm{CO}_n = C_n\left(\frac{\Delta P_n}{T_n} + \frac{\bar{P}_n}{\tau_n}\right)$$

where $\tau_n = R_n C_n$. The derivation of this estimate for CO, and the method for determining τ_n is described in more detail in Parlikar et al (2007). This is a better-conditioned estimate compared to simple formulations that estimate instantaneous CO from a single beat, without using information from preceding cardiac cycles. Moreover, total peripheral resistance (TPR) can be estimated as

$$R_n = \left(\frac{\bar{P}_n}{\mathrm{CO}_n - C_n \frac{\Delta P_n}{T_n}}\right)$$

Cardiac output is a very important parameter that clinicians use when assessing circulatory function and is normally in the range of 5 L/min. Unfortunately, CO cannot be measured directly in a noninvasive manner. Measurement of CO by thermodilution (TCO) involves the insertion of a catheter into the pulmonary artery and usually performed only on critically ill patients in the intensive care unit (ICU), and the procedure itself poses risk of complications. For this reason there is a great deal of interest in formulating noninvasive techniques for measuring CO.

The idea that changes in CO can be estimated from ABP has been around for a long time, and has attracted the attention of many investigators because peripheral ABP is routinely available in ICU patients. Several methods have been developed both in the academic literature and for commercial patient monitors. Despite these efforts, no single method has emerged as a leading candidate for use in the clinic. The study of Sun et al (2009) analyzed patient data from the MIMIC II database using a variety of these CO-from-ABP prediction algorithms in order to evaluate and compare their performance. The ICU physiologic data as well as the estimation algorithms (PhysioToolkit implemented in Matlab) used by these authors were made publicly available (http://www.physionet.org/physiotools/cardiac-output/). In this project, you will work with this data set and the authors' own tools in order to reproduce their results and gain a greater appreciation for what steps are required to make sense of this "real-world" data.

Project: Estimating CO from ABP

The documentation for PhysioTookit software for Matlab can be found at http://physionet.org/physiotools/matlab/. You may want to review this to gain some insight into the tools that are available for interacting with the MIMIC II datasets which are included within PhysioBank. This is just for informational purposes as the MIMIC II datasets needed for this project have already been extracted.

The data you will be using have been shared with you on OneDrive, a cloud computing resource available from your my.jh.edu portal. If you have not yet received an invitation to this shared drive, please see one of the instructors. This drive contains data for 282 MIMIC II patients. For each patient there is a numeric file (file names end with 'n') that contains a variety of vital signs measured once-per-minute (or less frequently). The first two rows of the file provide physiological variable name and units for each column. This file will contain one or more measurements of TCO for each patient. Please have a look at a few examples, and note that the files may not all contain the same set of variables, or list them in the same order for every patient. In addition to the numeric files, you will find files that contain time-series ABP data obtained a 125-Hz sampling rate (file names end with 'ABP'). Column 1 is time in seconds and column 2 is ABP in mmHg. Note that there are no headers in these time-series files. Also note that there may be time periods for which data do not exist in these files. Familiarize yourself with the structure of these data.

The original PhysioToolkit algorithms (Matlab) have been copied into folder 'PhysioToolkitCardiacOutput_MatlabCode' which you can find in the folder 'Project_Assignements_2016' in the class repository on GitHub. For this project we will be working with Matlab codes in the '2analyze' and '3estimate' subfolders. The shared data on OneDrive contains MIMIC II patients for which there are both CO measurements and ABP signals recorded.

1. We will first work with patient 's00020' (or #20). Read the description subtitled "ABP Signal Processing" of Sun et al (2009) in conjunction with the three Matlab functions in the '2analyze' folder. Use these functions to reproduce traces similar to those shown in their Figure 1 (including the ABP waveform and features denoted by the symbols). Do this for patient #20, showing the first 20 ABP pulses starting at 10 hours, and repeat for the first 20 pulses starting at 11 hours. Plot your results and prepare a documented version of your code to be submitted as part of the Coding Progress Assignment (described below).

2. Produce two or more additional example traces taken from one or more additional patients (over any appropriate timeframes). Observe the variations in the ABP waveform and features across measurements and patients.

3. All of the CO-from-ABP algorithms provide a relative estimate of CO which can be calibrated by one or more TCO measurements. We will use the calibration method called C2 in Sun et al (2009). Your task here is to reproduce the estimates of continuous CO, as well as the ABP features PP, MAP, and HR as shown in their Figure 4. First do this for patient #20 using the Liljestrand algorithm (estimator #5). For consistency, analyze results *only over the first 12 hours for this patient*. Plot your results and prepare a documented version of your code to be submitted as part of the Coding Progress Assignment.

Note that the Matlab function estimate_CO.m in the subfolder '3estimate' is not directly compatible (as provided by the original authors) with the output of the '2analyze' functions. A modified function estimateCO_v2.m has been provided in its place in folder 'Project_Assignements_2016'. This function requires, as output from the signal processing steps, a Matlab data file (.mat) with variables named: time, ABP, t_on, feat, and beatq. An alternative modified function estimateCO_v3.m has also been provided, which takes t_on, feat, and beatq as function arguments rather than loading these from a file. You may use either of these in place of the original. Please note that estimator #s 8, 9, and 11 cannot be used due to additional data requirements.

4. For patient #20, compare the Liljestrand algorithm to two other algorithms of your choice. Note the performance of the different algorithms. Repeat you your comparison of the three CO-from-ABP algorithms on the data analyzed above in problem 2.

Coding Progress Assignment (Due Monday Sept 19 at 12pm via GitHub):

Only a single submission is required for each group. Complete problems 1 and 3 only for this assignment (i.e. analysis of patient #20 only). You should have a single Matlab script that runs all functions needed to generate the results (plots) for patient #20. Document your Matlab code including any additional functions that you wrote. Your submitted code will be run and evaluated by the instructors. Simply upload all files to your group repository and the instructors will be able to retrieve them. All files should include your group name, "P1CA" (project 1 coding assignment), and "AofB" where A is file number and B is total number of files (e.g. "3of5") being submitted, within the file name so the instructors can identify all project files easily.

In problems 5 and 6 you will implement the CO estimator of Parlikar et al (2007). The authors have not provided code for this algorithm, so it will be necessary to write your own code based on the description of the algorithm in the paper. Note that it will be necessary to use least-square-error techniques for estimates of τ_n on each beat and it will be necessary to determine C_n by using a calibration method similar to that performed in problem 3. The Matlab functions provided by Sun et al (2009) for estimation of beat onset time and other features from ABP (in the '2analyze' folder) will be useful in your implementation of the Parkilar et al (2007) estimator.

5. Repeat the algorithm comparison performed in problem 4 using this new CO estimator, but with the same patient data sets. Reproduce the estimates of continuous CO using the Parlikar algorithm as shown in Figure 4 of Sun et al (2009). You may incorporate your code for the Parlikar estimator as a new function called by estimateCO_v2.m or estimateCO_v3.m (some modification will be necessary to pass needed data to your estimator function) or you may write a separate estimator function. Compare the performance of the Parlikar algorithm to the Liljestrand algorithm as well as the other algorithms used in problem 4. What difficulties did you encounter in the process of reproducing the coded algorithm directly from the description in the published paper?

6. For patient #20 and the same additional patient(s) studied above, estimate TPR over the same timeframe (over the first 12 hours for patient #20 as in problem 3). Important: When plotting TPR, use units of mmHg/(mL/s) similar to that shown in Fig 2 of Parlikar et al (2007). Why is TPR a clinically important quantity?

Oral Presentations (In Class Friday Sept 23, submit slides to GitHub by 11am):

The oral presentation covers your results for all aspects of this project, however, since time is limited, there is no need to include introductory or background materials. Each group will have a single team presentation which should be less than 15 slides and less than 10 minutes. Describe how you calculated and calibrated CO (and TPR). Focus on the results of your CO estimation algorithms and compare the performance of the algorithms tested. Be sure to include all results for patient #20 and answer questions noted in the problems. In addition, include a slide (or two) describing what you found most interesting about working with real patient data, any unexpected difficulties you encountered, and how you overcame them. On your final slide (which need not be presented orally), include a brief statement describing the contribution of each member of the group to the product. Slides must be submitted to GitHub by 11am on Sept 23, so that the instructors can download them to a common machine to be used for presentations.