

Optical system for assessment of atrial and ventricular mechanics in Langendorff-perfused mouse hearts

Guide to using algorithms for processing cardiac mechanics in MATLAB

Jorge Armenta Valenzuela¹ and Jesus Rodriguez Gonzalez¹

¹Centro de Investigación y de Estudios Avanzados del IPN, Unidad Monterrey, Vía del Conocimiento 201, Parque de Investigación e Innovación Tecnológica, 66600 Apodaca NL, México

* jrodriguez@cinvestav.mx

Table of Contents

1.	Video normalization	2
2.	Global mechanical dynamics analysis	4
3.	Segmental analysis of the ventricular mechanical dynamics	6
4.	Segmental analysis of the atria's mechanical dynamics	8
5.	Electro-mechanical analysis	11

Video normalization

Summary: The program **SupplementarynNormalizationMatlabFile4.m** performs normalization and segmentation on all selected videos.

- Prerequisites for Optimal Algorithm Performance:

- **Video Organization:** Place all videos obtained from the same experiment into a single folder. The program will automatically create a subfolder named **“VideoNorRes”** within that directory to save the results without modifying the original videos.
- **Required Files and Functions:** Ensure that the function file **“FucMasIm.m”** is in the same folder as this program. Otherwise, MATLAB will throw an error stating that the function does not exist.
- **Video Format:** The code is specifically designed to process files with the **“.avi”** extension. If you wish to analyze videos with a different extension, the necessary modifications must be made to the program.

- Variable configuration (Before starting)

Variable	Data Type	Description
“ref”	Char	Video Filter: The specific word that the names of the videos you wish to process must start with (e.g., 'Supple'). Any videos that do not begin with this text will be ignored.
“DivI”	Double (0.0 - 1.0)	Maximum Intensity: The maximum value to which the pixels will be scaled. A value of 1.0 is equivalent to full saturation (255). A value that eliminates reflections from saturating the object of interest is recommended.
“minP”	Int8 (0 - 255)	Background Threshold: Any pixel value below this limit will be removed (set to 0).
“a1”	int (0 or 1)	Activate Manual ROI: • “a1” = 1: Opens an interactive window on the first video to crop out unwanted areas. • “a1” = 0: Skips the manual selection and runs the process completely automatically.

Video normalization

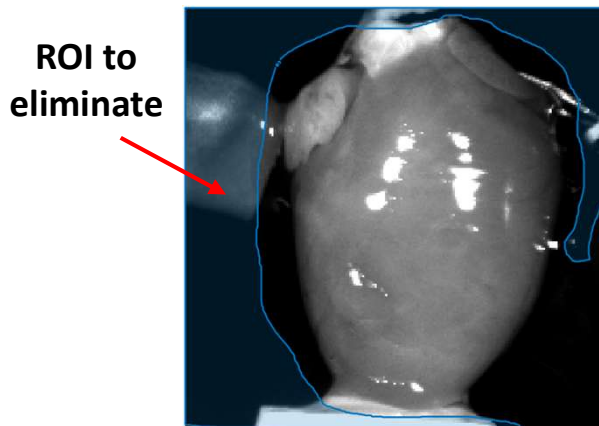
Execution Instructions:

1 Press "Run" in MATLAB: The first thing you will see on the screen is a system pop-up window ("*uigetdir*"). Select the folder where your experimental videos are saved and click Select Folder.

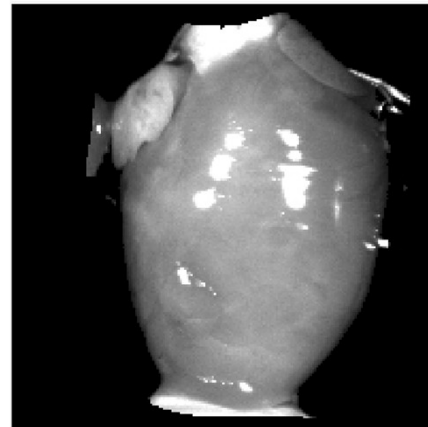
2. Region of Interest (ROI) Selection: If the variable "*a1*" = 1, the program will load a frame (video FPS + 1) from the first video on the list and open an interactive window:

Use the guided interactive tool ("*drawassisted*") to click around the perimeter of the heart or the specific area of interest you wish to keep (View the "Before" image).

Complete the path to close the polygon. The software will use this exact mask to eliminate external noise and will apply it identically to all subsequent videos to maintain geometric consistency (View the "After" image).



Before



After

3. Processing Monitoring: The program will automatically begin frame-by-frame analysis: morphological filtering, artifact removal, internal hole filling ("*imfill*"), and the calculation of the mathematical signal normalization. You can monitor the progress in the MATLAB Command Window, where the status of the videos will be printed.

4. Obtained Results: Once the program finishes the cycle, navigate to your videos' source folder using your file explorer. You will find a new folder named "**VideoNorRes**". Inside, you will see the new normalized videos exported in high-quality ".avi" format (Motion JPEG AVI at 100%). The modified files will carry the prefix assigned in the program (by default, "**ResNor[Original_Name].avi**").

Technical Note: The code utilizes an iterative nested loop (**for** in *i* and *j*) to perform pixel-by-pixel normalization applying the following formula:

$$VN = \frac{Im - m}{max - min}.$$

Depending on the video resolution and the number of frames (*N*), this process may take a few minutes per file due to the computational load on three-dimensional matrices.

Global mechanical dynamics analysis

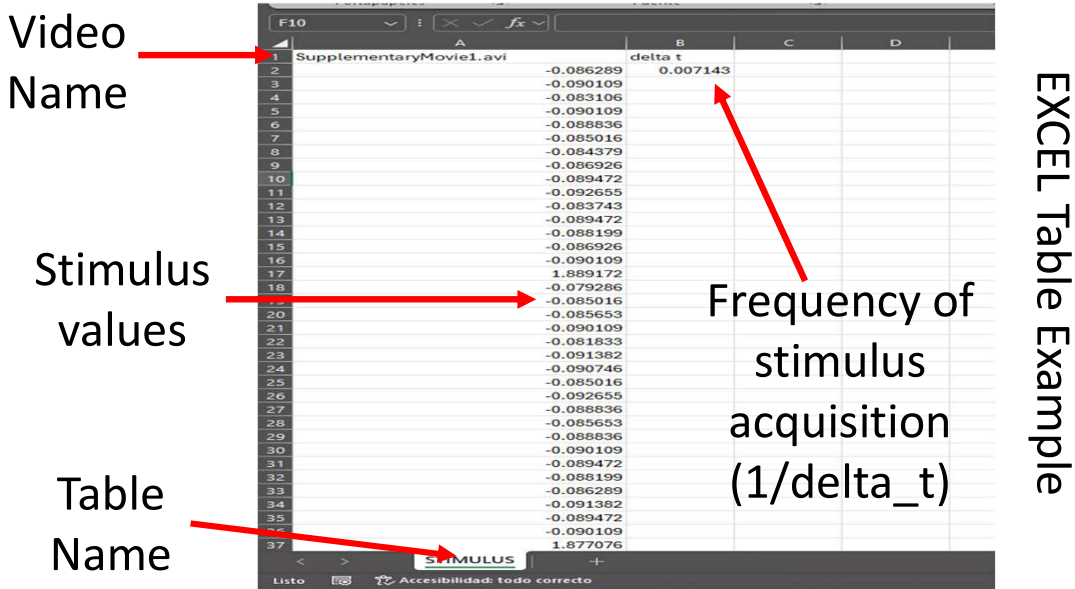
Summary: The program `SupplementaryHeartAreaMatlabFile5.m` calculates the variations in the total area of the heart in all frames of the selected videos.

1. Prerequisites for Optimal Algorithm Performance:

Input Videos: Must be in “.avi” format. The program supports the processing of either a single video or multiple videos simultaneously.

Stimulus File: An Excel file (.xlsx) is required, containing a sheet named exactly STIMULUS (refer to SupplementaryGuideAPPLabview.pdf).

Functions: The function file “*FucPoincare.m*” must be saved in the same directory as this main program.



- Variable configuration (Before starting)

Variable	Data Type	Description
"uhm"	Int8(0 - 255)	Intensity Threshold: Removes pixels with saturation or bright reflections in the fabric. Any pixel with a gray value above this threshold will be discarded (0).
"arRef"	Int	Minimum Area Filter: Removes isolated objects or background noise smaller than "arRef" pixels ² , leaving only the main structure of the heart.
"a3"	Int (0 or 1)	Poincare Graphs "a3" = 1: Generates and visually displays the point cloud of the Poincare map. "a3" = 0: Turn off the generation of these graphics to speed up processing.

Global mechanical dynamics analysis

Execution Instructions:

1. Video Selection:

- Execute the program in MATLAB (**Run**). A system pop-up window will open. Select the **“.avi”** video(s) you wish to analyze (you can hold down **“Ctrl”** to select multiple files).
- Immediately after, a second pop-up window will appear. Select the corresponding Excel file (**“.xlsx”**) containing the stimulus data.

2. Image Processing:

The program will automatically initiate a cycle to process each selected video. During this stage, the following operations are performed frame by frame:

- Conversion to grayscale and binarization.
- Image cleaning using morphological functions to isolate only the cardiac region.
- Measurement of the heart area (by counting pixels with values > 0).

3. Signal Analysis and Digital Filtering:

- **Signal Inversion:** Minimum values are transformed into relaxation and maximum values into contraction.
- **Band-pass Filter:** A 5th-order bidirectional Butterworth filter is applied between 3 and 100 Hz to suppress high-frequency noise and perfusion dripping artifacts.
- **Adaptive Smoothing:** A Savitzky-Golay filter (**“sgolayfilt”**) is implemented, with a window size that adjusts automatically based on the detected stimulus frequency.

4. Result Visualization:

- **Synchronization Plot:** Displays the smoothed heart area signal, the applied stimulus signal, and circular markers (**‘o’**) over the detected peaks in both signals. The first second of the video is automatically discarded to allow data stabilization.
- **Poincare Plot** (If **“a3”** = 1): Displays the dispersion of the Area-to-Area intervals (IAA_t vs IAA_{t+1}) to evaluate heart rate variability.

Saved Data

Upon completion of each video's processing, the program automatically saves the variables of interest.

- **Location:** Results are saved in the same directory where the original video file is located.
- **File Naming Convention:** A MATLAB data file (**“.mat”**) is generated using the following format:

“Area[Stimulus Frequency or Basal].mat”

(Example: If the stimulus frequency was 300 PPM, the file will be named **“Area300.mat”**. If no stimulus is detected, it will be saved as **“AreaBasal.mat”**).

Variables stored within the **“.mat”** file:

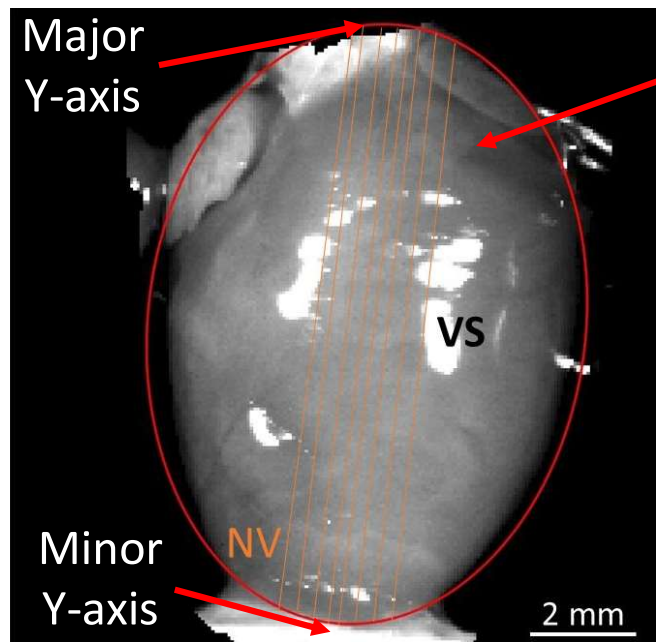
- **file3:** Name of the original processed video.
- **ID:** Code identifier (**‘Area’**).
- **CoraF:** Filtered heart area signal (unnormalized).
- **BPM:** Calculated heart rate in beats per minute.
- **EstiVec:** Vector containing the original stimulus signal.
- **freEs:** Frequency of the applied stimulus (converted to beats per minute).
- **t** and **tE:** Time vectors for the video and the stimulus, respectively.
- **FreVi** and **Fm:** Video sampling frequency (FPS) and stimulus acquisition rate (Hz), respectively.

Segmental analysis of the ventricular mechanical dynamics

Summary: The program **SupplementaryVentriclesMatlabFile6.m** estimates an ellipse that fits as closely as possible to the heart's perimeter and calculates ventricular shortening by taking the maximum and minimum points of the ellipse along the Y-axis.

1. Prerequisites for Optimal Algorithm Performance:

- **Input Videos:** Must be in **“.avi”** format. The program supports the processing of either a single video or multiple videos simultaneously.
- **Stimulus File:** An Excel file **(.xlsx)** is required, containing a sheet named exactly **STIMULUS** (refer to **SupplementaryGuideAPPLabview.pdf**).
- **Functions:** The function file **“filterPeaks.m”** must be saved in the same directory as this main program.



Representative image of the ellipse estimate

Note: To determine the value of **“CanE”** depends on the number of points that are used to estimate the ellipse (Example: if the ellipse has 200 points 10% of its value is placed, this takes half of the data from the bottom and top)

- Variable configuration (Before starting)

Variable	Data Type	Description
“canE”	Int	Lines: Number of maximum and minimum points of the ellipse along the Y-axis (NV) that will be used to calculate and average the shortening.
“uhm”	Int8(0 - 255)	Threshold: Eliminates pixels with values above this threshold to suppress saturation or light reflections on the cardiac tissue.
“arRef”	Int	Area Filter: Morphological threshold to discard artifacts or floating objects with areas smaller than this value.
“uhmAm”	Double (0.0 - 1.0)	Relative Amplitude Threshold: Internal filter passed to “filterPeaks” that discards events whose relative amplitude is lower than this value.
“TiemMin”	Int	Exclusion Time (s): Automatically discards any event detected during a time frame below this value.
“a1”	int (0 or 1)	Record Representative Ellipse Video: “a1” = 1: Generates and saves a new video showing the frame-by-frame ellipse fitting. “a1” = 0: Does not save any video.
“ReFPS”	int	Representative Video Frequency: Frame rate per second (FPS) at which the video is resampled (if “a1” = 1).

Segmental analysis of the ventricular mechanical dynamics

Execution Instructions:

1. Video Selection:

- Execute the program in MATLAB (**Run**). A system pop-up window will open. Select the **“.avi”** video(s) you wish to analyze (you can hold down **“Ctrl”** to select multiple files).
- Immediately after, a second pop-up window will appear. Select the corresponding Excel file (**“.xlsx”**) containing the stimulus data.

2. Image Processing and Ellipse Fitting:

The program will automatically initiate a cycle to process each selected video. During this stage, the following operations are performed frame by frame:

- **Morphological filtering.**
- Localization of the object with the largest area using region properties (**“regionprops”**).
- **Ellipse Fitting:** The centroid, orientation, major axis, and minor axis of the heart are calculated.
- **Ventricular Shortening (VS) Calculation:** The geometric boundaries along the Y-axis are extracted to determine the shortening distance.

3. Signal Analysis and Digital Filtering:

- **Signal Inversion:** Minimum values are transformed into relaxation and maximum values into contraction.
- **Band-pass Filter:** A 5th-order bidirectional Butterworth filter is applied between 3 and 100 Hz to suppress high-frequency noise and perfusion dripping artifacts.
- **Event Filtering:** The **“filterPeaks”** function processes event peaks by calculating individual amplitudes. Furthermore, it discards any event with a relative amplitude lower than the specified **“uhmAm”** value.

4. Result Visualization:

As each video is processed, a plot will be displayed on the screen containing:

- The normalized ventricular shortening signal over time.
- Circular markers (**‘o’**) indicating the ventricular contraction peaks that successfully met the filtering criteria.
- The applied stimulus signal with its respective peaks shown in parallel (if the video does not correspond to a basal recording).

Saved Data

Upon completion of each video's processing, the program automatically saves the variables of interest.

- **Location:** Results are saved in the same directory where the original video file is located.
- **File Naming Convention:** A MATLAB data file (**“.mat”**) is generated using the following format:

“Ventricles[Stimulus Frequency or Basal].mat”

(Example: If the stimulus frequency was 300 PPM, the file will be named **“Ventricles300.mat”**. If no stimulus is detected, it will be saved as **“VentriclesBasal.mat”**).

Variables stored within the **“.mat”** file:

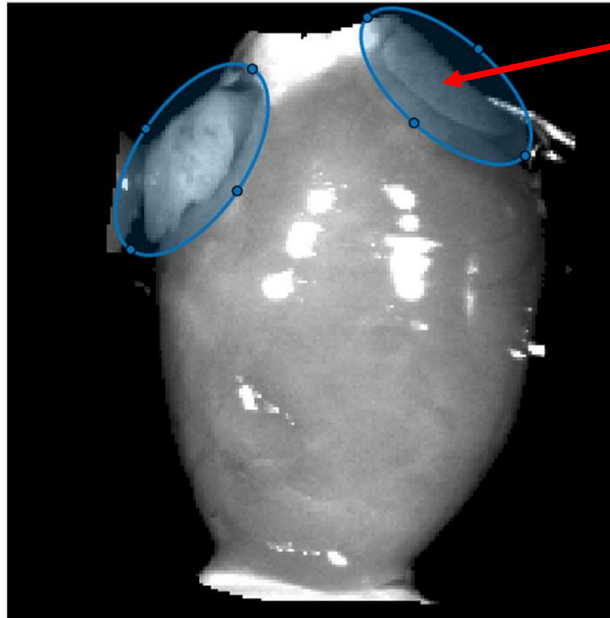
- **file3:** Name of the original processed video.
- **ID:** Code identifier (**‘Ventricles’**).
- **VenF:** Temporary sign of ventricular shortening (not normalized).
- **BPM:** Calculated heart rate in beats per minute (BPM).
- **EstiVec:** Vector containing the original stimulus signal.
- **freEs:** Frequency of the applied stimulus (converted to BPM).
- **inter:** Expected time interval between stimuli(s).
- **t** and **tE:** Time vectors for the video and the stimulus, respectively.
- **FreVi** and **Fm:** Video sampling frequency (FPS) and stimulus acquisition rate (Hz), respectively.

Segmental analysis of the atria's mechanical dynamics

Summary: The program `SupplementaryAtriaMatlabFile7.m` calculates the area of each atrium over time and obtains an average of the atrial response.

1. Prerequisites for Optimal Algorithm Performance:

- **Input Videos:** Must be in “.avi” format. The program supports the processing of either a single video or multiple videos simultaneously.
- **Stimulus File:** An Excel file (.xlsx) is required, containing a sheet named exactly STIMULUS (refer to `SupplementaryGuideAPPLabview.pdf`).
- **Functions:** The functions file “*filterPeaks.m*” (Event Filter) and “*AumCont.m*” (Contrast Enhancement) must be saved in the same directory as this main program.



Representative image of ROIs

Note: When placing the Regions of Interest (ROIs), ensure that the atrium is centered and occupies a maximum of 80% of the total ROI area.

- Variable configuration (Before starting)

Variable	Data Type	Description
“ <i>uhm</i> ”	Int8(0 - 255)	Threshold: Eliminates pixels with values above this threshold to suppress saturation or light reflections on the cardiac tissue.
“ <i>TiemMin</i> ”	Int	Exclusion Time (s): Automatically discards any event detected during a time frame below this value.
“ <i>mask</i> ”	Int (0 or 1)	ROI Selection Mode: “ <i>mask</i> ” = 0: Opens an interactive window in the first video in the 1s frame to draw the ROIs of the atria manually. “ <i>mask</i> ” = 1: Skips the manual step and asks you to upload a previous .mat file to reuse already saved masks.
“ <i>uhmAm</i> ”	Double (0.0 - 1.0)	Relative Amplitude Threshold: Internal filter passed to “ <i>filterPeaks</i> ” that discards events whose relative amplitude is lower than this value.

Segmental analysis of the atria's mechanical dynamics

Execution Instructions:

1. Video Selection:

- Execute the program in MATLAB (**Run**). A system pop-up window will open. Select the **".avi"** video(s) you wish to analyze (you can hold down **"Ctrl"** to select multiple files).
- Immediately after, a second pop-up window will appear. Select the corresponding Excel file (**".xlsx"**) containing the stimulus data.

2. Definition of the Regions of Interest (ROIs):

The program will use the first frame of the first second of the first video as a reference to isolate the atria:

- If **"mask" = 0 (Manual Selection)**:
 - A **Figure** window displaying the heart image will open. The cursor will change to allow you to draw an ellipse. Trace the ellipse over **Atrium 1** and press the Spacebar or Enter.
 - The message **"Atrium 1"** will appear in the Command Window.
 - The cursor will reactivate. Draw the second ellipse over **"Atrium 2"** and press Spacebar or Enter again.
 - A new **Figure** window will open, showing a visual confirmation of the two isolated regions against a black background. The code will apply these exact coordinates identically to all other selected videos.
- If **"mask" = 1 (Load Previous Masks)**:
 - The program will open a system pop-up window prompting you to select the **".mat"** file containing the previously saved **"b1"** and **"b2"** variables (ROIs for **Atrium 1** and **Atrium 2**, respectively).

3. Image Processing:

The program will automatically initiate a cycle to process each selected video. During this stage, the following operations are performed frame by frame:

- Processing of the **Atrium 1** and **Atrium 2** area separately.
- **Image Optimization**: Application of digital sharpening (**"imsharpen"**) and local adaptive contrast enhancement using the **"AumCont"** function.
- **Edge Detection**: Implementation of the **Canny** algorithm (**"edge"**) to precisely delineate the perimeter of the atrial tissue.
- **Area Calculation**: Estimation of the area by counting and quantifying the pixels that make up the detected edges in motion.

4. Signal Analysis and Digital Filtering:

Once the temporal area curves for both atria are obtained, the following procedure is executed:

- **Band-pass Filter**: A 5th-order bidirectional Butterworth filter is applied between 3 and 100 Hz to suppress high-frequency noise and perfusion dripping artifacts.
- **Smoothing**: Implementation of a Savitzky-Golay filter (**"sgolayfilt"**) dynamically adjusted based on the detected external stimulus frequency.
- **Signal Inversion**: Inversion of the mechanical vectors so that the maximum peaks represent the contraction phase and the minimum points represent the relaxation phase.
- **Atrial Average Signal ("AFS")**: Calculation of a unified signal by averaging the individual signals from **Atrium 1** and **Atrium 2** (**"AFS1"** and **"AFS2"**, respectively) using the following formula:

$$AFS = \frac{AFS1 + AFS2}{2}$$

- **Event Filtering**: The **"filterPeaks"** function processes event peaks by calculating individual amplitudes. Furthermore, it discards any event with a relative amplitude lower than the specified **"uhmAm"** value.

Segmental analysis of the atria's mechanical dynamics

Saved Data

Upon completion of each video's processing, the program automatically saves the variables of interest.

- **Location:** Results are saved in the same directory where the original video file is located.
- **File Naming Convention:** A MATLAB data file (".mat") is generated using the following format:

"Atria[Stimulus Frequency or Basal].mat"

(Example: If the stimulus frequency was 300 PPM, the file will be named "**Atria300.mat**". If no stimulus is detected, it will be saved as "**AtriaBasal.mat**").

Variables stored within the ".mat" file:

- **file3:** Name of the original processed video.
- **ID:** Code identifier ('**Atria**').
- **b1** and **b2:** Binary matrices of the elliptical masks used (useful for future executions with "**mask**" = 1).
- **AFS1** and **AFS2:** Filtered and smoothed temporal signals from **Atrium 1** and **Atrium 2** (not normalized).
- **NorA:** Average and normalized signal in the range [0, 1].
- **BPM:** Calculated heart rate in beats per minute (BPM).
- **EstiVec:** Vector containing the original stimulus signal.
- **freEs:** Frequency of the applied stimulus (converted to BPM).
- **inter:** Expected time interval between stimuli (s).
- **t** and **tE:** Time vectors for the video and the stimulus, respectively.
- **FreVi** and **Fm:** Video sampling frequency (FPS) and stimulus acquisition rate (Hz), respectively.

Electro-mechanical analysis

Summary: The program **SupplementaryCalculationDelayAVMatlabFile8.m** calculates the delay between the onset of atrial and ventricular events (AV delay).

1. Prerequisites for Optimal Algorithm Performance:

To ensure proper software functionality and optimal data synchronization, the following prerequisites must be met:

- **Input Files (".mat"):** You must have the data files previously generated by the individual atria (**Atria[Frequency].mat**) and ventricles (**Ventricles[Frequency].mat**) processing codes.
- **File Selection:** Both files must be selected simultaneously within the user interface and must correspond strictly to the same experimental video. The program will read the internal **"ID"** variable to automatically distinguish and identify each structure.
- **Required Functions:** The external function file **"filterPeaks.m"** must be located within the same directory as this main program.

- Variable configuration (Before starting)

Variable	Data Type	Description
"TiemMin"	Int	Exclusion Time (s): Automatically discards any event detected during a time frame below this value.
"uhmAm"	Double (0.0 - 1.0)	Relative Amplitude Threshold: Internal filter passed to "filterPeaks" that discards events whose relative amplitude is lower than this value.
"we4"	Double	Physiological Delay Window: On the final lines, the code discards any delay calculations that are outside the given range ensuring that only physiologically viable ranges are averaged.

Electro-mechanical analysis

Execution Instructions:

1. Data Selection:

Execute the script in MATLAB (**Run**).

- A system pop-up window will open. Navigate to your results folder and select both **“.mat”** files simultaneously (hold down **“Ctrl”** to click and select both files, e.g., **Atria180.mat** and **Ventricles180.mat**).
- **Automated Validation:** The program will internally verify that:
 - Exactly two files have been selected (an error will be thrown if only one file is chosen).
 - The original video filenames (**“file3”**) match in both files (an error will be thrown if you attempt to combine the ventricles of one experiment with the atria of a different one).

2. Signal Analysis and Filtering:

- **Atrial Reconstruction:** Averages the signals from both atria (**“AFS1”** and **“AFS2”**) to obtain the unified atrial signal.
- **Phase Inversion:** Inverts both signals (atrial and ventricular) so that the mechanical onset is positioned at the peak of the waveform.
- **Smoothing:** Implements an adaptive Savitzky-Golay filter (**“sgolayfilt”**) based on the stimulation frequency (**“inter”**) saved during previous experiments.

3. Event Detection (Coupling Algorithm):

To prevent errors arising from missed or double events, the program executes a conditional matching loop:

- **Onset Verification:** It ensures that the first event registered on the timeline is an atrial contraction. If an orphan ventricular peak is detected at the very beginning, it is discarded.
- **Pair Validation:** It verifies that each ventricular contraction is preceded by a single atrial contraction. If multiple atrial peaks are detected within a single cycle, only the first one is retained, and the others are removed from the equation. If a ventricular contraction lacks a preceding atrial contraction, the entire event is deleted.

4. Graphical Link Display:

The program will automatically display a control viewport:

- Plot of the temporal curve of the ventricle (**“VeF”**) and the atrium (**“AurF”**).
- It superimposes circles (**'o'**) on the validated peaks of both fabrics so that you can visually certify that the algorithm detected the waves in parallel and correctly.

Results:

Upon completion of filtering the events together, the code will calculate the time difference between each validated pair (**Ventricle Time - Atrial Time**) and directly display the results in the MATLAB Command Window:

- **PAV** (Average AV Delay): The average delay time expressed in seconds (e.g., a value of 0.045 represents an average atrioventricular delay of 45 ms).
- **DAV** (AV Delay Standard Deviation): The variability of the delay throughout the batch of beats analyzed (expressed in seconds).