Simulink Documentation

Part 1: For all implemented components you will need to document your design and implement the design in Simulink.

Document the design as follows:

1. List all requirements. This should be done formally (as in not just a bunch of bullet points), but as a group you can decide what formalism you want to use (mathematical, informal, somewhere in between).
2. Requirements must be concise and disjoint. They should also be traceable to the design and vice versa
3. List all design decisions.
4. The Simulink diagram must include necessary annotation to understand the model. There should be screenshots of the simulink model in the documentation. Provide an additional section in the document that describes testing you performed, and the results.

This document defines the functions and operating characteristics that the system must perform and provides a description of the system's environmental performance parameters and critical interactions.

# Part 1: Requirements and design

## Requirements

These are all the requirements for the Simulink implementation of the pacemaker and its modes.

### Overall Pacemaker System

The pacemaker should provide rate adaptive bradycardia pacing support, historical data on device implementation, and user diagnostics through brady analysis function. In addition, the bradycardia analysis functions should allow pacing measurements such as lead impedance, pacing threshold, P and R wave measurement, battery status, temporary brady pacing, motion sensor trending, and tests to be performed.

### Device Operation

* The device should monitor and handle a patient's heart rate by detecting and providing therapy for bradycardia.
* The device should provide programmable, single-chamber, rate-adaptive pacing, and permanent state.
* The device should be programmed and controlled through the Device Controller-Monitor (DCM).
* The device should provide history data such as output rate histograms (atrial and ventricular) and sensor output data.

### Pulse Pacing (Atrial and Ventricular)

The device should result in pulses with programmable voltages and widths for atrial and ventricular, which provide electrical heart-pacing stimulation.

### Amplitude and Width

Both amplitude and width for pulse pacing should be independently programmable for each chamber.

### Rate Sensing

Bipolar electrodes and a sensing circuit should operate rate sensing. Rate detection should be based on the measured cardiac cycle measurements of the sensed rhythm and assessed on an interval-by-interval basis.

### Operation Modes

The DCM will be used to select a permanent bradycardia operating mode for the pacemaker at startup. At present, it will choose between the modes AOO, VOO, AAI, and VVI. Table 1.0 outlines the features present in each mode.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | LRL | URL | Atrial Amplitude | Ventricular Amplitude | Atrial Pulse With | Ventricular Pulse Width | Atrial Sensitivity | Ventricular Sensitivity | ARP | VRP |
| AOO | O | O | O |  | O |  |  |  |  |  |
| VOO | O | O |  | O |  | O |  |  |  |  |
| AAI | O | O | O |  | O |  | O |  | O |  |
| VVI | O | O |  | O |  | O |  | O |  | O |

Table 1.0 Programmable Parameters for AOO, VOO, AAI, and VVI modes

### Bradycardia Therapy

### Lower Rate Limit (LRL)

The number of generator pace pulses delivered (per minute) should be affected by following the requirements.

1. LRL should be the longest allowable definition of pacing interval.
2. The LRL interval should begin at a ventricular sensed or paced event in VII and VOO mode.
3. The LRL interval should begin at an atrial sensed or paced event in AII and AOO mode.

### Upper Rate Limit (URL)

The Upper Rate Limit (URL) refers to the highest pace at which the heart's ventricles will follow the detected atrial signals. The URL interval represents the shortest duration between one ventricular event and the subsequent ventricular pacing.

### Refractory Periods

1. Ventricular Refractory Period (VRP): It should be the set duration of time after a ventricular event during which neither the sensing of ventricular activity will prevent nor initiate ventricular pacing.
2. Atrial Refractory Period (ARP): In single-chamber atrial modes, it should be the preset time duration following an atrial event, within which atrial events will neither stop nor initiate pacing.
3. Post Ventricular Atrial Refractory Period (PVARP): It must be available in modes with ventricular pacing and atrial sensing. It is a customizable period of time after a ventricular event during which an atrial heart event should neither block an atrial pace nor activate a ventricular pace.

### Programmable Parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Parameter | Programmable Values | Increment | Nominal | Tolerance |
| Modes | AOO | – | DDD | – |
| VOO |
| AAI |
| VVI |
| Lower Rate Limit | 30-50 ppm | 5 ppm | 60 ppm | 8 ms |
| 50-90 ppm | 1 ppm |
| 90-175 ppm | 5 ppm |
| Upper Rate Limit | 50-175 ppm | 5 ppm | 120 ppm | 8 ms |
| A or V Pulse Amplitude Regulated | Off, 0.5 – 3.2 V | 0.1 V | 3.5V | 12 % |
| 3.5 – 7.0 V | 0.5 V |
| A or V pulse Width | 0.05 ms | – | 0.4 ms | 0.2 ms |
| 0.1 – 1.9 ms | 0.1 ms |
| Ventricular Refractory Period | 150 – 500 ms | 10 ms | 320 ms | 8 ms |
| Atrial Refractory Period | 150 – 500 ms | 10 ms | 250 ms | 8 ms |

## Design decisions

A subsystem is used to map the pins inputting data for use in the program to their names as defined in Table 1 of Pacemaker Shield Explained. Another subsystem is used to map the output pins to their names in the same table. This allows the variables used within the program to be called “ATR\_CMP\_DETECT” instead of D0 everywhere, which makes it more readable, and hides the software from the hardware.

Constants are used for the programmable parameters to allow their values to be easily changed between simulation runs. In future, these values will be provided by the DCM through serial communication.

A stateflow diagram is used to branch into the correct operating modes based on the inputs given. Further stateflows within are used to implement the various states required for each pacing mode.

## Simulink diagram and testing

Pt2:

1. List likely changes to requirements
2. List all design decisions likely to change
3. For each module:
   1. Describe purpose
   2. List public functions and parameters
   3. Describe black box behaviour
   4. Describe global variables (state variables)
   5. List private functions in module
   6. Describe internal behaviour of functions

# Part 2: Future flexibility and modules

## 2.1 Requirements likely to change

In the future, more modes will need to be added. Subsystems are used to encapsulate the different operating modes to make it easier to add new ones.

## 2.2 Design decisions likely to change

The decision to use constants for user values will change in Assignment 2 as there will be serial communication between the DCM and Simulink model, allowing those values to be provided through a GUI.

## 2.3 MIS and MID

### 2.3.1 Module Interface Specification

Describes the purpose, public functions, and black-box behaviour of each function.

#### 2.3.1.1 Input Pins Subsystem

Maps MCU input pins to more descriptive internal variables to enable hardware hiding.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Type | Output | Type |
| PTC16 (D0) | bool | ATR\_CMP\_DETECT | bool |
| PTC17 (D1) | bool | VENT\_CMP\_DETECT | bool |

#### 2.3.1.2 Output Pins Subsystem

Maps internal variables to MCU output pins to enable hardware hiding.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Type | Output | Type |
| PACE\_CHARGE\_CTRL | bool | PTB9 (D2) | bool |
| Z\_ATR\_CTRL | bool | PTB23 (D4) | bool |
| PACING\_REF\_PWM | single | PTA2 (D5) | single |
| Z\_VENT\_CTRL | bool | PTC3 (D7) | bool |
| ATR\_PACE\_CTRL | bool | PTA0 (D6) | bool |
| VENT\_PACE\_CTRL | bool | PTC4 (D9) | bool |
| PACE\_GND\_CTRL | bool | PTD0 (D10) | bool |
| ATR\_GND\_CTRL | bool | PTD2 (D11) | bool |
| VENT\_GND\_CTRL | bool | PTD3 (D12) | bool |

#### 2.3.1.3 Main Stateflow

Allows the program to branch to different operating modes. Organizes sub-stateflows and is extensible for future modes.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Type | Output | Type |
| ATR\_CMP\_DETECT | bool | PACE\_CHARGE\_CTRL | bool |
| VENT\_CMP\_DETECT | bool | Z\_ATR\_CTRL | bool |
| Mode | int8 | PACING\_REF\_PWM | single |
| Rate | single | Z\_VENT\_CTRL | bool |
| Atrial\_Amplitude | single | ATR\_PACE\_CTRL | bool |
| Atrial\_Pulse\_Width | single | VENT\_PACE\_CTRL | bool |
| Ventricular\_Amplitude | single | PACE\_GND\_CTRL | bool |
| Ventricular\_Pulse\_Width | single | ATR\_GND\_CTRL | bool |
| ARP | single | VENT\_GND\_CTRL | bool |
| VRP | single |  |  |

#### 2.3.1.4 AOO Stateflow

Implements the AOO operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create paces.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Type | Output | Type |
| Atrial\_Amplitude | single | PACE\_CHARGE\_CTRL | bool |
| Atrial\_Pulse\_Width | single | Z\_ATR\_CTRL | bool |
| Rate\_ms | single | PACING\_REF\_PWM | single |
|  |  | Z\_VENT\_CTRL | bool |
|  |  | ATR\_PACE\_CTRL | bool |
|  |  | VENT\_PACE\_CTRL | bool |
|  |  | PACE\_GND\_CTRL | bool |
|  |  | ATR\_GND\_CTRL | bool |
|  |  | VENT\_GND\_CTRL | bool |

#### 2.3.1.5 VOO Stateflow

Implements the VOO operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create paces.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Type | Output | Type |
| Ventricular\_Amplitude | single | PACE\_CHARGE\_CTRL | bool |
| Ventricular\_Pulse\_Width | single | Z\_ATR\_CTRL | bool |
| Rate\_ms | single | PACING\_REF\_PWM | single |
|  |  | Z\_VENT\_CTRL | bool |
|  |  | ATR\_PACE\_CTRL | bool |
|  |  | VENT\_PACE\_CTRL | bool |
|  |  | PACE\_GND\_CTRL | bool |
|  |  | ATR\_GND\_CTRL | bool |
|  |  | VENT\_GND\_CTRL | bool |

#### 2.3.1.6 AAI Stateflow

Implements the AAI operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create and respond to paces.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Type | Output | Type |
| Atrial\_Amplitude | single | PACE\_CHARGE\_CTRL | bool |
| Atrial\_Pulse\_Width | single | Z\_ATR\_CTRL | bool |
| Rate\_ms | single | PACING\_REF\_PWM | single |
| ATR\_CMP\_DETECT | bool | Z\_VENT\_CTRL | bool |
| ARP | single | ATR\_PACE\_CTRL | bool |
|  |  | VENT\_PACE\_CTRL | bool |
|  |  | PACE\_GND\_CTRL | bool |
|  |  | ATR\_GND\_CTRL | bool |
|  |  | VENT\_GND\_CTRL | bool |

#### 2.3.1.7 VVI Stateflow

Implements the AAI operating mode, taking in appropriate programmable parameters, and controlling its outputs accordingly to create and respond to paces.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Type | Output | Type |
| Ventricular\_Amplitude | single | PACE\_CHARGE\_CTRL | bool |
| Ventricular\_Pulse\_Width | single | Z\_ATR\_CTRL | bool |
| Rate\_ms | single | PACING\_REF\_PWM | single |
| VENT\_CMP\_DETECT | bool | Z\_VENT\_CTRL | bool |
| VRP | single | ATR\_PACE\_CTRL | bool |
|  |  | VENT\_PACE\_CTRL | bool |
|  |  | PACE\_GND\_CTRL | bool |
|  |  | ATR\_GND\_CTRL | bool |
|  |  | VENT\_GND\_CTRL | bool |

### 2.3.2 Module Internal Design

Describes the state variables, private functions, and internal behaviour of all functions.

#### 2.3.2.1 Input Pins Subsystem

#### 2.3.2.2 Output Pins Subsystem

#### 2.3.2.3 Main Stateflow

#### 2.3.2.4 AOO Stateflow

#### 2.3.2.5 VOO Stateflow

#### 2.3.2.6 AAI Stateflow

#### 2.3.2.7 VVI Stateflow