IBEHS 3A03 Assignment #2:

LTI System Responses and Convolution

David Nasri (400323835)

Daniel Wong (400308249)

Kevin Zhang (400319666)

Team 16

October 26th, 2022

[**PURPOSE**](#_izo5lmpum1z2) **2**

[**METHODOLOGY**](#_wn6tj5j1tm86) **2**

[Part I) System Output When Input Is Unit Impulse Function](#_ej4c55aagh1h) 2

[Part II) System Output When Input Is Unit Step Function](#_4euadb731lu7) 2

[Part III) Proof: Unit Step Function Output = Cumulative Sum of Unit Impulse Response](#_ablryc612c6p) 3

[Part IV) Proof: Unit Impulse Function Output = First Difference of Unit Step Function Response](#_ke8cvtumjlqm) 3

[Part V) System Output For ECG Signals](#_5ujk0btptcjv) 4

[Part VI) System Output For Respiration Signals](#_i3dy01nr1x9o) 4

[Part VII) Proof: Output of Signals in V & VI = Convolution of Signals in V & VI with Impulse Response computed in I](#_31o84rbmidsw) 5

[BONUS) Formal Logical Test for III, IV and VII](#_a02ymlwp9vga) 6

[**RESULTS**](#_7q4q4ak6x5ml) **10**

[System A](#_qpqyzuoojmmf) 10

[Part I)](#_kjkmy64phuww) 10

[Part II)](#_fgq46wv77xb1) 10

[Part III)](#_ycwmz2frg34q) 11

[Part IV)](#_njlb33p6itmd) 11

[Part V)](#_k2ra8q23p9ma) 12

[Part VI)](#_v8eogsmuy93u) 12

[Part VII)](#_dnsy9sg7vd4o) 13

[System B](#_slbz8xrurtup) 13

[Part I)](#_wvyvop8yrd3p) 14

[Part II)](#_1i3a95nhap1) 14

[Part III)](#_sm59zieh451l) 15

[Part IV)](#_k7ofnd2f7oar) 15

[Part V)](#_qcgi7xlc9y3j) 16

[Part VI)](#_xvahy9jctbwm) 17

[Part VII)](#_dclhqp20o34g) 17

[System C](#_7b1kaiwqsc2h) 18

[Part I)](#_vew6kmgued7) 19

[Part II)](#_tyos12flbro0) 19

[Part III)](#_3amc42g8zg7u) 20

[Part IV)](#_aj87s3g2b1d4) 20

[Part V)](#_qurnc7ucidoh) 21

[Part VI)](#_b26xwwt953tv) 21

[Part VII)](#_jtrrtzz0jcjj) 22

[Bonus](#_1rp0i6vbgwf4) 22

[Part III)](#_d237vws0x2tu) 23

[Part IV)](#_49c8fsrjwgin) 23

[Part VII)](#_uw04iyatb0al) 23

[Summary of Findings](#_qex4i2dysf2q) 24

[**CONCLUSION**](#_p80j6zkapl9x) **24**

[**REFERENCES**](#_lffflox0qrw4) **25**

# PURPOSE

Based on knowledge about mathematical relationships between the impulse and step responses of linear time-invariant (LTI) systems, and how convolving an LTI system's impulse response with any input can yield the system's output, we tested a series of theories for three different black-box discrete-time LTI systems.

# METHODOLOGY

## Part I) System Output When Input Is Unit Impulse Function

For Part I, we obtained the unit impulse response **y** by inputting the unit impulse function **x** with domain **n**. We then graphed the unit impulse response **y** using *figure*. Table 1 describes the variables used for this task.

**Table 1.** Variables used in Part I in order of implementation.

| **File** | **Variable** | **Assignment** | **Description** |
| --- | --- | --- | --- |
| ALL | n | -4:15 | discrete step |
| x | [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] | unit impulse function |
| SystemA | y | ltisystemA(n,x) | unit impulse response |
| SystemB | ltisystemB(n,x) |
| SystemC | ltisystemC(n,x) |

## Part II) System Output When Input Is Unit Step Function

For Part II, the unit step response **y** is gained after inputting the unit step function **x** into the systems, with domain **n**. We then graphed the unit step response **y** using *figure*. Table 2 describes the variables used for this task.

**Table 2.** Variables used in Part II in order of implementation.

| **File** | **Variable** | **Assignment** | **Description** |
| --- | --- | --- | --- |
| ALL | n | -4:15 | discrete step |
| x | n>=0 | unit step function |
| SystemA | y | ltisystemA(n,x) | unit step response |
| SystemB | ltisystemB(n,x) |
| SystemC | ltisystemC(n,x) |

## Part III) Proof: Unit Step Function Output = Cumulative Sum of Unit Impulse Response

For Part III, we found the unit impulse response **y1** and the unit step response **y2** by feeding the systems with their respective inputs (unit impulse function **x1** and unit step function **x2** with domains **n1** and **n2**). We used *cumsum( )* to find the cumulative sum of the unit impulse response (**sum**)to determine whether this equated to the unit step response **y2**. The cumulative sum of the unit impulse response and the unit step response were graphed using *figure*. Table 3 describes the variables used for this task.

**Table 3.** Variables used in Part III in order of implementation.

| **File** | **Variable** | **Assignment** | **Description** |
| --- | --- | --- | --- |
| ALL | n1 | -4:15 | discrete step |
| x1 | [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] | unit impulse function |
| n2 | -4:15 | discrete step |
| x2 | n2>=0 | unit step function |
| SystemA | y1 | ltisystemA(n1,x1) | unit impulse response |
| SystemB | ltisystemB(n1,x1) |
| SystemC | ltisystemC(n1,x1) |
| SystemA | y2 | ltisystemA(n2,x2) | unit step response |
| SystemB | ltisystemB(n2,x2) |
| SystemC | ltisystemC(n2,x2) |
| ALL | sum | cumsum(y1) | cumulative sum of the impulse response |

## Part IV) Proof: Unit Impulse Function Output = First Difference of Unit Step Function Response

For Part IV, we used unit impulse function **x1** (domain **n1**) and unit step function **x2** (domain **n2**) as input functions. In return, we receive the unit impulse response **y1** and unit step response **y2** after calling the systems. Using *diff( )*, we then found the first difference of the unit step response (**difference**) to determine whether this equated to the unit impulse response **y1**. The unit impulse response **y1** and the first difference of the unit step response (**difference**) were graphed using *figure*. Table 4 describes the variables used for this task.

**Table 4.** Variables used in Part IV in order of implementation.

| **File** | **Variable** | **Assignment** | **Description** |
| --- | --- | --- | --- |
| ALL | n1 | -4:15 | discrete step |
| x1 | [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] | unit impulse function |
| n2 | -4:15 | discrete step |
| x2 | n2>=0 | unit step function |
| SystemA | y1 | ltisystemA(n1,x1) | unit impulse response |
| SystemB | ltisystemB(n1,x1) |
| SystemC | ltisystemC(n1,x1) |
| SystemA | y2 | ltisystemA(n2,x2) | unit step response |
| SystemB | ltisystemB(n2,x2) |
| SystemC | ltisystemC(n2,x2) |
| ALL | difference | diff(y2) | first difference of the unit step response |

## Part V) System Output For ECG Signals

For Part V, we used *load( )* so the systems could read the ECG data from *ECG\_assignment2.mat* as the input **ecg\_x**. We selected a discrete step **ecg\_n** that aligned with the input function **ecg\_x** and called each system to obtain the ECG response **y**, which was then graphed using *figure*. Table 5 describes the variables used for this task.

**Table 5.** Variables used in Part V in order of implementation.

| **File** | **Variable** | **Assignment** | **Description** |
| --- | --- | --- | --- |
| ALL | ecg\_n | 0:length(ecg\_x)-1 | discrete step |
| ecg\_x | x in *ECG\_assignment2.mat* | ECG values |
| SystemA | y | ltisystemA(ecg\_n, ecg\_x) | ECG response |
| SystemB | ltisystemB(ecg\_n, ecg\_x) |
| SystemC | ltisystemC(ecg\_n, ecg\_x) |

## Part VI) System Output For Respiration Signals

For Part VI, the systems read the respiration data from *respiration\_assignment2.mat* as the input **resp\_x**. We selected a discrete step **resp\_n** that aligned with the input function **resp\_x** and called each system to obtain the respiration response **y**, which was then graphed using *figure*. Table 6 describes the variables used for this task.

**Table 6.** Variables used in Part VI in order of implementation.

| **File** | **Variable** | **Assignment** | **Description** |
| --- | --- | --- | --- |
| ALL | resp\_n | 0:length(resp\_x)-1 | discrete step |
| resp\_x | x in *respiration\_assignment2.mat* | respiration values |
| SystemA | y | ltisystemA(resp\_n, resp\_x) | respiration response |
| SystemB | ltisystemB(resp\_n, resp\_x) |
| SystemC | ltisystemC(resp\_n, resp\_x) |

## Part VII) Proof: Output of Signals in V & VI = Convolution of Signals in V & VI with Impulse Response computed in I

For Part VII, the systems read the ECG and respiration data as two input functions (**ecg\_x** and **resp\_x**). The two input functions were convolved with the unit impulse response **y**, resulting in **conv\_ecg** and **conv\_resp**. Using *figure,* graphs were made to compare the convolved systems with their respective responses. From lecture, we learned convolution: inputting any function into a unit impulse response function with no initial conditions will yield the respective response from the input function [1]. We applied the same theory to Part VII. Therefore, the unit impulse response convolved with ECG data has the same graphical output as the ECG output. Similarly, the impulse response convolved with respiratory data has the same graphical output as the respiration output. Table 7 describes variables used for this task.

**Table 7.** Variables used in Part VII in order of implementation.

| **File** | **Variable** | **Assignment** | **Description** |
| --- | --- | --- | --- |
| ALL | n | -4:15 | discrete step |
| x | [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] | unit impulse function |
| ecg\_n | 0:length(ecg\_x)-1 | ECG discrete step |
| ecg\_x | x in ECG\_assignment2.mat | ECG values |
| resp\_n | 0:length(resp\_x)-1 | resp. discrete step |
| resp\_x | x in *respiration\_assignment2.mat* | resp. values |
| SystemA | y | ltisystemA(n, x) | unit impulse response |
| SystemB | ltisystemB(n, x) |
| SystemC | ltisystemC(n, x) |
| SystemA | ecg\_y | ltisystemA(ecg\_n, ecg\_x) | ECG response |
| SystemB | ltisystemB(ecg\_n, ecg\_x) |
| SystemC | ltisystemC(ecg\_n, ecg\_x) |
| SystemA | resp\_y | ltisystemA(resp\_n, resp\_x) | resp. response |
| SystemB | ltisystemB(resp\_n, resp\_x) |
| SystemC | ltisystemC(resp\_n, resp\_x) |
| ALL | conv\_ecg | conv(ecg\_x,y) | impulse response convolved with ECG data |
| conv\_resp | conv(resp\_x,y) | impulse response convolved with resp. data |

## BONUS) Formal Logical Test for III, IV and VII

All BONUS tasks were performed in the *bonus.m* file. For BONUS Part III, we implemented the same code as Part III and compared the similarity by using *isequal( )* which determines array equality. We also rounded the unit step response and the cumulative sum to 5 decimal digits to account for MATLAB’s finite precision mathematics. We examined the MATLAB Workspace, to determine the value of this variable equated to the *isequal( )* command. A value of 1 means “true” while a value of 0 represents “false.” Table 8 describes the variables used for this task.

**Table 8.** Variables used in BONUS Part III.

| **Variable** | **Assignment** | **Description** |
| --- | --- | --- |
| n | -4:15 | discrete step |
| x | [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] | unit impulse function |
| y1A | ltisystemA(n,x) | unit impulse response from system A |
| y1B | ltisystemB(n,x) | unit impulse response from system B |
| y1C | ltisystemC(n,x) | unit impulse response from system C |
| n2 | -4:15 | discrete step |
| x2 | n>=0 | unit step function |
| y2A | ltisystemA(n2,x2) | unit step response from system A |
| y2B | ltisystemB(n2,x2) | unit step response from system B |
| y2C | ltisystemC(n2,x2) | unit step response from system C |
| sumA | cumsum(y1A) | cumulative sum of unit impulse response (System A) |
| sumB | cumsum(y1B) | cumulative sum of unit impulse response (System B) |
| sumC | cumsum(y1C) | cumulative sum of unit impulse response (System C) |
| Q3a | isequal(round(y2A,5),round(sumA,5)) | verifying if unit step response = cumulative sum of unit impulse response (System A) |
| Q3b | isequal(round(y2B,5),round(sumB,5)) | verifying if unit step response = cumulative sum of unit impulse response (System B) |
| Q3c | isequal(round(y2C,5),round(sumB,5)) | verifying if unit step response = cumulative sum of unit impulse response (System C) |

A similar methodology was applied for BONUS Part IV, where we compared array similarity by using *isequal( )*. Table 9 describes the variables used for this task.

**Table 9.** Variables used in BONUS Part IV.

| **Variable** | **Assignment** | **Description** |
| --- | --- | --- |
| n | -4:15 | discrete step |
| x | [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] | unit impulse function |
| y1A | ltisystemA(n,x) | unit impulse response from system A |
| y1B | ltisystemB(n,x) | unit impulse response from system B |
| y1C | ltisystemC(n,x) | unit impulse response from system C |
| n2 | -4:15 | discrete step |
| x2 | n>=0 | unit step function |
| y2A | ltisystemA(n2,x2) | unit step response from system A |
| y2B | ltisystemB(n2,x2) | unit step response from system B |
| y2C | ltisystemC(n2,x2) | unit step response from system C |
| diffA | diff(y2A) | first difference of the unit step response (System A) |
| diffB | diff(y2B) | first difference of the unit step response (System B) |
| diffC | diff(y2C) | first difference of the unit step response (System C) |
| Q4a | isequal(round(y1A,5),round([0 diffA],5)) | verifying if unit step response = first difference of impulse response (System A) |
| Q4b | isequal(round(y1B,5),round([0 diffB],5)) | verifying if unit step response = first difference of impulse response (System B) |
| Q4c | isequal(round(y1B,5),round([0 diffB],5)) | verifying if unit step response = first difference of impulse response (System C) |

For BONUS Part VII, we also used the same methodology of comparing similarity by using *isequal( )*. We also applied a time shift and truncated each convolution so that the discrete steps values were consistent with the discrete steps values of their ECG/respiration response. For ECG-related verifications, we rounded variables to 2 decimal digits, while for respiratory-related verifications, we rounded variables to 0 decimal digits. This was done because further accounting for MATLAB’s finite precision mathematics was required. Table 10 describes the variables used for this task.

**Table 10.** Variables used in BONUS Part VII.

| **Variable** | **Assignment** | **Description** |
| --- | --- | --- |
| ecg\_n | 0:length(ecg\_x)-1 | discrete step |
| ecg\_x | x in ECG\_assignment2.mat | ECG values |
| ecg\_yA | ltisystemA(ecg\_n, ecg\_x) | ECG response from system A |
| ecg\_yB | ltisystemB(ecg\_n, ecg\_x) | ECG response from system B |
| ecg\_yC | ltisystemC(ecg\_n, ecg\_x) | ECG response from system C |
| n | -4:15 | discrete step |
| x | [0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0] | unit step function |
| yA | ltisystemA(n,x) | impulse response from system A |
| yB | ltisystemB(n,x) | impulse response from system B |
| yC | ltisystemC(n,x) | impulse response from system C |
| conv\_ecgA | conv(ecg\_x,yA)(5:length(ecg\_yA)+4) | impulse response (System A) convolved with the ECG data |
| conv\_ecgB | conv(ecg\_x,yB)(5:length(ecg\_yB)+4) | impulse response (System B) convolved with the ECG data |
| conv\_ecgC | conv(ecg\_x,yC)(5:length(ecg\_yC)+4) | impulse response (System C) convolved with the ECG data |
| Q7ECGa | isequal(round(ecg\_yA,2),round(conv\_ecgA,2)) | verifying if ECG response = convolution of unit impulse response with ECG data |
| Q7ECGb | isequal(round(ecg\_yB,2),round(conv\_ecgB,2)) | verifying if ECG response = convolution of unit impulse response with ECG data |
| Q7ECGc | isequal(round(ecg\_yC,2),round(conv\_ecgC,2)) | verifying if ECG response = convolution of unit impulse response with ECG data |
| resp\_n | 0:length(resp\_x)-1 | discrete step |
| resp\_x | x in respiration\_assignment2.mat | Respiration values |
| resp\_yA | ltisystemA(resp\_n, resp\_x) | Respiration response from system A |
| resp\_yB | ltisystemB(resp\_n, resp\_x) | Respiration response from system B |
| resp\_yC | ltisystemC(resp\_n, resp\_x) | Respiration response from system C |
| conv\_respA | conv(resp\_x,yA)(5:length(resp\_yA)+4) | impulse response (System A) convolved with respiration data |
| conv\_respB | conv(resp\_x,yB)(5:length(resp\_yA)+4) | impulse response (System B) convolved with respiration data |
| conv\_respC | conv(resp\_x,yC)(5:length(resp\_yA)+4) | impulse response (System C) convolved with respiration data |
| Q7RESPa | isequal(round(resp\_yA),round(conv\_respA)) | verifying if respiration response = convolution of unit impulse response with respiration data |
| Q7RESPb | isequal(round(resp\_yB),round(conv\_respB)) | verifying if respiration response = convolution of unit impulse response with respiration data |
| Q7RESPc | isequal(round(resp\_yC),round(conv\_respC)) | verifying if respiration response = convolution of unit impulse response with respiration data |

# RESULTS

## System A

### Part I)

|  |
| --- |

**Figure 1.** SystemA’s response to an impulse input.

### Part II)

|  |
| --- |

**Figure 2.** SystemA’s response to a unit step input.

### Part III)

|  |
| --- |

**Figure 3.** SystemA’s unit step response and the cumulative sum of SystemA’s impulse response.

### Part IV)

|  |
| --- |

**Figure 4.** SystemA’s impulse response and the first difference of SystemA’s unit step response (shifted right by 1 to align with the impulse response).

### Part V)

|  |
| --- |

**Figure 5.** SystemA’s response to an ECG signal input.

### Part VI)

|  |
| --- |

**Figure 6.** SystemA’s response to a respiration signal input.

### Part VII)

|  |
| --- |
|  |

**Figure 7.** ECG and respiration signals convolved with SystemA’s impulse response compared with SystemA’s response to ECG and respiration signal inputs.

As seen in Figure 1, the impulse response of SystemA is a finite time response. Furthermore Figures 3, 4, and 7 confirms that SystemA follows the expected properties of a LTI system. Figure 3 shows that the unit step response is equivalent to the cumulative sum of the impulse response, Figure 4 shows that the impulse response is equivalent to the first difference of the unit step response, and Figure 7 shows that any response is equivalent to the convolution of the impulse response with the input function.

## System B

### Part I)

|  |
| --- |

**Figure 8.** SystemB’s response to an impulse input.

### Part II)

|  |
| --- |

**Figure 9.** SystemB’s response to a unit step input.

### Part III)

|  |
| --- |

**Figure 10.** SystemB’s unit step response and the cumulative sum of SystemB’s impulse response.

### Part IV)

|  |
| --- |

**Figure 11.** SystemB’s impulse response and the first difference of SystemB’s unit step response (shifted right by 1 to align with the impulse response).

### 

### Part V)

|  |
| --- |

**Figure 12.** SystemB’s response to an ECG signal input.

### Part VI)

|  |
| --- |

**Figure 13.** SystemB’s response to a respiration signal input.

### Part VII)

|  |
| --- |
|  |

**Figure 14.** ECG and respiration signals convolved with SystemB’s impulse response compared with SystemB’s response to ECG and respiration signal inputs.

As seen in Figure 8, the impulse response of SystemB is a finite time response. Furthermore Figures 10, 11, and 14 confirms that SystemB follows the expected properties of a LTI system. Figure 10 shows that the unit step response is equivalent to the cumulative sum of the impulse response, Figure 11 shows that the impulse response is equivalent to the first difference of the unit step response, and Figure 14 shows that any response is equivalent to the convolution of the impulse response with the input function.

## System C

### Part I)

|  |
| --- |

**Figure 15.** SystemC’s response to an impulse input.

### Part II)

|  |
| --- |

**Figure 16.** SystemC’s response to a unit step input.

### Part III)

|  |
| --- |

**Figure 17.** SystemC’s unit step response and the cumulative sum of SystemC’s impulse response.

### Part IV)

|  |
| --- |

**Figure 18.** SystemC’s impulse response and the first difference of SystemC’s unit step response (shifted right by 1 to align with the impulse response).

### Part V)

|  |
| --- |

**Figure 19.** SystemC’s response to an ECG signal input.

### Part VI)

|  |
| --- |

**Figure 20.** SystemC’s response to a respiration signal input.

### Part VII)

|  |
| --- |

**Figure 21.** ECG and respiration signals convolved with SystemC’s impulse response compared with SystemC’s response to ECG and respiration signal inputs.

As shown in Figure 15 (and verified by examining values of y), the impulse response of System C does not equal zero at n ≥ 0. This indicates that System C has an infinite time response. Furthermore, Figures 17, 18, and 21 confirm that SystemC follows the expected properties of an LTI system: Figure 17 shows that the unit step response is equivalent to the cumulative sum of the impulse response, Figure 18 shows that the impulse response is equivalent to the first difference of the unit step response and Figure 21 shows that any response is equivalent to the convolution of the impulse response with the input function.

## Bonus

No figures were required to produce for the formal logical test. To verify the proofs of III, IV and VII for each system, the function **isequal( )** was used. The following figures were captured from the MATLAB workspace. Rounding was implemented to account for MATLAB’s finite precision mathematics.

### Part III)

|  |
| --- |

**Figure 22.** Output of true (i.e., equals 1) for each system that the unit step response equals the cumulative sum of the impulse response.

### Part IV)

|  |
| --- |

**Figure 23.** Output of true (i.e., equals 1) for each system that the impulse response equals the first difference of the unit step response.

### Part VII)

|  |
| --- |

**Figure 24.** Output of true (i.e., equals 1) for each system that the output of signals in V) and VI) is equivalent to the convolution of signals in V) and VI) with the impulse response of the system computed in I).

## **Summary of Findings**

Table 11 below summarizes our findings after testing theories on the three systems.

**Table 11.** Properties of the LTI black-box systems A, B, and C.

| **System** | **Finite or Infinite?** |
| --- | --- |
| System A | Finite |
| System B | Finite |
| System C | Infinite |

# CONCLUSION

We successfully tested theoretical concepts on three black-box discrete-time LTI systems using MATLAB 2022a. In some tests, we applied our understanding of LTI systems by testing discrete-time signals from ECG and respiration data. We finally conducted a formal logical test for specific proofs, while taking into account the finite precision mathematics of MATLAB 2020a.

# REFERENCES

1. I. Bruce, “Lecture 07: Computing the Discrete-Time Convolution,” 20-Sep-2022.