

IMP 301

Progress Test 2

Lê Đức Anh - Anhldhe150197 - AI1602

**Q1 (1.5pts) Implement Ideal band reject/pass filter.**

**1.1 ideal band reject filter**

| Source code | **def** IBF**(**F**,** C0**,** W**):**  M**,**N **=** F**.**shape  H **=** np**.**zeros**((**M**,** N**),** dtype**=**np**.**complex128**)**  **for** u **in** **range(**M**):**  **for** v **in** **range(**N**):**  D **=** np**.**sqrt**((**u**-**M**/**2**)\*\***2 **+** **(**v**-**N**/**2**)\*\***2**)**  **if** D **<=** C0 **-** W**/**2 **or** D **>=** C0 **+** W**/**2**:**  H**[**u**,**v**]** **=** 1  **else:**  H**[**u**,**v**]** **=** 0  **return** H |
| --- | --- |
| Input | Output |
|  | |

**1.2 ideal high pass filter**

| Source code | **def** IHPF**(**F**,** D0**):**  M**,**N **=** F**.**shape  H **=** np**.**zeros**((**M**,** N**))**  **for** u **in** **range(**M**):**  **for** v **in** **range(**N**):**  D **=** np**.**sqrt**((**u **-** M **/** 2**)** **\*\*** 2 **+** **(**v **-** N **/** 2**)** **\*\*** 2**)**  **if** D **>** D0**:**  H**[**u**,** v**]** **=** 1  **else:**  H**[**u**,** v**]** **=** 0  G **=** F **\*** H  **return** G |
| --- | --- |
| Input | Output |
|  | |

**1.3 ideal low pass filter**

| Source code | **def** ILPF**(**F**,** D0**):**  M**,**N **=** F**.**shape  H **=** np**.**zeros**((**M**,** N**))**  **for** u **in** **range(**M**):**  **for** v **in** **range(**N**):**  D **=** np**.**sqrt**((**u **-** M **/** 2**)** **\*\*** 2 **+** **(**v **-** N **/** 2**)** **\*\*** 2**)**  **if** D **<=** D0**:**  H**[**u**,** v**]** **=** 1  **else:**  H**[**u**,** v**]** **=** 0  G **=** F **\*** H  **return** G |
| --- | --- |
| Input | Output |
|  | |

**Q2 (1.5pts) butterworth band reject/pass filter.**

**2.1 butterworth band reject filter**

| Source code | **def** BBF**(**F**,** C0**,** W**,** n**):**  M**,**N **=** F**.**shape  H **=** np**.**zeros**((**M**,** N**),** dtype**=**np**.**complex128**)**  **for** u **in** **range(**M**):**  **for** v **in** **range(**N**):**  D **=** np**.**sqrt**((**u**-**M**/**2**)\*\***2 **+** **(**v**-**N**/**2**)\*\***2**)**  H**[**u**,**v**]** **=** 1**/(**1 **+** **((**D**\***W**)/(**D**\*\***2 **-** C0**\*\***2**))\*\*(**2**\***n**))**  **return** H |
| --- | --- |
| Input | Output |
|  | |

**2.2 butterworth band high pass filter**

| Source code | **def** BHPF**(**F**,**D0**):**  M**,**N **=** F**.**shape  H **=** np**.**zeros**((**M**,** N**))**  **for** u **in** **range(**M**):**  **for** v **in** **range(**N**):**  D **=** np**.**sqrt**((**u **-** M **/** 2**)** **\*\*** 2 **+** **(**v **-** N **/** 2**)** **\*\*** 2**)**  H**[**u**,** v**]** **=** 1**/(**1**+(**D0**/**D**)\*\***2**)**  G **=** F **\*** H  **return** G |
| --- | --- |
| Input | Output |
|  | |

**2.3 butterworth band low pass filter**

| Source code | **def** BLPF**(**F**,**D0**):**  M**,**N **=** F**.**shape  H **=** np**.**zeros**((**M**,** N**))**  **for** u **in** **range(**M**):**  **for** v **in** **range(**N**):**  D **=** np**.**sqrt**((**u **-** M **/** 2**)** **\*\*** 2 **+** **(**v **-** N **/** 2**)** **\*\*** 2**)**  H**[**u**,** v**]** **=** 1**/(**1**+(**D**/**D0**)\*\***2**)**  G **=** F **\*** H  **return** G |
| --- | --- |
| Input | Output |
|  | |

**Q3 (1.5pts) guassian band reject/pass filter.**

**3.1 guassian band reject filter**

| Source code | **def** GBF**(**F**,** C0**,** W**):**  M**,**N **=** F**.**shape  H **=** np**.**zeros**((**M**,** N**),** dtype**=**np**.**complex128**)**  **for** u **in** **range(**M**):**  **for** v **in** **range(**N**):**  D **=** np**.**sqrt**((**u**-**M**/**2**)\*\***2 **+** **(**v**-**N**/**2**)\*\***2**)**  H**[**u**,**v**]** **=** 1 **-** np**.**exp**(-**1**/**2**\*((**D**\*\***2 **-** C0**\*\***2**)/(**D**\***W**))\*\***2**)**  **return** H |
| --- | --- |
| Input | Output |
|  | |

**3.2 gaussian band high pass filter**

| Source code | **def** GHPF**(**F**,**D0**):**  M**,**N **=** F**.**shape  H **=** np**.**zeros**((**M**,** N**))**  **for** u **in** **range(**M**):**  **for** v **in** **range(**N**):**  D **=** np**.**sqrt**((**u **-** M **/** 2**)** **\*\*** 2 **+** **(**v **-** N **/** 2**)** **\*\*** 2**)**  H**[**u**,** v**]** **=** 1 **-** np**.**exp**(-((**D**/**D0**)\*\***2**))**  G **=** F **\*** H  **return** G |
| --- | --- |
| Input | Output |
|  | |

**3.3 gaussian band low pass filter**

| Source code | **def** GLPF**(**F**,**D0**):**  M**,**N **=** F**.**shape  H **=** np**.**zeros**((**M**,** N**))**  **for** u **in** **range(**M**):**  **for** v **in** **range(**N**):**  D **=** np**.**sqrt**((**u **-** M **/** 2**)** **\*\*** 2 **+** **(**v **-** N **/** 2**)** **\*\*** 2**)**  H**[**u**,** v**]** **=** np**.**exp**(-((**D**/**D0**)\*\***2**))**  G **=** F **\*** H  **return** G |
| --- | --- |
| Input | Output |
|  | |