



Electrical Engineering Department,
Fourth Year - Communications & Electronics.

EE 466 ANTENNA

Lab Assignment-2 (GUI)

| PREPARED BY | SECTION | SEAT.NO. |
|------------------------------|---------|----------|
| Mahmoud Mohamed Kamal Ismail | 7 | 250 |
| Mahmoud AbdElHady Mahmoud | 7 | 248 |
| Ahmed Mohamed Abdelakher | 1 | 37 |

➤ Contents

| | |
|--|----------------|
| 1. Interface design & Code | Page 4 |
| 1.1. Select | Page 4 |
| 1.2. Sketch | Page 5 |
| 1.3. Reset | Page 10 |
| 1.4. Special Cases | Page 11 |
| 2. Examples | Page 13 |
| 2.1. Part 1: linear antenna (dipole of general length) | Page 13 |
| 2.2. Part 2: Uniform linear antenna array (ULA) | Page 15 |
| 2.3. Part 3: Nonuniformly-Fed linear antenna array | Page 16 |
| A. Binomial Arrays | Page 16 |
| B. Dolph-Tschebysceff Arrays | Page 17 |
| 2.4. Special Case for dipole antenna | Page 19 |
| A. Infinite small dipole | Page 19 |
| B. Short dipole | Page 19 |
| C. $\lambda/2$ dipole | Page 20 |

➤ List of Figures

| | |
|---|-------------------|
| Figure 1: Interface design - linear antenna (dipole of general length). | Page 4 |
| Figure 2: Select. | Page 4 |
| Figure 3: Interface design - Uniform linear antenna array (ULA). | Page 5 |
| Figure 4: Interface design - Nonuniformly-Fed linear antenna array (Binomial Arrays). | Page 5 |
| Figure 5: Interface design - Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays). | Page 6 |
| Figure 6: Code – Select. | Page 6 |
| Figure 7: Sketch Button. | Page 7 |
| Figure 8: Code - Sketch. | Page 7 : Page 10 |
| Figure 9: Reset. | Page 10 |
| Figure 10: Code - Reset. | Page 10 |
| Figure 11: Special Cases. | Page 11 |
| Figure 12: Code - Special Cases. | Page 11 : Page 12 |
| Figure 13: Example 1 - Part 1: linear antenna (dipole of general length). | Page 13 |
| Figure 14: Example 2 (Errors) - Part 1: linear antenna (dipole of general length). | Page 14 |
| Figure 15: Example 1 - Part 2: Uniform linear antenna array (ULA). | Page 15 |

| | |
|--|---------|
| Figure 16: Example 2 (Errors) - Part 2: Uniform linear antenna array (ULA). | Page 16 |
| Figure 17: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays). | Page 16 |
| Figure 18: Example 2 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays). | Page 17 |
| Figure 19: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays). | Page 17 |
| Figure 20: Example 2 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays). | Page 18 |
| Figure 21: Infinite small dipole. | Page 19 |
| Figure 22: Short dipole. | Page 19 |
| Figure 23: $\lambda/2$ dipole. | Page 20 |

https://github.com/MahmoudFierro98/Antenna_Lab/

1. Interface design & Code

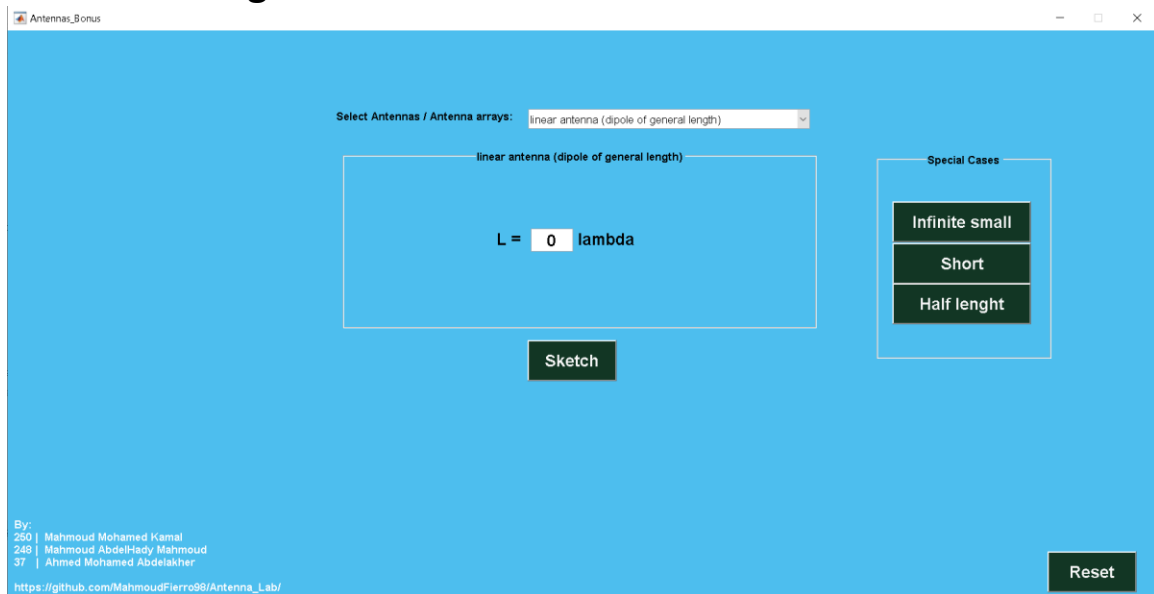


Figure 1: Interface design - linear antenna (dipole of general length).

1.1. Select

To choose any type of antennas you want.

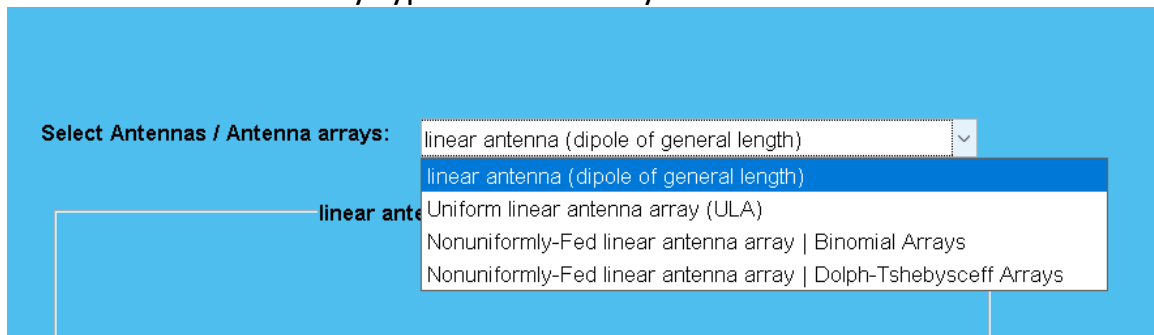


Figure 2: Select.

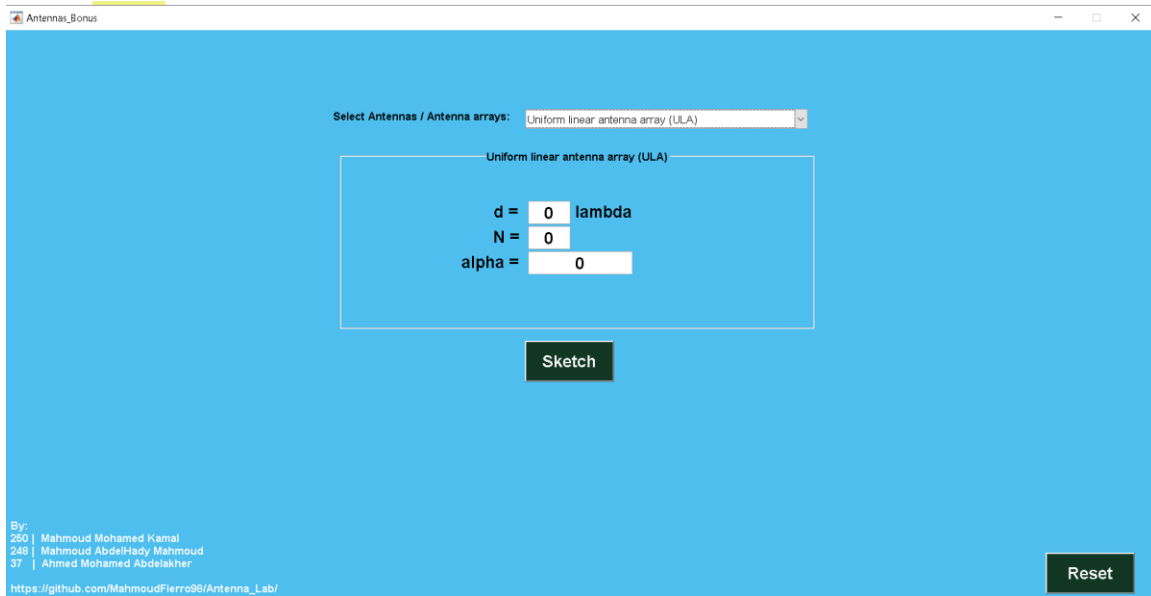


Figure 3: Interface design - Uniform linear antenna array (ULA).

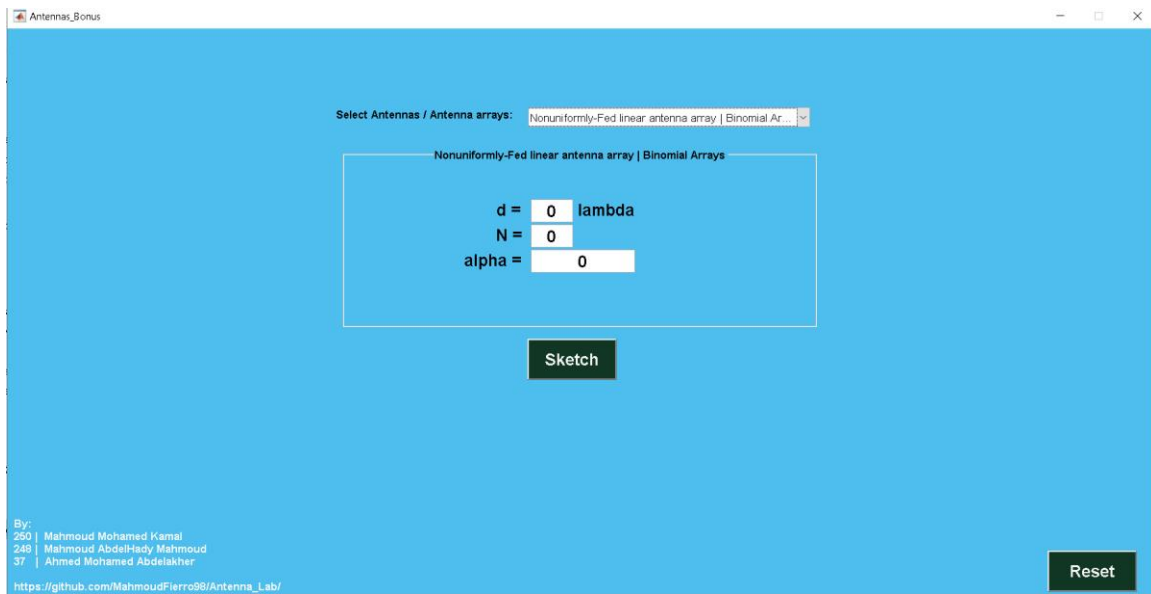


Figure 4: Interface design - Nonuniformly-Fed linear antenna array (Binomial Arrays).

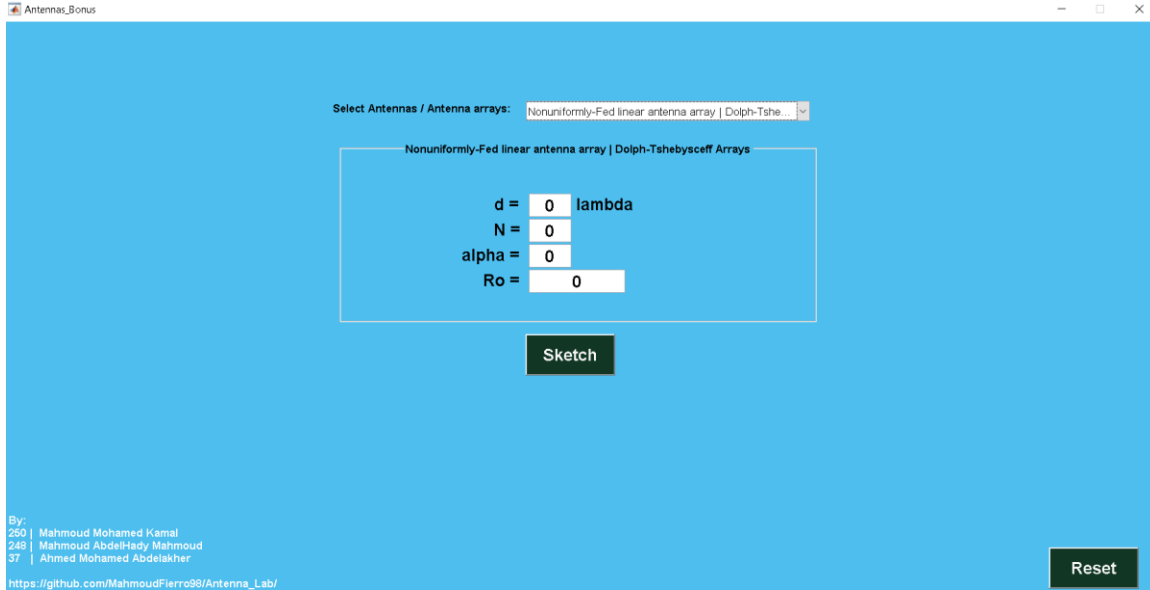


Figure 5: Interface design - Nonuniformly-Fed linear antenna array (Dolph-Tschebyscheff Arrays).

```

99 % --- Executes on selection change in select.
100 function select_Callback(hObject, eventdata, handles)
101     select = get(handles.select, 'Value');
102     set(handles.part1, 'Visible', 'off');
103     set(handles.part2, 'Visible', 'off');
104     set(handles.part3a, 'Visible', 'off');
105     set(handles.part3b, 'Visible', 'off');
106     set(handles.special_case, 'Visible', 'off');
107     set(handles.s1p, 'Visible', 'off');
108     set(handles.s2p, 'Visible', 'off');
109     set(handles.s3p, 'Visible', 'off');
110     set(handles.str, 'Visible', 'off');
111     set(handles.l_1, 'String', '0');
112     set(handles.d_2, 'String', '0');
113     set(handles.N_2, 'String', '0');
114     set(handles.alpha_2, 'String', '0');
115     set(handles.d_3a, 'String', '0');
116     set(handles.N_3a, 'String', '0');
117     set(handles.alpha_3a, 'String', '0');
118     set(handles.d_3b, 'String', '0');
119     set(handles.N_3b, 'String', '0');
120     set(handles.alpha_3b, 'String', '0');
121     set(handles.Ro_3b, 'String', '0');
122     switch select
123     case 1
124         set(handles.part1, 'Visible', 'on');
125         set(handles.special_case, 'Visible', 'on');
126     case 2
127         set(handles.part2, 'Visible', 'on');
128     case 3
129         set(handles.part3a, 'Visible', 'on');
130     case 4
131         set(handles.part3b, 'Visible', 'on');
132     end

```

Figure 6: Code – Select.

1.2. Sketch

Sketch

Figure 7: Sketch Button.

```
346 % --- Executes on button press in sketch_button.
347 function sketch_button_Callback(hObject, eventdata, handles)
348     Lambda = 1;
349     B = (2*pi)/Lambda;
350     select = get(handles.select, 'Value');
351     set(handles.special_case, 'Visible', 'off');
352     set(handles.s1p, 'Visible', 'off');
353     set(handles.s2p, 'Visible', 'off');
354     set(handles.s3p, 'Visible', 'off');
355     set(handles.str, 'Visible', 'off');
356     figure(1);
357     figure(2);
358     figure(3);
359     switch select
360     case 1
361         set(handles.special_case, 'Visible', 'on');
362         l_1 = get(handles.l_1, 'String');
363         Theta = linspace(-pi, pi, 350);
364         Phi = linspace(-2*pi, 2*pi, 350);
365         L = str2num(l_1) * Lambda;
366         En = abs((cos((B*L)/2) * cos(Theta)) - cos((B*L)/2)) ./ sin(Theta);
367         if (L < 0)
368             set(handles.str, 'Visible', 'on');
369             set(handles.str, 'ForegroundColor', 'r');
370             set(handles.str, 'String', 'Error :: (L >= 0)');
371             close.figure(1);
372             close.figure(2);
373             close.figure(3);
374         else
375             set(handles.str, 'Visible', 'on');
376             set(handles.str, 'ForegroundColor', 'g');
377             set(handles.str, 'String', 'Done');
378             figure(1);
379             polar(Theta, En);
380             view([90 90]);
381             title('linear antenna (dipole of general length) - The 2D pattern of the dipole');
382             Phi_3D = meshgrid(Phi);
383             Theta_3D = meshgrid(Theta);
384             En_3D = meshgrid(En);
385             X = En_3D.*sin(Theta_3D).*cos(Phi_3D);
386             Y = En_3D.*sin(Theta_3D).*sin(Phi_3D);
387             Z = En_3D.*cos(Theta_3D);
388             figure(2);
389             surf(X, Y, Z, 'EdgeColor', 'interp', 'FaceAlpha', 0.1);
390             axis vis3d;
391             axis equal;
392             lighting gouraud;
393             title('linear antenna (dipole of general length) - The 3D pattern of the dipole');
394             close.figure(3);
395         end
396     case 2
397         d_2 = get(handles.d_2, 'String');
398         N_2 = get(handles.N_2, 'String');
399         alpha_2 = get(handles.alpha_2, 'String');
400         d = str2num(d_2) * Lambda;
401         N = str2num(N_2);
402         alpha = str2num(alpha_2);
403         Gamma = linspace(-pi, pi, 6000);
404         Phi = linspace(-pi, pi, 6000);
405         ebsi = B*d*cos(Gamma) + alpha;
406         AF = abs(sin((N*ebsi)/2) ./ (N * sin(ebsi/2)));
407         if (d < 0)
408             set(handles.str, 'Visible', 'on');
409             set(handles.str, 'ForegroundColor', 'r');
410             set(handles.str, 'String', 'Error :: (d >= 0)');
411             close.figure(1);
```



```

412 -         close(ffigure(2));
413 -         close(ffigure(3));
414 -     elseif (N < 0)
415 -         set(handles.str, 'Visible','on');
416 -         set(handles.str, 'ForegroundColor','r');
417 -         set(handles.str, 'String', 'Error :: (N >= 0)');
418 -         close(ffigure(1));
419 -         close(ffigure(2));
420 -         close(ffigure(3));
421 -     else
422 -         set(handles.str, 'Visible','on');
423 -         set(handles.str, 'ForegroundColor','g');
424 -         set(handles.str, 'String','Done');
425 -         figure(1);
426 -         plot(ebsi,AF);
427 -         title('Uniform linear antenna array (ULA) - array factor vs ebsi');
428 -         xlabel('ebsi','fontsize',10);
429 -         ylabel('AF','fontsize',10);
430 -         figure(2);
431 -         polar(Gamma,AF);
432 -         view([90 90]);
433 -         title('Uniform linear antenna array (ULA) - The 2D pattern of the array');
434 -         Phi_3D = meshgrid(Phi);
435 -         Gamma_3D = meshgrid(Gamma);
436 -         AF_3D = meshgrid(AF);
437 -         X = AF_3D.*sin(Gamma_3D).*cos(Phi_3D');
438 -         Y = AF_3D.*sin(Gamma_3D).*sin(Phi_3D');
439 -         Z = AF_3D.*cos(Gamma_3D);
440 -         figure(3);
441 -         surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
442 -         lighting gouraud;
443 -         title('Uniform linear antenna array (ULA) - The 3D pattern of the array');
444 -     end
445 - case 3
446 -     d_3a = get(handles.d_3a, 'String');
447 -     N_3a = get(handles.N_3a, 'String');
448 -     alpha_3a = get(handles.alpha_3a, 'String');
449 -     d = str2num(d_3a) * Lambda;
450 -     N = str2num(N_3a);
451 -     alpha = str2num(alpha_3a);
452 -     Theta = linspace(-pi,pi,6000);
453 -     Phi = linspace(-pi,pi,6000);
454 -     u = (B*d*cos(Theta) + alpha)/2;
455 -     AF = abs(cos(u).^(N-1));
456 -     if (d < 0)
457 -         set(handles.str, 'Visible','on');
458 -         set(handles.str, 'ForegroundColor','r');
459 -         set(handles.str, 'String', 'Error :: (d >= 0)');
460 -         close(ffigure(1));
461 -         close(ffigure(2));
462 -         close(ffigure(3));
463 -     elseif (N < 0)
464 -         set(handles.str, 'Visible','on');
465 -         set(handles.str, 'ForegroundColor','r');
466 -         set(handles.str, 'String', 'Error :: (N >= 0)');
467 -         close(ffigure(1));
468 -         close(ffigure(2));
469 -         close(ffigure(3));
470 -     else
471 -         set(handles.str, 'Visible','on');
472 -         set(handles.str, 'ForegroundColor','g');
473 -         set(handles.str, 'String','Done');
474 -         figure(1);
475 -         plot(u,AF);
476 -         title('Nonuniformly-Fed linear antenna array | Binomial Arrays - array factor vs u');
477 -         xlabel('u','fontsize',10);
478 -         ylabel('AF','fontsize',10);

```

```

478 -         ylabel('AF','fontsize',10);
479 -         figure(2);
480 -         polar(Theta,AF);
481 -         view([90 90]);
482 -         title('Nonuniformly-Fed linear antenna array | Binomial Arrays - The 2D pattern of the array');
483 -         Phi_3D = meshgrid(Phi);
484 -         Theta_3D = meshgrid(Theta);
485 -         AF_3D = meshgrid(AF);
486 -         X = AF_3D.*sin(Theta_3D).*cos(Phi_3D');
487 -         Y = AF_3D.*sin(Theta_3D).*sin(Phi_3D');
488 -         Z = AF_3D.*cos(Theta_3D);
489 -         figure(3);
490 -         surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
491 -         axis vis3d;
492 -         axis equal;
493 -         lighting gouraud;
494 -         title('Nonuniformly-Fed linear antenna array | Binomial Arrays - The 3D pattern of the array');
495 -     end
496 - case 4
497 -     d_3b = get(handles.d_3b, 'String');
498 -     N_3b = get(handles.N_3b, 'String');
499 -     alpha_3b = get(handles.alpha_3b, 'String');
500 -     Ro_3b = get(handles.Ro_3b, 'String');
501 -     d = str2num(d_3b) * Lambda;
502 -     N = str2num(N_3b);
503 -     alpha = str2num(alpha_3b);
504 -     Ro = str2num(Ro_3b);
505 -     M = N - 1;
506 -     Zo = cosh((1/M)*acosh(Ro));
507 -     Z = linspace(-Zo,Zo,6000);
508 -     u_up = acos(Z./Zo);
509 -     u_down = -u_up;
510 -     u = [u_down ; u_up];
511 -     Theta1 = acos((2.*u_down)-alpha)/(B*d);
512 -     Theta2 = -Theta1;
513 -     Phi = linspace(-pi,pi,6000);
514 -     AF = abs(cosh(M.*acosh(Z)));
515 -     if (d < 0)
516 -         set(handles.str, 'Visible','on');
517 -         set(handles.str, 'ForegroundColor','r');
518 -         set(handles.str, 'String', 'Error :: (d >= 0)');
519 -         close(figure(1));
520 -         close(figure(2));
521 -         close(figure(3));
522 -     elseif (N < 0)
523 -         set(handles.str, 'Visible','on');
524 -         set(handles.str, 'ForegroundColor','r');
525 -         set(handles.str, 'String', 'Error :: (N >= 0)');
526 -         close(figure(1));
527 -         close(figure(2));
528 -         close(figure(3));
529 -     elseif (Ro <= 1)
530 -         set(handles.str, 'Visible','on');
531 -         set(handles.str, 'ForegroundColor','r');
532 -         set(handles.str, 'String', 'Error :: always (Ro > 1)');
533 -         close(figure(1));
534 -         close(figure(2));
535 -         close(figure(3));
536 -     else
537 -         set(handles.str, 'Visible','on');
538 -         set(handles.str, 'ForegroundColor','g');
539 -         set(handles.str, 'String','Done');
540 -         figure(1);
541 -         plot(Z,AF);
542 -         title('Nonuniformly-Fed linear antenna array | Dolph-Tshebyscheff Arrays - array factor vs Z');
543 -         xlabel('Z','fontsize',10);

```

```

533 -         close(fgure(1));
534 -         close(fgure(2));
535 -         close(fgure(3));
536 -     else
537 -         set(handles.ctr, 'Visible','on');
538 -         set(handles.ctr, 'ForegroundColor','g');
539 -         set(handles.ctr, 'String','Done');
540 -         fgure(1);
541 -         plot(Z,AF);
542 -         title('Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays - array factor vs Z');
543 -         xlabel('Z','fontsize',10);
544 -         ylabel('AF','fontsize',10);
545 -         fgure(2);
546 -         polar(Theta1,AF,'-b');
547 -         hold on;
548 -         polar(Theta2,AF,'-b');
549 -         view([90 90]);
550 -         title('Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays - The 2D pattern of the array');
551 -         Phi_3D = meshgrid(Phi);
552 -         Theta_3D = meshgrid(Theta1);
553 -         AF_3D = meshgrid(AF);
554 -         X = AF_3D.*sin(Theta_3D).*cos(Phi_3D');
555 -         Y = AF_3D.*sin(Theta_3D).*sin(Phi_3D');
556 -         Z = AF_3D.*cos(Theta_3D);
557 -         fgure(3);
558 -         surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
559 -         axis vis3d;
560 -         axis equal;
561 -         lighting gouraud;
562 -         title('Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays - The 3D pattern of the array');
563 -     end
564 - end

```

Figure 8: Code - Sketch.

1.3. Reset

To reboot program and close all fgures.



Reset

Figure 9: Reset.

```

566 - % --- Executes on button press in reset_button.
567 - function reset_button_Callback(hObject, eventdata, handles)
568 -     set(handles.select, 'Value',1);
569 -     set(handles.part1, 'Visible','on');
570 -     set(handles.part2, 'Visible','off');
571 -     set(handles.part3a, 'Visible','off');
572 -     set(handles.part3b, 'Visible','off');
573 -     set(handles.special_case, 'Visible','on');
574 -     set(handles.s1p, 'Visible','off');
575 -     set(handles.s2p, 'Visible','off');
576 -     set(handles.s3p, 'Visible','off');
577 -     set(handles.ctr, 'Visible','off');
578 -     set(handles.l_1, 'String','0');
579 -     set(handles.d_2, 'String','0');
580 -     set(handles.N_2, 'String','0');
581 -     set(handles.alpha_2, 'String','0');
582 -     set(handles.d_3a, 'String','0');
583 -     set(handles.N_3a, 'String','0');
584 -     set(handles.alpha_3a, 'String','0');
585 -     set(handles.d_3b, 'String','0');
586 -     set(handles.N_3b, 'String','0');
587 -     set(handles.alpha_3b, 'String','0');
588 -     set(handles.Ro_3b, 'String','0');
589 -     close(fgure(1));
590 -     close(fgure(2));
591 -     close(fgure(3));

```

Figure 10: Code - Reset.

1.4. Special Cases

Show only if you select linear antenna (dipole of general length).

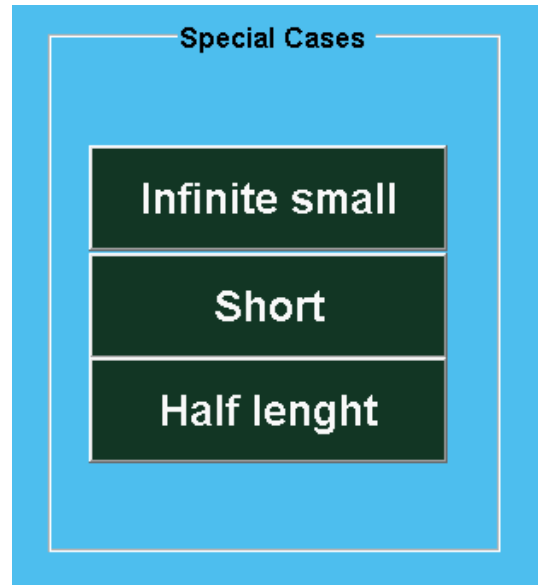


Figure 11: Special Cases.

```
594 % --- Executes on button press in s1.
595 function s1_Callback(hObject, eventdata, handles)
596     Lambda = 1;
597     B       = (2*pi)/Lambda;
598     set(handles.s1p, 'Visible', 'on');
599     set(handles.s2p, 'Visible', 'off');
600     set(handles.s3p, 'Visible', 'off');
601     set(handles.str, 'Visible', 'off');
602     Theta = linspace(-pi, pi, 350);
603     Phi    = linspace(-2*pi, 2*pi, 350);
604     L      = (1/50) * Lambda;
605     En     = abs(sin(Theta));
606     set(handles.str, 'Visible', 'on');
607     set(handles.str, 'ForegroundColor', 'g');
608     set(handles.str, 'String', 'Done');
609     figure(1);
610     polar(Theta, En);
611     view([90 90]);
612     title('Infinte small dipole - The 2D pattern of the dipole');
613     Phi_3D = meshgrid(Phi);
614     Theta_3D = meshgrid(Theta);
615     En_3D = meshgrid(En);
616     X      = En_3D.*sin(Theta_3D).*cos(Phi_3D);
617     Y      = En_3D.*sin(Theta_3D).*sin(Phi_3D);
618     Z      = En_3D.*cos(Theta_3D);
619     figure(2);
620     surf(X, Y, Z, 'EdgeColor', 'interp', 'FaceAlpha', 0.1);
621     axis vis3d;
622     axis equal;
623     lighting gouraud;
624     title('Infinte small dipole - The 3D pattern of the dipole');
625     close.figure(3);
```

```

627 % --- Executes on button press in s2.
628 function s2_Callback(hObject, eventdata, handles)
629 Lambda = 1;
630 B = (2*pi)/Lambda;
631 set(handles.s2p, 'Visible', 'on');
632 set(handles.s1p, 'Visible', 'off');
633 set(handles.s3p, 'Visible', 'off');
634 set(handles.str, 'Visible', 'off');
635 Theta = linspace(-pi,pi,350);
636 Phi = linspace(-2*pi,2*pi,350);
637 L = (1/10) * Lambda;
638 En = abs(sin(Theta));
639 set(handles.str, 'Visible', 'on');
640 set(handles.str, 'ForegroundColor', 'g');
641 set(handles.str, 'String', 'Done');
642 figure(1);
643 polar(Theta,En);
644 view([90 90]);
645 title('Short dipole - The 2D pattern of the dipole');
646 Phi_3D = meshgrid(Phi);
647 Theta_3D = meshgrid(Theta);
648 En_3D = meshgrid(En);
649 X = En_3D.*sin(Theta_3D).*cos(Phi_3D);
650 Y = En_3D.*sin(Theta_3D).*sin(Phi_3D);
651 Z = En_3D.*cos(Theta_3D);
652 figure(2);
653 surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
654 axis vis3d;
655 axis equal;
656 lighting gouraud;
657 title('Short dipole - The 3D pattern of the dipole');
658 close.figure(3));
659
660 % --- Executes on button press in s3.
661 function s3_Callback(hObject, eventdata, handles)
662 Lambda = 1;
663 B = (2*pi)/Lambda;
664 set(handles.s3p, 'Visible', 'on');
665 set(handles.s1p, 'Visible', 'off');
666 set(handles.s2p, 'Visible', 'off');
667 set(handles.str, 'Visible', 'off');
668 Theta = linspace(-pi,pi,350);
669 Phi = linspace(-2*pi,2*pi,350);
670 L = (1/2) * Lambda;
671 En = abs((cos((B*L)/2).*cos(Theta)) - cos((B*L)/2)) ./ sin(Theta);
672 set(handles.str, 'Visible', 'on');
673 set(handles.str, 'ForegroundColor', 'g');
674 set(handles.str, 'String', 'Done');
675 figure(1);
676 polar(Theta,En);
677 view([90 90]);
678 title('lambda/2 dipole - The 2D pattern of the dipole');
679 Phi_3D = meshgrid(Phi);
680 Theta_3D = meshgrid(Theta);
681 En_3D = meshgrid(En);
682 X = En_3D.*sin(Theta_3D).*cos(Phi_3D);
683 Y = En_3D.*sin(Theta_3D).*sin(Phi_3D);
684 Z = En_3D.*cos(Theta_3D);
685 figure(2);
686 surf(X,Y,Z,'EdgeColor','interp','FaceAlpha',0.1);
687 axis vis3d;
688 axis equal;
689 lighting gouraud;
690 title('lambda/2 dipole - The 3D pattern of the dipole');
691 close.figure(3));

```

Figure 12: Code - Special Cases.

2. Examples

2.1. Part 1: linear antenna (dipole of general length)

Example 1: $l = 2\lambda$

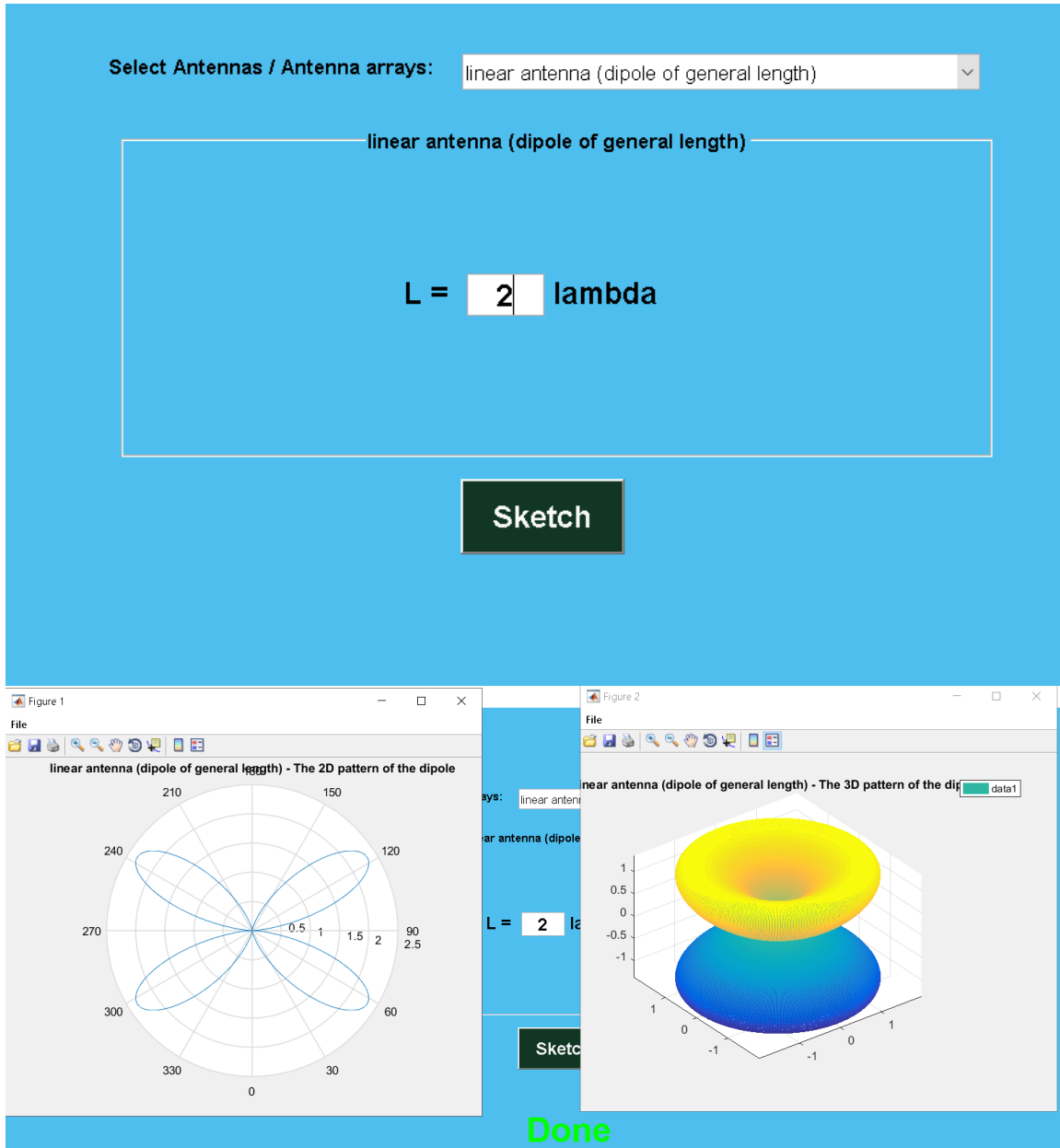



Figure 13: Example 1 - Part 1: linear antenna (dipole of general length).

Example 2: *Errors*
if $l < 0$

Select Antennas / Antenna arrays: linear antenna (dipole of general length) 

linear antenna (dipole of general length)

$L = -1 \text{ lambda}$

Sketch

Error :: ($L \geq 0$)

Figure 14: Example 2 (Errors) - Part 1: linear antenna (dipole of general length).

2.2. Part 2: Uniform linear antenna array (ULA)

Example 1: $d = \frac{4\lambda}{7}$, $N = 7$, $\alpha = \frac{-4\pi}{7}$

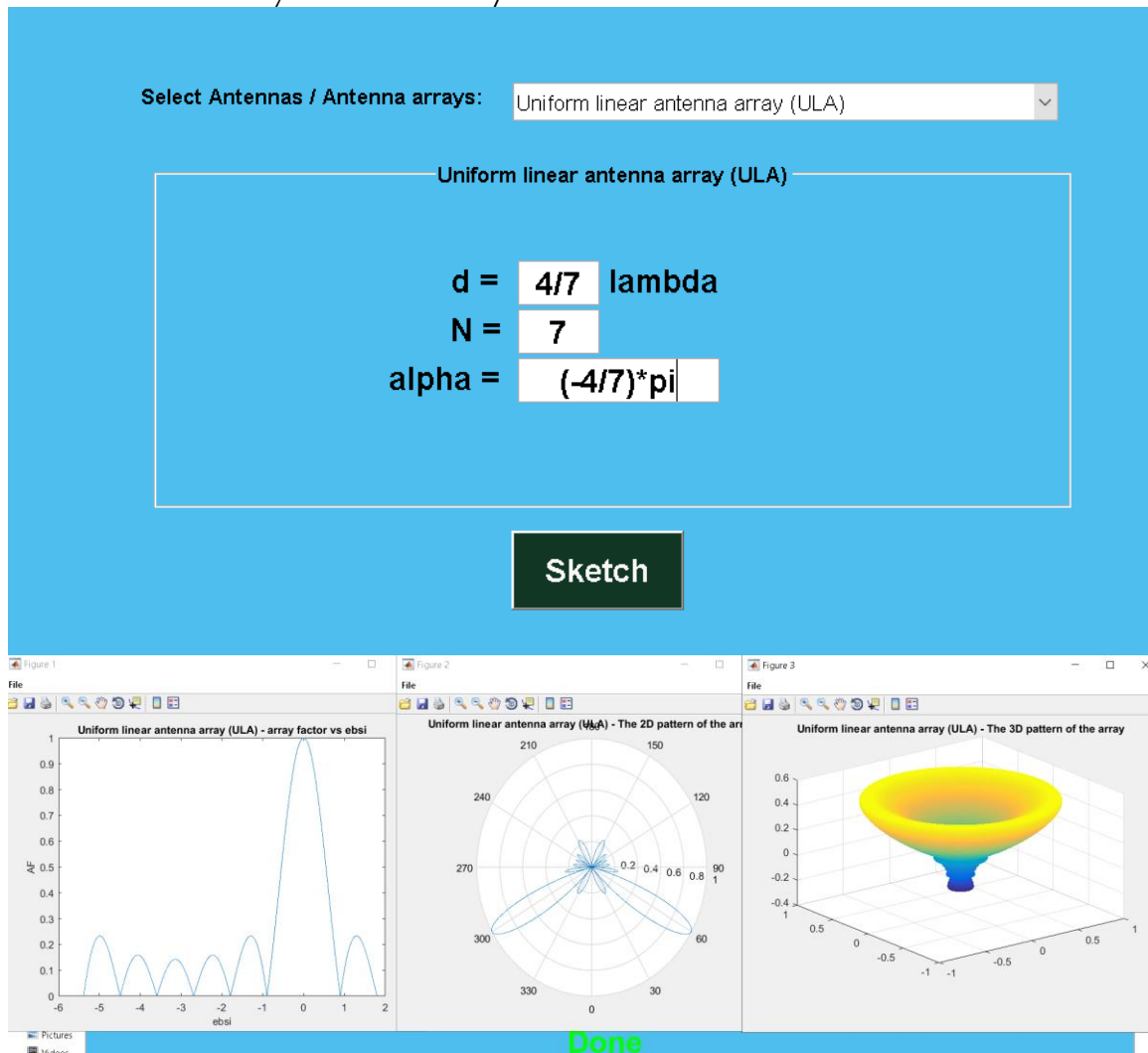


Figure 15: Example 1 - Part 2: Uniform linear antenna array (ULA).

Example 3: *Errors*
if $d < 0$ or $N < 0$.

The image shows two side-by-side screenshots of a software interface for a Uniform linear antenna array (ULA). Both screenshots have a dropdown menu set to 'Uniform linear antenna array (ULA)'.
 Left screenshot: The input fields show $d = -2$ lambda, $N = 0$, and $\alpha = 0$. Below the 'Sketch' button, a red error message reads 'Error :: (d >= 0)'.
 Right screenshot: The input fields show $d = 0$ lambda, $N = -1$, and $\alpha = 0$. Below the 'Sketch' button, a red error message reads 'Error :: (N >= 0)'.

Figure 16: Example 2 (Errors) - Part 2: Uniform linear antenna array (ULA).

2.3. Part 3: Nonuniformly-Fed linear antenna array

A. Binomial Arrays

Example 1: $d = \frac{3\lambda}{4}$, $N = 8$, $\alpha = 0$

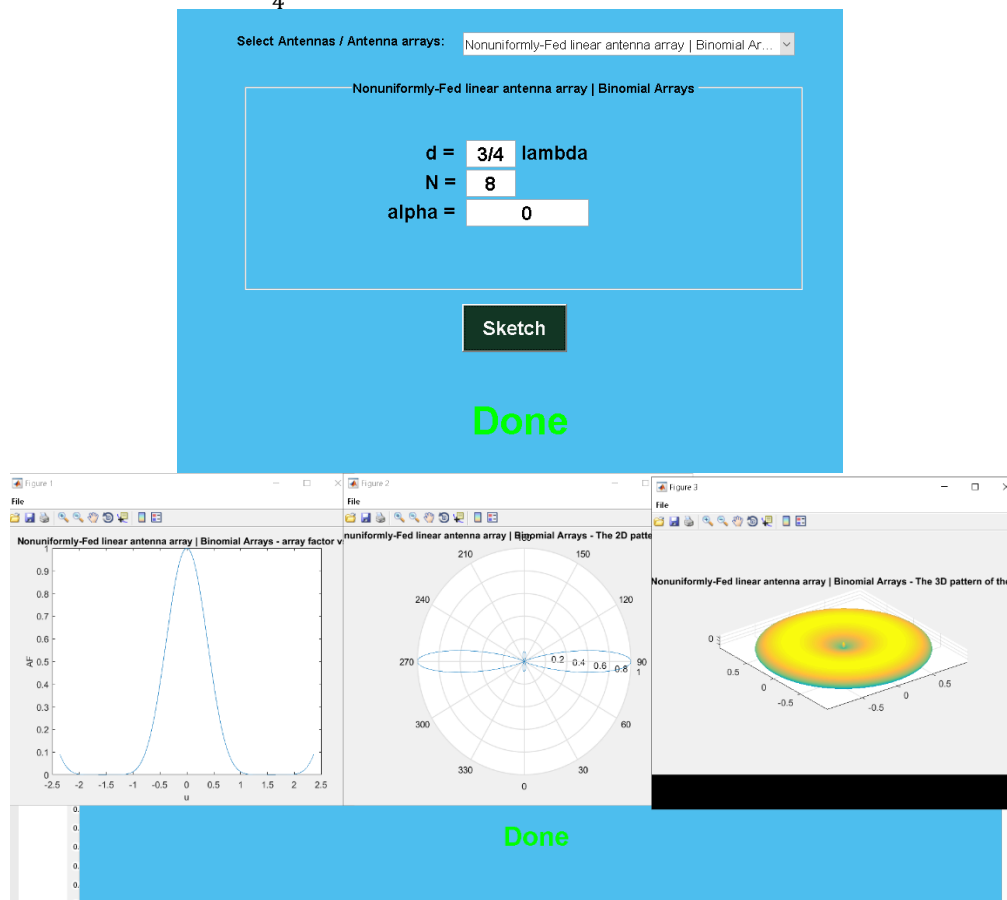


Figure 17: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).

Example 3: *Errors*
if $d < 0$ or $N < 0$.

Select Antennas / Antenna arrays: Nonuniformly-Fed linear antenna array | Binomial Ar...

Nonuniformly-Fed linear antenna array | Binomial Arrays

d = -5 lambda

N = 0

alpha = 0

Sketch

Error :: (d >= 0)

Select Antennas / Antenna arrays: Nonuniformly-Fed linear antenna array | Binomial Ar...

Nonuniformly-Fed linear antenna array | Binomial Arrays

d = 0 lambda

N = -2

alpha = 0

Sketch

Error :: (N >= 0)

Figure 18: Example 2 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Binomial Arrays).

B. Dolph-Tschebyscheff Arrays

Example 1: $d = \frac{\lambda}{2}$, $N = 6$, $\alpha = -\pi$, $R_0 = 10$.

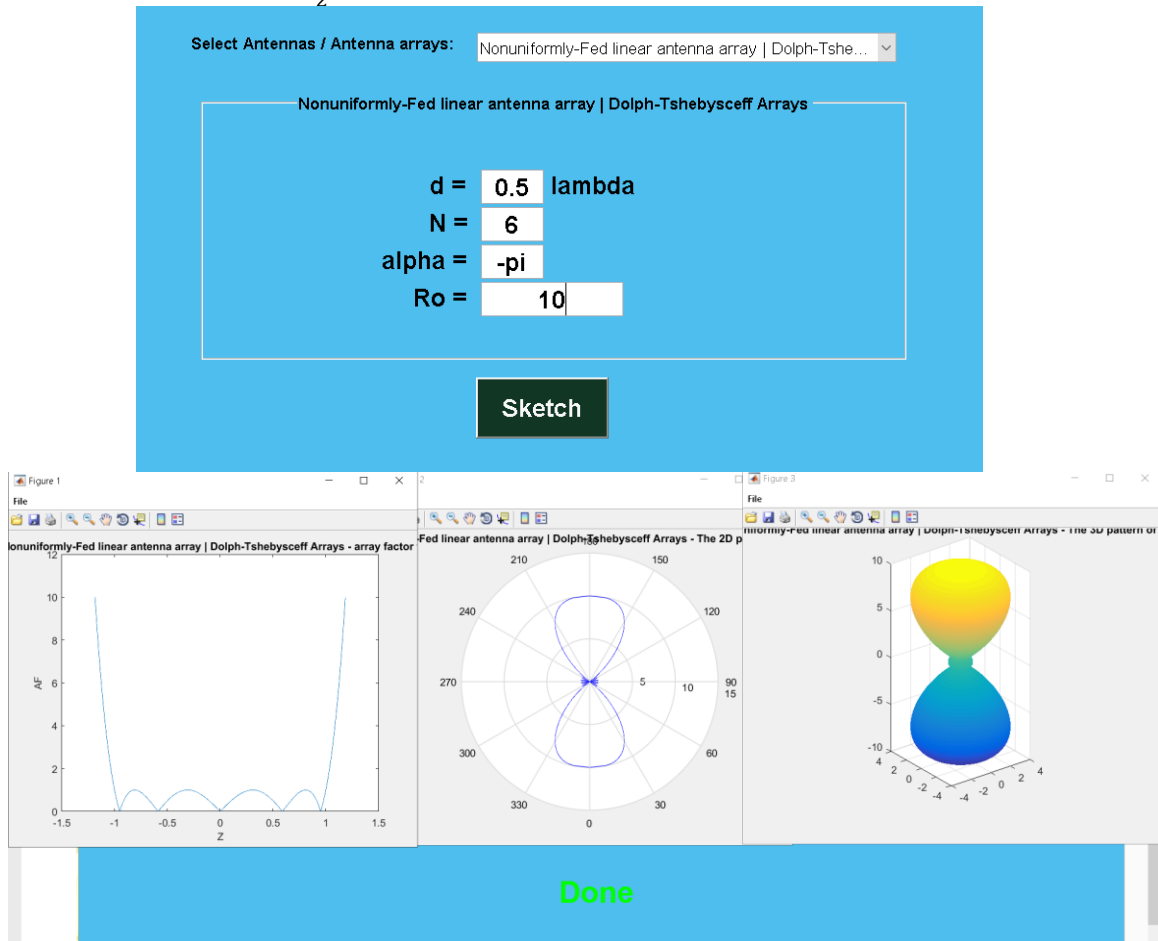


Figure 19: Example 1 - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebyscheff Arrays).

Example 3: *Errors*

if $d < 0$ or $N < 0$, we will ask you again.

Always $R_0 > 1$.

Select Antennas / Antenna arrays: Nonuniformly-Fed linear antenna array | Dolph-Tshe... ▾

Nonuniformly-Fed linear antenna array | Dolph-Tshebysceff Arrays

d = lambda

N =

alpha =

Ro =

Sketch

Error :: always ($R_0 > 1$)

Figure 20: Example 2 (Errors) - Part 3: Nonuniformly-Fed linear antenna array (Dolph-Tschebysceff Arrays).

2.4. Special Case for dipole antenna

A. Infinite small dipole

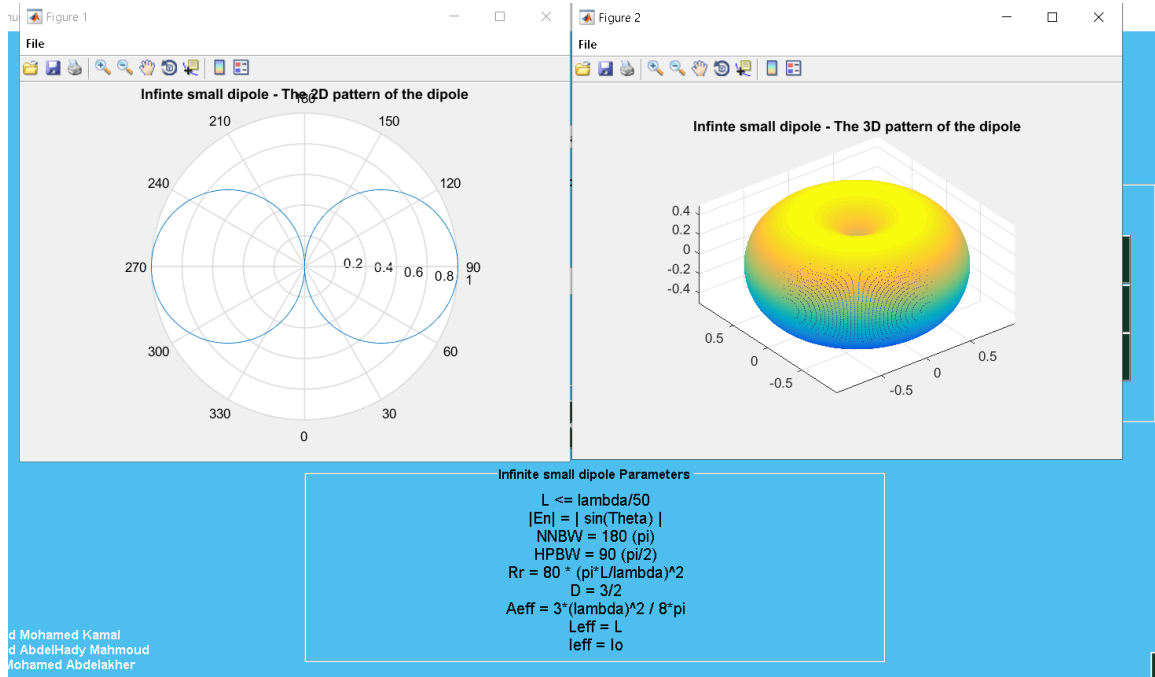


Figure 21: Infinite small dipole.

B. Short dipole

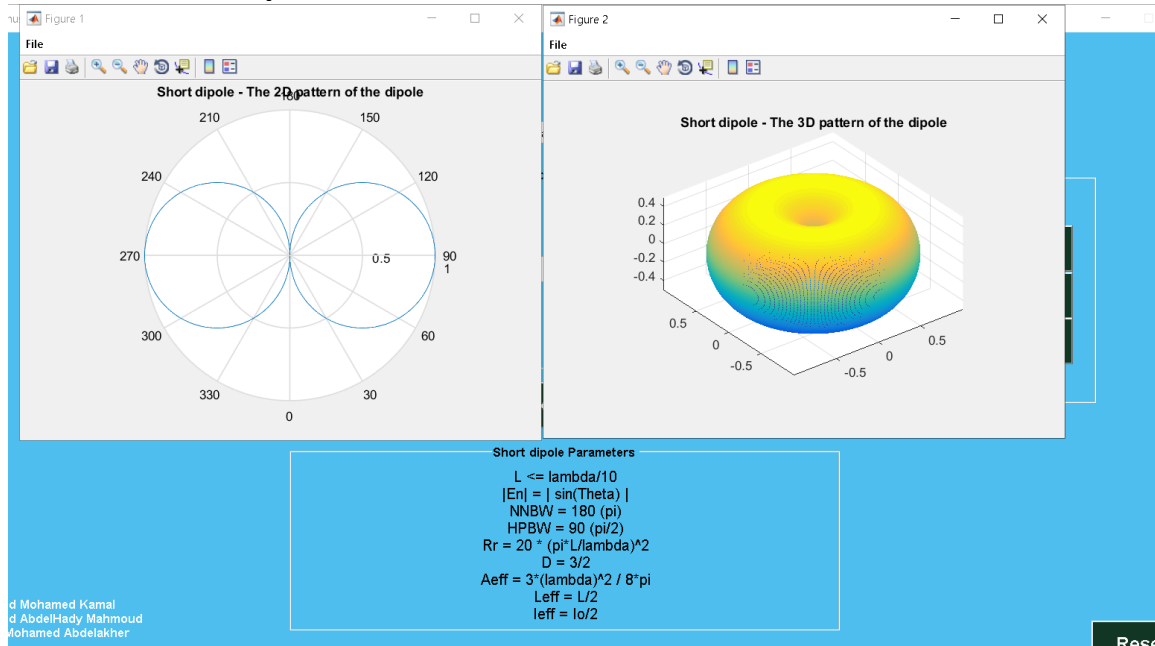


Figure 22: Short dipole.

C. $\lambda/2$ dipole

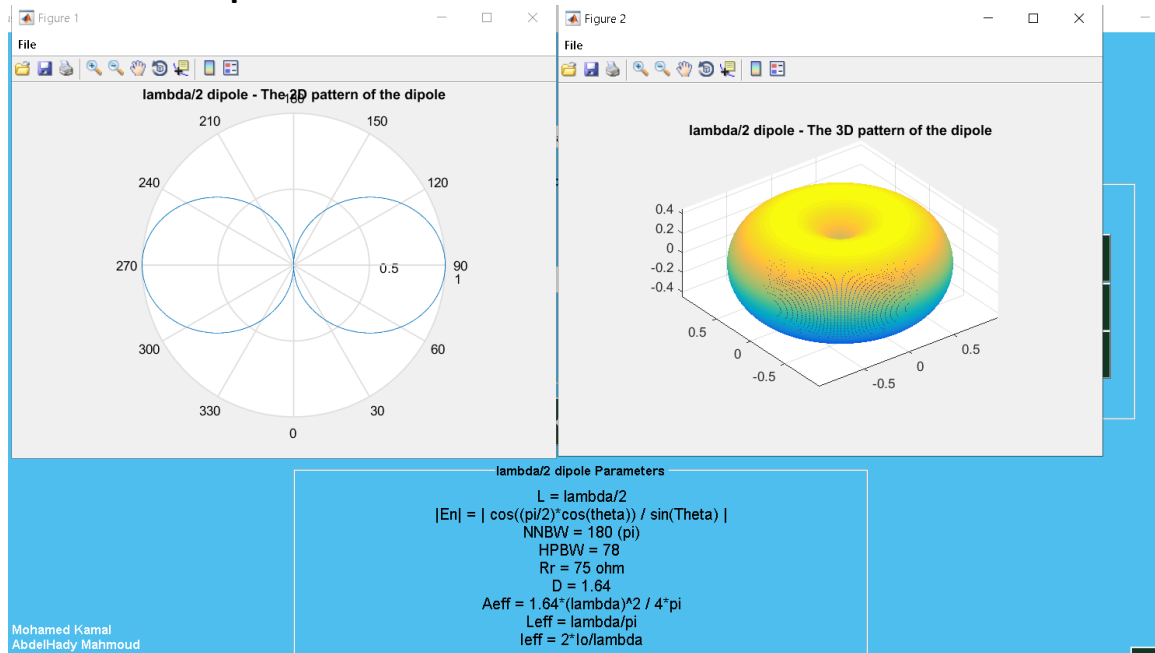


Figure 23: $\lambda/2$ dipole.



Electrical Engineering Department,
Fourth Year - Communications & Electronics.

EE 466 ANTENNA

Patch Antenna

| PREPARED BY | SECTION | SEAT.NO. |
|------------------------------|---------|----------|
| Mahmoud Mohamed Kamal Ismail | 7 | 250 |
| Mahmoud AbdElHady Mahmoud | 7 | 248 |
| Ahmed Mohamed Abdelakher | 1 | 37 |

➤ Contents

| | |
|----------------------------------|--------|
| 1. Introduction to Patch Antenna | Page 2 |
| 2. Matlab | Page 6 |
| 3. Copy of Code | Page 9 |

➤ List of Figures

| | |
|---|--------|
| Figure 1: Single Patch Antenna with 3D Radiation Pattern, Azimuth Plane Pattern and Elevation Plane Pattern. | Page 2 |
| Figure 2: A 4x4 Patch Array Antenna with 3D Radiation Pattern, Azimuth Plane Pattern and Elevation Plane Pattern. | Page 3 |
| Figure 3: Azimuth Plane Patterns of the 4 x 4 Patch Array in Polar and Rectangular Coordinates. | Page 4 |
| Figure 4: Elevation Plane Patterns of the 4 x 4 Patch Array in Polar and Rectangular Coordinates. | Page 4 |
| Figure 5: Code. | Page 6 |
| Figure 6: Create and view a microstrip patch with specified parameters. | Page 7 |
| Figure 7: Plot the radiation pattern of the antenna at a frequency of 1.67 GHz. | Page 7 |
| Figure 8: Calculate and plot the impedance of the antenna over the specified frequency range. | Page 8 |
| Figure 9: Patch 1X4 Antenna Array. | Page 8 |
| Figure 10: Create a microstrip patch antenna using 'FR4' as the dielectric substrate. | Page 8 |

https://github.com/MahmoudFierro98/Antenna_Lab/

1. Introduction to Patch Antenna

Directional antennas are used for coverage as well as point-to-point links. They can be patch antennas, dishes, horns or a whole host of other varieties. They all accomplish the same goal: radiating their energy out in a particular direction.

Patch Antennas

A patch antenna, in its simplest form, is just a single rectangular (or circular) conductive plate that is spaced above a ground plane. Patch antennas are attractive due to their low profile and ease of fabrication.

The radiation pattern of a single patch is characterized by a single main lobe of moderate beamwidth. Frequently, the beamwidths in the azimuth and elevation planes are similar, resulting in a fairly circular beam, although this is by no means universal. The beamwidths can be manipulated to produce an antenna with higher or lower gain, depending on the requirements. An antenna built with a single patch will have a maximum gain of about 9 dBi or a bit less.

The patch antenna in Figure 1 shows how simple these antennas can be. This is a simple rectangular patch built over a rectangular ground plane. The radiation patterns exhibit typical patch antenna characteristics. There is a single main lobe with a fairly wide beamwidth with shallow nulls pointing up and down from the antenna. Other than that, there aren't many features to the pattern. The one shown in Figure 1 is designed to have higher gain rather than symmetrical plane patterns. The gain is about 8.8 dBi with an azimuth plane beamwidth of 70 degrees and an elevation plane beamwidth of 57 degrees. These are not uncommon beamwidths for single patch antennas.

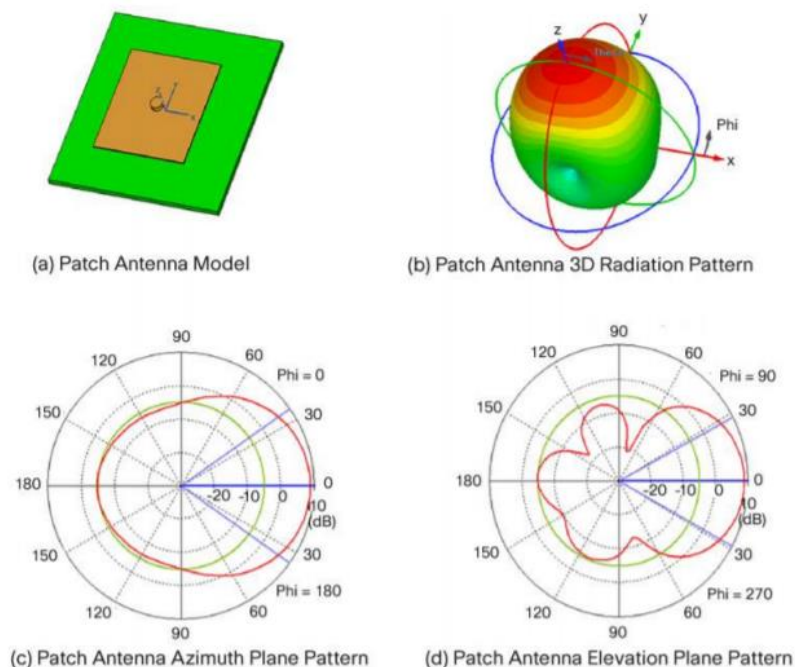


Figure 1: Single Patch Antenna with 3D Radiation Pattern, Azimuth Plane Pattern and Elevation Plane Pattern.

The azimuth and elevation plane patterns are derived by simply slicing through the 3D radiation pattern. In this case, the azimuth plane pattern is obtained by slicing through the x-z plane, and the elevation plane pattern is formed by slicing through the y-z plane. Note that there is one main lobe that is radiated out from the front of the antenna. There are three back lobes in the elevation plane (in this case), the strongest of which happens to be 180 degrees behind the peak of the main lobe, establishing the front-to-back ratio at about 14 dB. That is, the gain of the antenna 180 degrees behind the peak is 14 dB lower than the peak gain. Again, it doesn't matter if these patterns are shown pointing up, down, to the left or to the right. That is usually an artifact of the measurement system. A patch antenna radiates its energy out from the front of the antenna. That will establish the true direction of the patterns.

Patch Array Antennas

A patch array antenna is, in general, some arrangement of multiple patch antennas that are all driven by the same source. Frequently, this arrangement consists of patches arranged in orderly rows and columns (a rectangular array) as shown in Figure 2. The reason for these types of arrangements is higher gain. Higher gain commonly implies a narrower beamwidth and that is, indeed, the case with patch arrays. The array shown here has a gain of about 18 dBi with an azimuth and elevation plane beamwidth of about 20 degrees. Notice that the back lobes are very small and that the front-to-back ratio is about 30 dB. The first sidelobes are down from the peak about 14 dB.

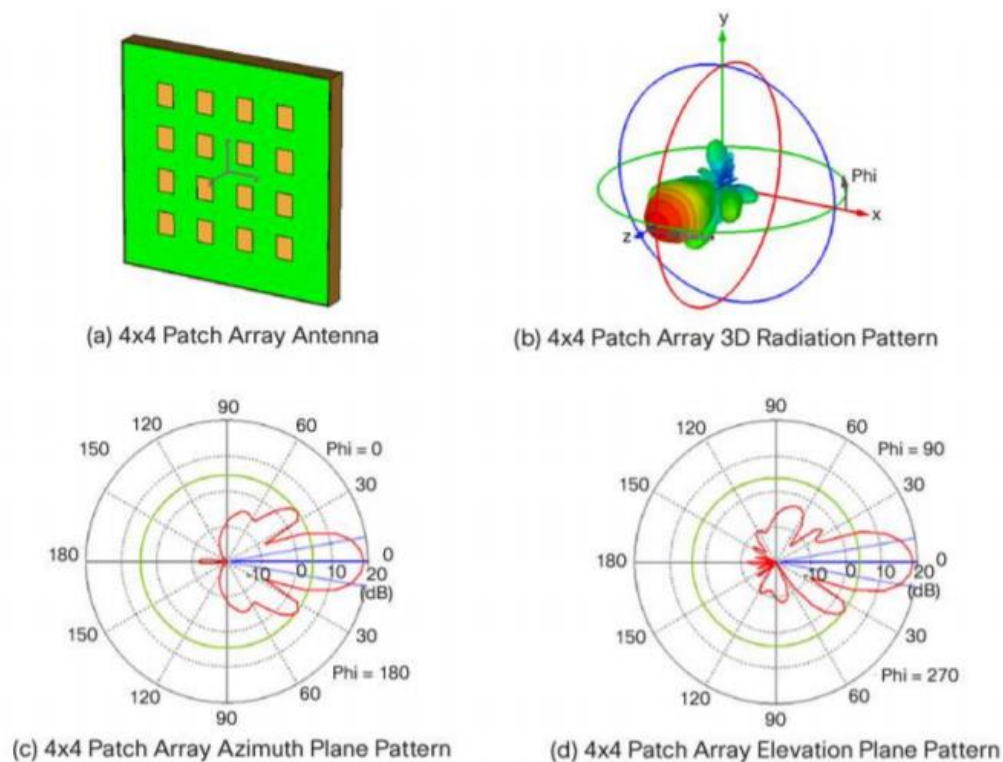


Figure 2: A 4x4 Patch Array Antenna with 3D Radiation Pattern, Azimuth Plane Pattern and Elevation Plane Pattern.

Antenna patterns are frequently shown normalized to the peak gain. The peak gain (in dBi) is simply subtracted from the gain at all the points on the curve and the pattern is plotted with the new values. These patterns are expressed in dB with 0 dB corresponding to the peak gain. A normalized pattern is especially useful when the sidelobe levels and the depth of the nulls are of interest since it's easier to read their respective levels. The patterns of the patch array shown here have enough lobes and features that a look at their normalized patterns in rectangular coordinates might be interesting. Figure 3 shows the azimuth plane in both polar and Cartesian (rectangular) coordinates. Figure 4 shows the elevation plane in both coordinate systems. The side lobe levels are easily readable from the rectangular plots. In the azimuth plane, the side lobes are down about 14 dB from the peak. The first side lobe levels are more than 14 dB down in the elevation plane. Note that the back lobe is 30 dB down from the peak. That means the front-to-back ratio is 30 dB. Of course, if the patterns are given in normalized form, the peak gain must be given to determine absolute levels of any of the pattern parameters. The side lobes are labeled in all the plots. Notice that the lower side lobes are to the left of the main beam in the Cartesian plots. These plots show the main beam at 0 degrees, so below the main beam would imply negative angle and above the main beam would imply positive angle.

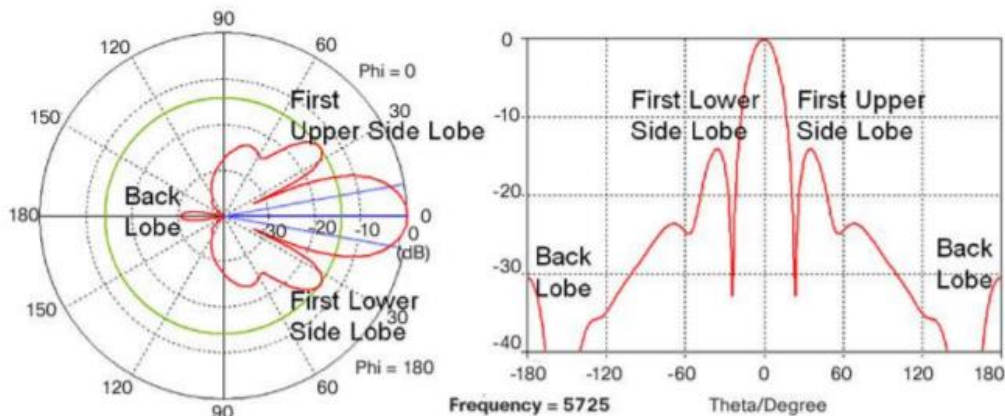


Figure 3: Azimuth Plane Patterns of the 4 x 4 Patch Array in Polar and Rectangular Coordinates.

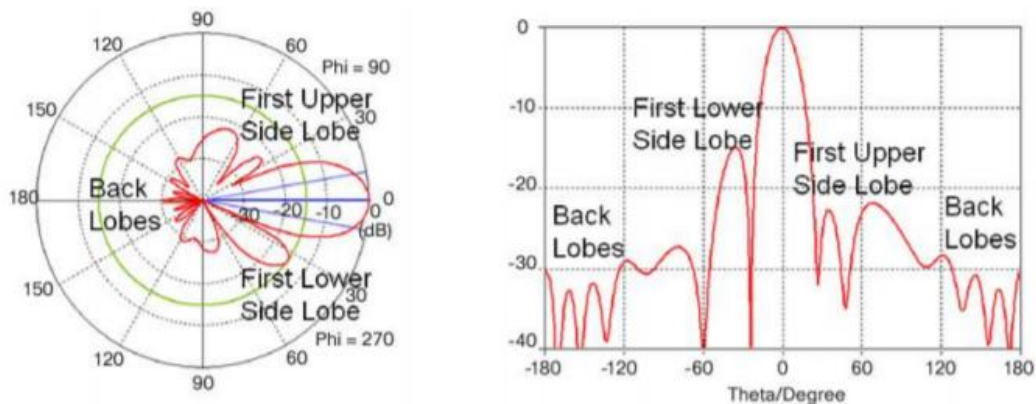


Figure 4: Elevation Plane Patterns of the 4 x 4 Patch Array in Polar and Rectangular Coordinates.

Microstrip antenna

In telecommunication, a microstrip antenna (also known as a printed antenna) usually means an antenna fabricated using photolithographic techniques on a printed circuit board (PCB). It is a kind of internal antenna. They are mostly used at microwave frequencies. An individual microstrip antenna consists of a patch of metal foil of various shapes (a patch antenna) on the surface of a PCB (printed circuit board), with a metal foil ground plane on the other side of the board. Most microstrip antennas consist of multiple patches in a two-dimensional array. The antenna is usually connected to the transmitter or receiver through foil microstrip transmission lines. The radio frequency current is applied (or in receiving antennas the received signal is produced) between the antenna and ground plane. Microstrip antennas have become very popular in recent decades due to their thin planar profile which can be incorporated into the surfaces of consumer products, aircraft and missiles; their ease of fabrication using printed circuit techniques; the ease of integrating the antenna on the same board with the rest of the circuit, and the possibility of adding active devices such as microwave integrated circuits to the antenna itself to make active antennas.

The most common type of microstrip antenna is commonly known as patch antenna. Antennas using patches as constitutive elements in an array are also possible. A patch antenna is a narrowband, wide-beam antenna fabricated by etching the antenna element pattern in metal trace bonded to an insulating dielectric substrate, such as a printed circuit board, with a continuous metal layer bonded to the opposite side of the substrate which forms a ground plane. Common microstrip antenna shapes are square, rectangular, circular and elliptical, but any continuous shape is possible. Some patch antennas do not use a dielectric substrate and instead are made of a metal patch mounted above a ground plane using dielectric spacers; the resulting structure is less rugged but has a wider bandwidth. Because such antennas have a very low profile, are mechanically rugged and can be shaped to conform to the curving skin of a vehicle, they are often mounted on the exterior of aircraft and spacecraft, or are incorporated into mobile radio communications devices.

2. Matlab

```

Patch_Antennam  x +
1  %%
2  % Alexandria University - Faculty of Engineering
3  % Electrical and Electronic Engineering Department - Fourth Year - Communications & Electronics
4  %
5  % Course: Antenna Lab
6  % Patch Antenna
7  %
8  % Name      : Mahmoud Mohamed Kamal Ismail - Mahmoud AbdElHady Mahmoud - Ahmed Mohamed Abdelakher
9  % Section   : 7                               - 7                               - 1
10 % Seat No.: 250                               - 248                              - 37
11
12 %%
13 clear;
14 close all;
15 clc;
16
17 %% Create and view a microstrip patch with specified parameters.
18 pm = patchMicrostrip('Length',75e-3, 'Width',37e-3, ...
19 'GroundPlaneLength',120e-3, 'GroundPlaneWidth',120e-3);
20 pm.Height = 0.01;
21 pm
22 figure(1);
23 show(pm);
24
25 %% Plot the radiation pattern of the antenna at a frequency of 1.67 GHz.
26 figure(2);
27 pattern(pm,1.67e9);
28
29 %% Calculate and plot the impedance of the antenna over the specified frequency range.
30 figure(3);
31 impedance(pm,linspace(0.5e9,2e9,50));
32
33 %% Patch Antenna Array
34 a = linearArray;
35 a.Element      = pm;
36 a.ElementSpacing = 0.1;
37 a.NumElements  = 4;
38 figure(4);
39 layout(a);
40 figure(5);
41 pattern(a,1.67e9);
42
43 %% Create a microstrip patch antenna using 'FR4' as the dielectric substrate.
44 d = dielectric('FR4');
45 pm = patchMicrostrip('Length',75e-3,'Width',37e-3, ...
46 'GroundPlaneLength',120e-3,'GroundPlaneWidth',120e-3, ...
47 'Substrate',d)
48 figure(6);
49 show(pm);

```

Figure 5: Code.

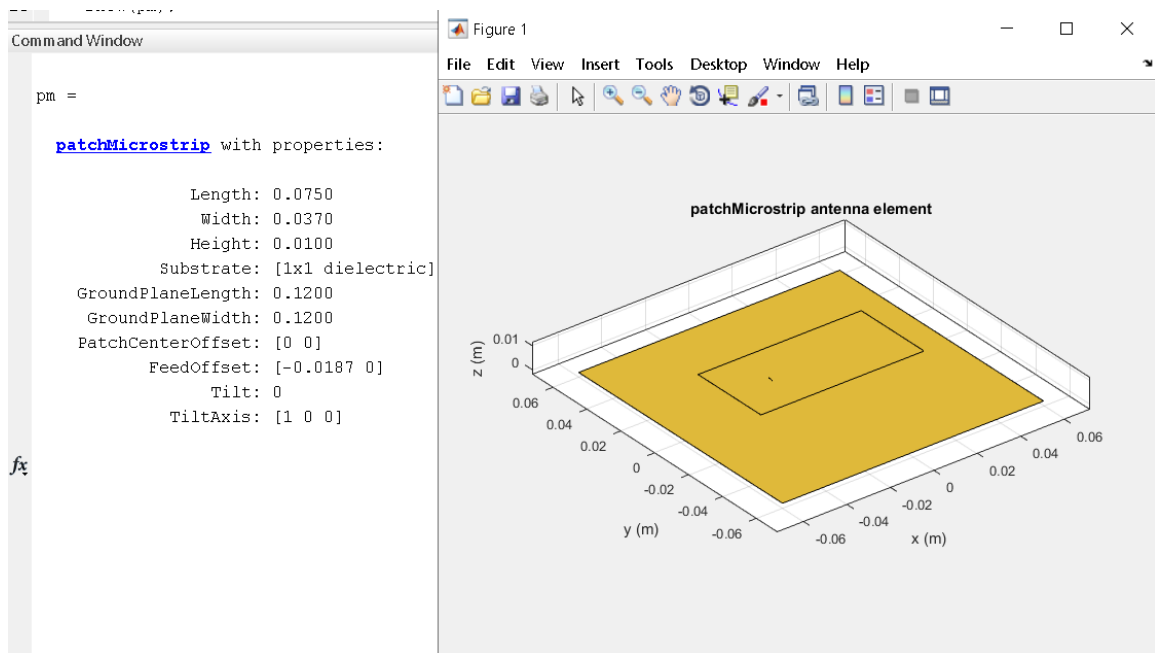


Figure 6: Create and view a microstrip patch with specified parameters.

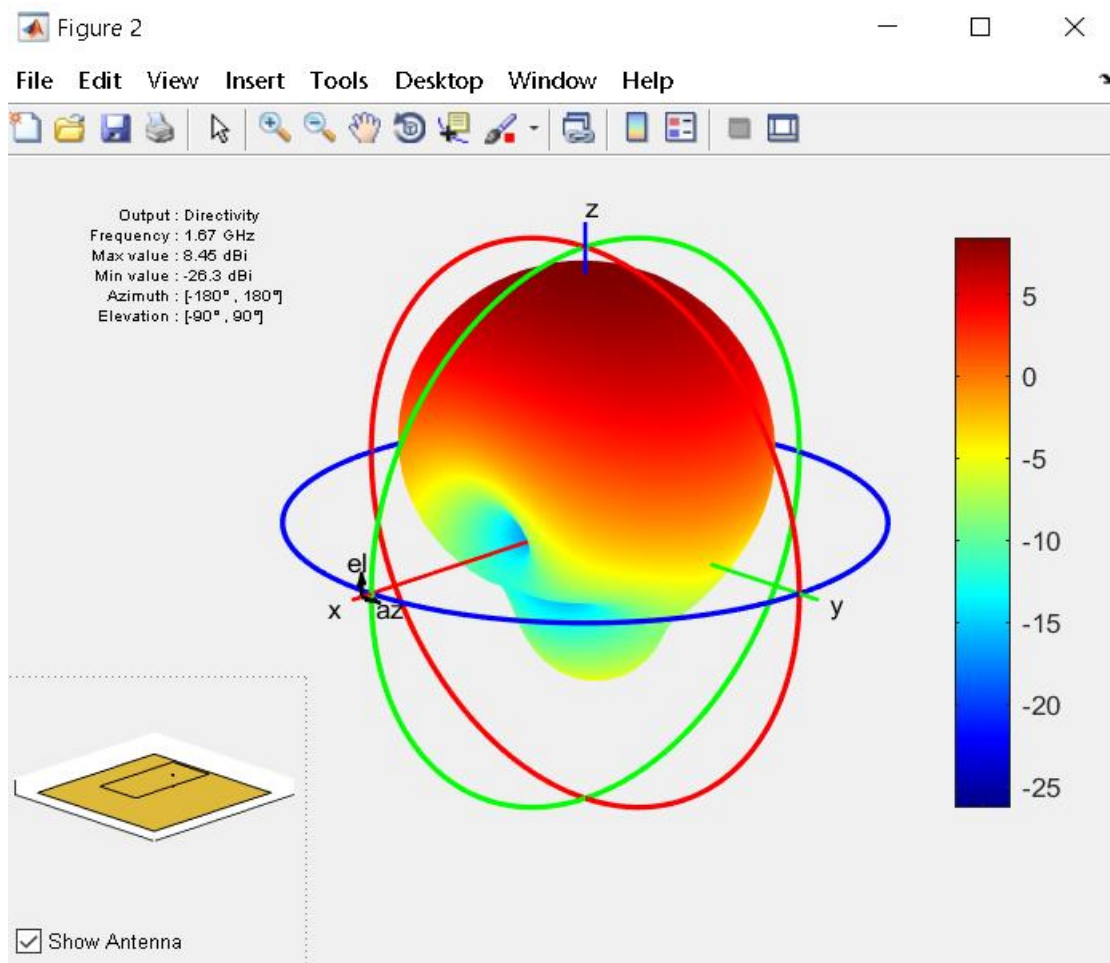


Figure 7: Plot the radiation pattern of the antenna at a frequency of 1.67 GHz.

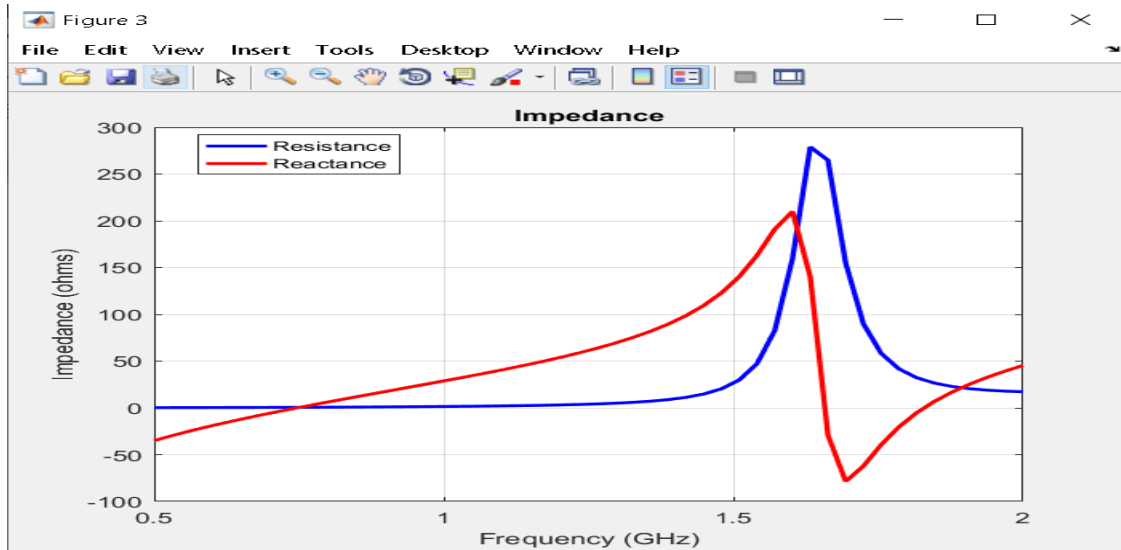


Figure 8: Calculate and plot the impedance of the antenna over the specified frequency range.

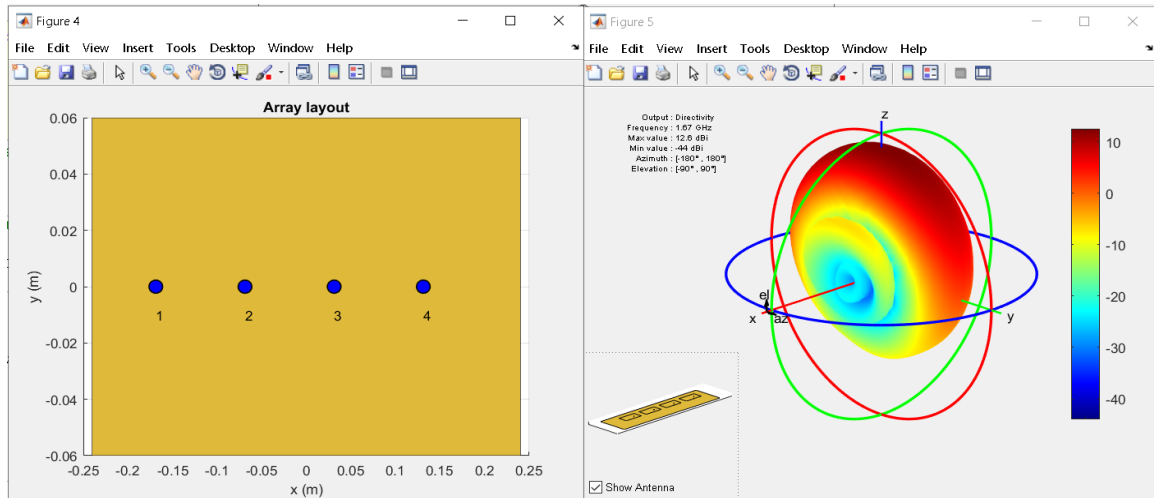


Figure 9: Patch 1X4 Antenna Array.

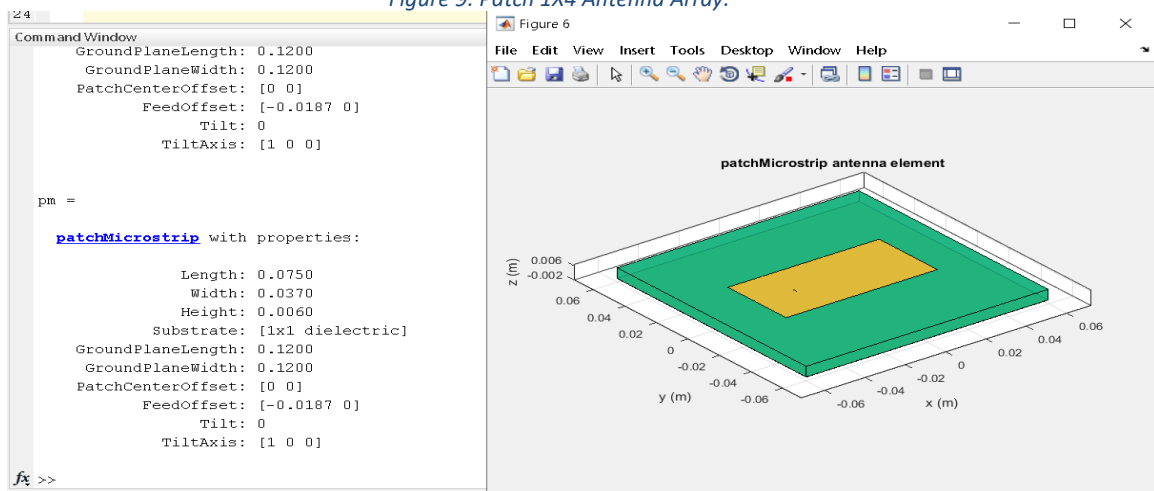


Figure 10: Create a microstrip patch antenna using 'FR4' as the dielectric substrate.

3. Copy of Code

```
%%
% Alexandria University - Faculty of Engineering
% Electrical and Electronic Engineering Department -
Fourth Year - Communications & Electronics
%
% Course: Antenna Lab
% Patch Antenna
%
% Name      : Mahmoud Mohamed Kamal Ismail - Mahmoud
AbdElHady Mahmoud - Ahmed Mohamed Abdelakher
% Section  : 7                      - 7
- 1
% Seat No.: 250                      - 248
- 37

%%
clear;
close all;
clc;

%% Create and view a microstrip patch with specified
parameters.
pm = patchMicrostrip('Length',75e-3, 'Width',37e-3,
...
    'GroundPlaneLength',120e-3,
    'GroundPlaneWidth',120e-3);
pm.Height = 0.01;
pm
figure(1);
show(pm);

%% Plot the radiation pattern of the antenna at a
frequency of 1.67 GHz.
figure(2);
pattern(pm,1.67e9);

%% Calculate and plot the impedance of the antenna over
the specified frequency range.
figure(3);
impedance(pm,linspace(0.5e9,2e9,50));

%% Patch Antenna Array
a = linearArray;
```

```

a.Element          = pm;
a.ElementSpacing   = 0.1;
a.NumElements      = 4;
figure(4);
layout(a);
figure(5);
pattern(a,1.67e9);

%% Create a microstrip patch antenna using 'FR4' as the
dielectric substrate.
d = dielectric('FR4');
pm = patchMicrostrip('Length',75e-3,'Width',37e-3,
...
    'GroundPlaneLength',120e-
3,'GroundPlaneWidth',120e-3, ...
    'Substrate',d)
figure(6);
show(pm);

```

References: <https://www.industrialnetworking.com/pdf/Antenna-Patterns.pdf>
<https://www.antenna-theory.com/antennas/patches/antenna.php>
https://en.wikipedia.org/wiki/Microstrip_antenna
<https://www.mathworks.com/help/antenna/ref/patchmicrostrip.html>