

## HW 6 Solutions

### Problem 1

#### Part a

Check the Grashof Criterion:

$$\begin{aligned}1.5 + 2.75 &< 2 + 2.5 \\4.25 &< 4.5\end{aligned}$$

This is a Grashof linkage. Since  $r_3 = r_s$ , then it is classified as a double-rocker (rocker-rocker.)

#### Part b

$$\begin{aligned}\Delta\theta_2 &= |\theta'_2 - \theta''_2| \\ \theta'_2 &= \arccos\left(\frac{r_1^2 + r_2^2 - (r_3 + r_4)^2}{2r_1r_2}\right) \\ &= \arccos\left(\frac{2^2 + 2.5^2 - (1.5 + 2.75)^2}{2 \times 2 \times 2.5}\right) \\ &= 141.4^\circ \\ \theta''_2 &= \arccos\left(\frac{r_1^2 + r_2^2 - (r_3 - r_4)^2}{2r_1r_2}\right) \\ &= \arccos\left(\frac{2^2 + 2.5^2 - (1.5 - 2.75)^2}{2 \times 2 \times 2.5}\right) \\ &= 29.7^\circ \\ \Delta\theta_2 &= |141.4 - 29.7| = 111.7^\circ\end{aligned}$$

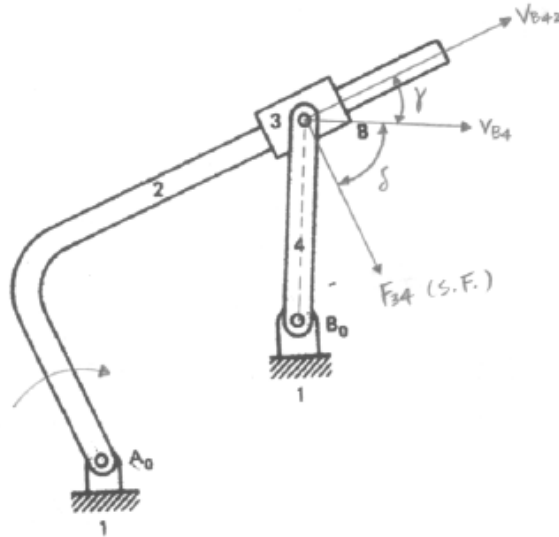
**Part c**

$$\begin{aligned}
 \Delta\theta_4 &= |\theta'_4 - \theta''_4| \\
 \theta'_4 &= \arccos\left(\frac{r_1^2 + r_4^2 - (r_3 + r_2)^2}{2r_1r_4}\right) \\
 &= \arccos\left(\frac{2^2 + 2.75^2 - (1.5 + 2.5)^2}{2 \times 2 \times 2.75}\right) \\
 &= 113.8^\circ \\
 \theta''_4 &= \arccos\left(\frac{r_1^2 + r_4^2 - (r_3 - r_2)^2}{2r_1r_4}\right) \\
 &= \arccos\left(\frac{2^2 + 2.75^2 - (1.5 - 2.5)^2}{2 \times 2 \times 2.75}\right) \\
 &= 16.2^\circ \\
 \Delta\theta_2 &= |113.8 - 16.2| = 97.6^\circ
 \end{aligned}$$

**Problem 2**

$$\theta_4 - \theta_3 = 105^\circ - 15^\circ = 90^\circ$$

**Problem 3**



*Image 1: Mechanism a, link 2 input*

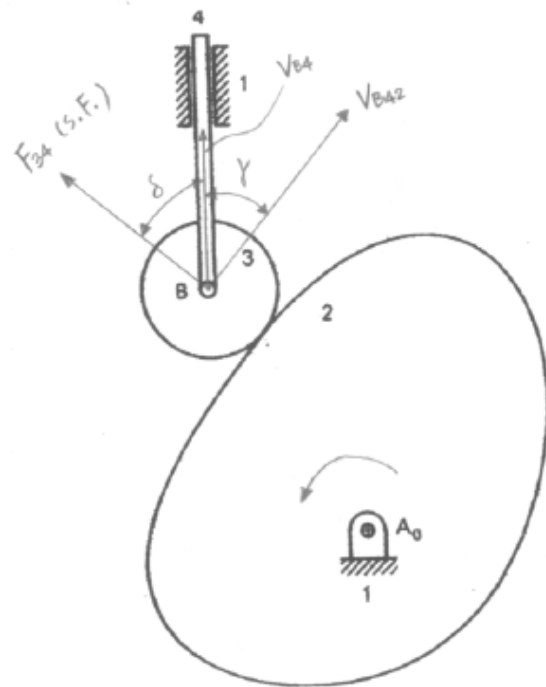


Image 2: Mechanism b, link 2 input

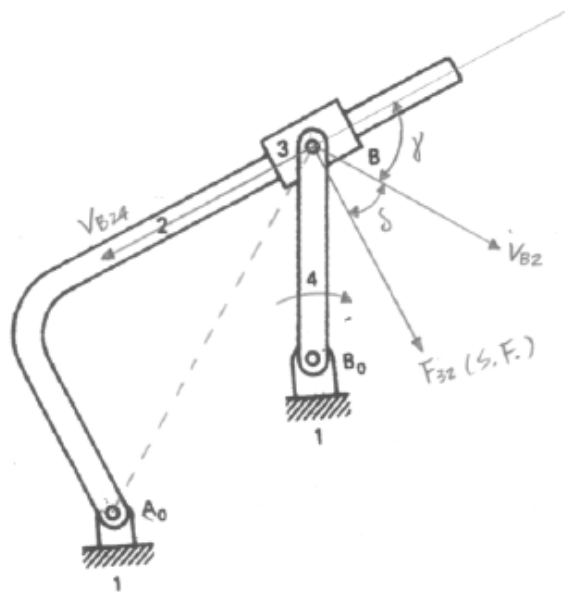


Image 3: Mechanism a, link 4 input

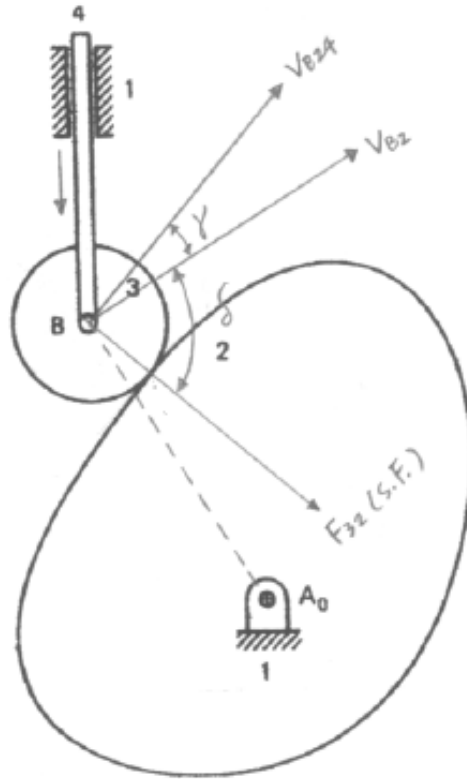


Image 4: Mechanism b, link 4 input

## Problem 4

### Part a

### Part b

```

1 At 45 degrees ,
2 branches = 2
3 Circuit 1:
4 theta3 = 19.634638
5 theta4 = 109.432520
6 theta5 = -11.129897
7 r6 = 7.790135
8 P = complex(2.632426,2.301831)
9
10 Circuit 2:
11 theta3 = -98.735229
12 theta4 = 171.466889
13 theta5 = 36.985943

```

```

14 r6 = 4.435564
15 P = complex(1.195465,-1.744731)
16
17 Circuit 3:
18 theta3 = 19.634638
19 theta4 = 109.432520
20 theta5 = -168.870103
21 r6 = -0.059402
22 P = complex(2.632426,2.301831)
23
24 Circuit 4:
25 theta3 = -98.735229
26 theta4 = 171.466889
27 theta5 = 143.014057
28 r6 = -1.954701
29 P = complex(1.195465,-1.744731)

```

### Part c

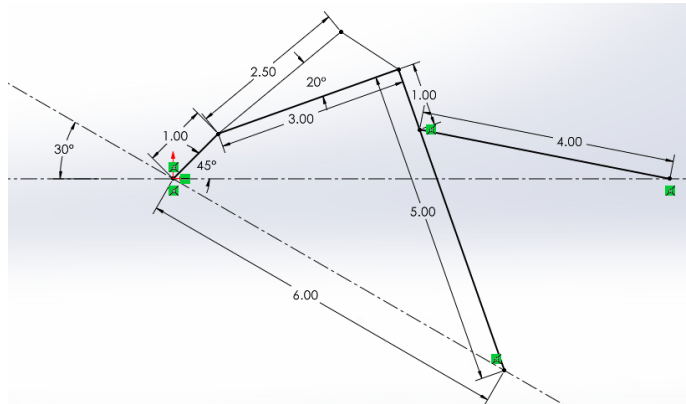


Image 5: Drawing to scale

### Part d

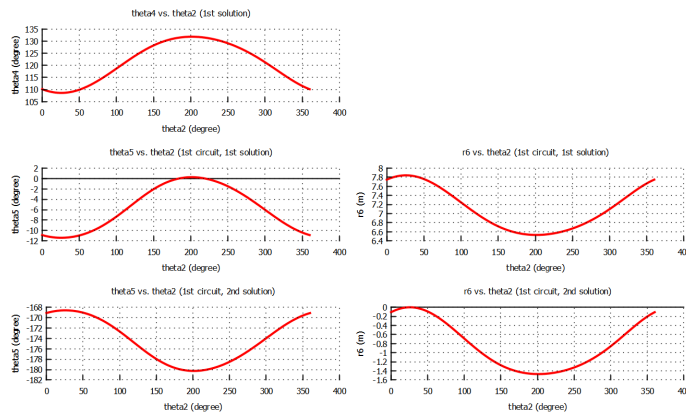


Image 6: First set of solutions

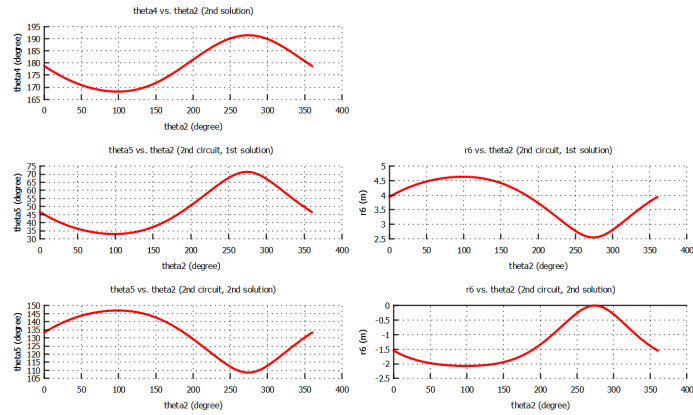


Image 7: Second set of solutions

## Code

```

1  /* h5p5.ch */
2  #include <math.h>
3  #include <complex.h>
4  #include <array.h>
5  #include <chplot.h>
6
7  #define NUMPOINTS 361
8  #define DEG2RAD(a) ((M_PI/180)*a)
9  #define RAD2DEG(a) ((180/M_PI)*a)
10
11 /* Consult Figure P3.50 in the Mechanism Design textbook. */
12
13 int main() {
14     int n1, n2;
15     double r[1:7], theta1, rp, beta, theta6;
16
17     int i;
18     double theta2[NUMPOINTS],
19            theta3_1[NUMPOINTS], theta3_2[NUMPOINTS],
20            x1, x2, x3, x4;
21     array double theta4_1[NUMPOINTS], theta4_2[NUMPOINTS],
22            theta5_1_1[NUMPOINTS], theta5_1_2[NUMPOINTS],
23            theta5_2_1[NUMPOINTS], theta5_2_2[NUMPOINTS];
24     double complex z,
25            P_1[NUMPOINTS], P_2[NUMPOINTS],
26            B_1[NUMPOINTS], B_2[NUMPOINTS];
27
28     array double r6_1_1[NUMPOINTS], r6_1_2[NUMPOINTS],
29            r6_2_1[NUMPOINTS], r6_2_2[NUMPOINTS];
30     class CPlot mainplot1;
31     class CPlot mainplot2;
32     class CPlot* subplot;
33     int branches[NUMPOINTS];
34
35     //-----Part A
36     -----

```

```

36 //Determine theta4, point B on the output link, and coupler
point P
37 //
-----

38
39 // Define the fourbar linkage
40 n1 = 2;
41 n2 = 4;
42 r[1] = 6;
43 r[2] = 1;
44 r[3] = 3;
45 r[4] = 5;
46 r[5] = 4;
47 r[7] = r[4] - 1; /* Length of B0 - C */
48 theta1 = DEG2RAD(-30);
49 rp = 2.5;
50 beta = DEG2RAD(20);
51 theta6 = 0;
52
53 // Analyze the fourbar with theta2 rotating from 0-360
54 for(i=0; i<=360; i++) {
55     theta2[i] = DEG2RAD(i);
56
57     // Determine theta3 and theta4
58     z = polar(r[1], theta1) - polar(r[2], theta2[i]);
59     branches[i] = complexsolve(n1, n2, r[3], -r[4], z, x1, x2,
x3, x4);
60
61     // First set of solutions
62     theta3_1[i] = x1;
63     theta4_1[i] = x2;
64     P_1[i] = polar(r[2], theta2[i]) + polar(rp, theta3_1[i]+
beta);
65     B_1[i] = polar(r[2], theta2[i]) + polar(r[3], theta3_1[i]);
66
67     // Second set of solutions
68     theta3_2[i] = x3;
69     theta4_2[i] = x4;
70     P_2[i] = polar(r[2], theta2[i]) + polar(rp, theta3_2[i]+
beta);
71     B_2[i] = polar(r[2], theta2[i]) + polar(r[3], theta3_2[i]);
72
73     // Find solutions for r5 and r6 : Circuit 1
74     complexsolve(1, 4, theta6, -r[5],
75         polar(r[1], theta1) + polar(r[7], theta4_1[i]),
76         x1, x2, x3, x4 );
77     r6_1_1[i] = x1;
78     theta5_1_1[i] = x2;
79     r6_1_2[i] = x3;
80     theta5_1_2[i] = x4;
81
82     // Find solutions for r5 and r6 : Circuit 2
83     complexsolve(1, 4, theta6, -r[5],
84         polar(r[1], theta1) + polar(r[7], theta4_2[i]),
85         x1, x2, x3, x4 );
86     r6_2_1[i] = x1;

```



```

87     theta5_2_1[i] = x2;
88     r6_2_2[i] = x3;
89     theta5_2_2[i] = x4;
90 }
91 // Smooth theta4_1, theta4_2
92 unwrap(theta4_1, theta4_1);
93 unwrap(theta4_2, theta4_2);
94 unwrap(theta5_1_2, theta5_1_2);
95
96 // Convert required angles from radian to degree
97 for(i=0; i<=360; i++) {
98     theta2[i] = RAD2DEG(theta2[i]);
99     theta3_1[i] = RAD2DEG(theta3_1[i]);
100    theta3_2[i] = RAD2DEG(theta3_2[i]);
101    theta4_1[i] = RAD2DEG(theta4_1[i]);
102    theta4_2[i] = RAD2DEG(theta4_2[i]);
103    theta5_1_1[i] = RAD2DEG(theta5_1_1[i]);
104    theta5_1_2[i] = RAD2DEG(theta5_1_2[i]);
105    theta5_2_1[i] = RAD2DEG(theta5_2_1[i]);
106    theta5_2_2[i] = RAD2DEG(theta5_2_2[i]);
107 }
108
109 printf("At 45 degrees, \n");
110 printf("branches = %d\n", branches[45]);
111 printf("Circuit 1:\n");
112 printf("theta3 = %lf\n", theta3_1[45]);
113 printf("theta4 = %lf\n", theta4_1[45]);
114 printf("theta5 = %lf\n", theta5_1_1[45]);
115 printf("r6 = %lf\n", r6_1_1[45]);
116 printf("P = %lf\n", P_1[45]);
117 printf("\nCircuit 2:\n");
118 printf("theta3 = %lf\n", theta3_2[45]);
119 printf("theta4 = %lf\n", theta4_2[45]);
120 printf("theta5 = %lf\n", theta5_2_1[45]);
121 printf("r6 = %lf\n", r6_2_1[45]);
122 printf("P = %lf\n", P_2[45]);
123 printf("\nCircuit 3:\n");
124 printf("theta3 = %lf\n", theta3_1[45]);
125 printf("theta4 = %lf\n", theta4_1[45]);
126 printf("theta5 = %lf\n", theta5_1_2[45]);
127 printf("r6 = %lf\n", r6_1_2[45]);
128 printf("P = %lf\n", P_1[45]);
129 printf("\nCircuit 4:\n");
130 printf("theta3 = %lf\n", theta3_2[45]);
131 printf("theta4 = %lf\n", theta4_2[45]);
132 printf("theta5 = %lf\n", theta5_2_2[45]);
133 printf("r6 = %lf\n", r6_2_2[45]);
134 printf("P = %lf\n", P_2[45]);
135
136 /* Display Plots */
137 mainplot1.subplot(3,2);
138 subplot=mainplot1.getSubplot(0,0);
139 subplot->data2D(theta2, theta4_1);
140 subplot->title("theta4 vs. theta2 (1st solution)");
141 subplot->label(PLOT_AXIS_X, "theta2 (degree)");
142 subplot->label(PLOT_AXIS_Y, "theta4 (degree)");
143

```

```

144 subplot=mainplot1.getSubplot(1,0);
145 subplot->data2D(theta2, theta5_1_1);
146 subplot->title("theta5 vs. theta2 (1st circuit, 1st solution)");
147 ;
148 subplot->label(PLOT_AXIS_X,"theta2 (degree)");
149 subplot->label(PLOT_AXIS_Y,"theta5 (degree)");
150
151 subplot=mainplot1.getSubplot(1,1);
152 subplot->data2D(theta2, r6_1_1);
153 subplot->title("r6 vs. theta2 (1st circuit, 1st solution)");
154 subplot->label(PLOT_AXIS_X,"theta2 (degree)");
155 subplot->label(PLOT_AXIS_Y,"r6 (m)");
156
157 subplot=mainplot1.getSubplot(2,0);
158 subplot->data2D(theta2, theta5_1_2);
159 subplot->title("theta5 vs. theta2 (1st circuit, 2nd solution)");
160 ;
161 subplot->label(PLOT_AXIS_X,"theta2 (degree)");
162 subplot->label(PLOT_AXIS_Y,"theta5 (degree)");
163
164 subplot=mainplot1.getSubplot(2,1);
165 subplot->data2D(theta2, r6_1_2);
166 subplot->title("r6 vs. theta2 (1st circuit, 2nd solution)");
167 subplot->label(PLOT_AXIS_X,"theta2 (degree)");
168 subplot->label(PLOT_AXIS_Y,"r6 (m)");
169
170 mainplot1.plotting();
171
172 /* Display Plots */
173 mainplot2.subplot(3,2);
174 subplot=mainplot2.getSubplot(0,0);
175 subplot->data2D(theta2, theta4_2);
176 subplot->title("theta4 vs. theta2 (2nd solution)");
177 subplot->label(PLOT_AXIS_X,"theta2 (degree)");
178 subplot->label(PLOT_AXIS_Y,"theta4 (degree)");
179
180 subplot=mainplot2.getSubplot(1,0);
181 subplot->data2D(theta2, theta5_2_1);
182 subplot->title("theta5 vs. theta2 (2nd circuit, 1st solution)");
183 ;
184 subplot->label(PLOT_AXIS_X,"theta2 (degree)");
185 subplot->label(PLOT_AXIS_Y,"theta5 (degree)");
186
187 subplot=mainplot2.getSubplot(1,1);
188 subplot->data2D(theta2, r6_2_1);
189 subplot->title("r6 vs. theta2 (2nd circuit, 1st solution)");
190 subplot->label(PLOT_AXIS_X,"theta2 (degree)");
191 subplot->label(PLOT_AXIS_Y,"r6 (m)");
192
193 subplot=mainplot2.getSubplot(2,0);
194 subplot->data2D(theta2, theta5_2_2);
195 subplot->title("theta5 vs. theta2 (2nd circuit, 2nd solution)");
196 ;
197 subplot->label(PLOT_AXIS_X,"theta2 (degree)");
198 subplot->label(PLOT_AXIS_Y,"theta5 (degree)");
199
200 subplot=mainplot2.getSubplot(2,1);

```

```
197 subplot->data2D(theta2, r6_2_2);
198 subplot->title("r6 vs. theta2 (2nd circuit, 2nd solution)");
199 subplot->label(PLOT_AXIS_X, "theta2 (degree)");
200 subplot->label(PLOT_AXIS_Y, "r6 (m)");
201
202 mainplot2.plotting();
203 return 0;
204 }
```