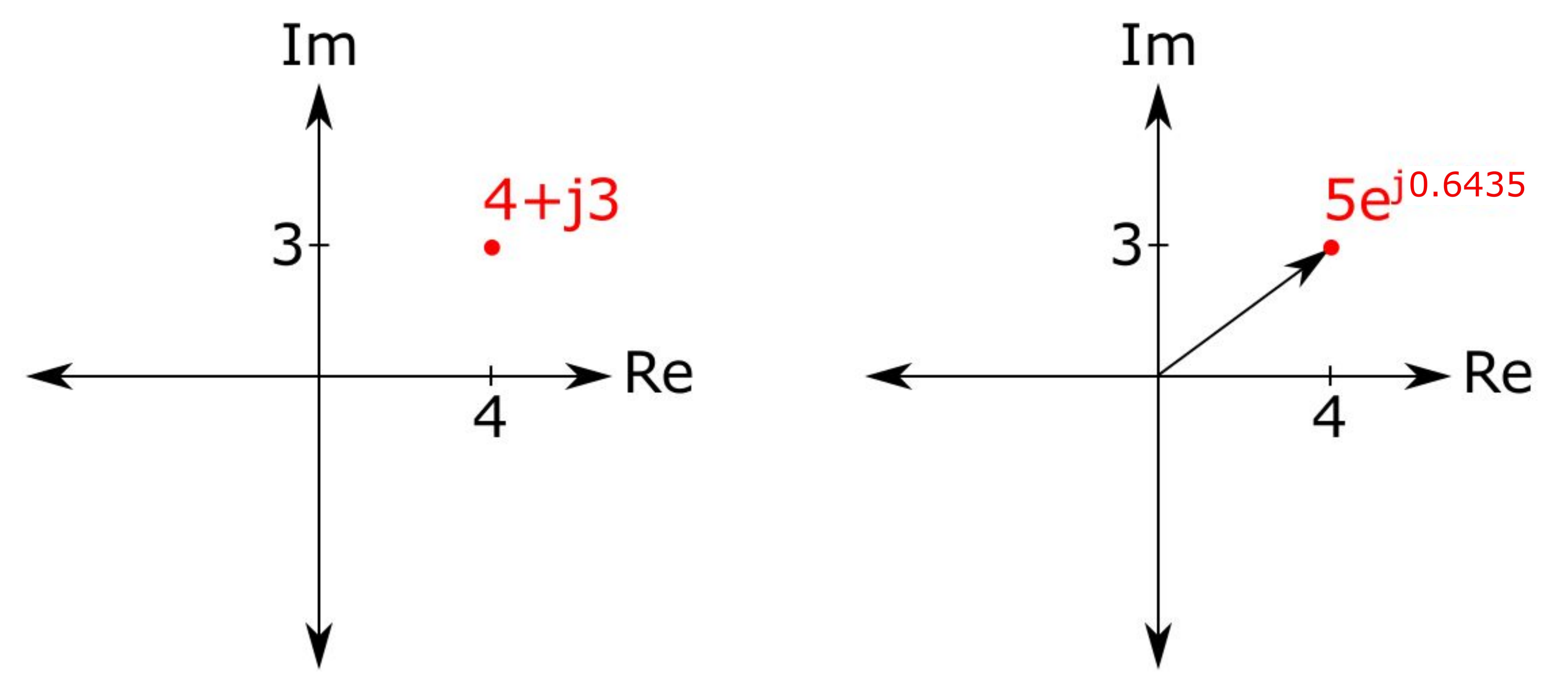
EME 152 Discussion 3

October 13, 2021

Agenda

- Review of complex numbers in Ch
- Solutions for complex equations
- Finding the degrees of freedom of a mechanism

Review



Review

$$3 + j4$$

$$2e^{\frac{j\pi}{4}}$$

$$j$$

$$3e^{-\frac{j\pi}{6}}$$

```c
complex double z;

z = complex(-3, 4);
// -3.000000 + i4.000000

z = polar(2, M_PI / 4.0);
// 1.414214 + i1.414214

z = I;
// 0.000000 + i1.000000

z = 3 * exp(-I * M_PI / 6.0);
// 2.598076 - i1.500000
```

Review

- For `exp()` and `polar()`, make sure that the angle is in radians!
    - Trigonometric functions in C/Ch/C++ also require inputs of radians
- You can use the handy function `deg2rad()` if your angle is in degrees

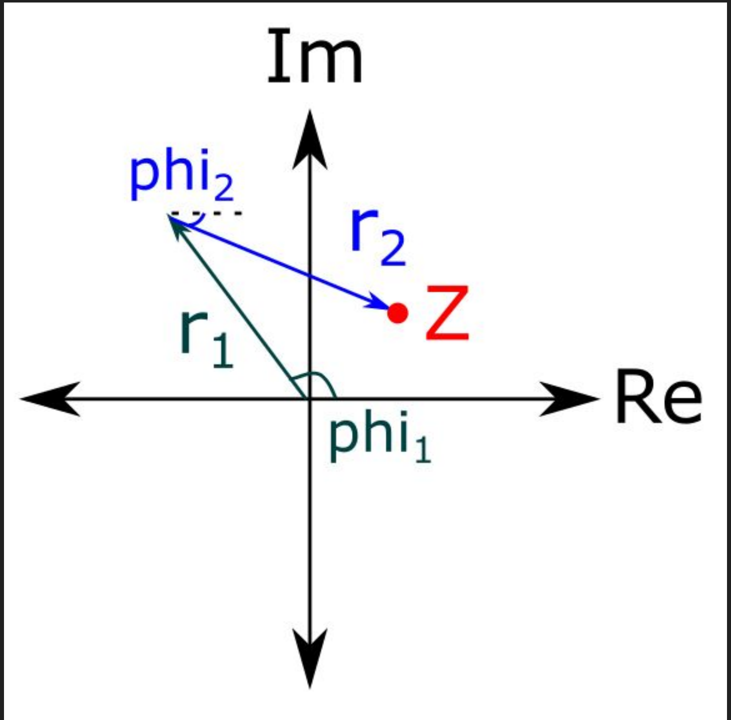
```
C:/Ch/extern/lib> deg2rad(30.0)

0.5236

C:/Ch/extern/lib> M_PI/6.0

0.5236
```

Complex equations with 2 unknowns

This following equation is the basis for analysis and design of planar mechanisms. Complex numbers are used to represent links in the mechanism. Assume Z is a known complex number. You might know both the magnitudes, both the angles, or one angle and one magnitude.

$$r_1 e^{i\phi_1} + r_2 e^{i\phi_2} = Z$$

Complex equations with 2 unknowns



Both magnitudes known, both angles unknown

```
complexsolveRR(r1, r2, Z, phi1_1, phi2_1, phi1_2, phi2_2);
```

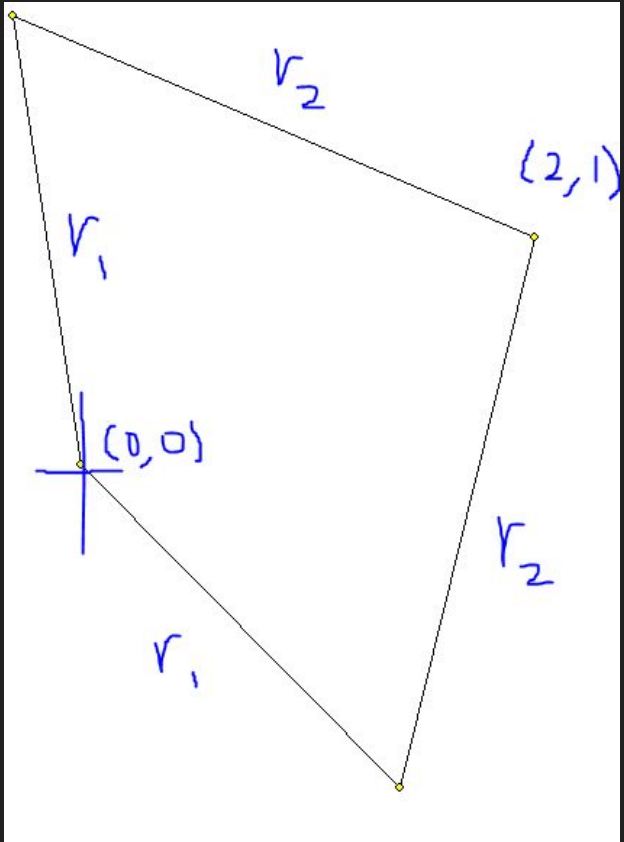
- `r1` and `r2` are the known magnitudes
- `phi1_1` and `phi2_1` are the angles to the first solution
- `phi1_2` and `phi2_2` are the angles to the second solution
- The parameters `phiN_N` are outputs of the function
- This function always returns the integer 2 (the number of solutions)

Both magnitudes known, both angles unknown

Known: $r_1 = 2.0$, $r_2 = 2.5$, $Z = 2 + i$

Find: phi_1, phi_2 (for both solutions)

Both magnitudes known, both angles unknown



Both magnitudes known, both angles unknown

Want to find the angle in degrees? You can use `rad2deg()`

```
C:/Ch/extern/lib> rad2deg(M_PI/6.0)

30.0000

C:/Ch/extern/lib> rad2deg(M_PI)

180.0000
```

One magnitude and one angle known

```
complexsolvePR(phi2, r1, Z, r2_1, phi1_1, r2_2, phi1_2);
```

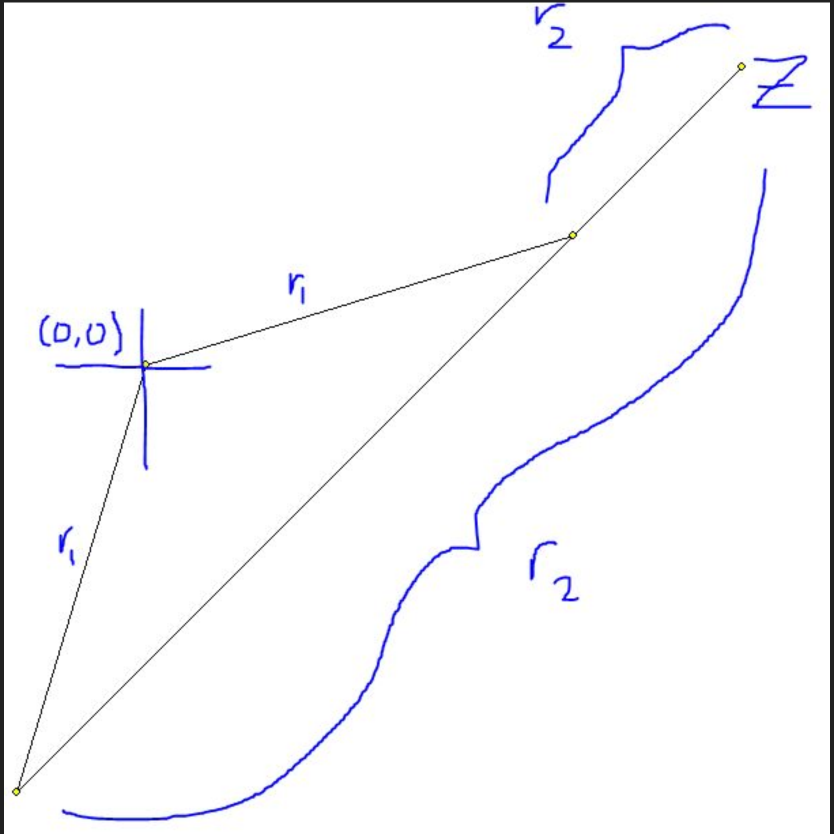
- `phi2` and `r1` are known
- r2_1 and phi1_1 represent the first set of solutions
- `r2_2` and `phi1_2` represent the second set of solutions
- This function always returns the integer 2 (the number of solutions)

One magnitude and one angle known

Known: $r_1 = 1.5$, $phi_2 = 45^o$, $Z = 2 + i$

Find: r_2 and phi_1 (for both solutions)

One magnitude and one angle known



Both angles known, both magnitudes unknown

```
complexsolvePP(phi1, phi2, Z, r1, r2);
```

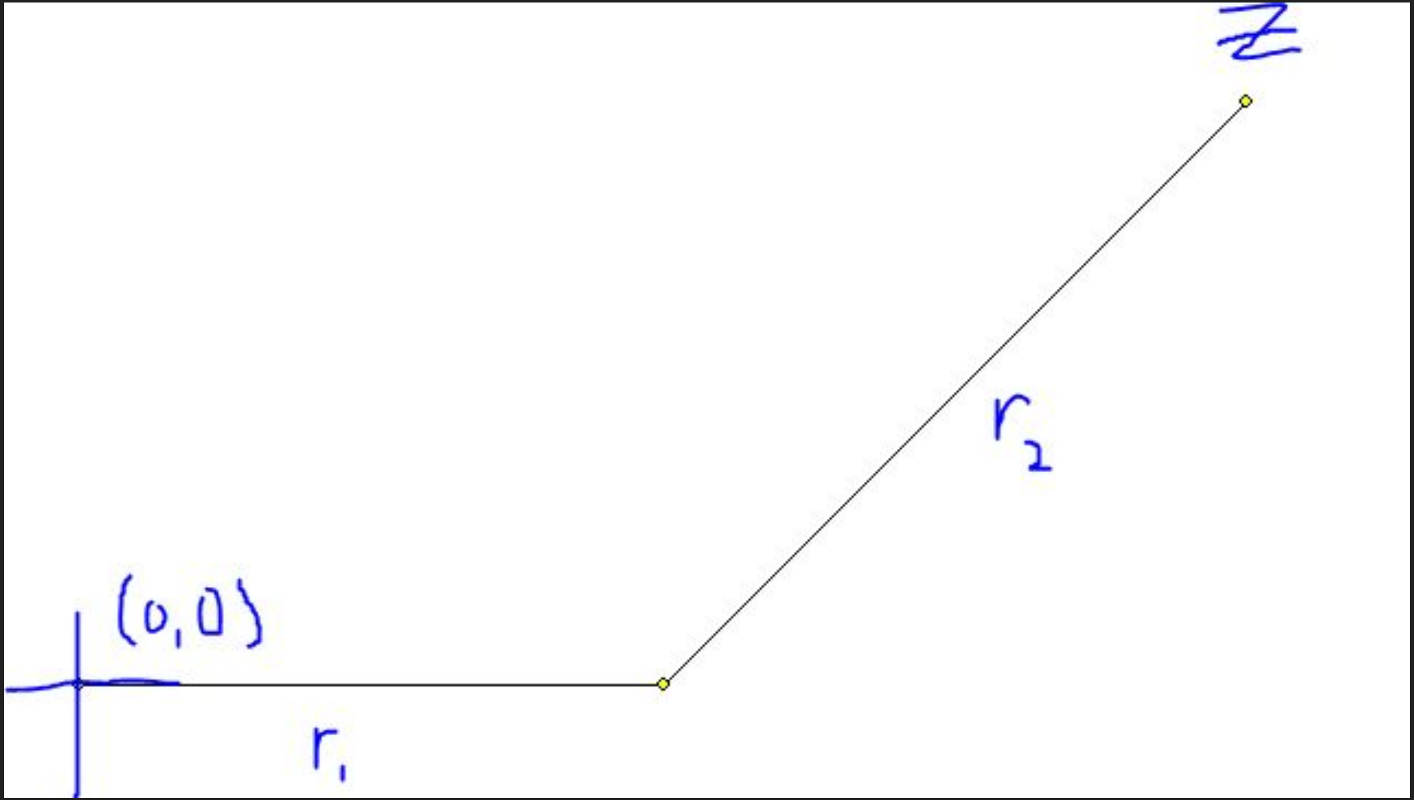
- `phi1` and `phi2` are known
- `r1` and `r2` are unknown, represent the first (and only) set of solutions
- This function always returns the integer 1 (the number of solutions)

Both angles known, both magnitudes unknown

Known: $phi_1 = 0^o$, $phi_2 = 45^o$, $Z = 2 + i$

Find: r_1 and r_2 (one solution)

Both angles known, both magnitudes unknown



One displacement and two angles unknown

The equation can also be solved in this format. It forces a 90º angle between linkages. For example, a cam design. A and Z are known, and r and theta are unknown.

$$(a + ir)\, e^{i\theta} = Z$$

One displacement and two angles unknown

```
complexsolveRP(a, Z, r_1, theta_1, r_2, theta_2);
```

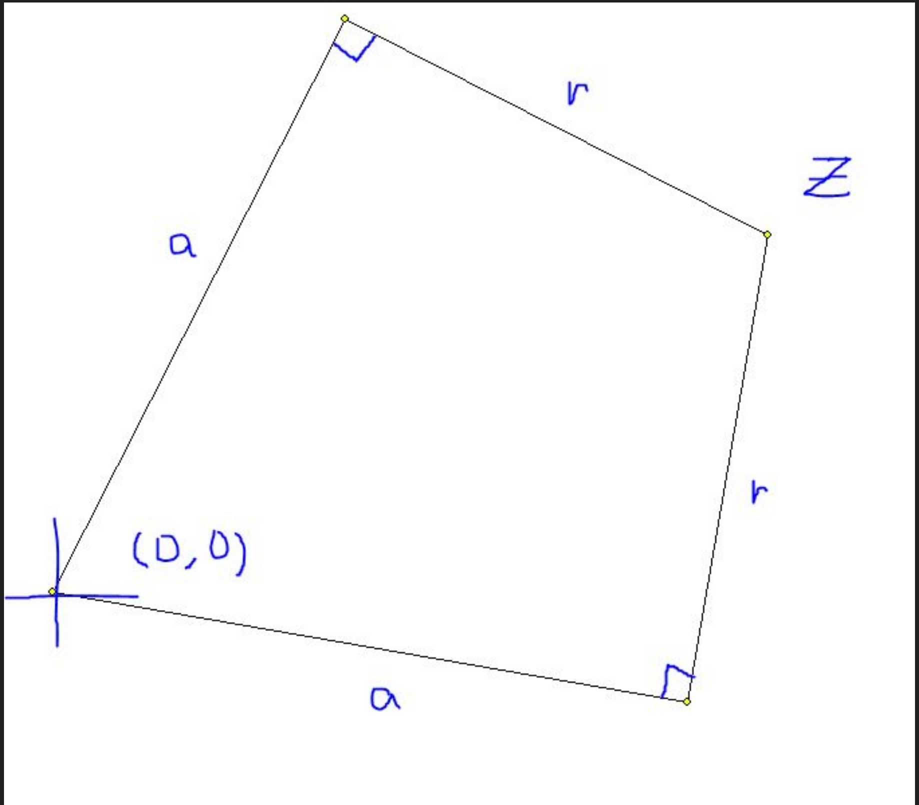
- a and Z are known
- `r_1` and `theta_1` represent the first set of solutions
- `r_2` and `theta_2` represent the second set of solutions
- Returns the integer -1 if no solutions were found, otherwise returns the integer 2

One displacement and two angles unknown

Known: $a = 1.8$, $Z = 2 + i$

Find: r, theta (both solutions)

One displacement and two angles unknown



Recap

Ch Function	Known	Unknown
`complexsolveRR`	r1, r2	phi1, phi2
`complexsolvePR`	r1, phi2	r2, phi1
`complexsolvePP`	phi1, phi2	r1, r2
`complexsolveRP`	a	r, theta

Refer to the Ch mechanism toolkit documentation for more details!

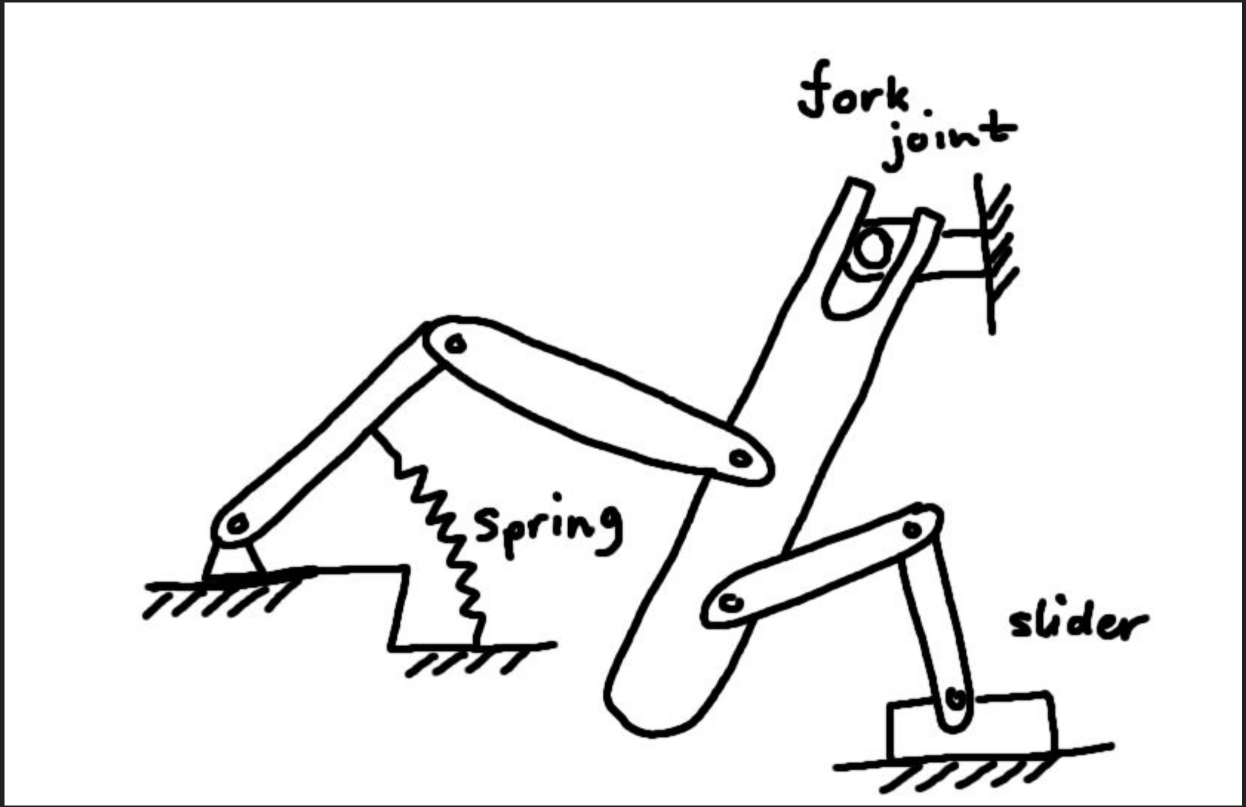
Degrees of Freedom

Use the Gruebler equation:

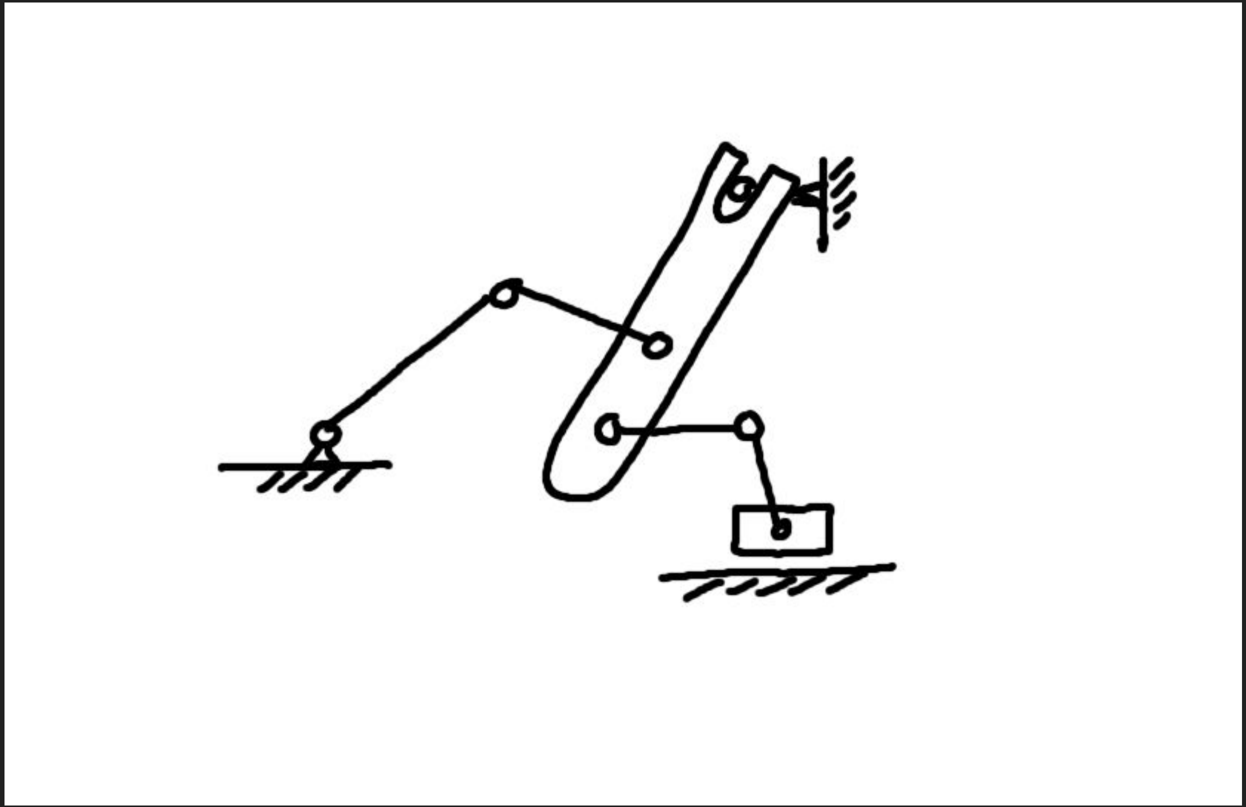
- DOF = degrees of freedom
- n = number of links
- f_1 = number of joints with 1 degree of freedom (e.g. slider, pin)
- f_2 = number of joints with 2 degrees of freedom (e.g. fork joint)

$$DOF = 3(n-1) - 2f_1 - f_2$$

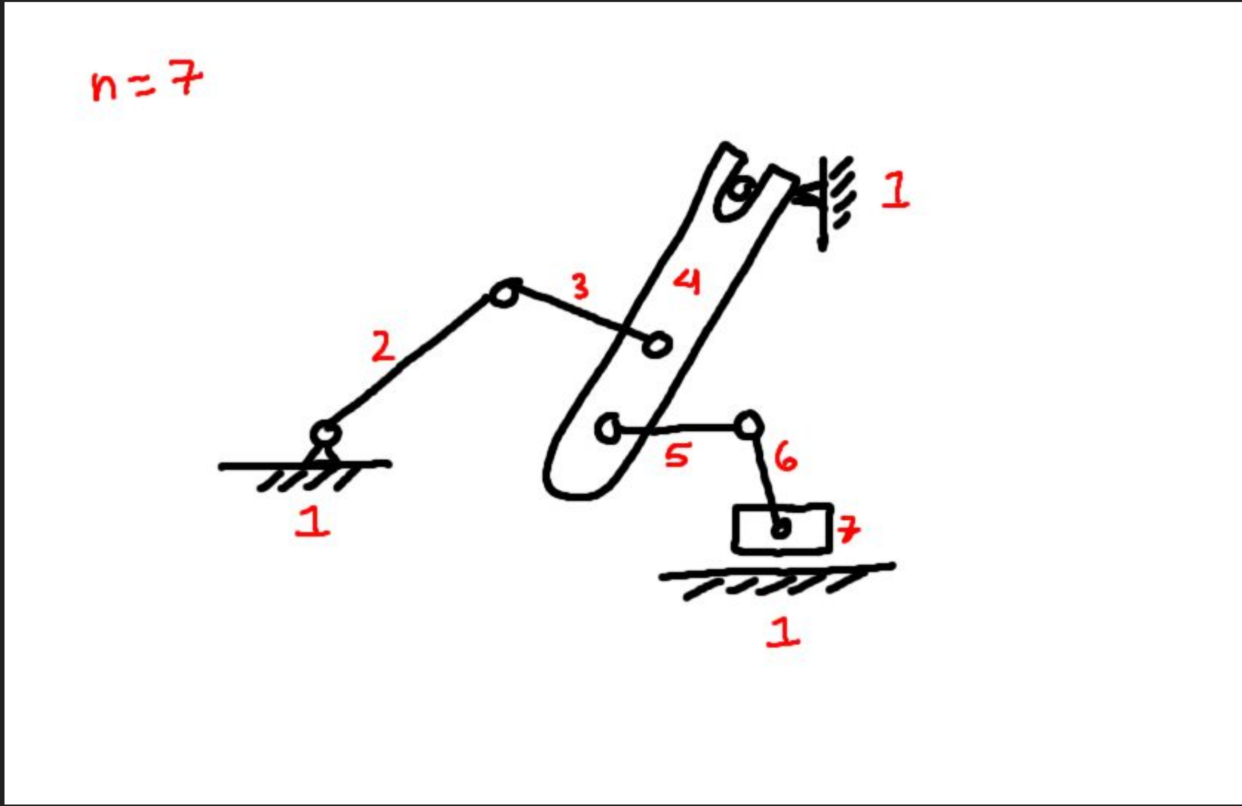
Degrees of Freedom



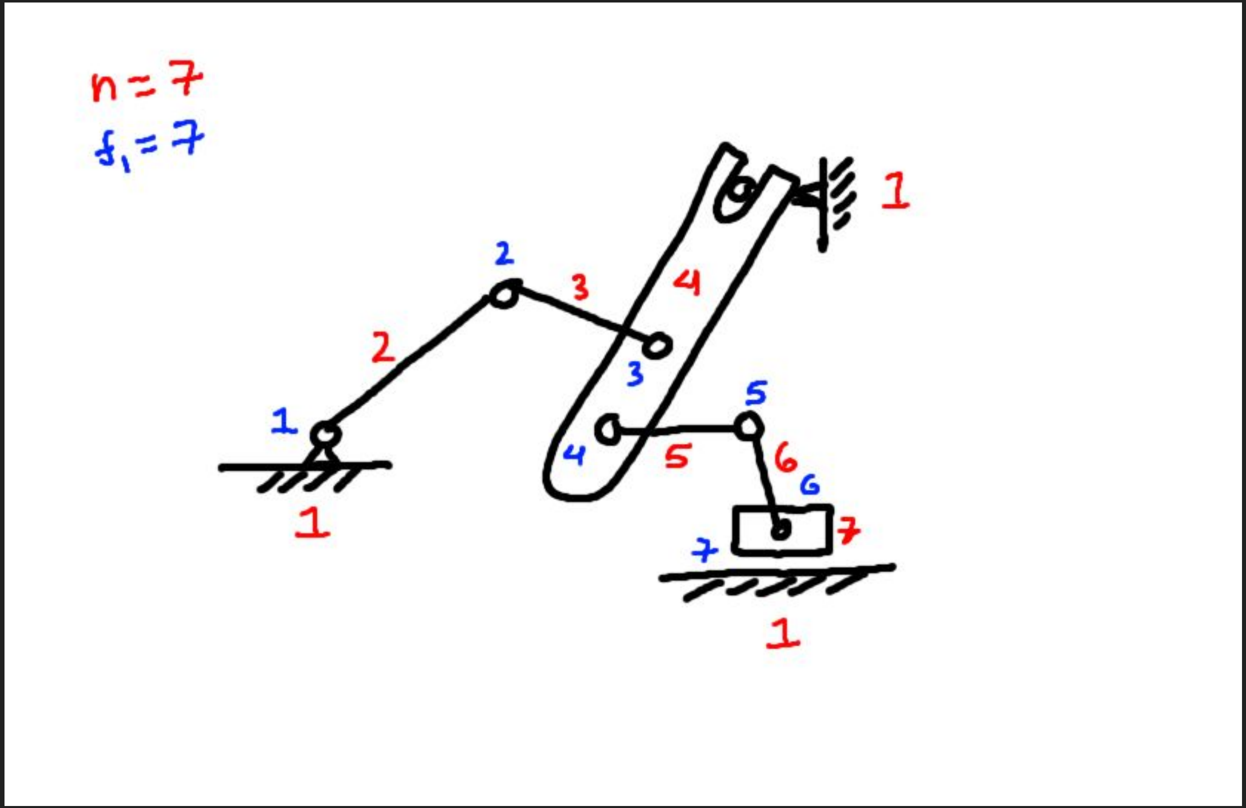
Degrees of Freedom



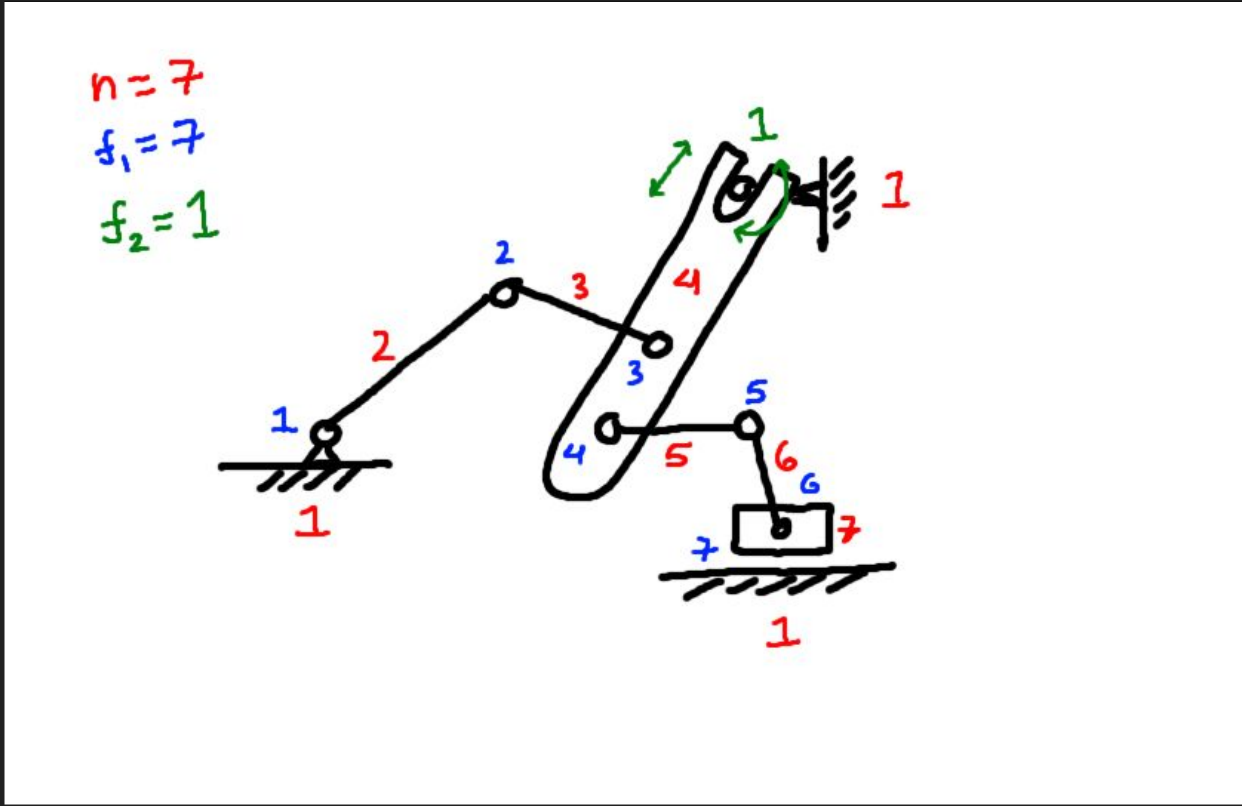
Degrees of Freedom



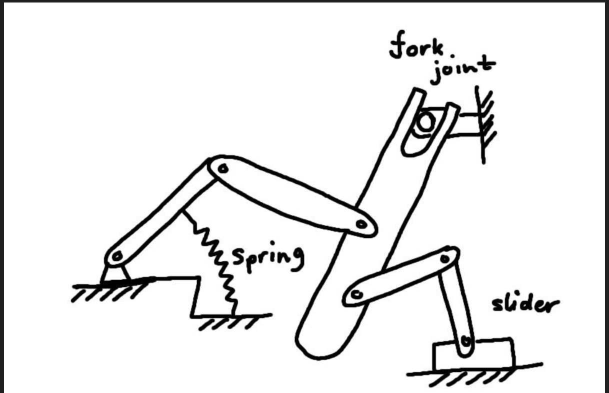
Degrees of Freedom



Degrees of Freedom



Degrees of Freedom



$$\mathrm{DOF} = 3(7-1) - 2 \cdot 7 - 1 = 3$$

Thank you!

Questions?