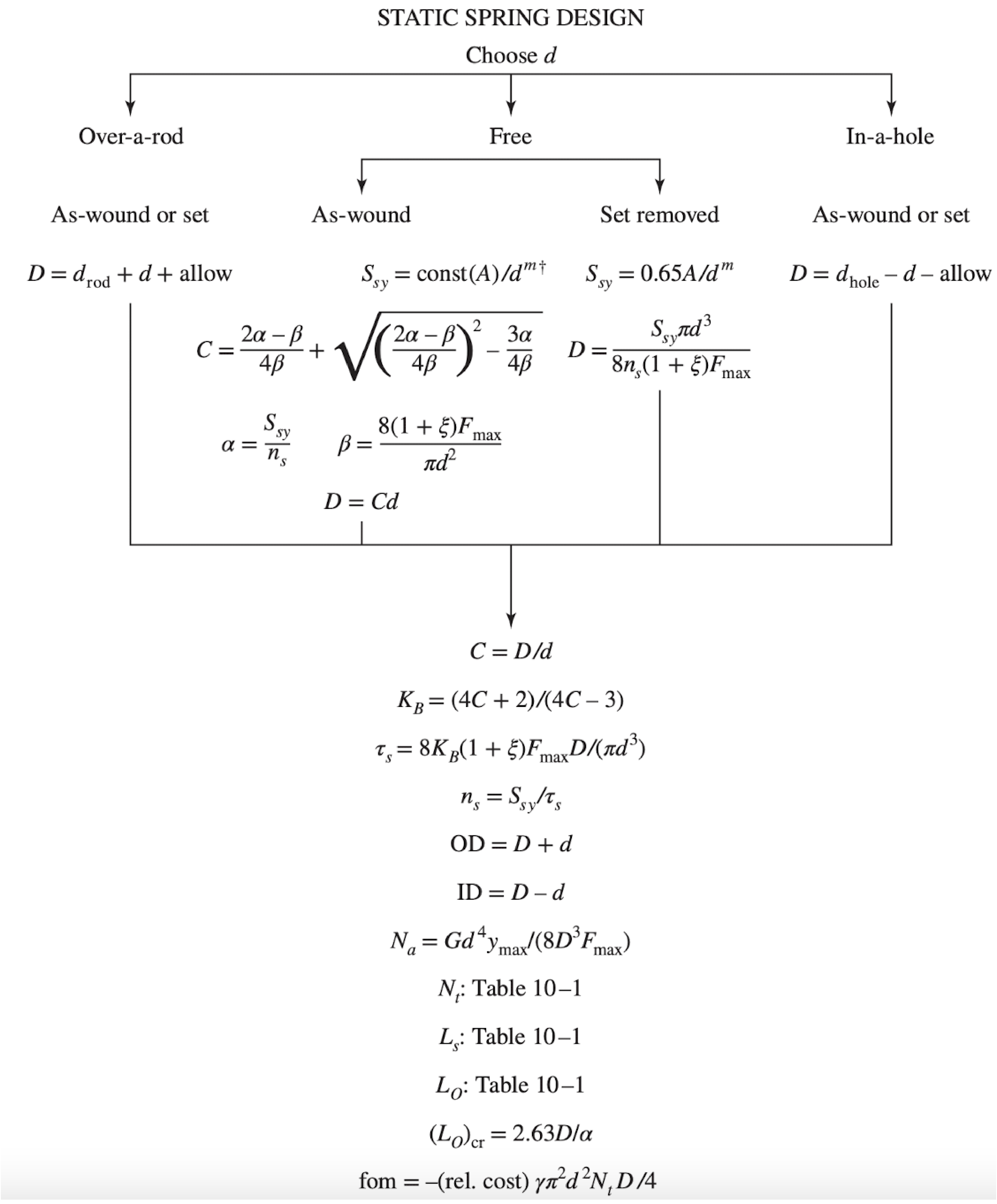
MEMS 1029 - Project 2 - Normal (Round Wire) Spring Design

Author: Ziang Cao [ZIC25]
Data: Feb 26 - Mar 10

In [ ]:
```python
import math
import numpy as np
pi = math.pi
```

Problem Statement

We start from the round wire Spring. As making a realword-product, instad of solving a textbook question: we simplify the design by applying "Over-a-rob" scenario -- which to get rid of the buckling, we add sticks inside of the wire:



Declare The Layout Constant

In [ ]:
```python
# We want to have 5 springs arranged in a parallel way to share the load
N_spring = 5
```

Section1: Find Equivalent K_eq

In [ ]:
```python
# U is in Nm; delta_x is in mm
def return_K(U, delta_x, N_spring=5):
  num = 2*U
  den = pow(delta_x,2)
  K_eq = round((num/den),4)
  text = "For U= {} Nm and delta_x = {} mm, the equivalent K_eq = {} kN/m".format(U, delta_x, K_eq)
  print(text)
  return K_eq      # unit: kN/mm
```

In [ ]:
```python
def get_K(U, delta_x_list = [30, 40, 50, 60, 70]):
  K_eq_list = [None] * len(delta_x_list)
  for i in range(len(delta_x_list)):
    K_eq_list[i] = return_K(U, delta_x_list[i])
  text = "\nThe minimal K is {} kN/mm under U= {} Nm and delta_x = {} mm ".format(min(K_eq_list), U, delta_x_l
  print(text)
  return K_eq_list
```

In [ ]:
```python
U = 10*3600 # Unit: Nm
delta_x_list = [30, 40, 50] # Unit: mm
K_eq_list = get_K(U = U, delta_x_list=delta_x_list)
```

```
For U= 36000 Nm and delta_x = 30 mm, the equivalent K_eq = 80.0 kN/m
For U= 36000 Nm and delta_x = 40 mm, the equivalent K_eq = 45.0 kN/m
For U= 36000 Nm and delta_x = 50 mm, the equivalent K_eq = 28.8 kN/m

The minimal K is 28.8 kN/mm under U= 36000 Nm and delta_x = 50 mm
```

Select the desire delta_x and corresponding K_eq, K_each

In [ ]:
```python
delta_x = 50   # mm
K_eq = K_eq_list[-1]   # kN/mm
K_each = K_eq/N_spring   # kN/mm
print("The K for each individual Spring is {} in kN/mm".format(K_each))

F_max = delta_x * K_each
print("\nThe max Force on each Spring become {} kN".format(F_max))

y_max = delta_x
print("\nThe max delta x on each Spring is {} mm".format(y_max))
```

```
The K for each individual Spring is 5.76 in kN/mm

The max Force on each Spring become 288.0 kN
```

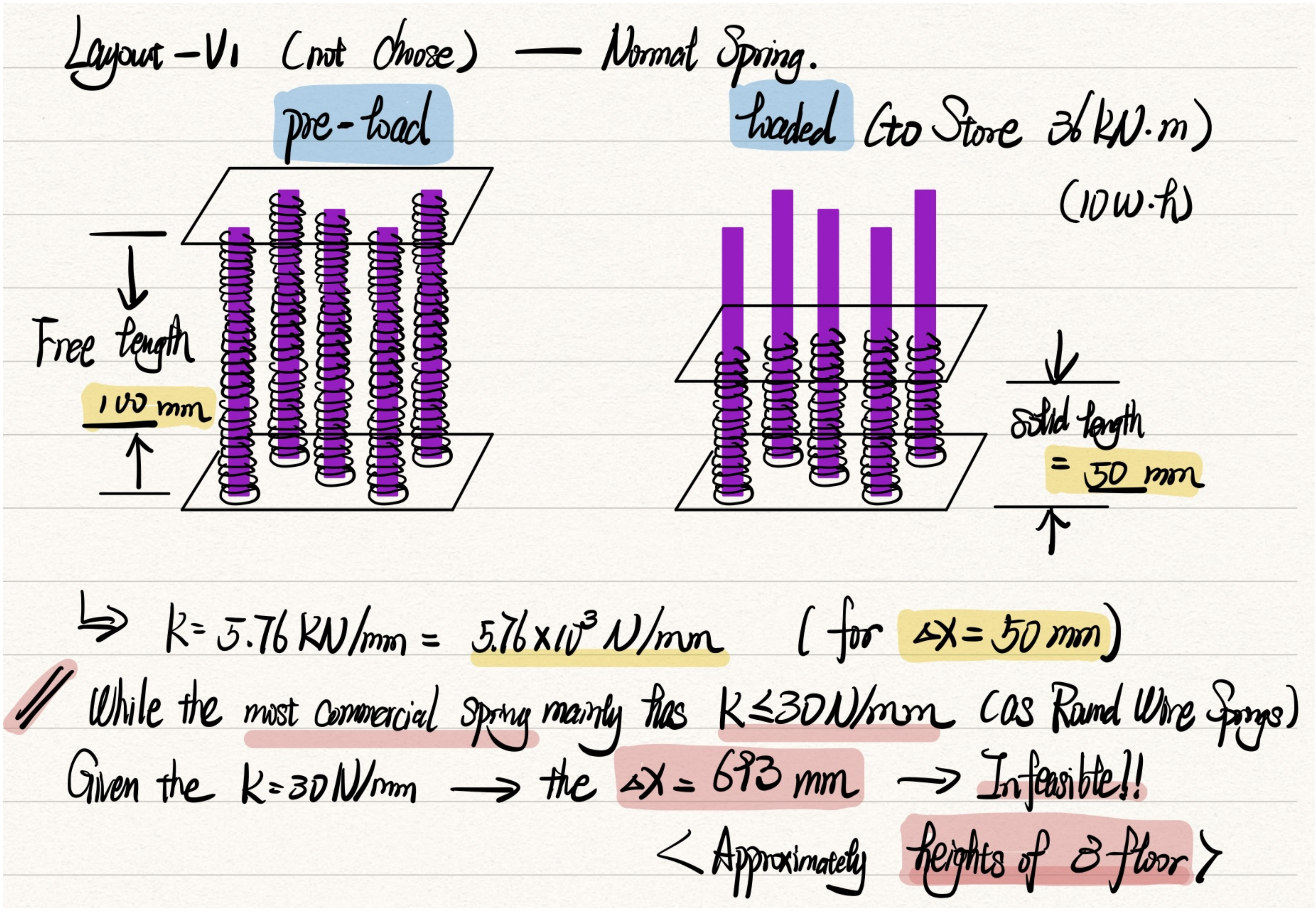
Section2: Find Equivalent delta_x

In [ ]:
```python
# U is in Nm; delta_x is in mm, K in in N/mm
def return_delta_x(U, K, N_spring=5):
  num = 2 * U * 1e3    # N*mm
  den = K * N_spring
  val = num / den
  delta_x = pow(val ,0.5)
  print("The required delta_x is {} mm, given by most commerical K={} N/mm".format(round(delta_x), round(K)))
  return delta_x
```

In [ ]:
```python
U_require = 10*3600 # Unit: Nm
K_usual = 30   # N/mm
delta_x_required = return_delta_x(U=U_require, K=K_usual, N_spring=5)
```

```
The required delta_x is 693 mm, given by most commerical K=30 N/mm
```

Overall Sketch of this Infeasible design



Stop for Infeasibility

As this design is indeasible at all, we switch to another types of Spring like "torque" and "rotor" Spring, and stop the analysis for round wire one.