

```

1  mcdp {
2      # We need to fly for this duration
3      provides endurance [s]
4      # While carrying this payload
5      # (in addition to battery)
6      provides extra_payload [kg]
7      # And providing this extra power
8      provides extra_power [W]
9
10     # Sub-design problem: choose the battery
11     sub battery = instance dp {
12
13         provides capacity [J]
14         requires mass [kg]
15
16         implemented-by load BatteryDP
17     }
18     # Sub-design problem: actuation
19     sub actuation = instance abstract mcdp {
20         # actuators need to provide this lift
21         provides lift [N]
22         # and will require power
23         requires power [W]
24         # simple model: quadratic
25         c = 0.002 W/N^2
26         power >= lift * lift * c
27     }
28     # Co-design constraint: battery must be large enough
29     power = actuation.power + extra_power
30     energy = power * endurance
31     battery.capacity >= energy
32
33     # Co-design constraint: actuators must be powerful enough
34     gravity = 9.81 m/s^2
35     weight = (battery.mass + extra_payload) * gravity
36     actuation.lift >= weight
37
38     # suppose we want to optimize for size of the battery
39     requires mass for battery
40 }

```