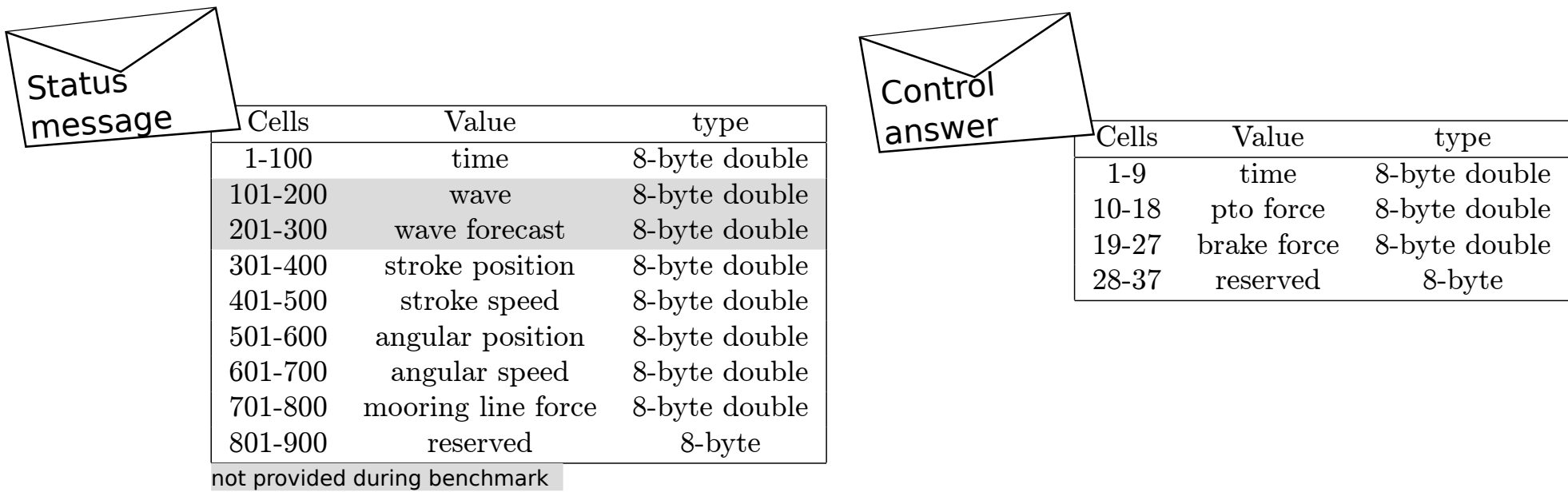


COERbuoy1: writing a controller

The control interface allows to write the controller independent from the model. This allows to implement controller for COERBuoy1 with minimal effort.

The interface consists of the status message send from the model to the controller and the control message, the controller's answer to the model. The communication runs over the TCP/IP interface, which can be used with every programming language.



Linear damping controller in python

```
1  #!/usr/bin/env python3
2  # -*- coding: utf-8 -*-
3  """
4  Simple linear damping
5  @author: Simon H. Thomas, COER laboratory, Maynooth University
6  """
7  import numpy as np;
8  from scipy.interpolate import interp1d;
9  import COERbuoy.connection as connection;
10
11 conn_model=connection.connection();#Initialize connection
12 conn_model.openC();#Use client mode
13
14 while msg:=conn_model.get_control():
15
16     ##Read incoming message
17     time =msg["time"]           #1x100 array with time series
18     wave =msg["wave"];          #1x100 array with wave data (related to time series)*
19     wave_f=msg["wave_forecast"];#1x100 array with wave forecast*
20     x     =msg["stroke_pos"];    #1x100 array with stroke position (related to time series)
21     dx    =msg["stroke_speed"];  #1x100 array with stroke speed (related to time series)
22     alpha =msg["angular_pos"];   #1x100 array with pitch angle (related to time series)
23     dalpha=msg["angular_speed"];#1x100 array with pitch angular speed (related to time
series)
24     force =msg["force"];         #1x100 array with force sensor data (related to time series)
25     #* not available during COERbuoy1 benchmark
26     now=time[-1]; #The last element contains the most recent data (except the wave forecast)
27
28     ##This part contains the controller's logic
29     ##TODO: replace logic with own controller idea
30
31     #(here we use a simple linear damping)
32     #Calculate PTO force
33     gamma=100000;#kNs/m
34     F_pto=-gamma*dx[-1];
35     #In this example we don't use the WECs brake
36     brake=0;#kNs/m
37
38     ##Write control message
39     answer={
40         "time":np.linspace(now,now+1,9),#1x9 array with time steps in the future
41         "pto":np.array([F_pto]*9),      #1x9 array with PTO force
42         "brake":np.zeros(9),            #1x9 array with brake force
43         "test":np.zeros(9),            #1x9 array; not used
44     }
45     #Send control message to model
46     conn_model.set_control(answer["time"],answer["pto"],answer["brake"],answer["test"]);
47
48 conn_model.close();
49
```

Control algorithm

COERbuoy1
(model, ODE solver, user interface)