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.

Status message

Cells	Value	type
1-100	time	8-byte double
101-200	wave	8-byte double
201-300	wave forecast	8-byte double
301-400	stroke position	8-byte double
401-500	stroke speed	8-byte double
501-600	angular position	8-byte double
601-700	angular speed	8-byte double
701-800	mooring line force	8-byte double
801-900	reserved	8-byte

not provided during benchmark

Control answer

Cells	Value	type
1-9	time	8-byte double
10-18	pto force	8-byte double
19-27	brake force	8-byte double
28-37	reserved	8-byte

Linear damping controller in python

```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Simple linear damping
@author: Simon H. Thomas, COER laboratory, Maynooth University
"""
import numpy as np;
from scipy.interpolate import interp1d;
import COERbuoy.connection as connection;

conn_model=connection.connection();#Initialize connection
conn_model.openC();#Use client mode

while msg:=conn_model.get_control():

  ##Read incoming message
  time  =msg["time"]          #1x100 array with time series
  wave  =msg["wave"];         #1x100 array with wave data (related to time series)*
  wave_f=msg["wave_forecast"];#1x100 array with wave forecast*
  x     =msg["stroke_pos"];   #1x100 array with stroke position (related to time series)
  dx    =msg["stroke_speed"]; #1x100 array with stroke speed (related to time series)
  alpha =msg["angular_pos"];  #1x100 array with pitch angle (related to time series)
  dalpha=msg["angular_speed"];#1x100 array with pitch angular speed (related to time
series)
  force =msg["force"];        #1x100 array with force sensor data (related to time series)
  #* not available during COERbuoy1 benchmark
  now=time[-1]; #The last element contains the most recent data (except the wave forecast)

  ##This part contains the controller's logic
  ##TODO: replace logic with own controller idea

  #(here we use a simple linear damping)
  #Calculate PTO force
  gamma=100000;#kNs/m
  F_pto=-gamma*dx[-1];
  #In this example we don't use the WECs brake
  brake=0;#kNs/m

  ##Write control message
  answer={
          "time":np.linspace(now,now+1,9),#1x9 array with time steps in the future
          "pto":np.array([F_pto]*9),      #1x9 array with PTO force
          "brake":np.zeros(9),            #1x9 array with brake force
          "test":np.zeros(9),             #1x9 array; not used
          }
  #Send control message to model
  conn_model.set_control(answer["time"],answer["pto"],answer["brake"],answer["test"]);

conn_model.close();
```

Control algorithm

Status message

Control answer

COERbuoy1 (model, ODE solver, user interface)