

海洋調査用ブイの水深制御 のためのPID制御 パラメータの求め方

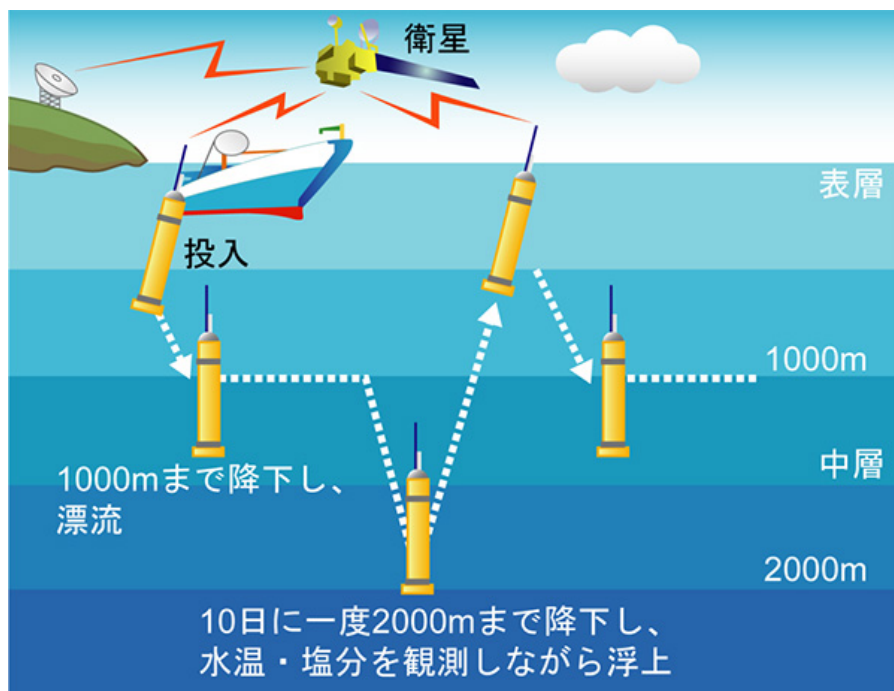
Katsuhiro MORISHITA

Created: 2014-02-01

Uupdate: 2017-03-16

背景

海洋調査では、アルゴフロートなどのブイが使われている。これらのブイでは、浮力を調整しながら観測水深を制御している。



画像参照元：
http://www.jamstec.go.jp/j/about/press_release/20160129/img/image002.jpg

目的

定水深制御にPID制御を利用し、その制御パラメータを求めるプログラムを作成する。

作成したプログラムPD.py

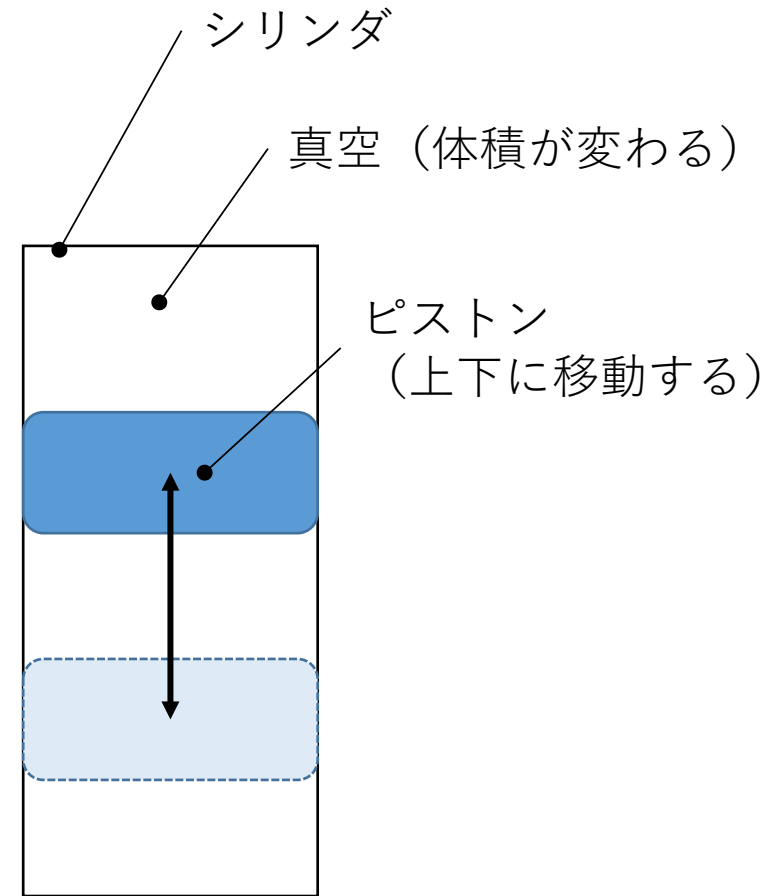
PIDの内、PDのパラメータを求めるPythonスクリプトを作成した。下記よりダウンロードできる。

Download:

<https://github.com/KatsuhikoMorishita/buoy/blob/master/PD.py>

プログラムの仮定

- 重力加速度
 - $9.80[\text{m/s}^2]$
- ブイの質量
 - $20.0 [\text{kg}]$
- 鉛直下向きを正
- 体積変化量の範囲
 - $0.00024 \sim -0.00016 [\text{m}^3]$
- 周囲流体の密度
 - $1035 [\text{kg/m}^3]$
- 水深によって水の密度を変えない
- 計算時間ステップ
 - $0.01 [\text{s}]$
- 最大移動速度（目安）
 - $0.5 [\text{m/s}]$
- 制御量の飽和をさせない
 - 別にしてもいいはずなので、緩和してもいいかも
- 中性浮力から計算をスタート
- ブイの初速は $0.0[\text{m/s}]$
- ブイ周囲の流体は上下に移動しない
- ブイ周囲の流体から浮力以外の力を受けない
 - 減衰項を無視



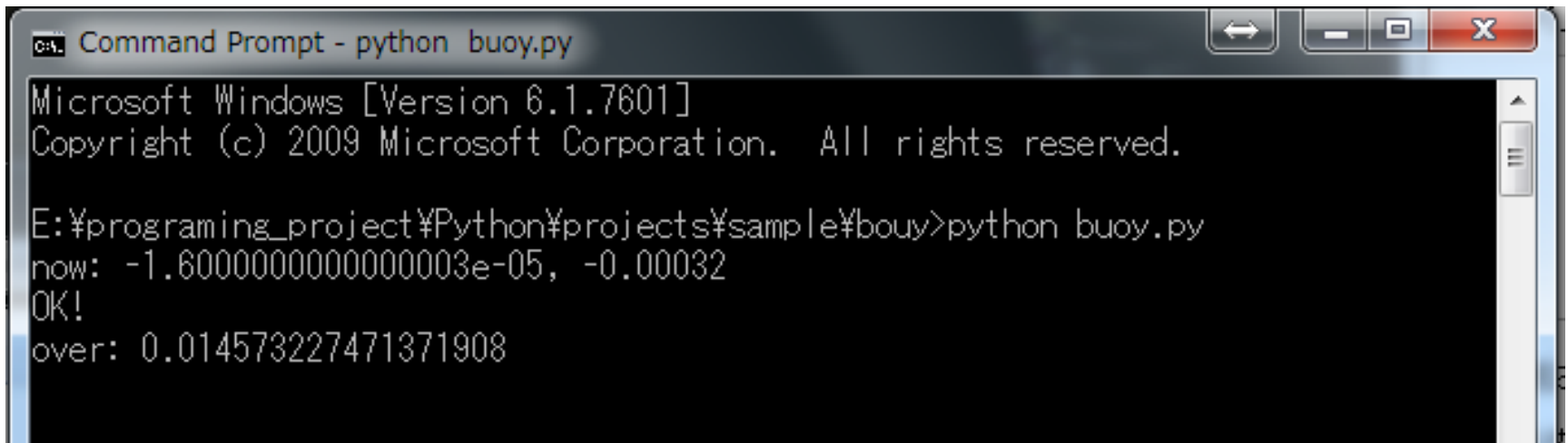
モータによる体積変化機構により浮力を変化させる構造を前提とする。

実行上の注意事項

- 大量のファイルが作成されます
- 数Gbyteを超えます

実行の方法

- Pythonをインストールしてください
- ターミナル/コマンドプロンプトから実行
 - `$ python buoy.py` + エンターキー
- 実行すると良さそうな結果をcsvに保存する

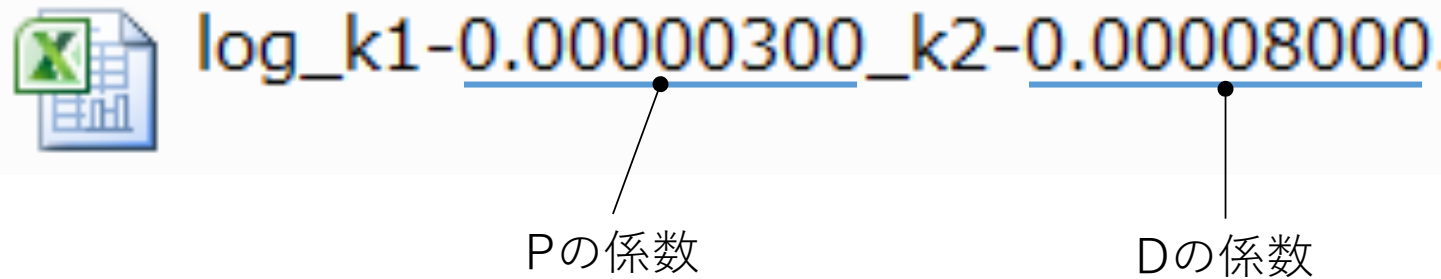
A screenshot of a Windows Command Prompt window. The title bar reads "Command Prompt - python buoy.py". The window has standard Windows window controls (back, forward, minimize, maximize, close) on the right. The text inside the window is as follows:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

E:\programing_project\Python\projects\sample\bouy>python buoy.py
now: -1.6000000000000003e-05, -0.00032
OK!
over: 0.014573227471371908
```

実行の様子

作成されるファイル名の意味



ファイル名がPID制御のPとDの係数を表している。

計算結果の見方

	A	B	C	D	E	F
1	0	0	0.015215	0	3.00E-05	0.30429
2	0.01	1.52E-06	0.015208	0.000152	3.00E-05	0.304166
3	0.02	4.56E-06	0.015202	0.000304	3.00E-05	0.304043
4	0.03	9.13E-06	0.015196	0.000456	3.00E-05	0.30392
5	0.04	1.52E-05	0.01519	0.000608	3.00E-05	0.303796

時間[s]

深さ[m]

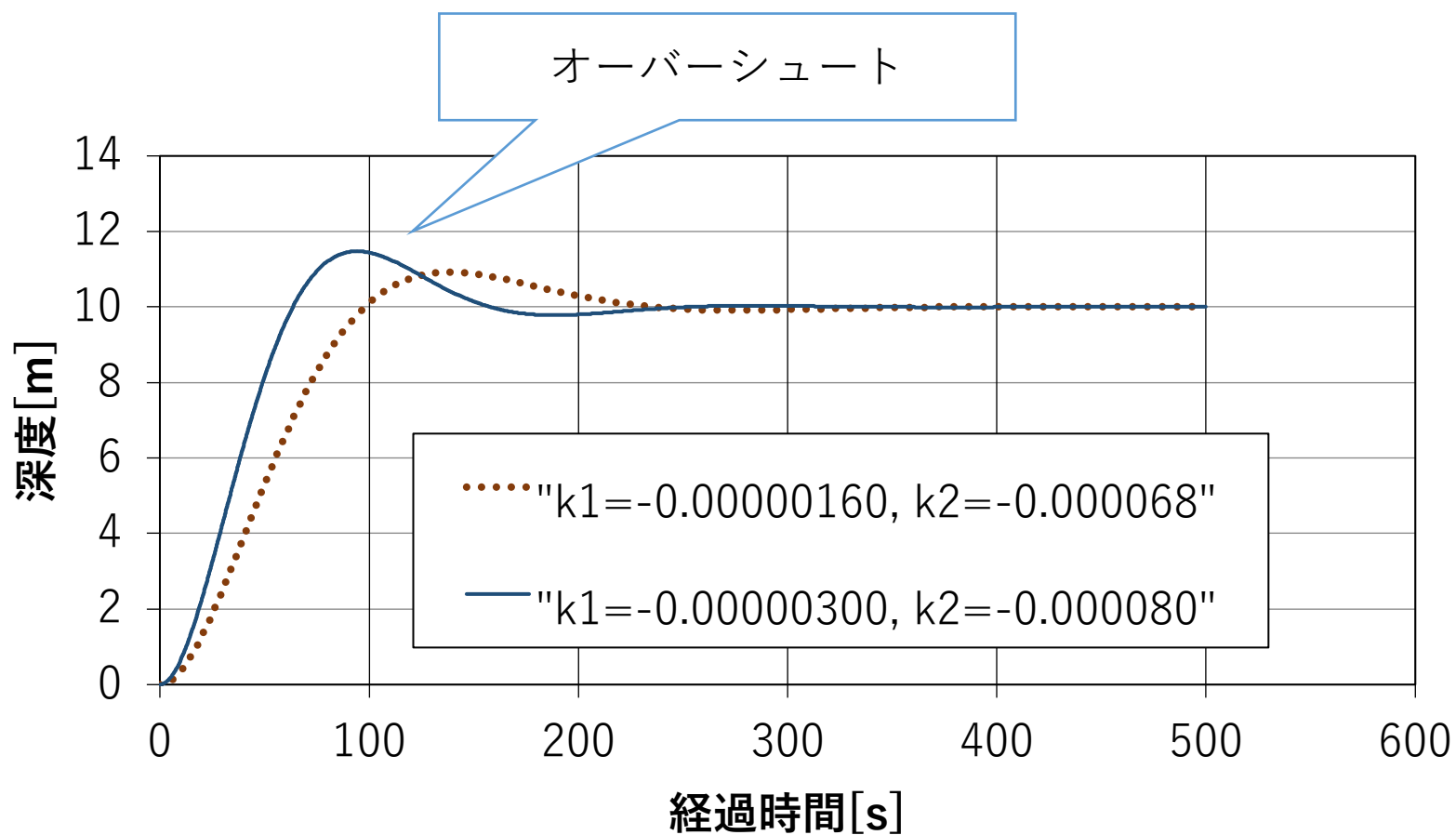
ブイの加
速度
[m/s²]

ブイの移
動速度
[m/s]

体積変化
量[m³]

浮力[N]

計算結果例



パラメータの選択方法

例えば、以下の指標を基に選択すれば良い

- 目標深度までの時定数
- オーバーシュート量

課題

ブイ周囲の流体の湧昇/下降流の影響の大きさによっては、ブイが永遠に目標深度に達さないこともあり得る。

- 対策としては、PIDのIの制御を入れることが考えられる
- また、デフォルトでは周囲の流体速度は0とし、かつ流体との相対速度による抗力を無視しているので、いくつか修正が必要
- なお、周囲の流体速度が大きい場合、現在仮定している最大移動速度 $0.5[\text{m/s}]$ では対応できないかもしれない

PIDの積分項Iを考慮する方法

一定時間の偏差（現時点での深度 - 目標深度）を積分し、これに係数 k_3 をかけたものを積分項として追加すれば良い

$$\text{delta_V_ideal} = k1 * \text{delta_z} + k2 * v$$

I成分を加えるために変更する箇所のコード

$$\text{delta_V_ideal} = k1 * \text{delta_z} + k2 * v + k3 * \text{delta_z_integral}$$

I成分を加えた例

ここで、積分項を`delta_z_integral`で表している。
`delta_z_integral`は別に計算しておく必要がある。

*Iを考慮すると計算時間が数十倍になり得るので注意のこと

流体速度の影響の追加の仕方

周囲の流体速度 v_e を定義し、抗力 cd を計算した後に加速度の式に加える。なお、抗力は流体との相対速度の関数である。

$$a = (\text{buoyant_force}) / m$$

$$a = (\text{buoyant_force} + cd) / m$$

周囲の流体から受ける抵抗を追加する例
(左：デフォルト、右：抗力を追加した様子)

移動速度の制限値を変更する場合

下記の“0.5”の部分を変更する

$$k2_search_width = 3 * \text{delta_V_small} / 0.5$$

制御量の飽和を許容する場合

下記のコード部分の係数を調整して下さい

- 係数は正の数値とする
- 2.0～3.0がいいかも

```
k1_search_width = 9 * delta_V_small / abs(z_0 - z_target)  
k2_search_width = 3 * delta_V_small / 0.5
```


その他

座標の計算を行うときに利用する移動速度 v と体積変化量の計算に利用する v を分けて、後者の v に観測値としての誤差を含めて応答を調べてもいいのかもしれない

周囲の状況に合わせて動的にパラメータを変更するなら・・・

- カルマンフィルタを利用
 - 難易度は高い
- MATLABの利用を推奨
- 実装の際は、センサーのヘルス状態の判定も必要になる