

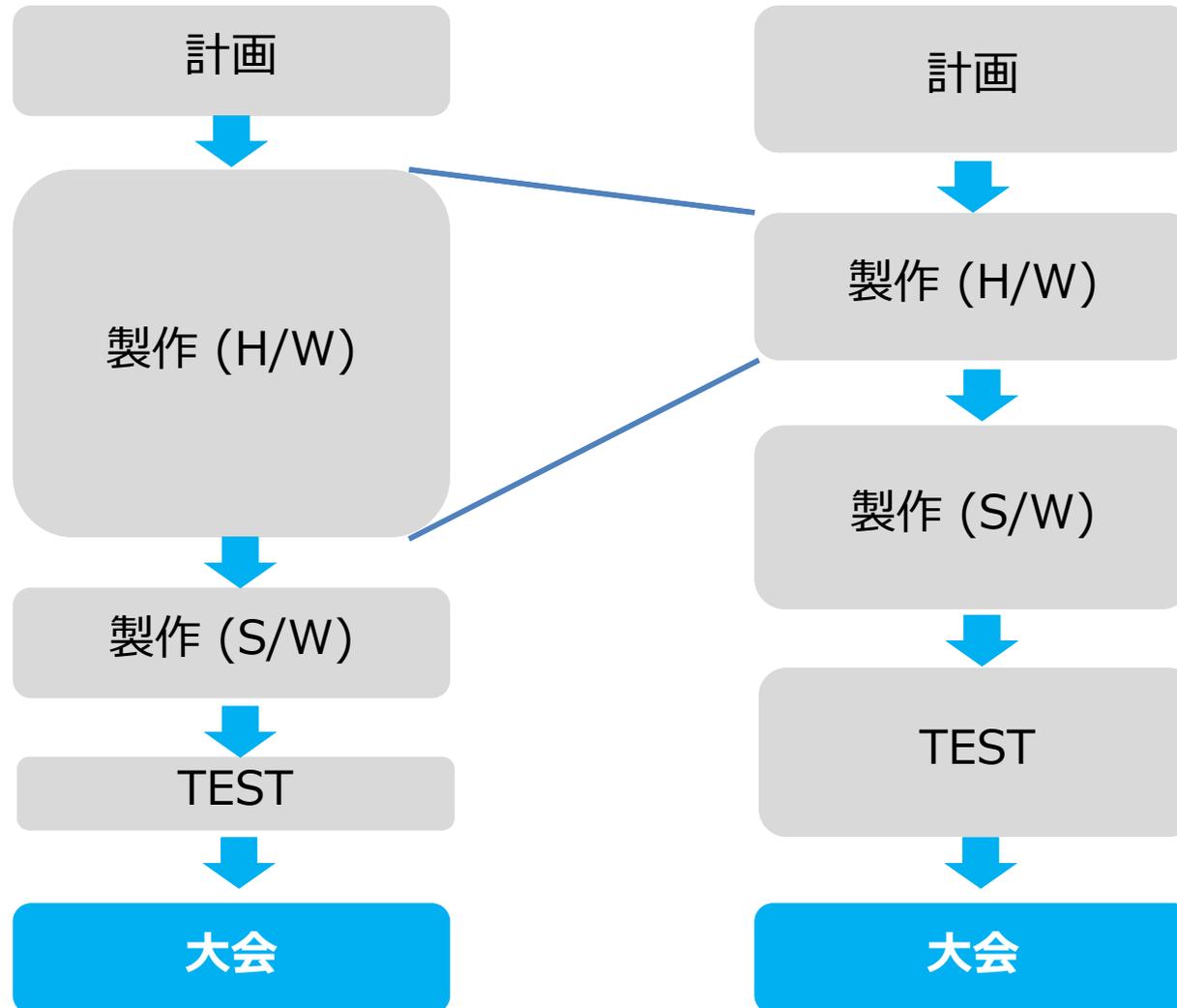
# 海中ロボットのデザイン

## —第1回目：システムデザインとハードウェア—

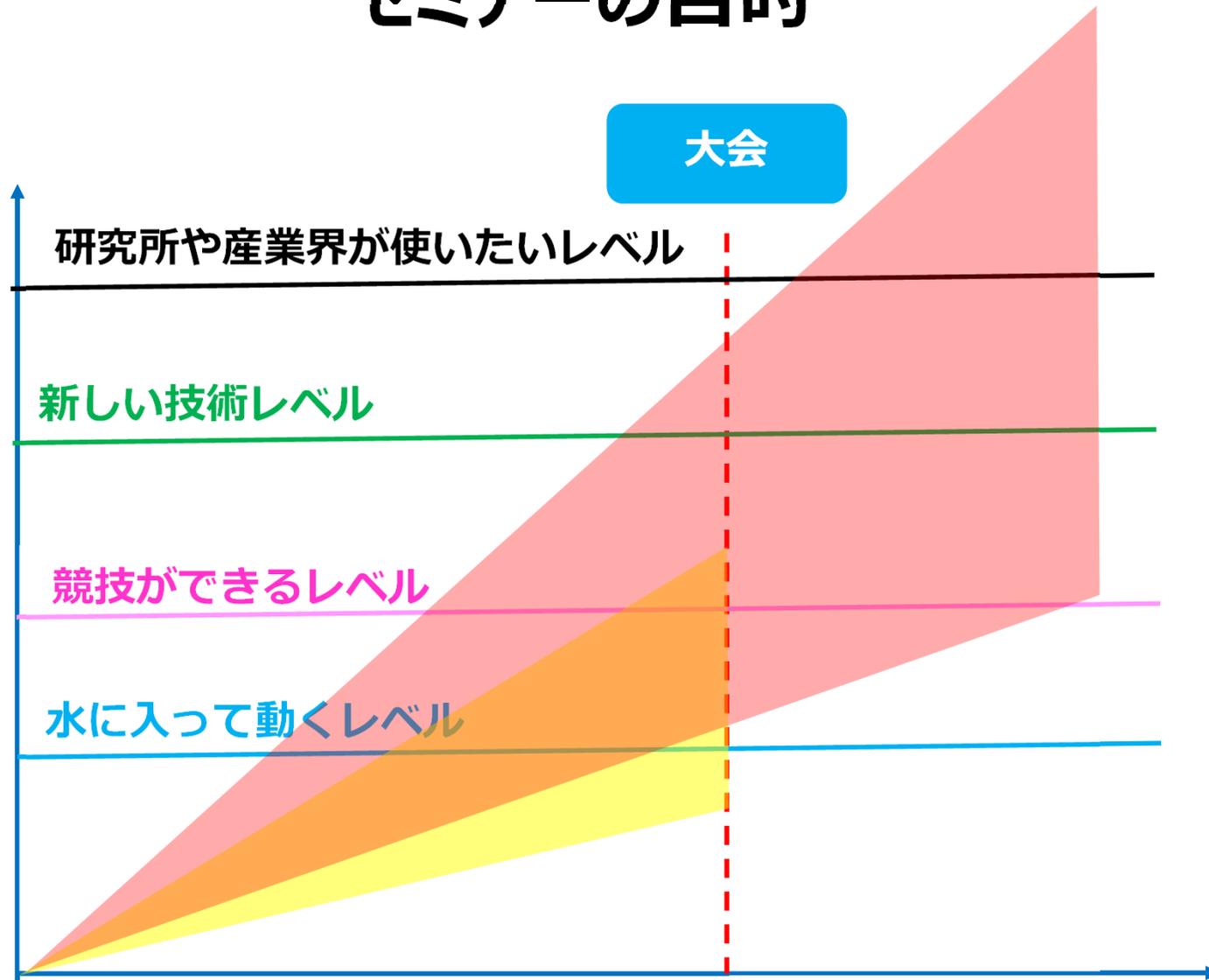
吉田 弘

2019年6月15日  
東京大学 生産技術研究所  
第1回セミナー

# セミナーの目的



# セミナーの目的



# 特に強化したいところ

知能化

センシング

# 5大 チェックポイント

# 1、海水に入れるまえに

君の作ったロボットが、海中で**電源を入**れて

**30分**以上無事だったら

製作の**5割**はクリアーできた！

海水に入れる前に

- ① **真水中で動か**し、浸水の無い事を確認
- ② **真水**のプールに**一晚**浸けてる

本番では

絶対に直前に**耐圧容器を開けるな**！！

## 2. フィールドで蓋をあけるな

ロボットは本番前に**ばらしてはいけない**  
たとえ、電源ONなどの理由があっても

### 電源ON/OFFは外からできるようにする

- マグネットスイッチ
- 赤外線・電波
- 有線（水中コネクター）

### ソフトのパラメータ変更も外からできるようにする

- 電波
- 有線（水中コネクター）

**先生に頼んで水中コネクターを買ってもらおう！**

水中レセプタクル 1個

水中ケーブル 2個（ソフト設定用&充電用、電源on用）

### 3. ソフト変更は、し易くない！

君たちは、ソフトは**簡単に変更できる**と思っているだろう

**勘違いです！**

いじるのは簡単だが**デバッグ**はタイヘン



パラメータは**ファイル読込**や**I/F画面**で変更できるようにする

海水に入れる前に

- ① 実環境で**2時間**以上うごかせ
- ② 本番と**同じ行動**をさせてチェック

本番では

絶対に直前に**ソフトウェアをいじるな！！**

## 4. 電源容量は本当に足りているか？

**モータ**が最大の電量消費パーツ  
次に大きいのはライト

チェックポイント

- ① モータのワット数 = **消費電力**ではない
- ② 電線の**太さ**は適正か？

海水に入れる前に

モータの消費電力は空気中と**水中**で  
**大きく違う**ことを理解しているか？

## 5. WiFiの罨におちるな

WiFiはとっても便利

しかし

2つの物理を把握しておけ！

**海水**は激しく電波を**減衰**する

**チャンネル**は**有限**



確認しておかないと、本番で使えない目にあう！

# 何がしたいのか？ (ミッション解析)

リクワイヤメント（大会の要求）を満足する前に、**デザイン**

- わたしは、**なぜ**（ロボットが）
- わたしは、作（望）
- ロボットの（メンする）

右脳で  
イメージング

→ 絵を**描いて**みる（ロボットの絵では無く、**使い方の絵**）

# システム設計 (リクワイヤメント分析)

大会**仕様** (リクワイヤメント) を**分析**せよ!

- ロボットの**機体**に要求されている何か?
- ロボットの**動作**は何か?
- 明確な**仕様**は何か?
- **理由**を調査すること
- 自分たちの**強み**は何か? → **スペシャル**な仕様

→必ず、「**計画書**」を書いておくこと

# システム設計 (リクワイヤメントの分析)

たとえば、会場の条件

波や潮は？  
透明度は？

操作卓から  
コースまでの  
距離は？

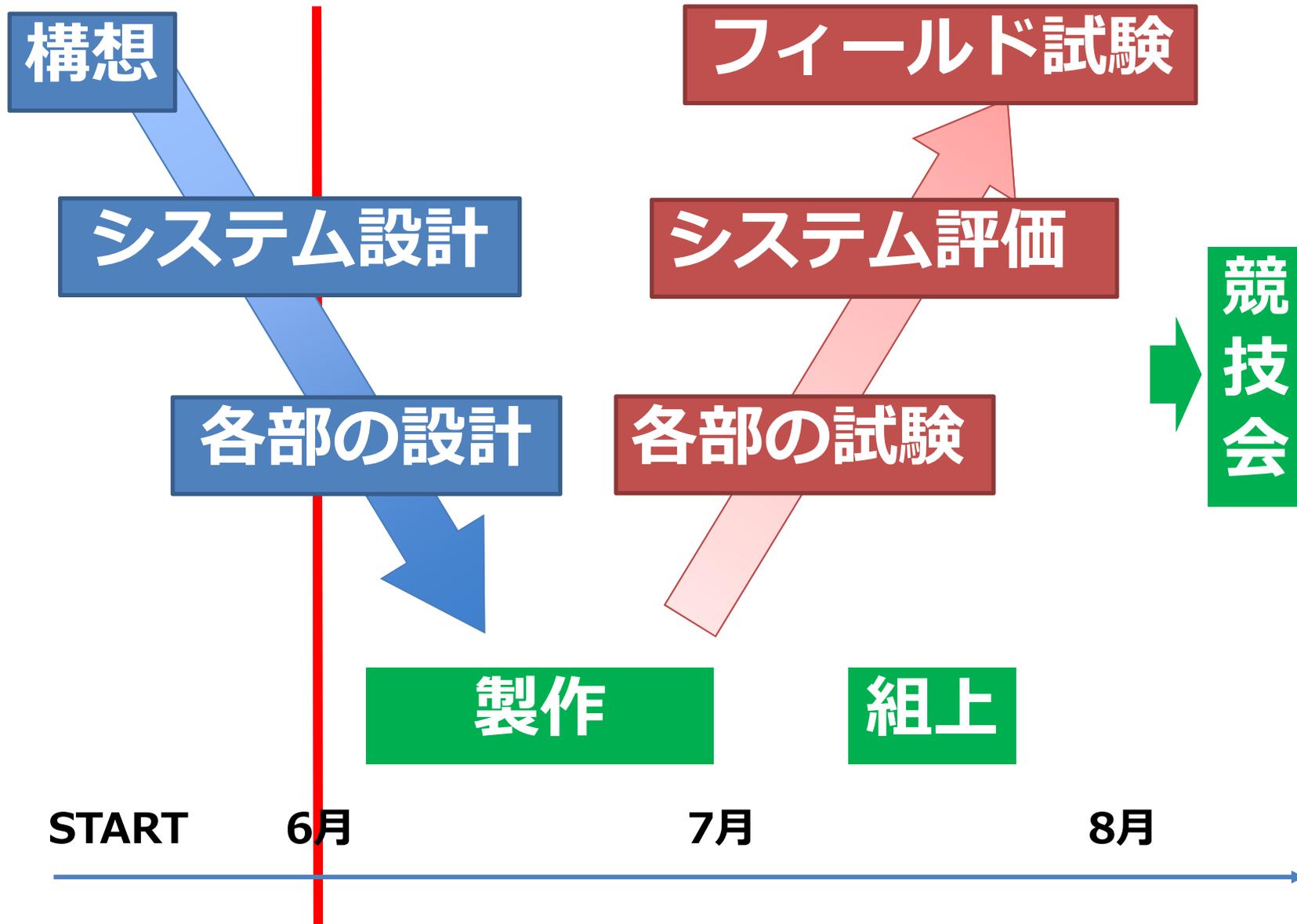
入水の  
仕方は？

電源条件？

明るくても  
画面は見えるか？



# システムデザインのプロセス



# サブシステム化と各部の設計

- ロボットは、必ずサブシステム化される。
- 基本設計をベースに**サブシステムの仕様**を決める
- **担当と工程**を決める

決まったことは、ドキュメントにまとめる  
**工程表・担当者表**

サブシステムのアウトプットも  
ドキュメント化して共有する  
**機械図面、回路図面、ソフト仕様書等**

# 皆がキッチンとできない検証工程

- 実はここまでは、言われれば皆できる
- 出来ないのが、**検証**工程
- いかにか、各部の検証をキッチンとやれるかが成功の鍵



**ハードウェアから作りたいヒトに**

# ロボットの基本設計の項目

- ① 潜航**深度**はどれくらいか？  
※ 10mは0.2MPaA、つまり1kg/cm<sup>2</sup>の圧力
- ② 航行**時間**はどれくらいか？
- ③ 航行**距離**はどれくらいか？（ROV：ケーブル長）
- ④ スラスト**パワー**はどれくらい必要か？
- ⑤ 運動性能は（スラスト&舵の**配置と数**）
- ⑥ **航行**センサーは何が必要か
- ⑦ 着水・揚収方法は（サイズと重量）  
【AUVのみ】
- ① スタート & ゴール**判断**はどのように？
- ② **位置**はどうやって知る？

# 1. 耐水3か条

- 市販の防水製品は「海中では**使えない**」
- 防水コネクタは「じゃぼじゃぼ**漏れる**」
- どれだけテープを巻いても海水は**止まらない**

# 耐水

- **海水**は電子回路にとって過酷な条件
- わずかでも浸水したら故障
- しかも直せない
- どんなすてきなロボットを作っても、  
**動かなければ意味が無い**
- 海水ショートは火災のネタでもある

エネルギーを注ぐのは**防水・耐水対策**

# 耐水

## 容器の防水

- 確実な海水止めは「**リング**」
- パッキンを使うときは、多点でしっかり締める
- **シリコングリース**（高粘性）を多用しろ！
- 接触面はゴミ一つ無いように
- 蓋締め**3種の神器**：
  - 1) 洗浄スプレー（またはエタノール）
  - 2) エアードスタースプレー
  - 3) シリコングリースこれにプラスして「キムワイプ」
- 簡易的修復はシリコンコンパウンド
- 小さな穴埋めはアロンアルファも有効
- 2液混合**エポキシ接着剤**は無敵（アラルダイト等）
- 海水に浸ける前には、必ず**真水水槽**に一晩浸けて確認を！！

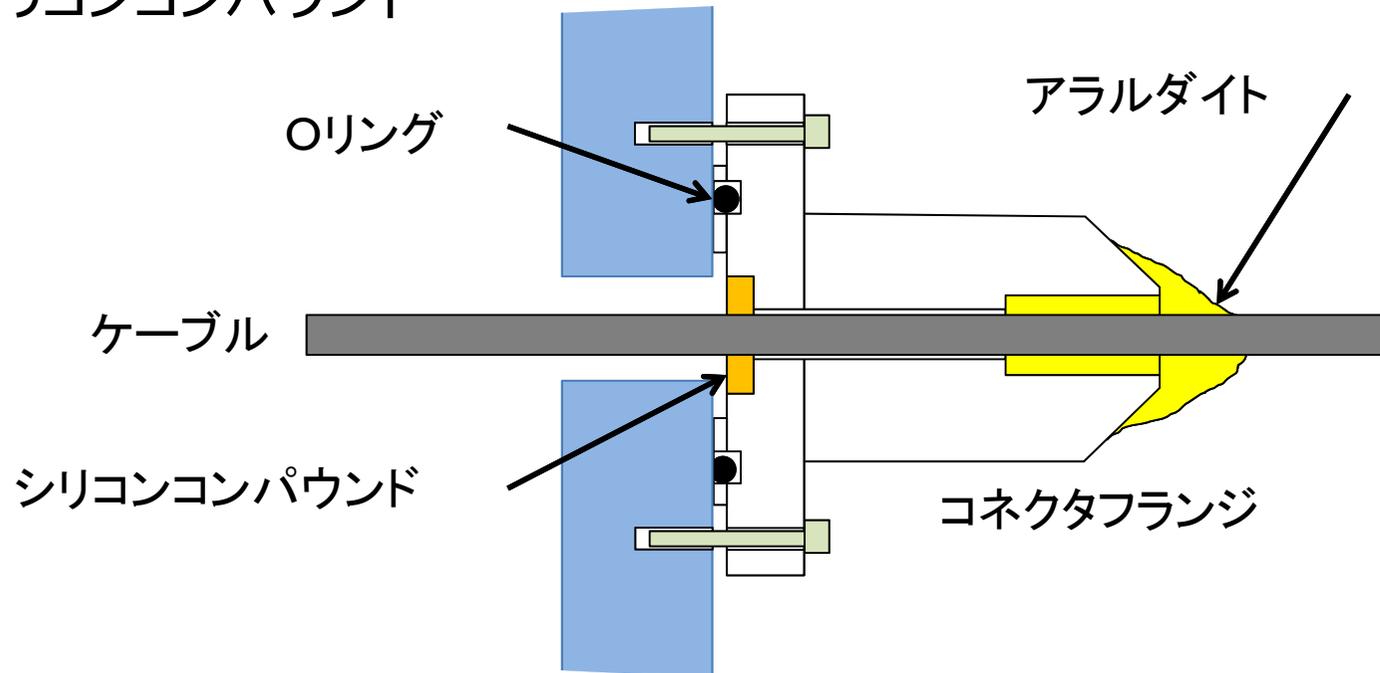


# 耐水

## 貫通型コネクタの製作

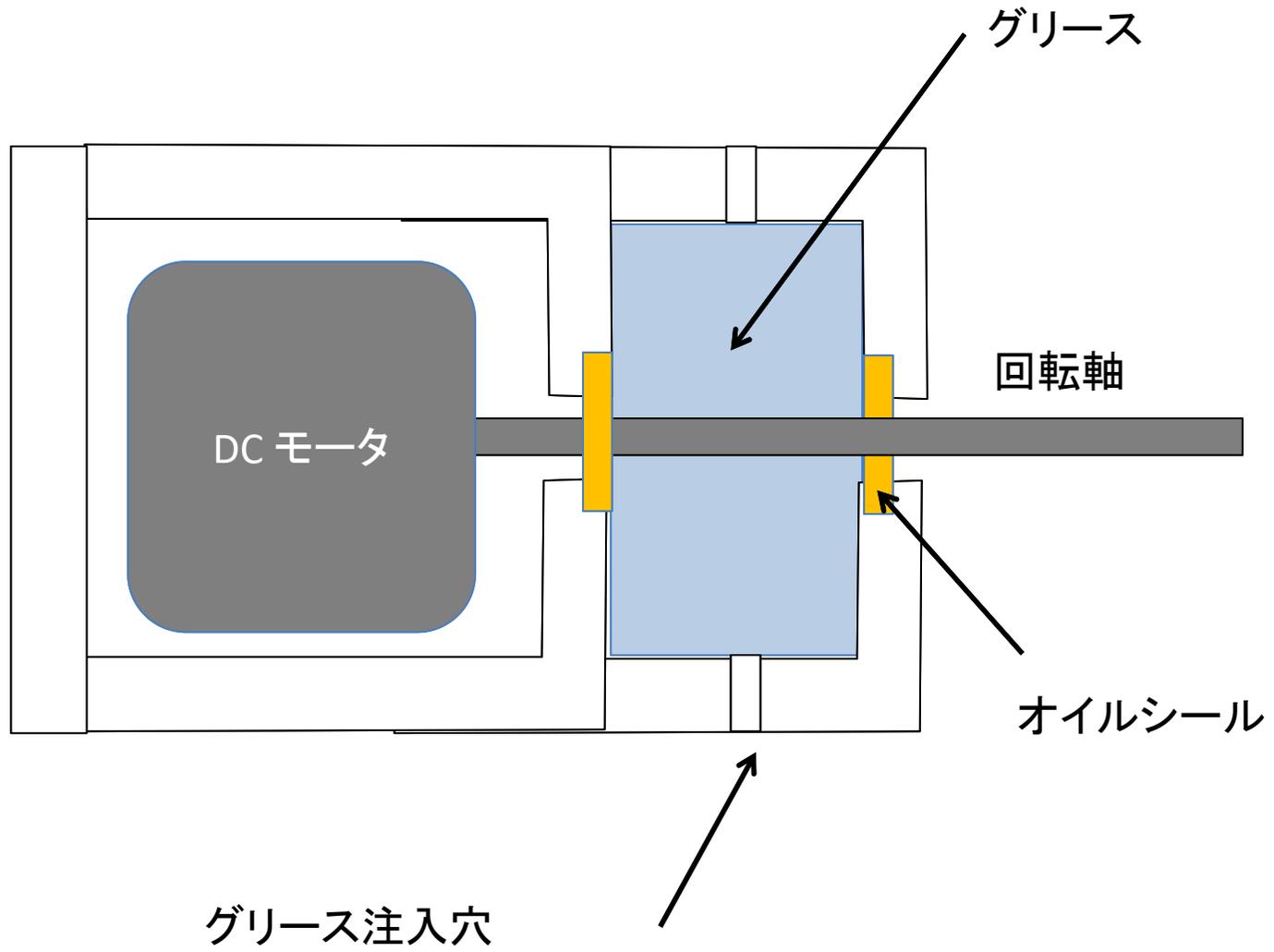
### 用意するもの

- コネクタフランジ（切削加工してください）
- 屋外用ケーブル（キャブタイヤケーブル：2PNCT等）
- アラルダイト
- シリコンコンパウンド



# 耐水

## スラストの耐水



## 耐水：水に入れるまえに

- 君の作ったロボットが、水中で電源を入れて、**30分以上無事**だったら、システムの問題の**5割**はクリアーできたと思うべし。
- でも、大会に出る前に必ずやろう！

**水中で動か**し、浸水の無い事を確認すること

**水**のプールに**一晩**浸けてみること

本番では

絶対に直前に**容器を開けるな**！！

# 耐水のために



洗浄スプレー



キムワイブ



グリース

- Oリングを閉じるときは、必ず
- ・ 洗浄（クリーナ+キムワイブ）
  - ・ グリースアップ

## 2. 耐圧設計

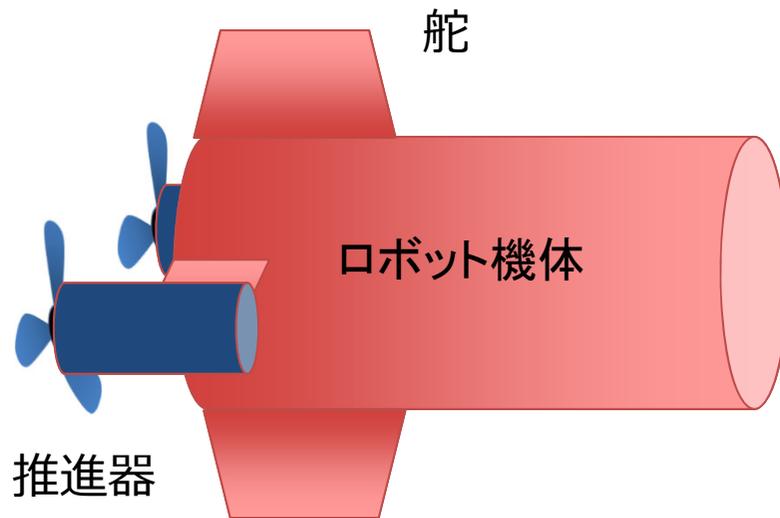
- 今日の座学は、「使える」ことが重要なので、難しい式は一切使わない。けれども、エンジニアとしては、「どうしてこうなるのか？」その物理は知っておくべきで、**各自勉強**されたい。
- さて、ネットから耐圧設計のソフトをダウンロードしてしまおう。
- 米国サイト、DeepSea Power&Light が無償で提供している、**Under Pressure** というソフトだ。すごく簡単に使えるし、設計精度が良い。
- ざっくり言えば、**外形120mm x 300 mm, t=3mm**の亚克力パイプは深度**20m**くらいまで使える。
- ただしエンドキャップは**t=7mm**は必要

### 3. 形状

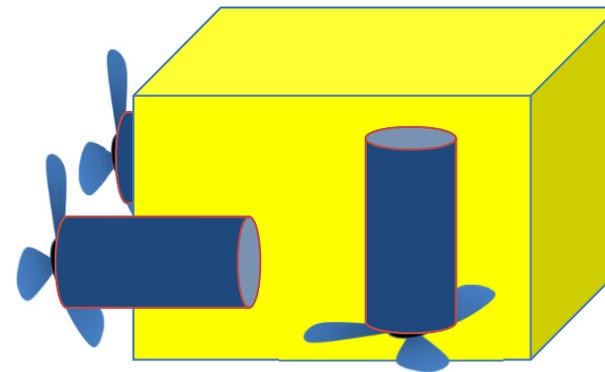
機体形状を大きく左右するのは

- **止まらず**に早く走るか？
- ゆっくり走り**止まる**事もあるか？

の2点. 最初のタイプをクルージングタイプ, 後者をワーキングタイプ (またはホバリングタイプ) という.



クルージングタイプ



ワーキングタイプ

## 4. 浮力をラフに求める

### 比重と密度

- **比重**：ロボットを構成する物体の密度と、真水の密度との比。
- **密度** = 物体の質量 ÷ 体積 であるが、ほぼ同じ値と扱って良い。
- 真水の密度は  $\rho \sim 1 \text{ g/cm}^3$ , 比重は 1
- 海水の密度は, **塩分濃度が35‰**のときに,  
 $\rho \sim \mathbf{1.025 \text{ g/cm}^3}$
- 実際には、密度は**温度で変化**するがラフエスティメーションではこれで十分。
- よく使う材料の大体の**比重**も参考に示そう。
  - ① アクリル  $\sim 1.2$
  - ② アルミ  $\sim 2.7$
  - ③ ステンレス  $\sim 7.8$  (鉄類は圧倒的に重いので錘に良い)

# 浮力をラフに求める

浮くかどうかのラフな計算をする

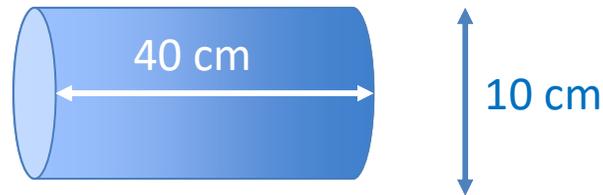
※**最初に計算**しておけば、あとで浮力調整に苦労しなくて済む。

## (1) 排水量

水を押しのける体積 → 計算はとても簡単

例題

外形:  $d = 10 \text{ cm}$ , 長さ:  $l = 40 \text{ cm}$ の円筒容器の排水量:  $V [\text{cc}]$  は?



答え

$$V = \pi \left(\frac{d}{2}\right)^2 \times l = 3141 \quad \rightarrow \quad \text{約3リットル}$$

真水の場合は比重~1であるから、**約3kgの浮力**が有る事になる。

# 浮力をラフに求める

浮くかどうかのラフな計算をする

## (2) ボディ重量の見積もり

機体重量 = 容器重量 + フレーム重量 + 外部機器重量 + 内部機器重量

これは設計段階では**簡単にもとまらない**。

2回、3回と計算をやり直そう。

初期設計では、

● 容器の大きさや材料 :

$$W_{PH} = 1 \text{ kg}$$

● 大体のフレームサイズや材料 :

$$W_{FM} = 0.5 \text{ kg}$$

● 主要機器 :

$$W_{EQUIP} = 1 \text{ kg}$$

を決めて、**だいたいの重さ**を見積もりましょう。

この例では、機体重量は**2.5kg**ですね。

# 浮力をラフに求める

浮くかどうかのラフな計算をする

## (3) 浮量のもとまる

浮力 = 排水量 - 重量

先ほどの例では、

$V=3$  リットル

$W=2.5$  kg

問題：

- 真水と海水の時の上記例の浮力をもとめなさい。

答え：

- 真水： $(V \times 1.0) - W = 500$  g となり、このロボットは浮く。
- 海水：さらに軽くなり： $(V \times 1.025) - W = 575$  g

正確には、外部のスラスタや舵、外部センサなどの体積も排水量に含まれるが、ここでは無視している

# 5. 浮心と重心

## 浮心と重心 (B-G) の関係

作ったロボットがひっくり返らないかどうか？ 傾いたりしないか？ これも重要だ。予めラフに計算しておくが良い。

ただし、ロボットの機械図面と各 부품の配置図を作る必要がある。集計にはExcelを使うが良い。

### 1. 重心(G)を求める

容器、フレーム、スラスタ、部品、錘など、個別にExcelでn個の部品リストを作って、

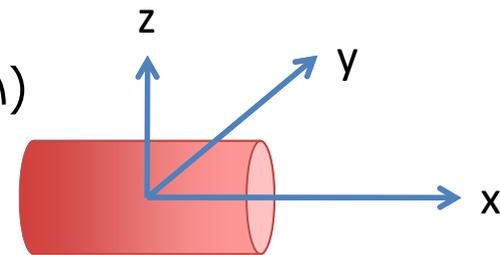
n番目の部品重量:  $W_n$  (秤で計っておこう)

n番目の部品の重心:  $X_n, Y_n, Z_n$  (大体でよい)

を記載する。

これらから、ロボット全体の重心点(x-y-z)を求める。

ロボットの前後をx軸、左右をy軸、上下をz軸としたとき、重心のxとyの位置は軸上にないと、ロボットは傾く事になる。



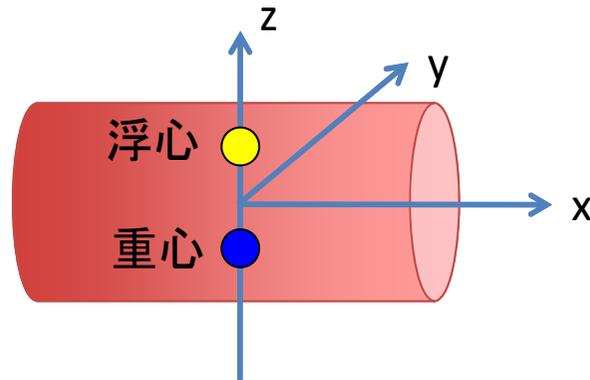
# 浮心と重心

## 2. 浮心(B)を求める

- 容器の排水量だけで浮力を得ている場合は、容器の中心が浮心になる。複数の浮力材を用いている場合は、重心の計算同様、浮力と位置をサムアップする。
- 浮心の水平面内(x-y)位置は重心と同じであるべきである。

## 3. 浮心と重心(B-G)

- Z軸上の浮心位置は、重心位置より必ず上になければならない。
- 浮心と重心が離れていれば、姿勢は非常に安定するが、姿勢の制御性が悪くなる。初心者は離しておいたほうがよいかもしれない。

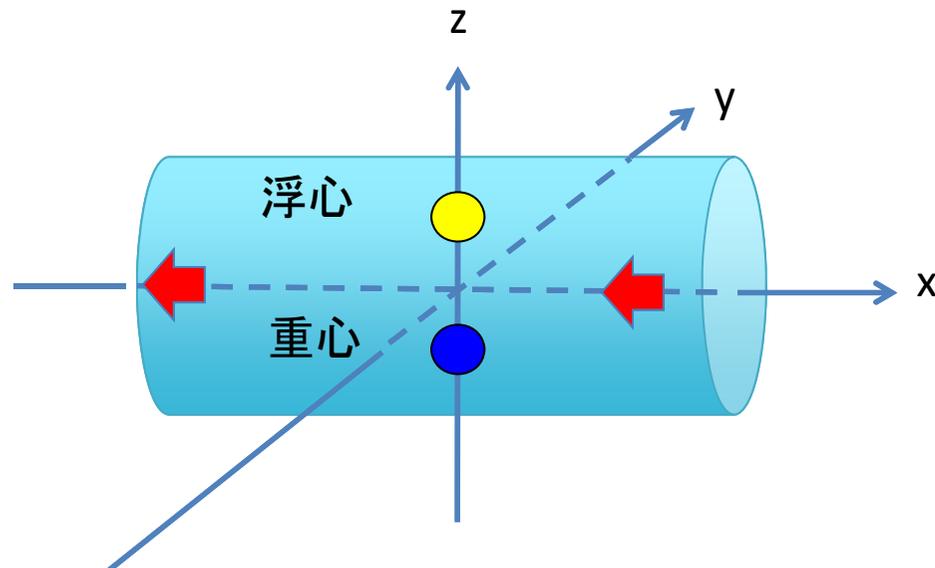


## 6. 推進器と舵

- 推進力の基本はスクルー（スラスト）だ
- 船尾に**1スラスト** + 水平垂直**ラダー**が**AUV**の基本形（制御し易い）
- しかし舵は低速では効かない
- **速度が遅い**なら4つのスラスト（2x前後スラスト、上下スラスト、左右スラスト）も基本形
- スラストや舵の取付位置は正確には計算で求めるが、ラフに決めても概ねOK

# 推進器と舵

- スラストの適正配置設計は難しいので、ラフにやる
- 水力差制御で航行する場合は、前部もしくは後部に配置
- 浮心・重心位置と、機体形状（流体抵抗）に影響を受ける
- 重心近くに配置すると不安定になる



# 共通項目

# ハードウェア組み立ての注意 #1

## 1. 振動を侮るな！

- 耐圧容器内の基板や部品を**宙ぶらりん**のまま使っているのを見かけます
- 直前に付け足す場合を除いてやめましょう。
- **振動**（やノイズ、温度環境の変化）により、動作が**不安定**になります。
- コネクタやポッティング技術を使え

**内部の基板・部品は必ず固定する**

## ハードウェア組み立ての注意 #2

### 2. 熱を気にせよ！

- スラスタドライバはもちろん、**CPU**やDCDCの**放熱**にも気を使おう
- いまさらですが、シャーシには**厚め（5mm以上）のアルミ板**を使うと良い。放熱できますから。
- もう無理、という方は、銅板（1mm以上）を、好みの形に切って、ドライバやCPUに密着させます。

発熱部品は、必ず**放熱対策**を

# ハードウェア組み立ての注意 #3

## 3. 配線に気を配れ！

- 学生さんの製作物でよく見かけますが、**配線**をバラバラ、ブラブラにしている。
- 絶対にやめましょう
- 電氣的・機械的に**弱**くなり、動作**不安定**を引き起こします
- 電源線は必ず**ツイストペア**に
- 信号線もグランド線を中心に束ねる
- 信号線と電源線は**離して配線**
- これらの配線はシャーシなどに固定する

**配線は、必ず束ねて固定**

# ソフトウェア製作の注意

- ソフトウェアは最後になりがち
- あと回しは絶対にやめよう
  
- 必ず**設計仕様書**を作る
- それをメンバーで**レビュー**する
  
- **マトリクス**を作成する
- **異常系**（予期せぬ操作・状態）での**動作を規定**する
- **関数チェック**をする

# ソフトウェア製作の注意

## C ベース

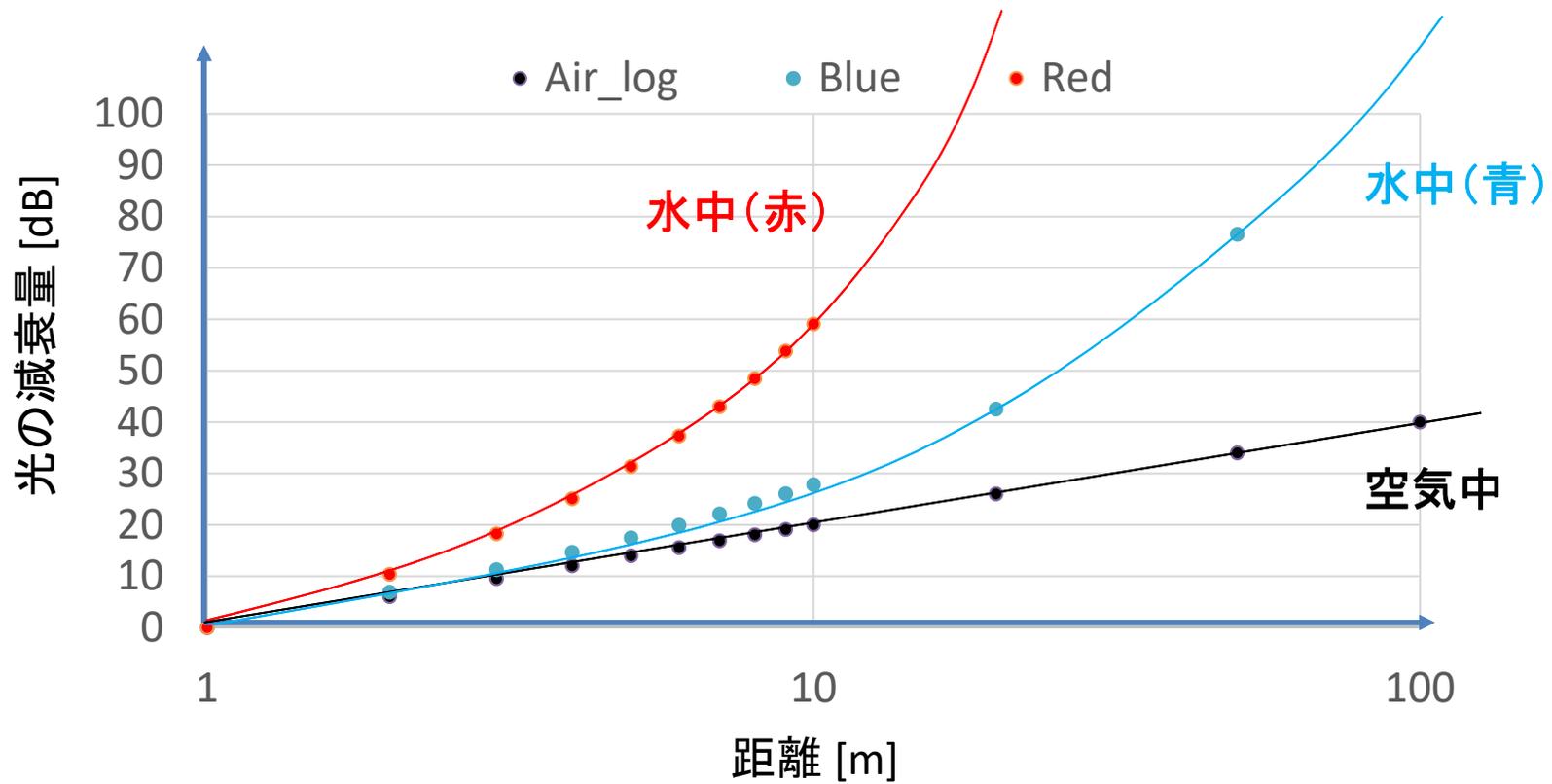
- メインの1本書きは避けよう
- If 分岐の多用と、グローバル変数のフラグはやめよう
- Case 文でステートマシンを作ろう
- コメントを必ず書こう
- 変数は、型と機能を明確に表す名称にする（例: iTimer10ms）
- 関数名はローカルかグローバルかと、機能を明確にする名称にする（例： g\_GetRs232cData）
- 各部にトレースを入れ、必ずログを取って、解析できるようにしよう

# 画像認識で注意すること



一様均一な画像が得られない

# 光の減衰の差



0m

0.1m

2m

5m

10m

# ソフトウェア製作の注意

## ソフトウェアは万能ではない

- OPEN CVのライブラリは**何でもできる**わけでは**無い**
- しっかり作った**学習済みデータのない**Deep Learningは何も学んでいないヒトと同じ
- 入力になる**ハードのスペック**に気を使え

## 位置推定の方法

- 画像認識によるSLAMはハードルが高い
- ソナーの画像解析はより難しい
- センサで位置を知るのが最も賢いやり方
- 短区間なら**方位 + 速度 × 走った時間**による位置推定が意外に有効

## 意外にある誤解

- **ジャイロ**を搭載しているから**位置**が分かる
- **4K**カメラを搭載しているので**画像が良い**
- 画像マッピングで走るのが一番
- **万一**の時は**CPU**が対応してくれる
- プールで試験したから大丈夫

# とにかく信頼性向上！！

- ここまで話してきたことは、実はすべて**作ったロボットの信頼性の向上**の手法です。
- そう、皆さんのロボットが現場で動かない、それどころか**壊れちゃう**のうは、信頼性が**低く**つられているからです。
- でも、しかたありません。アマチュアの製作物には信頼性なんて概念がありませんから。
- **一歩先**を行くためには、**信頼性を考慮**しましょう。そんな君たちが、勝てます！

# テストの期間をタツプり取れ！

- 大会**直前**まで作っていたら、**負け**たも同然
- **自然**環境は「**まぐれ**」を許さない
- **フィールドテスト**（海中実験）をどれだけやったか？その**回数**が**優勝の確率**だ！

# 頑張れ，未来のエンジニア！

- 第一回目ということで，海中ロボットの機体の作り方の基本をレクチャーしました。
- もっと，知りたい方はどこかで私のレクチャーを受け  
るか， [yoshidah@jamstec.go.jp](mailto:yoshidah@jamstec.go.jp) まで
- ただし，忙しいので1回メールしたくらいじゃ返事は無いと思ってください。しつこくね！