自律ECLSS SW開発

12/16-12/20

開発会の目的

一番議論したいこと

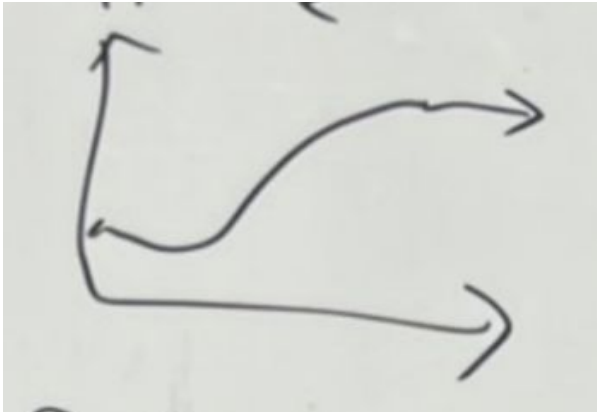
- ECLSS OSSに対するPoC (Proof of Concept)・仮説検証
  - ECLSS OSSは次世代宇宙ステーション開発のプラットフォームになりうるのか？
  - 公開する価値は?
  - どのような開発になるのか？

→プロトタイプを作成

→ECLSS のOSSの設計要件明確化

開発マイルストーン

1) ハビタット(w/ CDRA)のモデル化
2) 故障のシミュレーション(ppCO2, Humidity, Temperature)
3) ECLSS OSの設計



ECLSS シミュレーション開発

VHABによるモデリング

有人宇宙基地に特化した化学プロセスシミュレーター(Matlab言語)

FIG 4. Matter Store

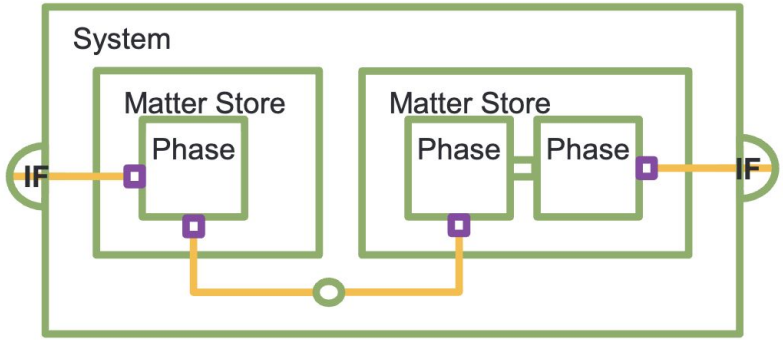


FIG 8. V-HAB System with Branches

マイルストーン

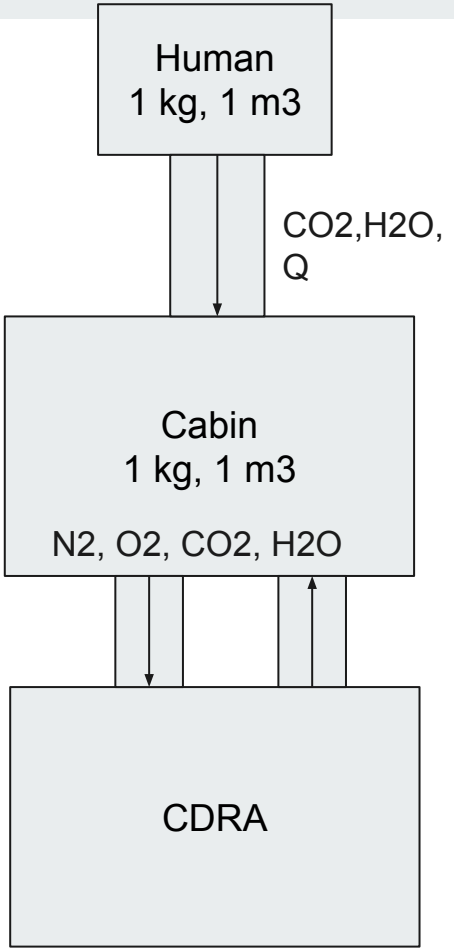
① キャビン(制御装置なし)のモデル化

・クルーによる一定のCO2, H2O, Heat供給

・ppCO2, Humidity, Temperature 単調増加 (大気圧は微量増加)

② CDRAのモデル化

③ OSSのプロトタイプ作成

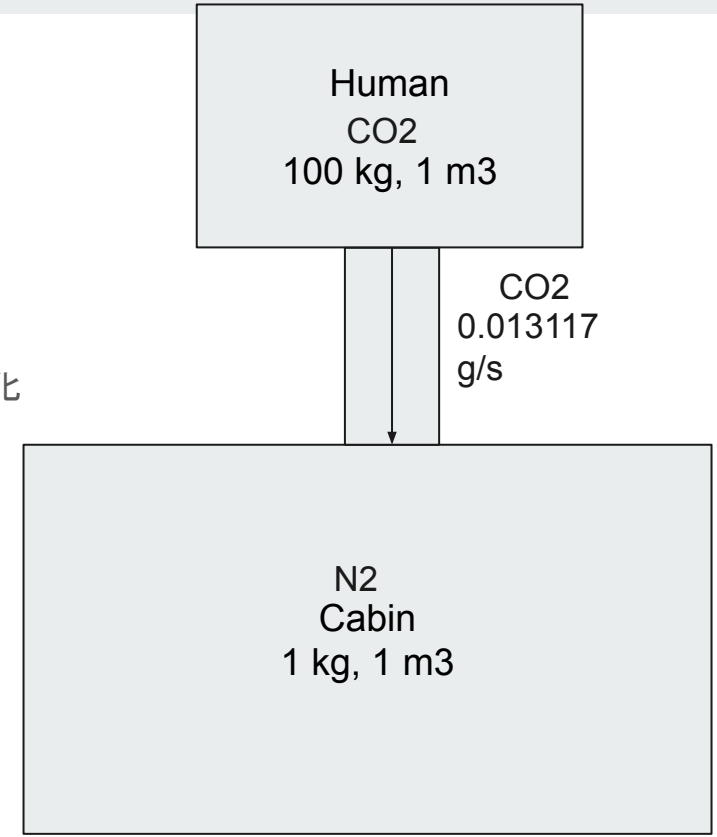


①-1 シンプルなキャビンのモデル化

Humanストアからキャビンストアに対する二酸化炭素の移動をモデル化

PV=nRT

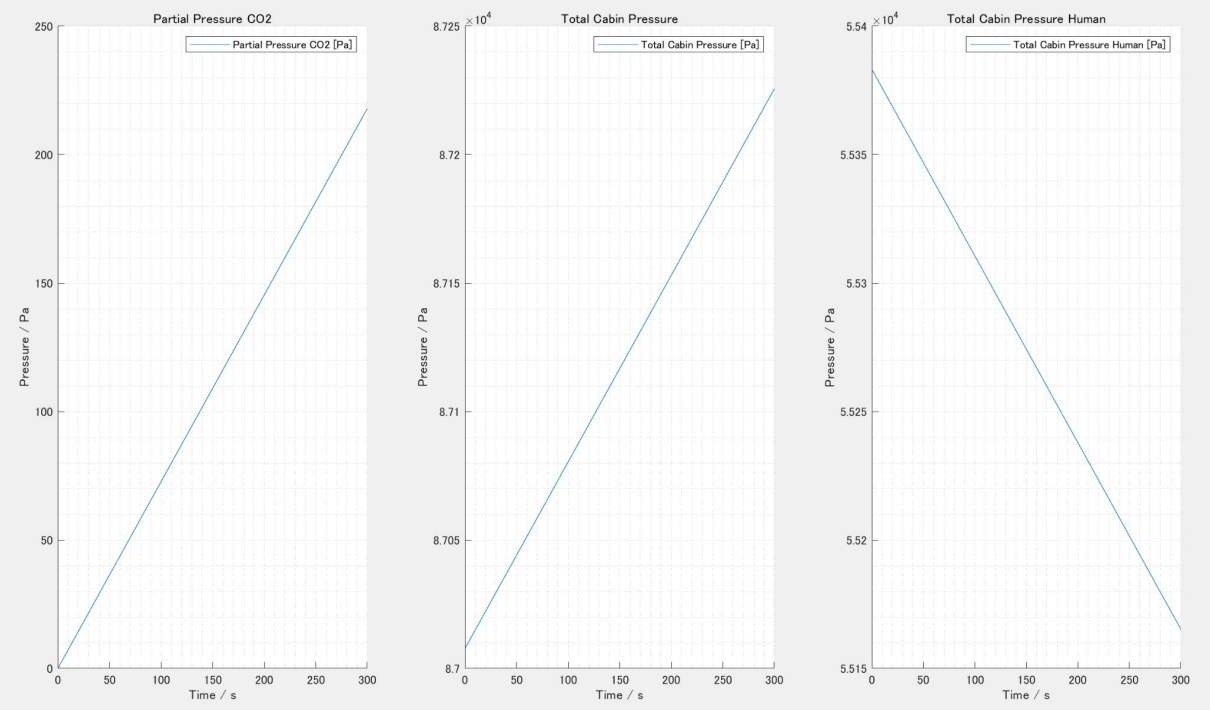
二酸化炭素モル質量 44g/mol



コード

```
classdef ExampleHERA < vsys
properties (SetAccess = protected, GetAccess = public)


end

methods
    function this = ExampleHERA(oParent, sName)
        this@vsys(oParent, sName, 30);%exec()関数に使用する時間ステップを定義。この例では30秒が使用されており、これはexec()関数が30秒ごとに呼び出される
        eval(this.oRoot.oCfgParams.configCode(this));

    end

    function createMatterStructure(this)
        createMatterStructure@vsys(this);

        %ストアの定義
        matter.store(this, 'Cabin', 1);
        matter.phases.gas(this.toStores.Cabin, 'CabinAir', struct('N2', 1), 1, 293.15);

        matter.store(this, 'Human', 1);
        matter.phases.gas(this.toStores.Human, 'HumanCO2', struct('CO2', 1), 1, 293.15);

        %ブランチの定義
        matter.branch(this, this.toStores.Human.toPhases.HumanCO2, {}, this.toStores.Cabin.toPhases.CabinAir, 'CO2_Human_to_Cabin');
    end

    function createSolverStructure(this)
        createSolverStructure@vsys(this);

        solver.matter.manual.branch(this.toBranches.CO2_Human_to_Cabin');
        %this.toBranches.CO2_Human_to_Cabin.oHandler.setVolumetricFlowRate(0.01); %分岐を通じて一定の容積流量を確保
        this.toBranches.CO2_Human_to_Cabin.oHandler.setFlowRate(0.01); %分岐を通じて一定の容積流量を確保  0.000013117kg/s

        % 熱ソルバー
        this.setThermalSolvers();
    end
end

 methods (Access = protected)
    function exec(this, ~)
        exec@vsys(this);

    end
 end

 end
```

シミュレーション結果


Partial Pressure CO2
Partial Pressure CO2 [Pa]
Pressure / Pa
Time / s
Total Cabin Pressure
Total Cabin Pressure [Pa]
Pressure / Pa
Time / s
Total Cabin Pressure Human
Total Cabin Pressure Human [Pa]
Pressure / Pa
Time / s

考察

流入速度は問題ないことが確認できた

1m^3のキャビンでは10分程度で二酸化炭素濃度が許容上限値4000ppm)を超える

①-2 P2Pを用いたキャビンのモデル化

同一ストア間のphaseとphaseどうしの物質移動

→時間切れでギブアップ

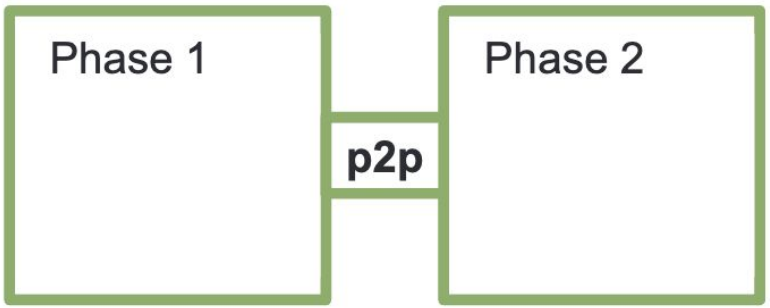


FIG 6. Phase-to-phase processor

② CDRAのモデル化

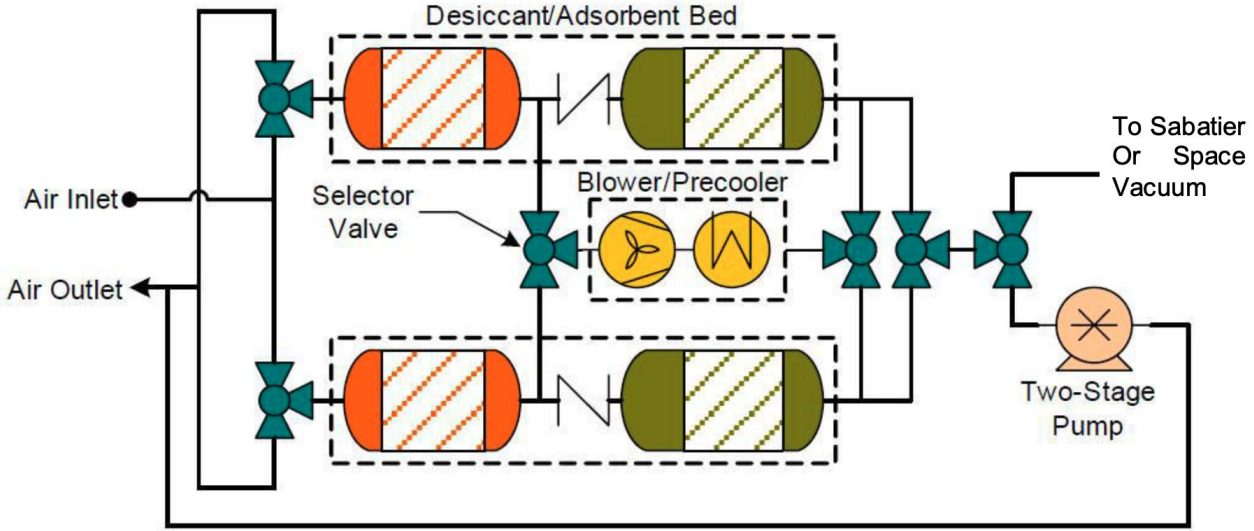
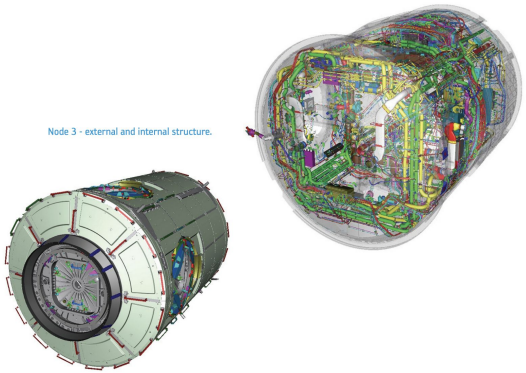


Figure 1. Integration of the CDRA Components[9]

シミュレーション結果

後で写真貼り付け

On-Orbit ISS ECLS Hardware Distribution as of February 2010



Node 3 - external and internal structure.

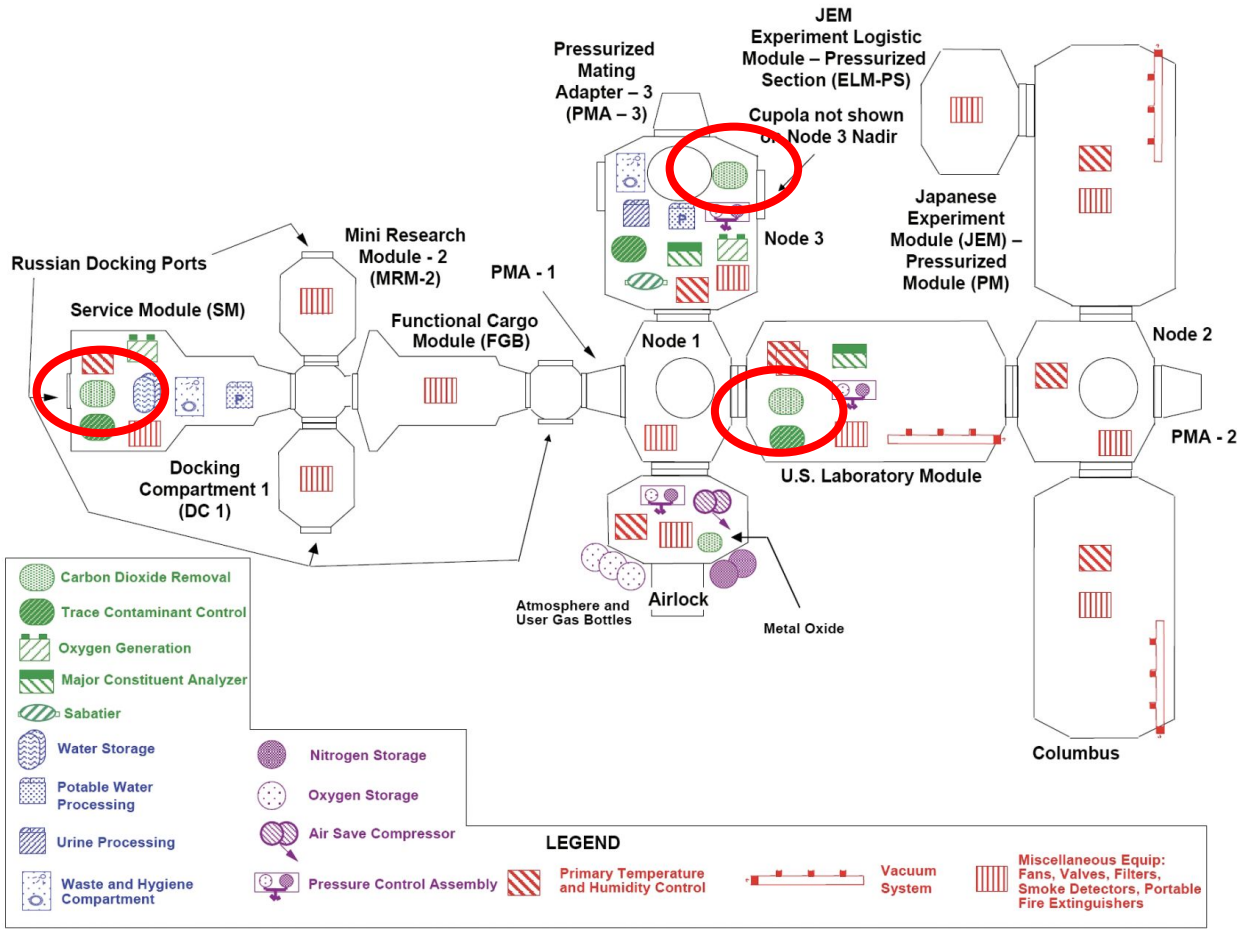
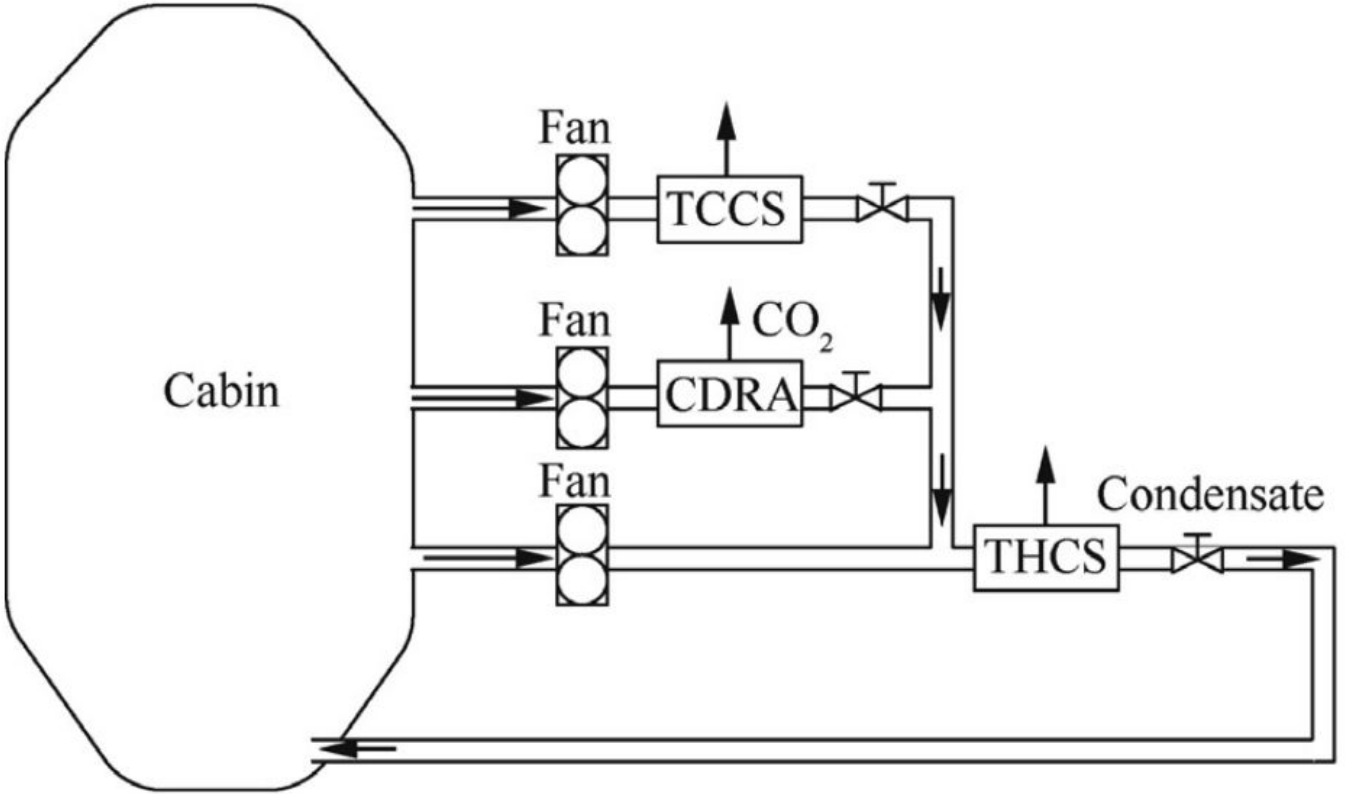


Diagramme: NASA/ISEC



(b) Simplified ventilation pipe network

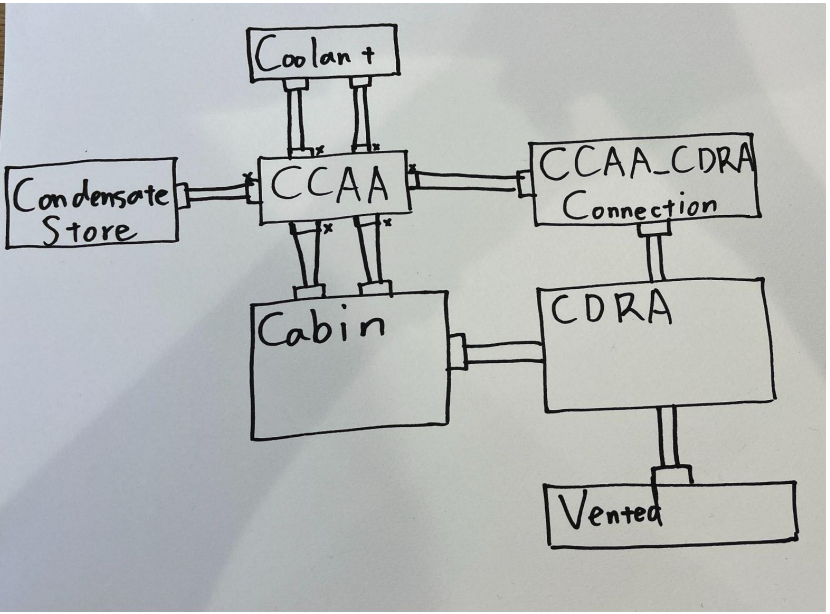
Fig. 1 Distribution and flow diagram of ECLSS.

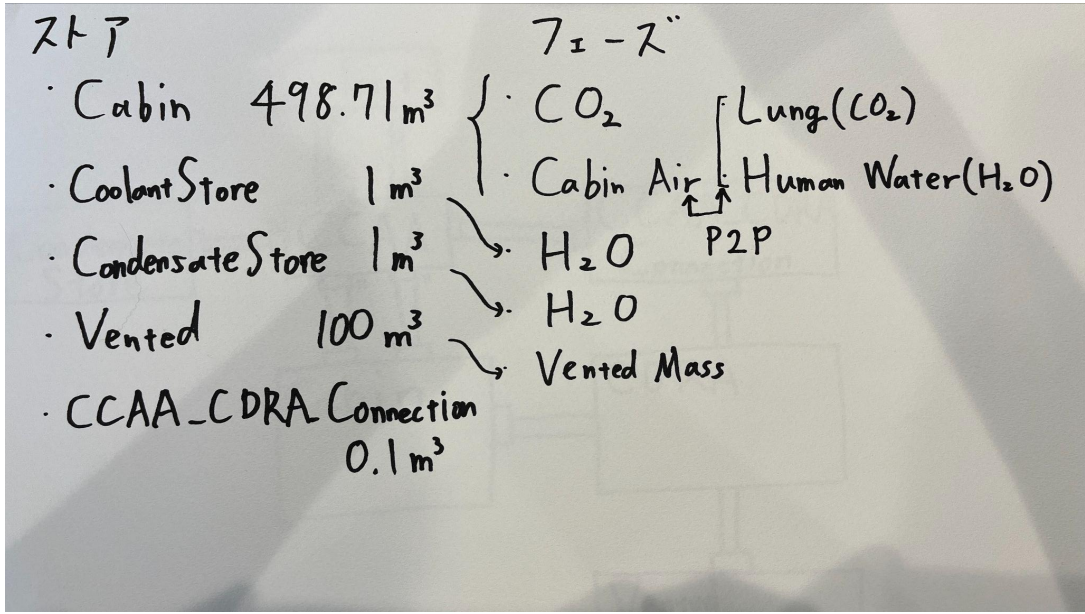
昨夜の進捗

CDRAを使ったシミュレーションに挑戦→デモを実行→シミュレーションが終わらなさそう→有人宇宙講義→佐藤くん就寝→講義継続(EVA)→蓮見くん就寝→講義継続(アナログ)→横関くん小倉くんと日本の有人宇宙開発について大議論(日本人全体の有人宇宙に対する根本的なモチベーションの欠落→シミュレーション再開→シミュレーション時間を小さくしたら実行結果が出た→グラフに対してなんやかんや議論→横関くん就寝

CDRAを使ったシミュレーションは一応成功→結果の解釈(ECLSS OS設計への活用について)→故障のイベントの模擬できるか怪しい→

CDRAデモンストレーション





CCAA

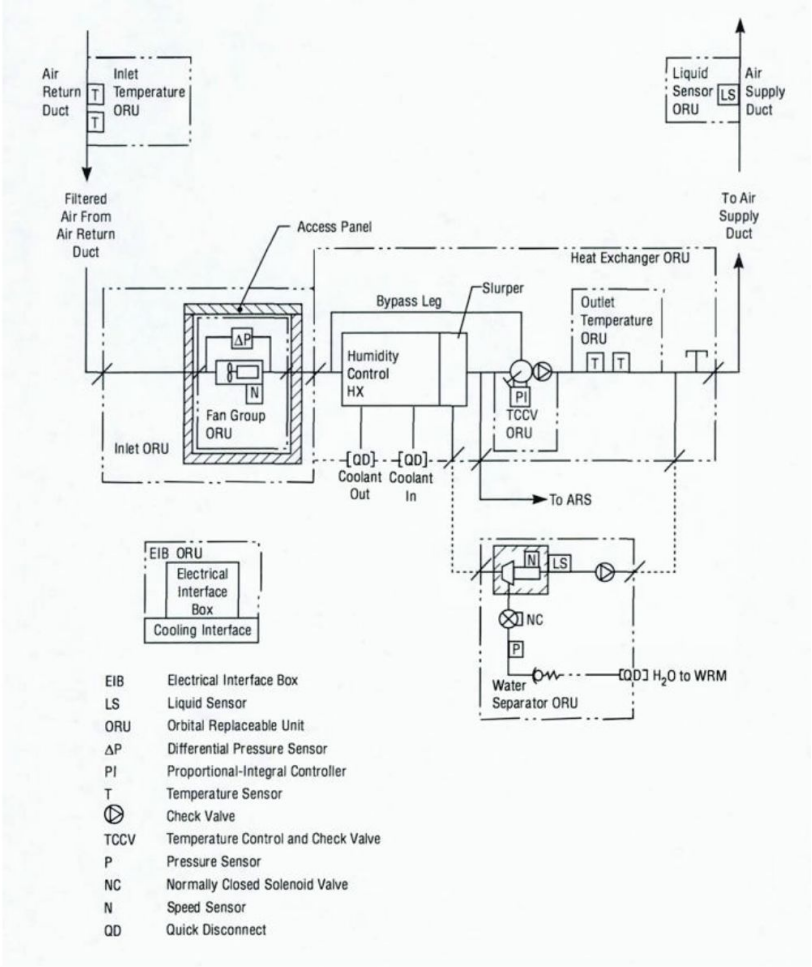
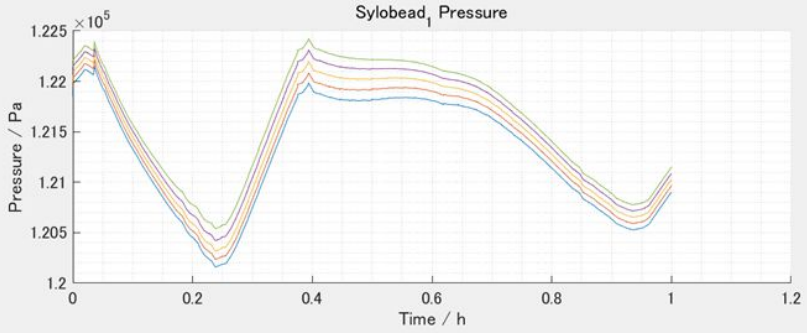
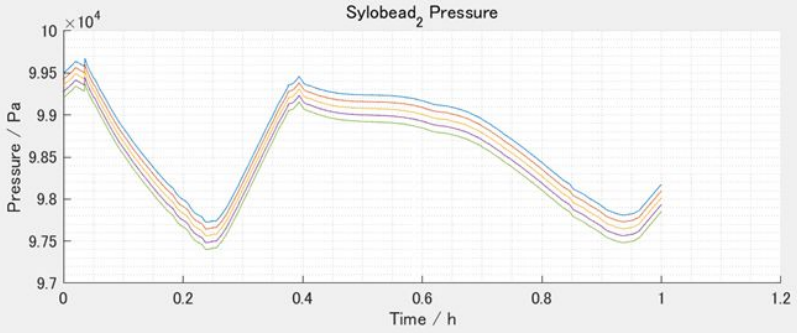
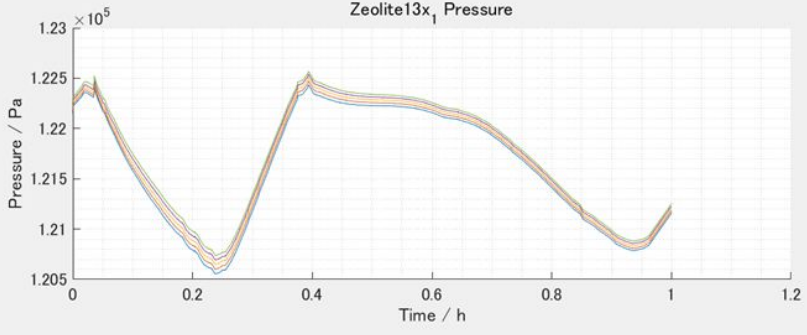
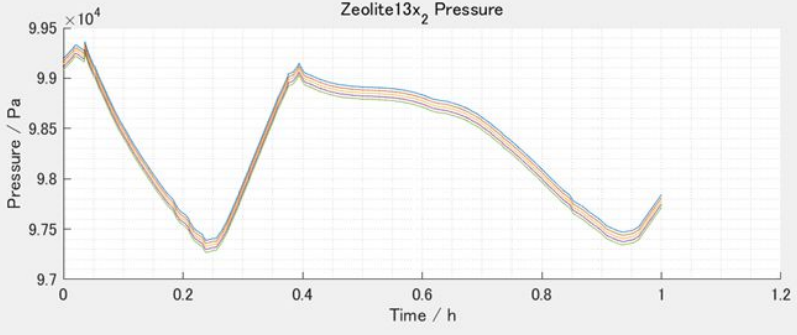


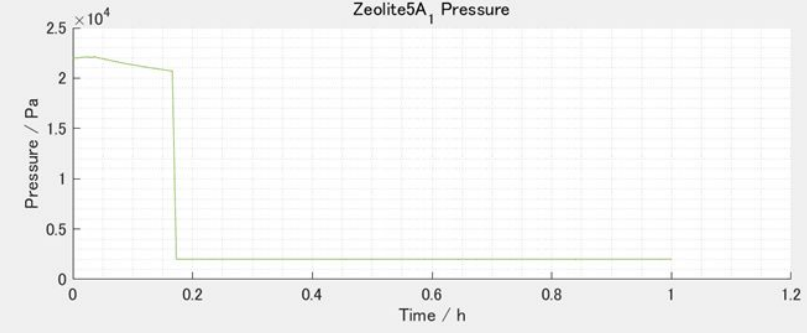
FIG 11. Common Cabin Air Assembly Process Schematic [21]

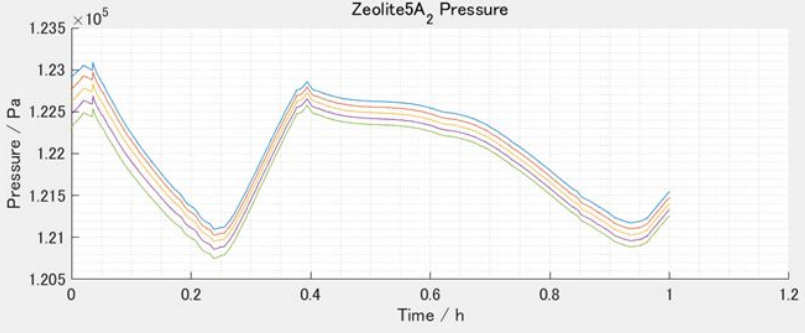

Sylobead1 Pressure
×10^5
1.225
1.22
1.215
1.21
1.205
1.2
Pressure / Pa
0
0.2
0.4
0.6
0.8
1
1.2
Time / h

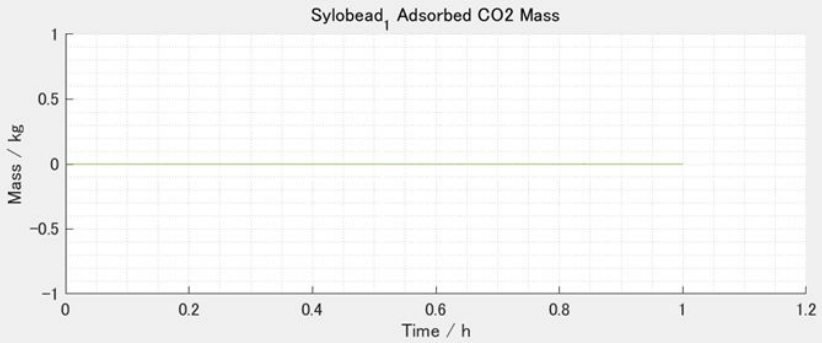

Sylobead2 Pressure
×10^4
10
9.95
9.9
9.85
9.8
9.75
9.7
Pressure / Pa
0
0.2
0.4
0.6
0.8
1
1.2
Time / h

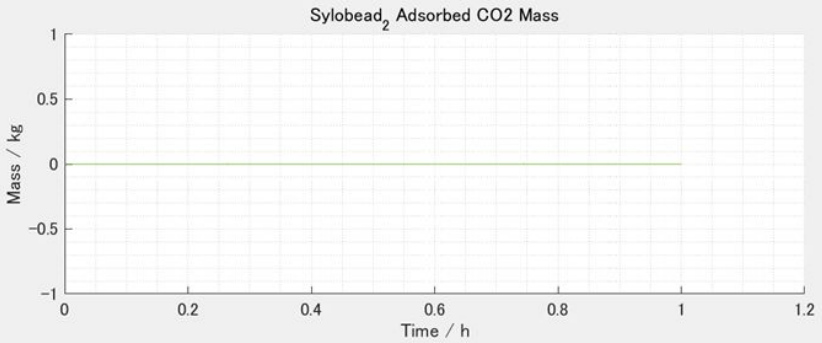

Zeolite13x1 Pressure
×10^5
1.23
1.225
1.22
1.215
1.21
1.205
Pressure / Pa
0
0.2
0.4
0.6
0.8
1
1.2
Time / h

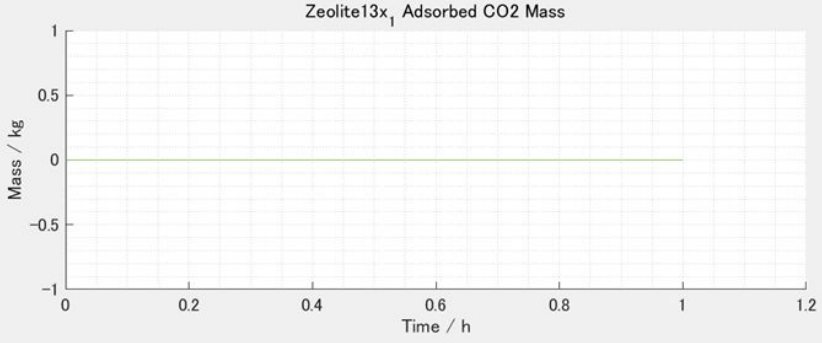

Zeolite13x2 Pressure
×10^4
9.95
9.9
9.85
9.8
9.75
9.7
Pressure / Pa
0
0.2
0.4
0.6
0.8
1
1.2
Time / h

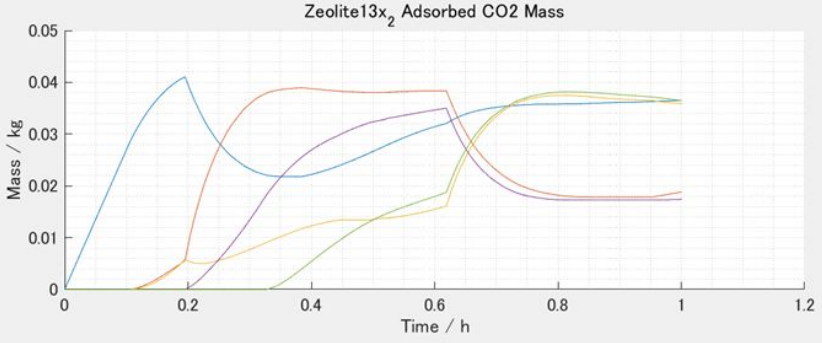

Zeolite5A1 Pressure
×10^4
2.5
2
1.5
1
0.5
0
Pressure / Pa
0
0.2
0.4
0.6
0.8
1
1.2
Time / h

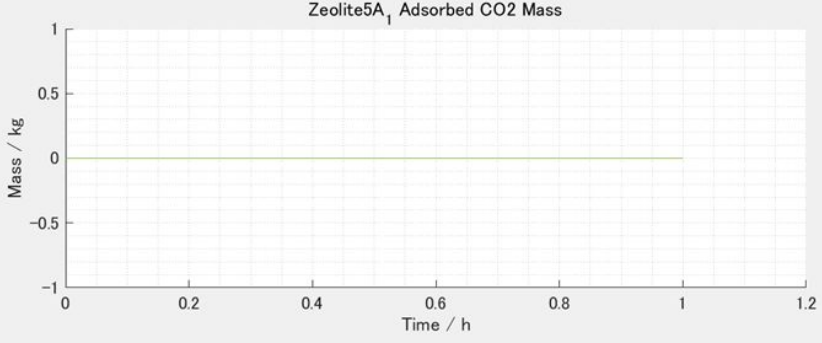

Zeolite5A2 Pressure
×10^5
1.235
1.23
1.225
1.22
1.215
1.21
1.205
Pressure / Pa
0
0.2
0.4
0.6
0.8
1
1.2
Time / h

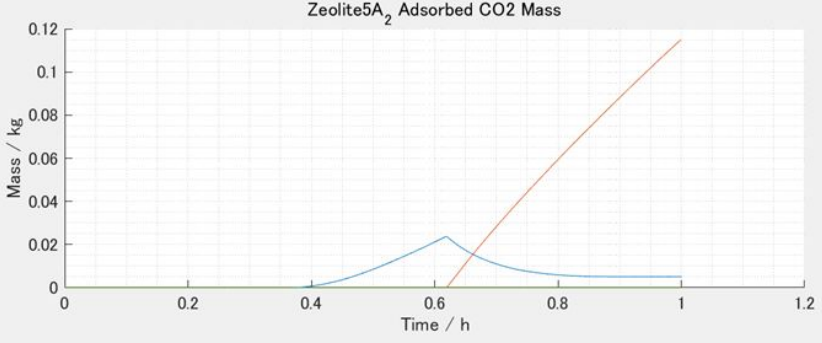

Sylobead$_1$ Adsorbed CO2 Mass
Mass / kg
Time / h

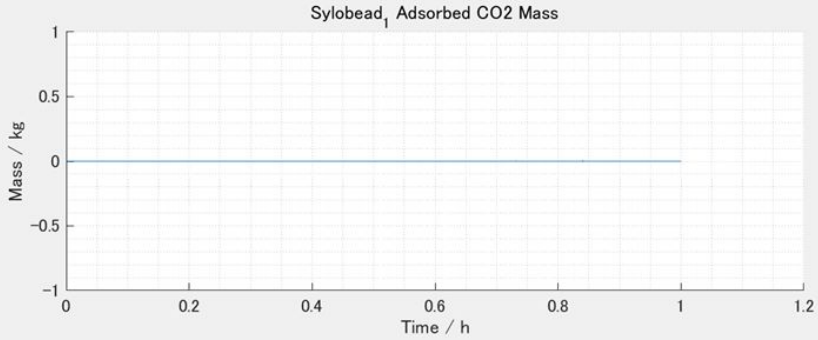

Sylobead$_2$ Adsorbed CO2 Mass
Mass / kg
Time / h

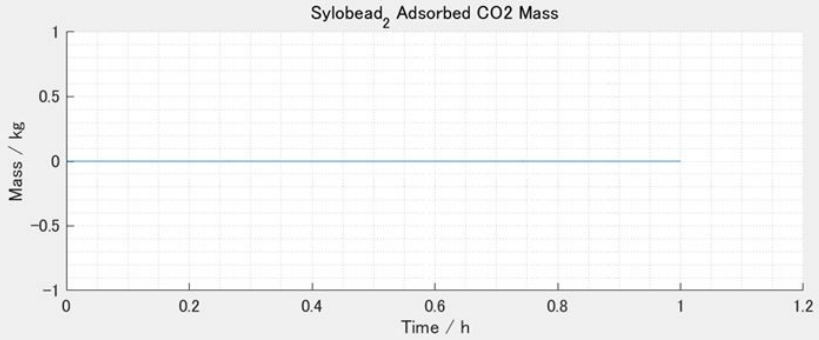

Zeolite13x$_1$ Adsorbed CO2 Mass
Mass / kg
Time / h

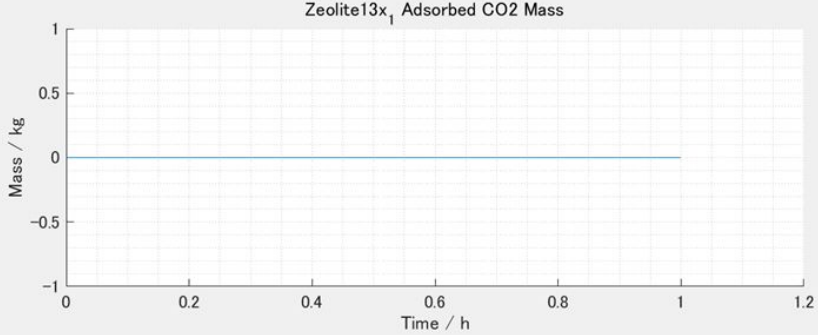

Zeolite13x$_2$ Adsorbed CO2 Mass
Mass / kg
Time / h

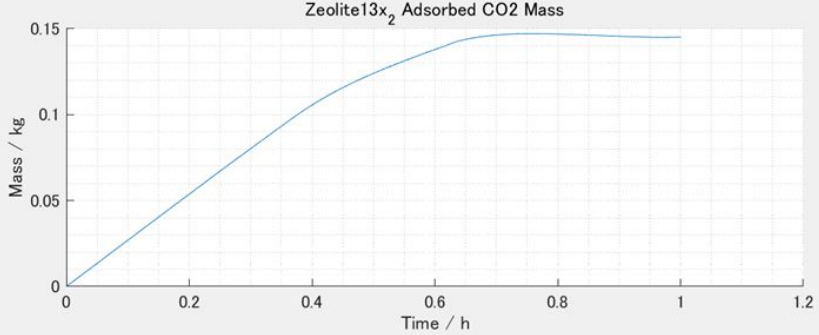

Zeolite5A$_1$ Adsorbed CO2 Mass
Mass / kg
Time / h

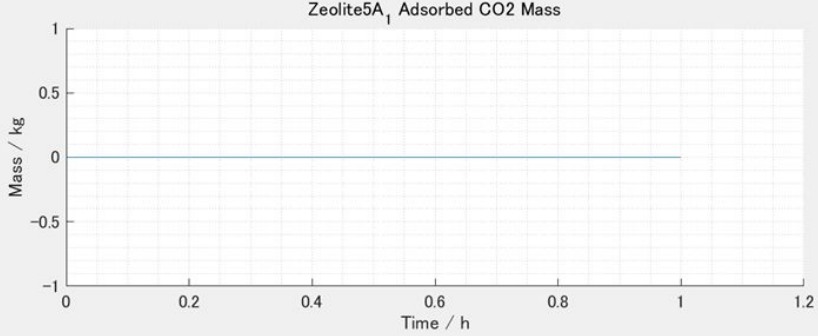

Zeolite5A$_2$ Adsorbed CO2 Mass
Mass / kg
Time / h

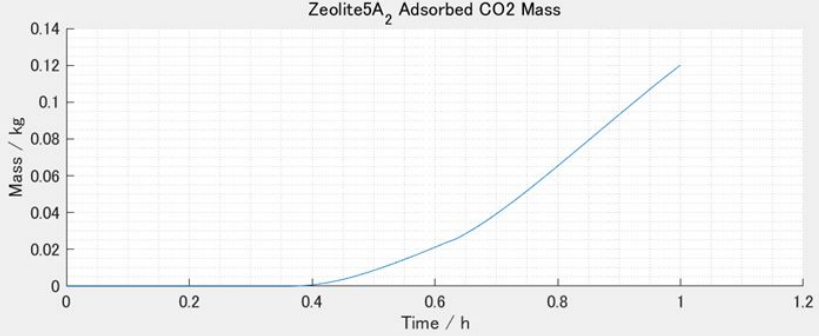

Sylobead1 Adsorbed CO2 Mass
Mass / kg
Time / h

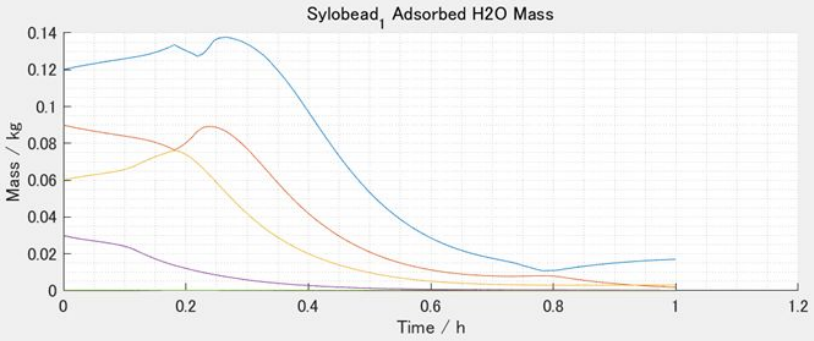

Sylobead2 Adsorbed CO2 Mass
Mass / kg
Time / h

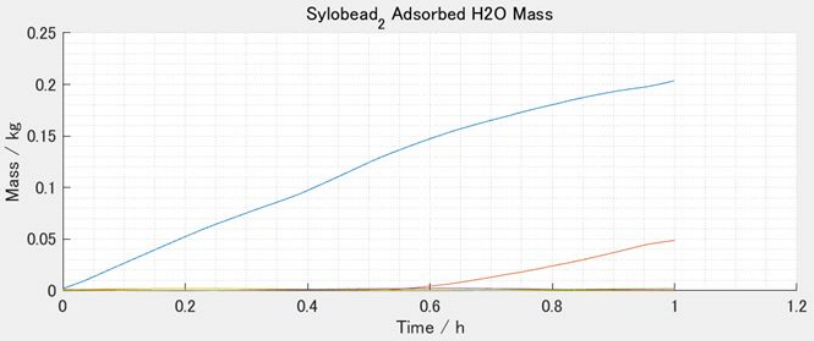

Zeolite13x1 Adsorbed CO2 Mass
Mass / kg
Time / h

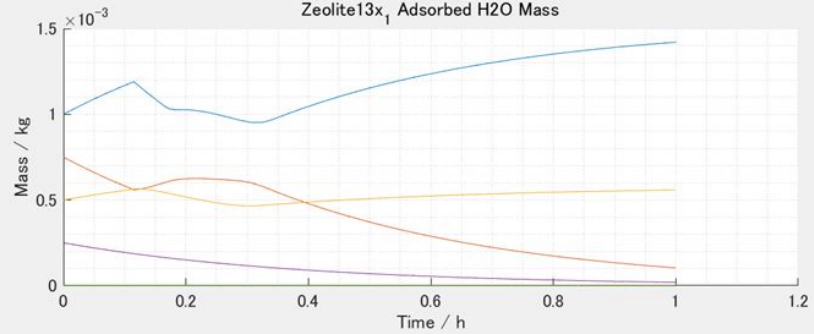

Zeolite13x2 Adsorbed CO2 Mass
Mass / kg
Time / h

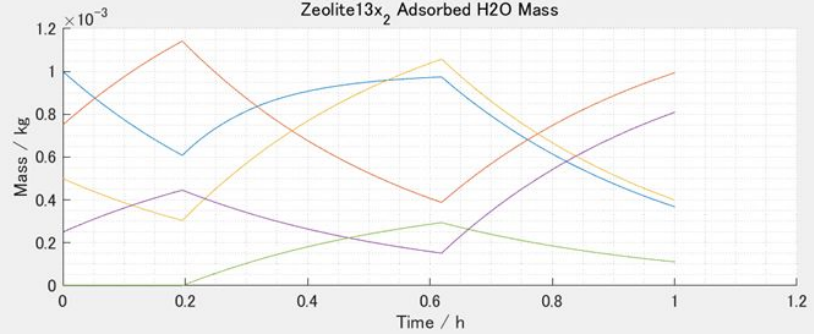

Zeolite5A1 Adsorbed CO2 Mass
Mass / kg
Time / h

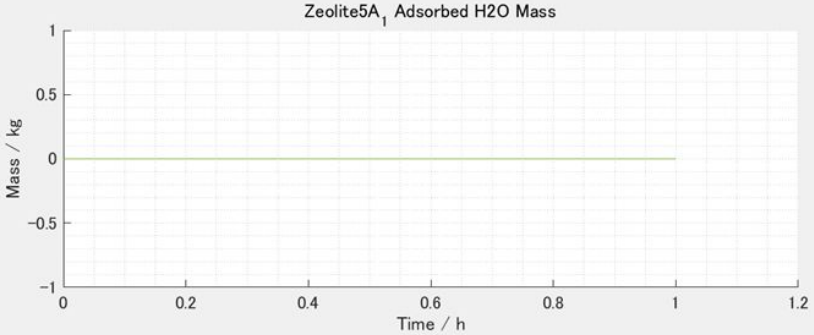

Zeolite5A2 Adsorbed CO2 Mass
Mass / kg
Time / h

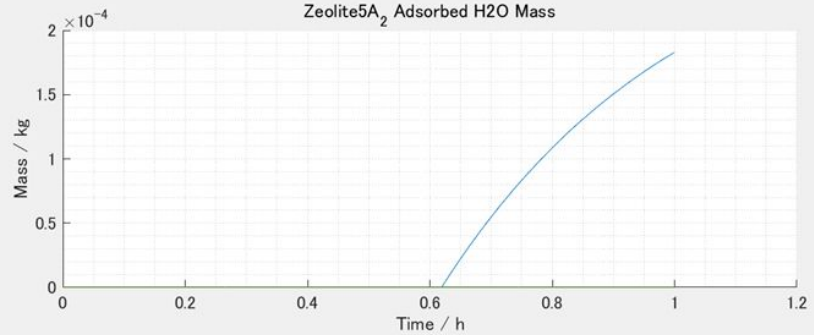

Sylobead$_1$ Adsorbed H2O Mass
Mass / kg
Time / h

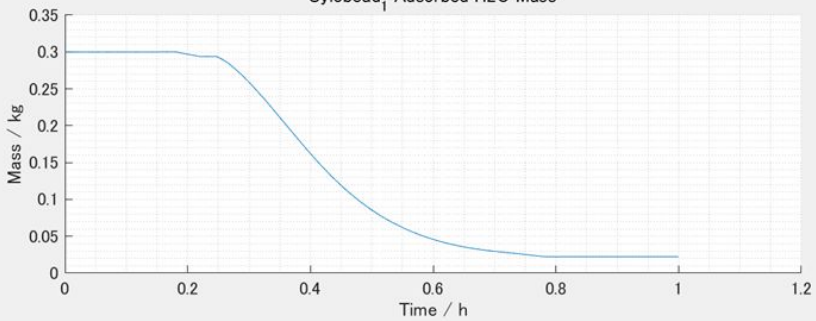

Sylobead$_2$ Adsorbed H2O Mass
Mass / kg
Time / h

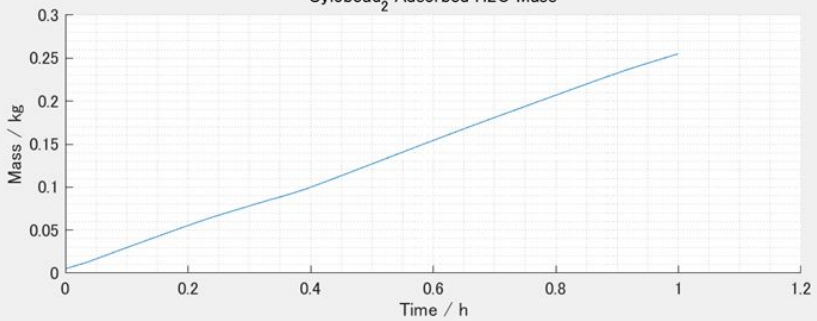

Zeolite13x$_1$ Adsorbed H2O Mass
$\times 10^{-3}$
Mass / kg
Time / h

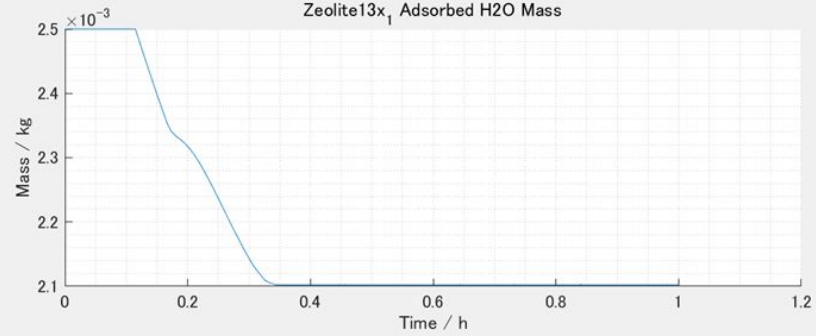

Zeolite13x$_2$ Adsorbed H2O Mass
$\times 10^{-3}$
Mass / kg
Time / h

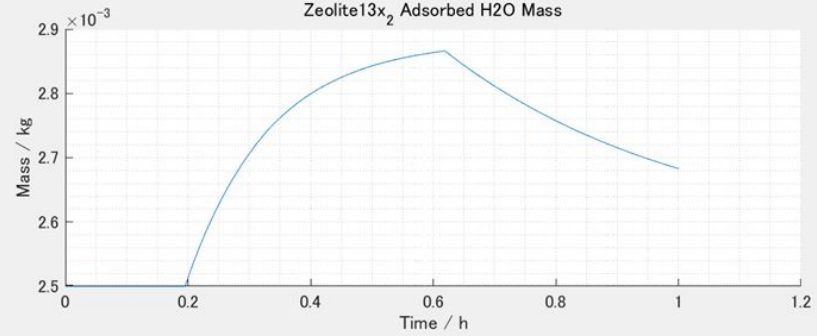

Zeolite5A$_1$ Adsorbed H2O Mass
Mass / kg
Time / h

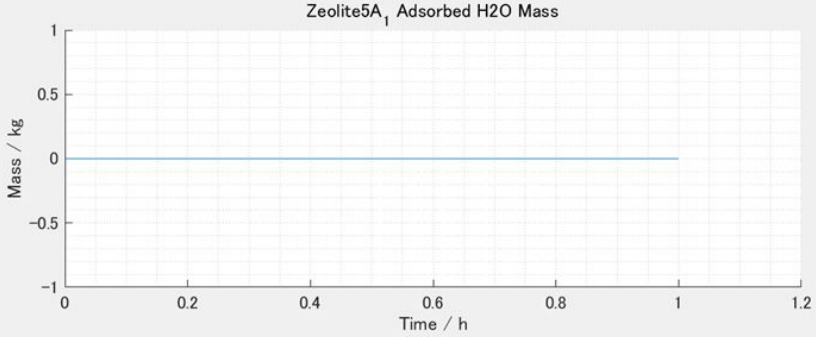

Zeolite5A$_2$ Adsorbed H2O Mass
$\times 10^{-4}$
Mass / kg
Time / h

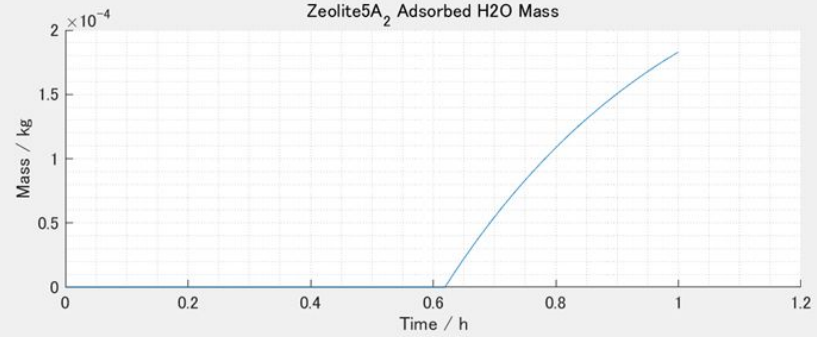

Sylobead$_1$ Adsorbed H2O Mass
Mass / kg
Time / h

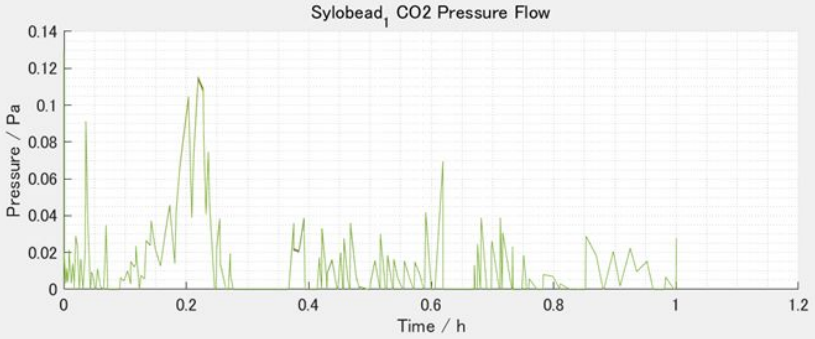

Sylobead$_2$ Adsorbed H2O Mass
Mass / kg
Time / h

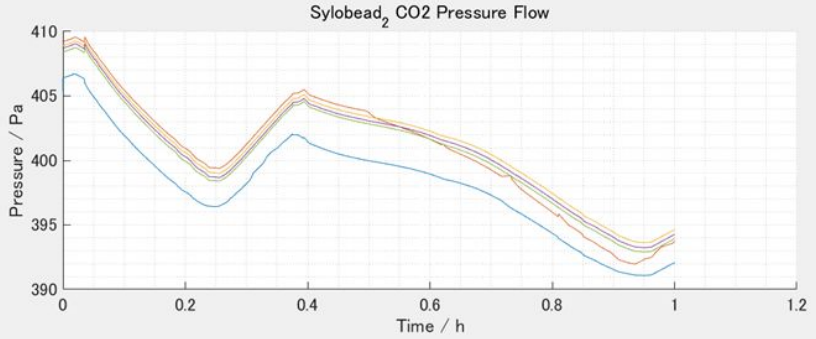

Zeolite13x$_1$ Adsorbed H2O Mass
$\times 10^{-3}$
Mass / kg
Time / h

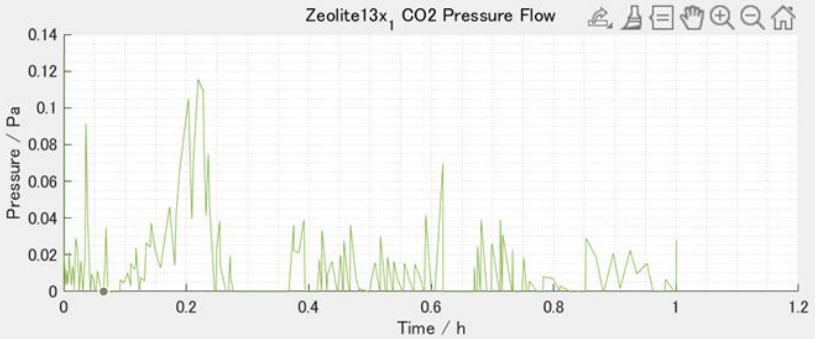

Zeolite13x$_2$ Adsorbed H2O Mass
$\times 10^{-3}$
Mass / kg
Time / h

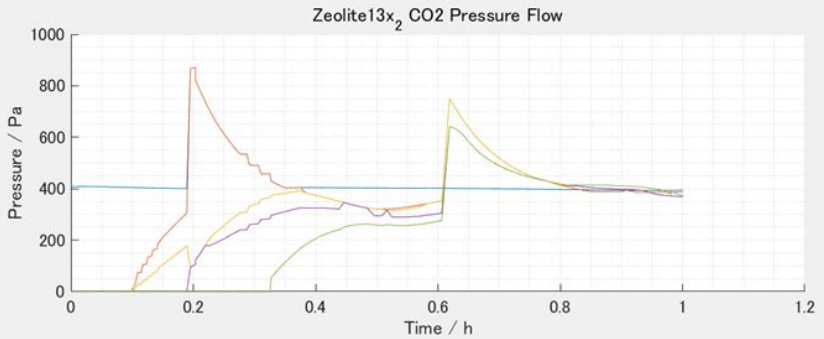

Zeolite5A$_1$ Adsorbed H2O Mass
Mass / kg
Time / h

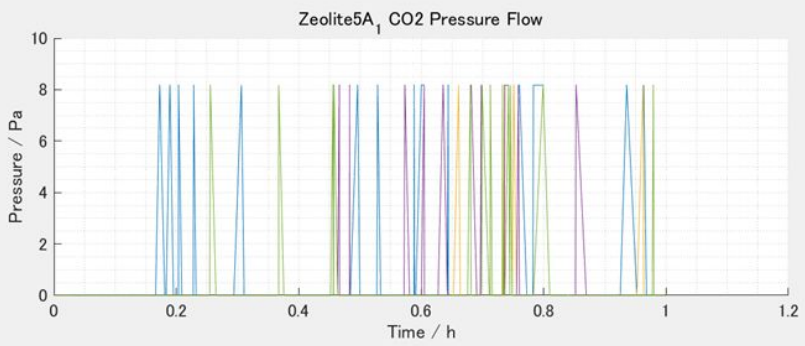

Zeolite5A$_2$ Adsorbed H2O Mass
$\times 10^{-4}$
Mass / kg
Time / h

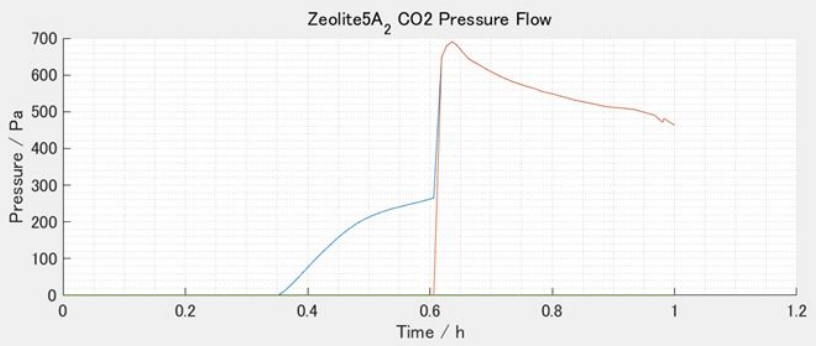

Sylobead1 CO2 Pressure Flow
Pressure / Pa
Time / h

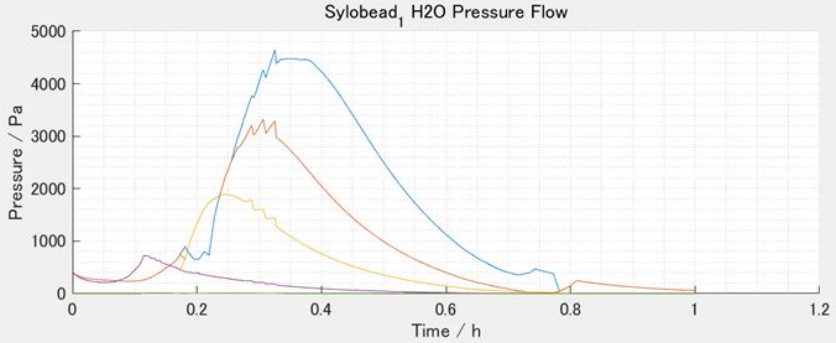

Sylobead2 CO2 Pressure Flow
Pressure / Pa
Time / h

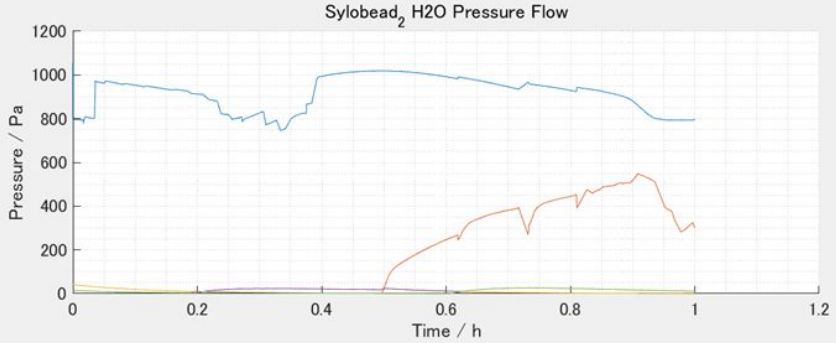

Zeolite13x1 CO2 Pressure Flow
Pressure / Pa
Time / h

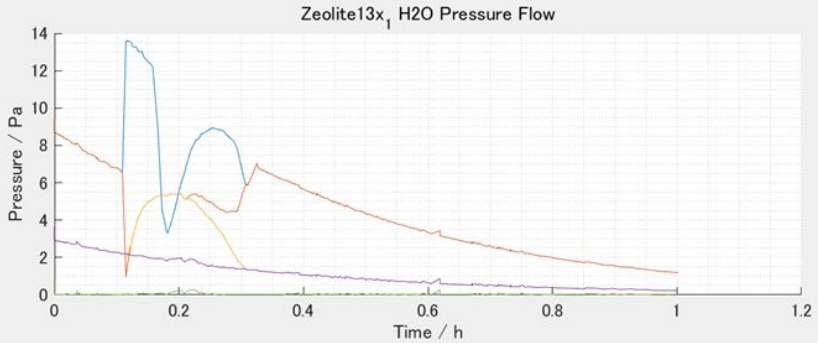

Zeolite13x2 CO2 Pressure Flow
Pressure / Pa
Time / h

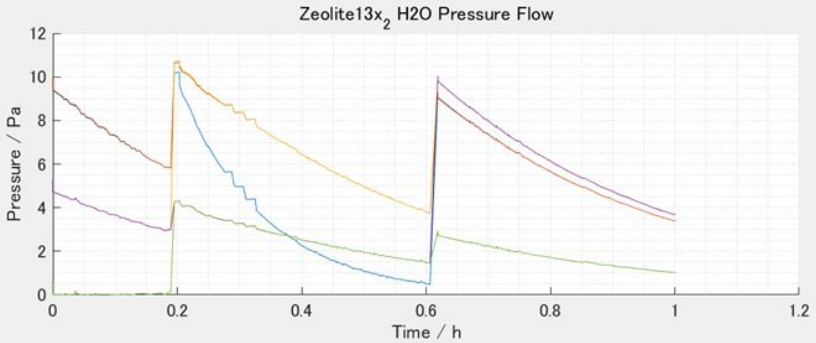

Zeolite5A1 CO2 Pressure Flow
Pressure / Pa
Time / h

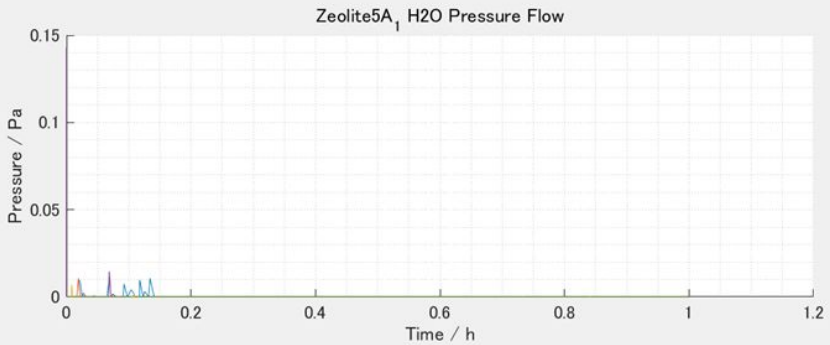

Zeolite5A2 CO2 Pressure Flow
Pressure / Pa
Time / h

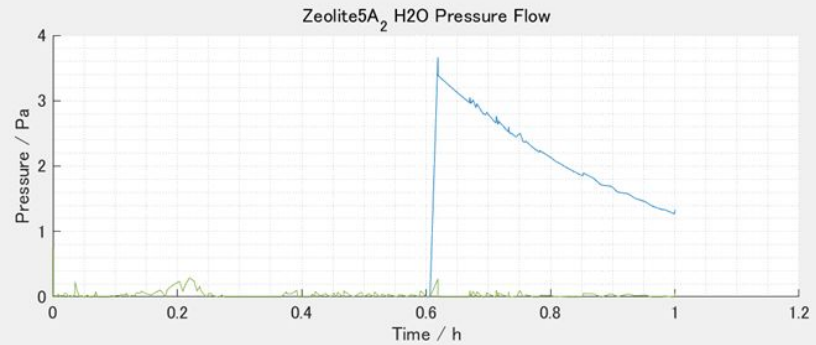

Sylobead$_1$ H2O Pressure Flow
Pressure / Pa
Time / h

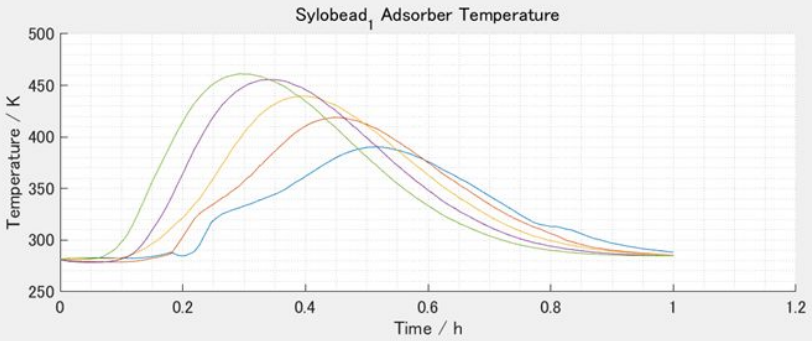

Sylobead$_2$ H2O Pressure Flow
Pressure / Pa
Time / h

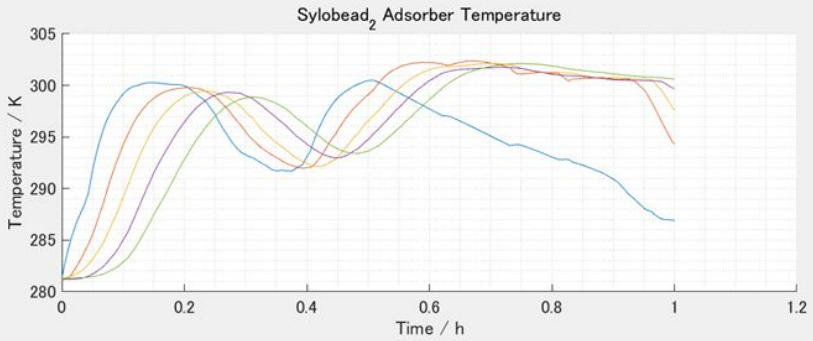

Zeolite13x$_1$ H2O Pressure Flow
Pressure / Pa
Time / h

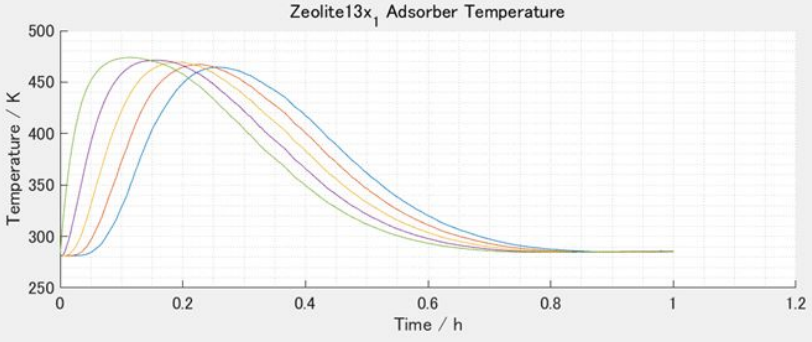

Zeolite13x$_2$ H2O Pressure Flow
Pressure / Pa
Time / h

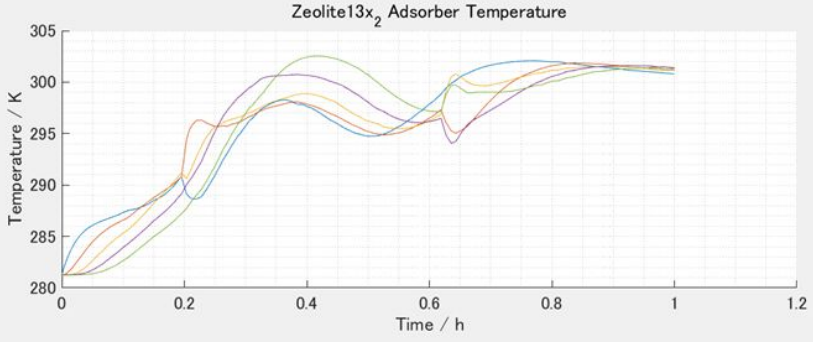

Zeolite5A$_1$ H2O Pressure Flow
Pressure / Pa
Time / h

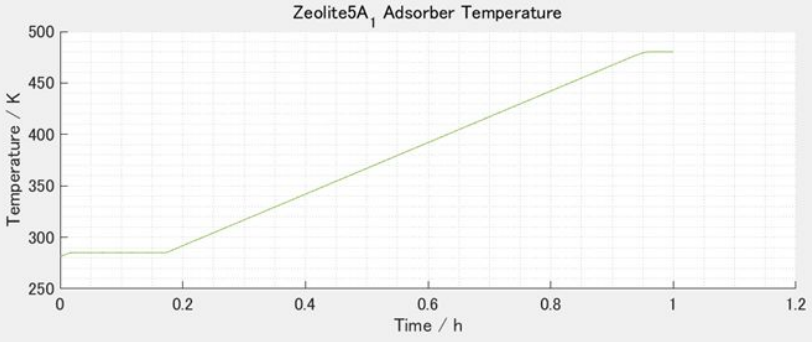

Zeolite5A$_2$ H2O Pressure Flow
Pressure / Pa
Time / h

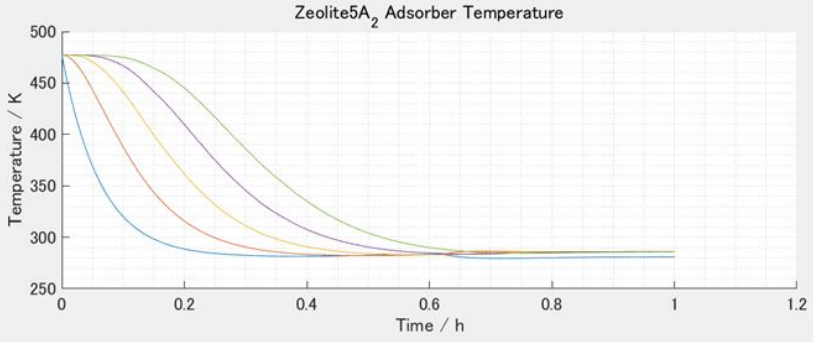

Sylobead$_1$ Adsorber Temperature
Temperature / K
Time / h

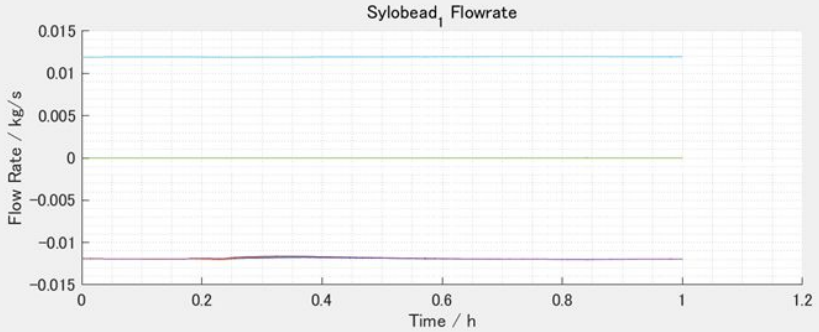

Sylobead$_2$ Adsorber Temperature
Temperature / K
Time / h

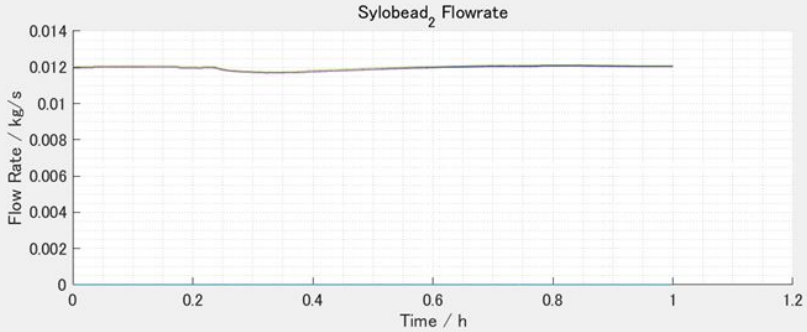

Zeolite13x$_1$ Adsorber Temperature
Temperature / K
Time / h

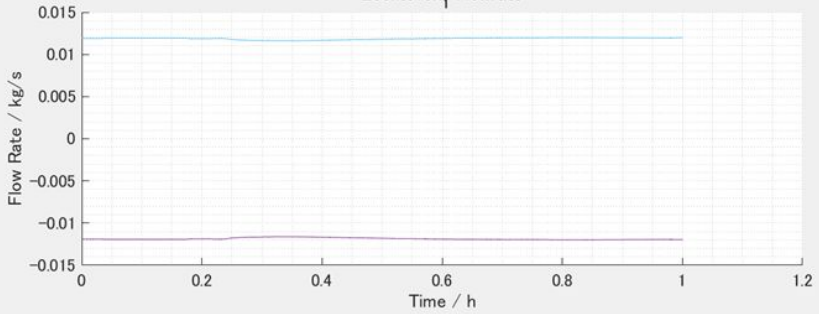

Zeolite13x$_2$ Adsorber Temperature
Temperature / K
Time / h

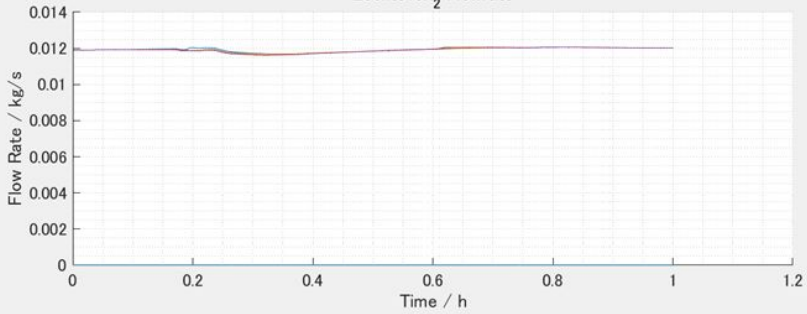

Zeolite5A$_1$ Adsorber Temperature
Temperature / K
Time / h

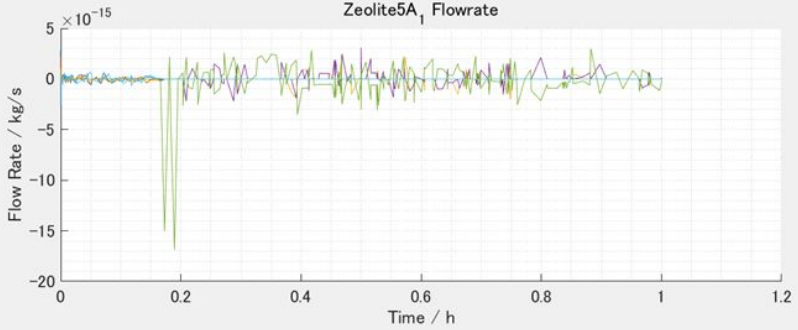

Zeolite5A$_2$ Adsorber Temperature
Temperature / K
Time / h

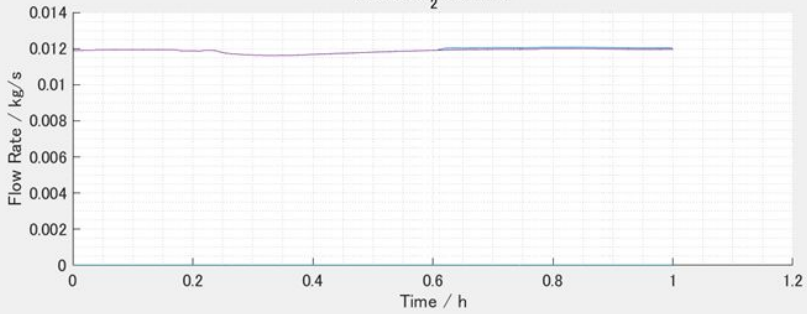

Sylobead1 Flowrate
Flow Rate / kg/s
Time / h

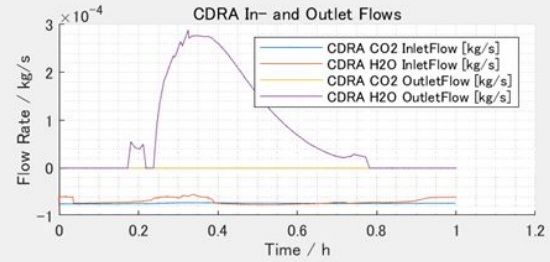

Sylobead2 Flowrate
Flow Rate / kg/s
Time / h

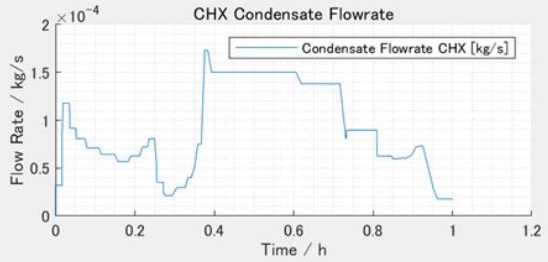

Zeolite13x1 Flowrate
Flow Rate / kg/s
Time / h

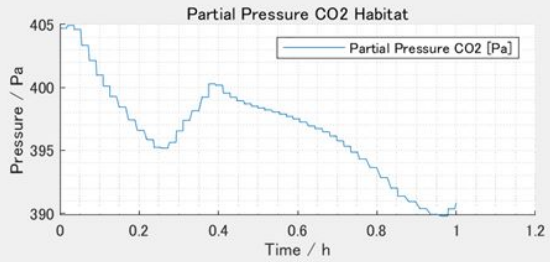

Zeolite13x2 Flowrate
Flow Rate / kg/s
Time / h

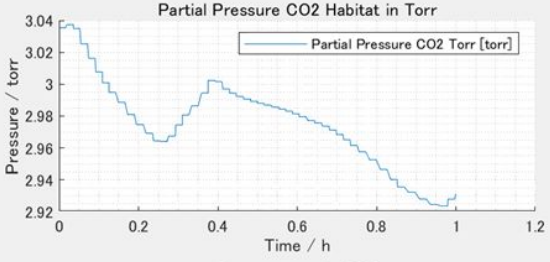

Zeolite5A1 Flowrate
×10−15
Flow Rate / kg/s
Time / h

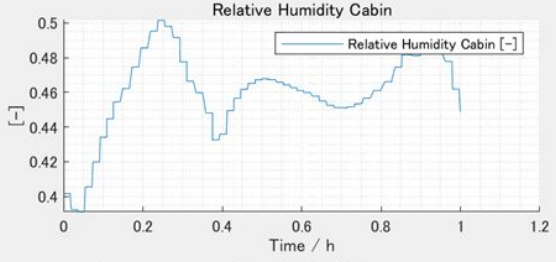

Zeolite5A2 Flowrate
Flow Rate / kg/s
Time / h

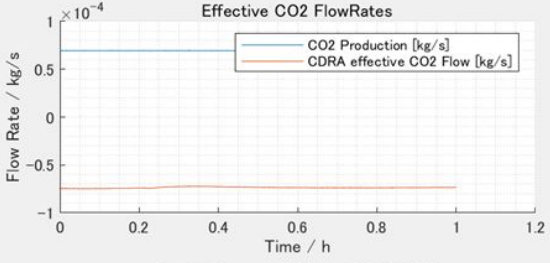

CDRA In- and Outlet Flows
×10^-4
Flow Rate / kg/s
Time / h
CDRA CO2 InletFlow [kg/s]
CDRA H2O InletFlow [kg/s]
CDRA CO2 OutletFlow [kg/s]
CDRA H2O OutletFlow [kg/s]

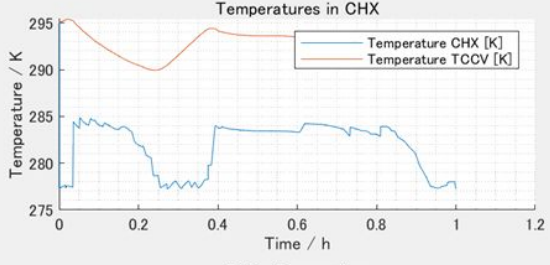

CHX Condensate Flowrate
×10^-4
Flow Rate / kg/s
Time / h
Condensate Flowrate CHX [kg/s]

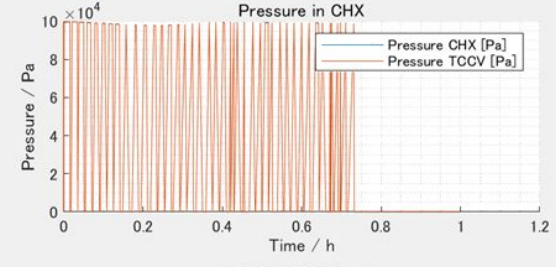

Partial Pressure CO2 Habitat
Pressure / Pa
Time / h
Partial Pressure CO2 [Pa]

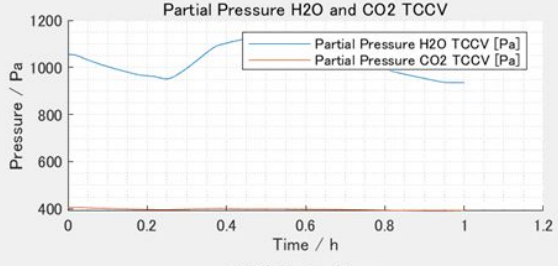

Partial Pressure CO2 Habitat in Torr
Pressure / torr
Time / h
Partial Pressure CO2 Torr [torr]

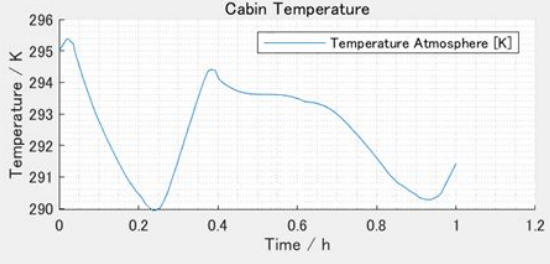

Relative Humidity Cabin
[-]
Time / h
Relative Humidity Cabin [-]

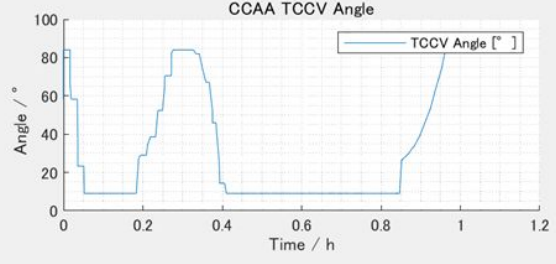

Effective CO2 FlowRates
×10^-4
Flow Rate / kg/s
Time / h
CO2 Production [kg/s]
CDRA effective CO2 Flow [kg/s]

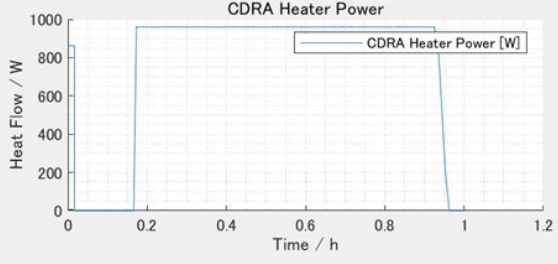

Temperatures in CHX
Temperature / K
Time / h
Temperature CHX [K]
Temperature TCCV [K]

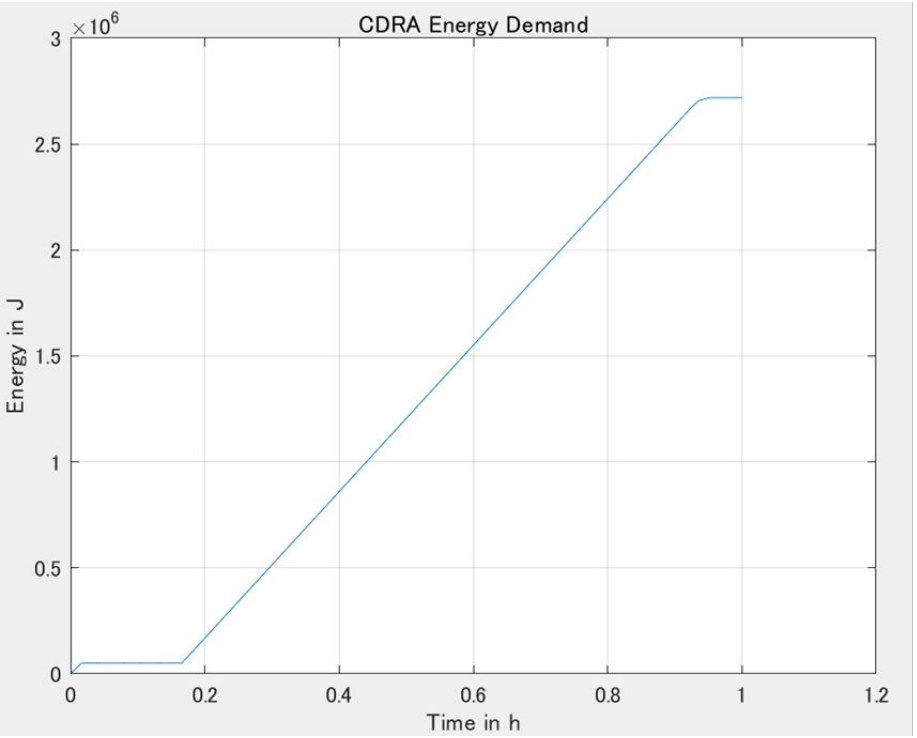

Pressure in CHX
×10^4
Pressure / Pa
Time / h
Pressure CHX [Pa]
Pressure TCCV [Pa]

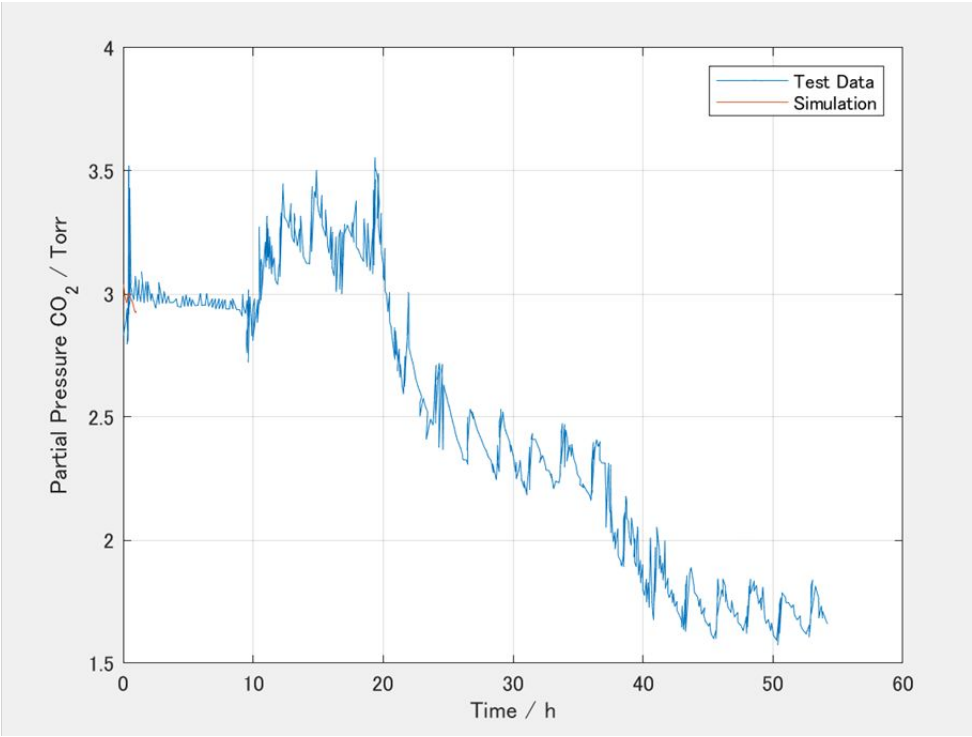

Partial Pressure H2O and CO2 TCCV
Pressure / Pa
Time / h
Partial Pressure H2O TCCV [Pa]
Partial Pressure CO2 TCCV [Pa]


Cabin Temperature
Temperature / K
Time / h
Temperature Atmosphere [K]


CCAA TCCV Angle
Angle / °
Time / h
TCCV Angle [°]

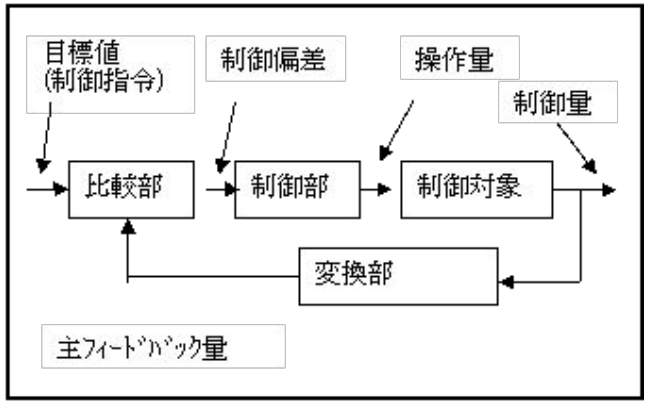

CDRA Heater Power
Heat Flow / W
Time / h
CDRA Heater Power [W]

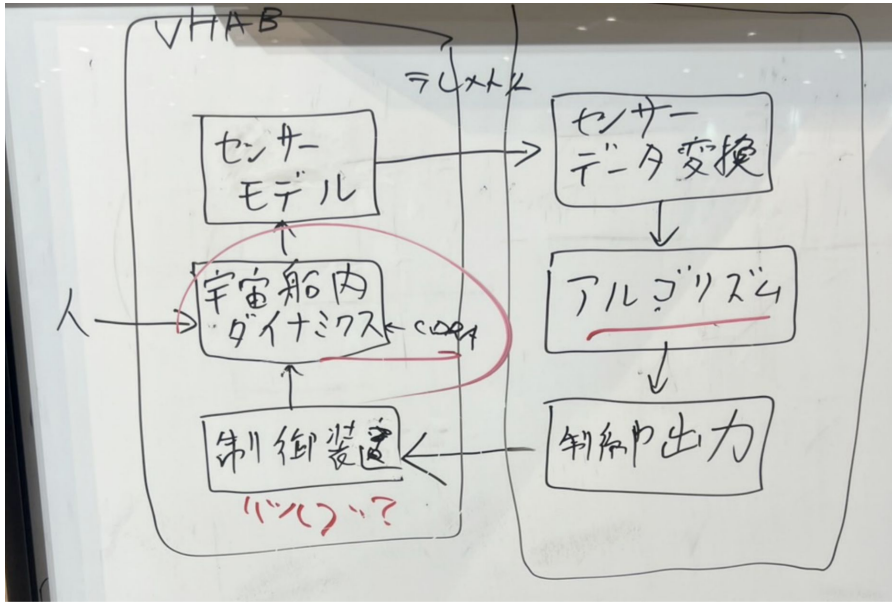

CDRA Energy Demand
×10^6
Energy in J
Time in h
0
0.5
1
1.5
2
2.5
3
0.2
0.4
0.6
0.8
1.2

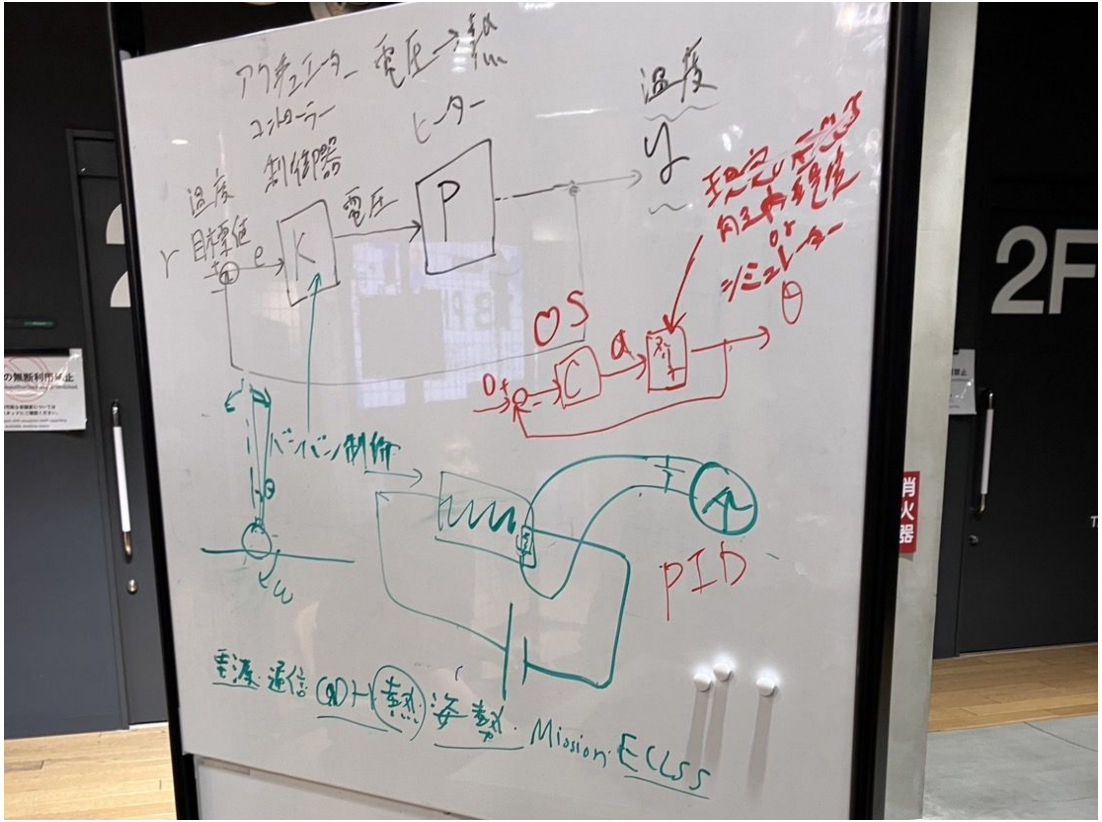

Test Data
Simulation
Partial Pressure CO2 / Torr
Time / h
1.5
2
2.5
3
3.5
4
0
10
20
30
40
50
60

ECLSS OS開発

ECLSS OSの果たす役割 ―制御系とは―







アクチュエーター 電圧→熱
温度
コントローラー
ヒーター
制御器
温度 目標値
電圧
K
P
y
r
e
OS
C
シミュレーター
バンバン制御
PID
電源 通信 OBC 熱 姿勢 Mission ECLSS
2F
消火器

ECLSS制御の現状調査

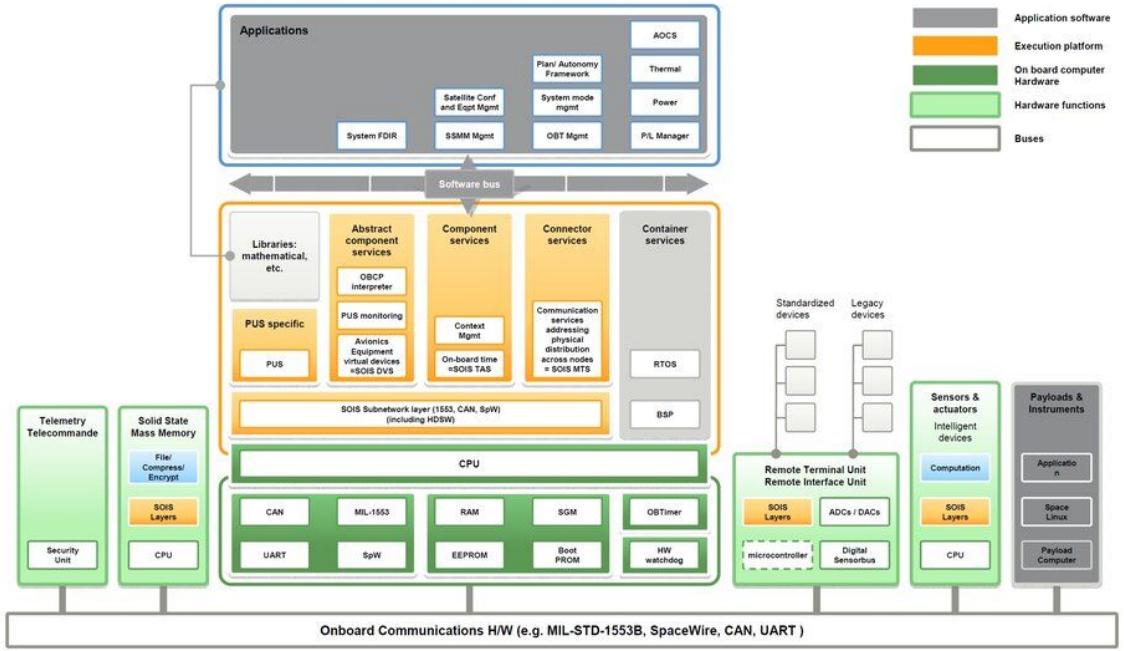
今のECLSSはどれくらい自動で動いているのか？

→クルーが自動で制御してる部分は？

→地上局が遠隔で制御している部分は？

→軌道上コンポが自動的にモードや制御量を切り替えている部分は？

ISS・Space ShuttleのOS (Onboard Computer)

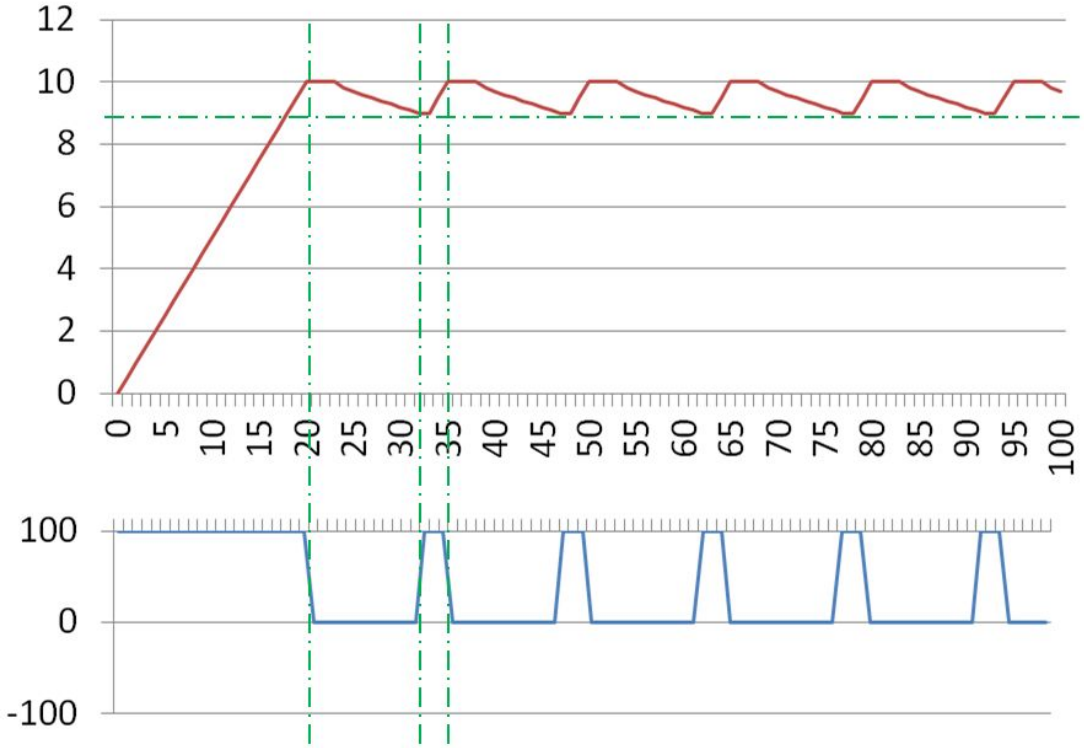


（ISS想定）

機能	制御	イベント	OS	クルー	地上局
CDRA	自動 ON/OFF	$ppCO_2$ 閾値逸脱	異常検知・制御指令	モニタリング 動作中の違和感を報告	モニタリング
OGS 酸素発生装置	自動 ON/OFF	ppO_2 閾値逸脱			
空気循環	流量制御	気圧閾値逸脱			
水再生	多段階プロセス	水タンクの水位モニター			
CDRA・除去タンク	手動 ON/OFF	故障・メンテナンス	異常検知・クルー通知	修復・交換作業	モニタリング
CDRA	機器出力レベル手動	非定常状態			
	吸着剤交換	吸着性能低下通知			
TCCS	フィルター交換	吸着性能低下通知			
全体システム	制御ループの設定 動作スケジュール・パラメータ変更	高度な診断・問題特定(センサー有・総合的判断あり)	-	-	高度な診断

人工衛星の熱制御にも使われる

Bang-Bang制御



main.m

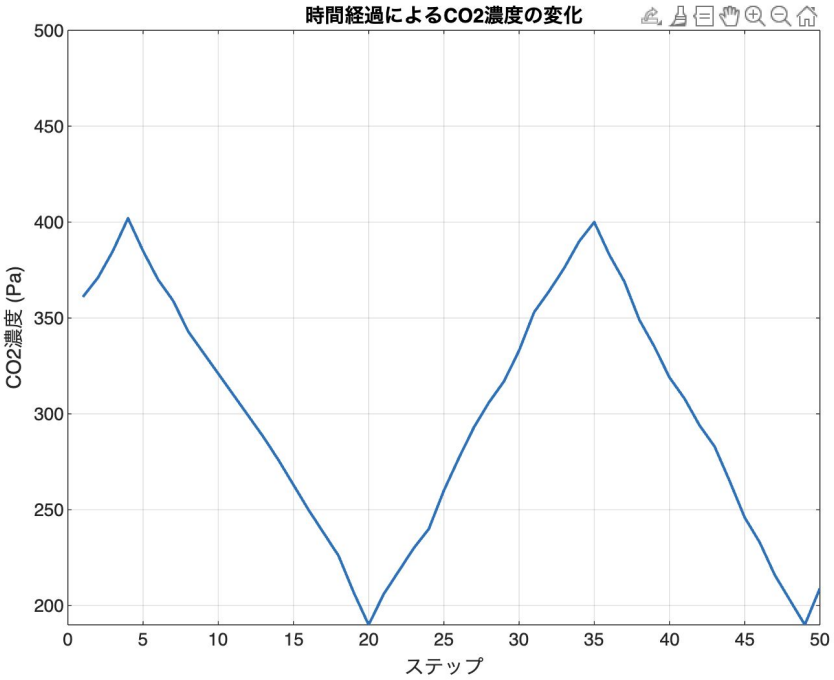
```
% インスタンス作成
cdra = CDRACo2Control;  % CDRA制御インスタンス
sensor = CO2Sensor;     % CO2センサーインスタンス

% シミュレーション設定
numSteps = 50; % シミュレーションステップ数

% データ保存用
co2Values = zeros(1, numSteps); % CO2濃度の履歴
cdraStates = zeros(1, numSteps); % CDRAのON/OFF状態の履歴

% シミュレーションループ
for i = 1:numSteps
    % センサーが次のCO2濃度を生成
    sensor = sensor.generateNextCO2(cdra.CDRAstate, cdra.ppCO2UpperLimit, cdra.ppCO2LowerLimit);
    currentCO2 = sensor.getCO2();

    % CO2濃度を用いてCDRAを制御
    cdra = cdra.updateState(currentCO2);

    % データを保存
    co2Values(i) = currentCO2;
    cdraStates(i) = cdra.CDRAstate; % true/false を 1/0 に変換して保存

    % 結果を表示
    fprintf('ステップ %d - CO2濃度: %d Pa, CDRA状態: %s\n', i, currentCO2, cdra.getState());
end

% グラフ1: CO2濃度
figure;
plot(1:numSteps, co2Values, '-', 'LineWidth', 1.5);
xlabel('ステップ');
ylabel('CO2濃度 (Pa)');
title('時間経過によるCO2濃度の変化');
ylim([190 500]);
grid on;

% グラフ2: CDRAのON/OFF状態
figure;
stairs(1:numSteps, cdraStates, 'LineWidth', 1.5);
xlabel('ステップ');
ylabel('CDRAの状態 (1=ON, 0=OFF)');
title('時間経過によるCDRA状態の変化');
ylim([-0.5 1.5]);
grid on;
```

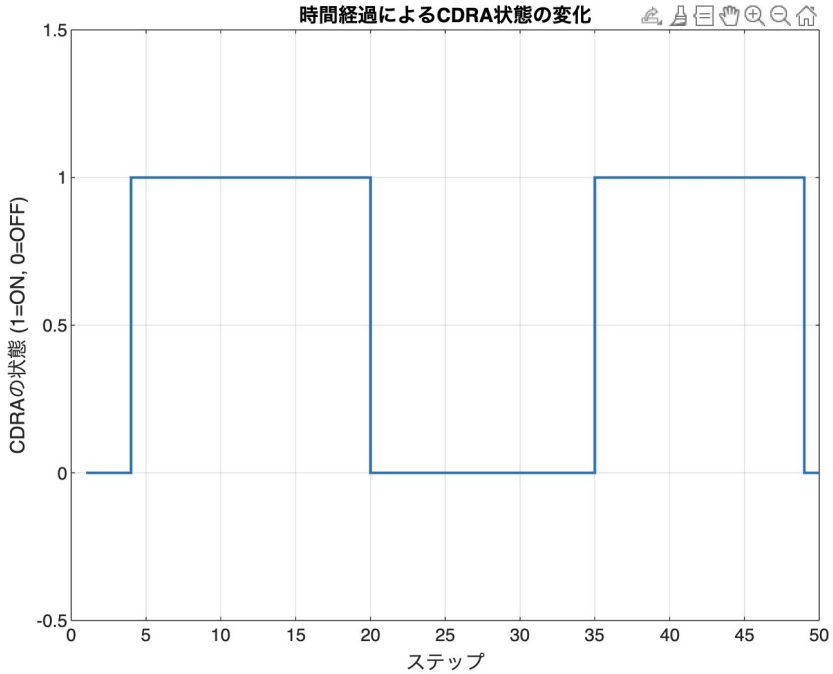
CDRACo2Control.m

```
classdef CDRACo2Control
    properties
        ppCO2UpperLimit = 400;  % CO2がこの値を超えたらCDRAをON (Pa)
        ppCO2LowerLimit = 200;  % CO2がこの値を下回ったらCDRAをOFF (Pa)
        CDRAstate = false;      % CDRAの状態 (false: OFF, true: ON)
    end

    methods
        % コンストラクタ
        function obj = CDRACo2Control(torigerOn, torigerOff)
            if nargin > 0
                obj.ppCO2UpperLimit = torigerOn;
                obj.ppCO2LowerLimit = torigerOff;
            end
        end

        % CDRAの状態を更新
        function obj = updateState(obj, currentCO2)
            if currentCO2 >= obj.ppCO2UpperLimit
                obj.CDRAstate = true; % ON
            elseif currentCO2 <= obj.ppCO2LowerLimit
                obj.CDRAstate = false; % OFF
            end
        end

        % 現在の状態を文字列で返す
        function state = getState(obj)
            if obj.CDRAstate
                state = 'ON';
            else
                state = 'OFF';
            end
        end
    end
end
```

CO2Sensor.m

```
classdef CO2Sensor
    properties
        currentCO2 = 350; % 初期CO2濃度 (Pa)
    end

    methods
        % 現在のCO2濃度を取得
        function co2 = getCO2(obj)
            co2 = obj.currentCO2;
        end

        % CO2濃度を設定
        function obj = setCO2(obj, newCO2)
            if newCO2 < 200 || newCO2 > 500
                error('CO2 concentration must be between 200 and 500 Pa.');
            end
            obj.currentCO2 = newCO2;
        end

        % CO2濃度を制御状態に基づいて更新
        function obj = generateNextCO2(obj, cdraState, ppCO2UpperLimit, ppCO2LowerLimit)
            % 制御状態がONの場合
            if cdraState
                % OFFになるまでCO2を下げる
                obj.currentCO2 = obj.currentCO2 - randi([10, 20]); % ランダムに減少
                obj.currentCO2 = max(obj.currentCO2, ppCO2LowerLimit - 10); % 下限を超えない
            else
                % ONになるまでCO2を上げる
                obj.currentCO2 = obj.currentCO2 + randi([10, 20]); % ランダムに増加
                obj.currentCO2 = min(obj.currentCO2, ppCO2UpperLimit + 10); % 上限を超えない
            end
        end
    end
end
```


時間経過によるCO2濃度の変化
CO2濃度 (Pa)
ステップ
500
450
400
350
300
250
200
0
5
10
15
20
25
30
35
40
45
50


時間経過によるCDRA状態の変化
CDRAの状態 (1=ON, 0=OFF)
ステップ
1.5
1
0.5
0
-0.5
0
5
10
15
20
25
30
35
40
45
50