```
clear;clc;close all

% note: these dynamics are valid only for regular waves, since I haven't
% implemented the convolution integral (memory effect).

plot_hydro = false;
RM5_spring_string = false; % whether RM5 uses a string-spring PTO

disp('Running RM5')
run_wec(5,RM5_spring_string,plot_hydro) % RM5
disp('Running RM3')
run_wec(3,RM5_spring_string,plot_hydro) % RM3
```

```
Running RM5
```

```
function run_wec(dof,RM5_spring_string,plot_hydro)
    [gear_ratio, spring_size, omega_test, lambda, T_amp, H, p] = get_numerical_inputs(dof, RM5_spring_string);

    powered = true;
    inner_loop_qty = H;
    run_sweep(powered,gear_ratio,spring_size,omega_test,inner_loop_qty,lambda,plot_hydro,p);

    powered = false;
    inner_loop_qty = T_amp;
    run_sweep(powered,gear_ratio,spring_size,omega_test,inner_loop_qty,lambda,plot_hydro,p);
end

function [gear_ratio, spring_size, omega_test, lambda, T_amp, H, p] = get_numerical_inputs(dof, RM5_spring_string)
    lambda = 50;
    H = [0.02,0.06,0.10,0.136]; % wave heights - from Olivia slack message 8/1/24
    T_amp = .2 : .2 : .6; %1:4; % torque amplitude for radiation tests (Nm)

    m_rotor = .354;
    r_rotor = .087/2;
    I_g = 1/2 * m_rotor * r_rotor^2; % moment of inertia of generator/powertrain
    % parameters
    p = struct('Ig',I_g,...
               'H',H,...
               'tau_max_Nm',4, 'motor_max_rpm',3000,... % motor max torque and speed
               ...%'T_s',0.05,'T_d',0.05,'b',.1,...  % static and dynamic friction torques and viscous friction coefficient
               'T_s',0,'T_d',0,'b',0,...
               'dof',dof,'string_spring',RM5_spring_string);

    % set gear ratios and springs to sweep over
    if dof==5
        gear_ratio = 1 : 2 : 9; % gear ratio range
    elseif dof==3
        pinion_radius = 0.010 : .005 : 0.025; % m
        gear_ratio = 1./pinion_radius; % 1/m
    end
    if dof == 3 || ~RM5_spring_string
        spring_size = 0;
    elseif dof == 5 && RM5_spring_string
        spring_size = 1:1:10;  % constant force spring size [N-m]
    end

    % set frequencies to sweep over
    % g = 9.8;
    % depth = 1.36; % Oregon tank depth [m]
    % w_min_deepwater = sqrt(2*g./depth);
    % w_max = 10; % rad/s
    % omega_idx_min = find(w_scaled > w_min_deepwater,1,'first');
    % omega_idx_max = find(w_scaled < w_max,          1,'last');
    % omega_idxs = omega_idx_min : omega_idx_max;
    omega_test = [7.07, 8.49, 9.83]; % override with actual test values
end

function [] = run_sweep(powered,gear_ratio,spring_size,omega_test,inner_loop_qty,lambda,plot_hydro,p)
    [w_scaled, A_scaled, B_scaled, K_scaled, gamma_scaled] = coeffs(p.dof,lambda,plot_hydro);

    if p.dof == 5
        K_scaled = 0.8580; % override since K_scaled is somehow negative
        I_scaled = 0.0372; % override to match actual ptype moment of inertia
        mI_A_scaled = A_scaled + I_scaled;
    elseif p.dof == 3
        m_scaled = 1.208;
        mI_A_scaled = A_scaled + m_scaled;
    end
    w_diff = abs(w_scaled - omega_test');
    [~,omega_idxs] = find(w_diff == min(w_diff,[],2));
    omegas = w_scaled(omega_idxs);

    if powered
        powered_label = 'powered';
        inner_loop_label = 'H - regular wave height (m)';
    else
```

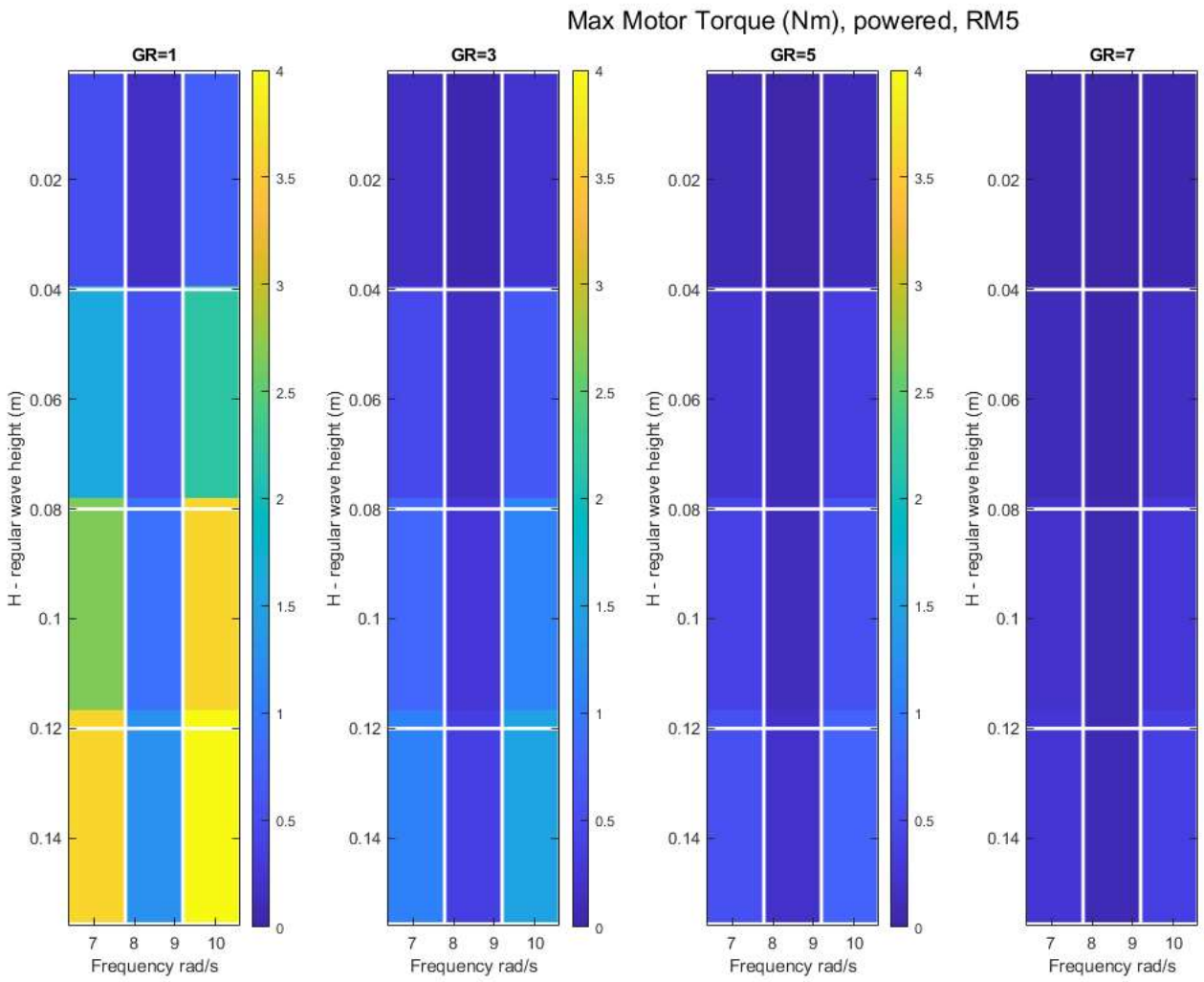
```
    powered_label = 'forced oscillation';
    inner_loop_label = 'Generator torque amplitude (Nm)';
end
RM_label = [', RM' num2str(p.dof)];

non_slack_combo  = zeros(length(gear_ratio),length(spring_size),length(omega_idxs),length(inner_loop_qty));
max_motor_torque = zeros(length(gear_ratio),length(spring_size),length(omega_idxs),length(inner_loop_qty));
amplitude        = zeros(length(gear_ratio),length(spring_size),length(omega_idxs),length(inner_loop_qty));
power            = zeros(length(gear_ratio),length(spring_size),length(omega_idxs),length(inner_loop_qty));
plot_timeseries = false;

[spring_mesh, omega_mesh] = meshgrid(spring_size,omegas);

p.Kh = K_scaled; % hydrodynamic stiffness
fig_start_number = numel(findobj('Type', 'figure'));
for i = 1:length(gear_ratio)

    p.GR = gear_ratio(i); % gear ratio
    if p.dof == 5
        GR_label = ['GR=' num2str(gear_ratio(i))];
    else
        GR_label = ['Pinion radius = ' num2str(1000/gear_ratio(i)) ' mm'];
    end

    for j = 1:length(spring_size)

        p.T_spring = spring_size(j); % spring torque
        if p.dof == 5 && p.string_spring
            spring_label = [', spring = ' num2str(spring_size(j)) ' Nm'];
        else
            spring_label = '';
        end

        for k = 1:length(omega_idxs)
            omega_idx = omega_idxs(k);
            p.Bh = B_scaled(1,omega_idx); % hydrodynamic damping
            p.I = mI_A_scaled(1,omega_idx); % moment of inertia of flap
            p.w = w_scaled(1,omega_idx); % wave frequency

            for inner_idx = 1:length(inner_loop_qty)
                if powered
                    % impedance matching
                    p.Kp = p.I * p.w^2 - p.Kh;
                    p.Bp = p.Bh;
                    H = inner_loop_qty(inner_idx);
                    p.Fh = H/2 * gamma_scaled(1,omega_idx); % exciting force amplitude
                    odefun = @(t,y,yp)dynamics_power(t,y,yp,p);
                else
                    p.T_gen_amplitude = inner_loop_qty(inner_idx);
                    odefun = @(t,y,yp)dynamics_radiation(t,y,yp,p);
                end

                [non_slack_combo(i,j,k,inner_idx), ...
                    max_motor_torque(i,j,k,inner_idx),...
                    amplitude(i,j,k,inner_idx),...
                    power(i,j,k,inner_idx)   ] = run_sim(odefun,p,plot_timeseries);
            end
        end

        % results for this GR and spring, over each frequency and amplitude
        each_test_non_slack = squeeze(non_slack_combo(i,j,:,:))';
        each_test_torque    = squeeze(max_motor_torque(i,j,:,:))';
        each_test_amplitude = squeeze(amplitude(i,j,:,:))';
        each_test_power     = squeeze(power(i,j,:,:))';

        % titles
        GR_spring_title = [GR_label spring_label];
        omegas_label = 'Frequency rad/s';

        % plots for this GR and spring, over each frequency and amplitude
        subplot_number = i + (j-1)*(length(gear_ratio));

        f1 = figure(fig_start_number + 1);
        f1.WindowState = 'maximized';
        subplot(length(spring_size),length(gear_ratio),subplot_number)
        sgtitle(['Max Motor Torque (Nm), ' powered_label RM_label])
        plot_results(omegas,omegas_label, inner_loop_qty,inner_loop_label, each_test_torque, GR_spring_title,[0 p.tau_max_Nm])

        f2 = figure(fig_start_number + 2);
        f2.WindowState = 'maximized';
        subplot(length(spring_size),length(gear_ratio),subplot_number)
        sgtitle(['WEC amplitude, ' powered_label RM_label])
        plot_results(omegas,omegas_label, inner_loop_qty,inner_loop_label, each_test_amplitude,GR_spring_title,[0 max(each_test_amplitude,[],'all')])

        f3 = figure(fig_start_number + 3);
        f3.WindowState = 'maximized';
        subplot(length(spring_size),length(gear_ratio),subplot_number)
        sgtitle(['WEC power (W), ' powered_label RM_label])
        plot_results(omegas,omegas_label, inner_loop_qty,inner_loop_label, each_test_power,    GR_spring_title,[0 max(each_test_power,[],'all')])

        if p.dof == 5 && p.string_spring
```

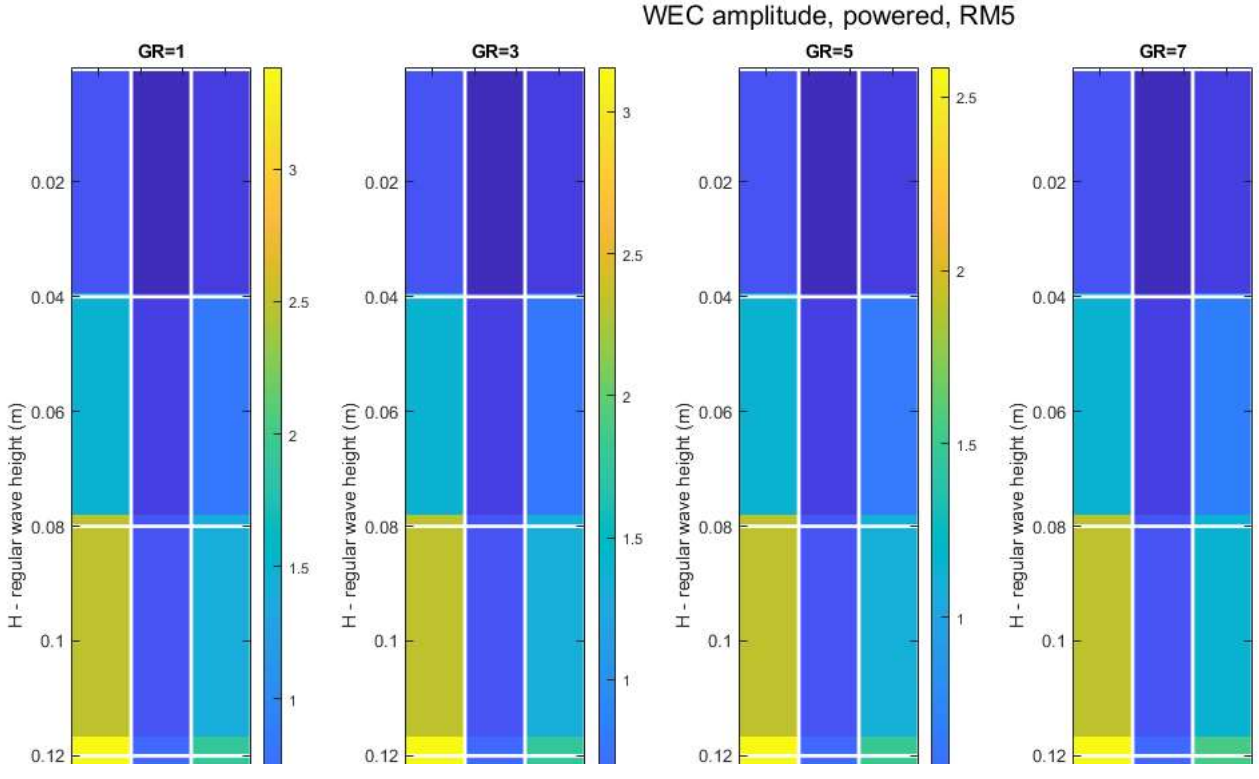
```
                f4 = figure(fig_start_number + 4);
                f4.WindowState = 'maximized';
                subplot(length(spring_size),length(gear_ratio),subplot_number)
                sgtitle(['Acceptable combinations for preventing slackness, ' powered_label RM_label])
                plot_results(omegas,omegas_label, inner_loop_qty,inner_loop_label, each_test_non_slack,GR_spring_title,[0 1])
            end

            each_test_non_slack(isnan(each_test_non_slack)) = 0;
            each_test_torque(isnan(each_test_torque)) = 0;
            all_tests_ok = all(each_test_non_slack & each_test_torque<=p.tau_max_Nm,'all');

        end

    end

    % overall comparison of which GRs and springs work for every test
    if p.dof == 3
        GR_label = 'Pinion radius (mm)';
        x_GR = 1000./gear_ratio;
    elseif p.dof == 5
        GR_label = 'GR (-)';
        x_GR = gear_ratio;
    end
    figure
    plot_results(x_GR,GR_label,spring_size,'spring torque (Nm)',all_tests_ok,...
        ['Acceptable for all tests - ' powered_label RM_label],[0 1])

end

function plot_results(x,x_label,y,y_label,z,z_title,z_lim)
    h = imagesc(x, y, z);
    colorbar
    xlabel(x_label)
    ylabel(y_label)
    title(z_title)
    if z_lim(1) ~= z_lim(2) && all(~isnan(z_lim))
        clim(z_lim)
    end
    draw_lines(x,y)
    set(h, 'AlphaData', ~isnan(z))
end

function [non_slack, max_motor_torque, amplitude, power] = run_sim(odefun,p,plot_timeseries)
    % ode inputs
    T = 2*pi/p.w;
    y0 = [0;0];
    yp0 = [0;0];
    tspan = [0 5*T];

    % ode solve
    options = odeset('MaxStep',T/20);

    %[y0,yp0] = decic(odefun, 0, y0, [1,1], yp0, [0,0]);
    sol = ode15i(odefun,tspan,y0,yp0,options);
    t = linspace(tspan(1),tspan(end));
    try
        [y,yp] = deval(sol,t);
        [err,P,T_gen,T_fric,T_string] = odefun(t,y,yp);

        if p.dof == 5 && p.string_spring && any(T_string <= 0)
            non_slack = false;
            max_motor_torque = NaN;
            %fprintf('UNACCEPTABLE GR %f and Spring Force %f and Frequency %f\n',gear_ratio(i),spring_size(j),w_scaled(1,omega_idx))
        else
            non_slack = true;
            max_motor_torque = max(abs(T_gen));
            %fprintf('  ACCEPTABLE GR %f and Spring Force %f and Frequency %f\n',gear_ratio(i),spring_size(j),w_scaled(1,omega_idx))

            if plot_timeseries
                % plot solution
                figure
                plot(t,y,  t,T_gen, t,T_fric, t,T_string, t,P)
                xlabel('Time (s)')
                legend('$\theta$','$\dot{\theta}$','$T_{gen}$','$T_{fric}$','$T_{string}$', ...
                    'P','interpreter','latex','FontSize',14)

                figure
                xlabel('Time (s)')
                ylabel('Integration Error')
                plot(t,err(1,:),t,err(2,:))
                legend('err_T (Nm)','err_v (m/s)')
            end
        end

        power = max(P);
        amplitude = max(y(1,:)) - min(y(1,:));
    catch exception
        if strcmp(exception.identifier, 'MATLAB:deval:SolOutsideInterval')
            %fprintf('SIM FAILED   GR %f and Spring Force %f and Frequency %f\n',gear_ratio(i),spring_size(j),w_scaled(1,omega_idx))
            non_slack = NaN;
            max_motor_torque = NaN;
```

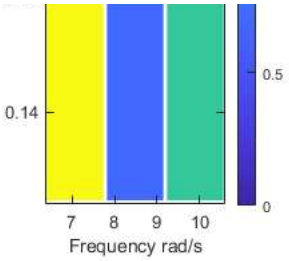
```
            power = NaN;
            amplitude = NaN;
        else
            rethrow(exception);
        end
    end
end

function draw_lines(x,y)
    hold on
    if length(x)>1
        d_x = diff(x(1:2))/2;
    else
        d_x = .5;
    end
    if length(y)>1
        d_y = diff(y(1:2))/2;
    else
        d_y = .5;
    end

    for xi = x
        plot(d_x+[xi xi],[min(y)-d_y max(y)+d_y],'w','LineWidth',2) % vert line
    end
    xlim([min(x)-d_x, max(x)+d_x])

    for yi = y
        plot([min(x)-d_x max(x)+d_x],d_y+[yi yi],'w','LineWidth',2) % horz line
    end
    ylim([min(y)-d_y, max(y)+d_y])
end

% dynamics for forced oscillation radiation test
function [err,P,T_gen,T_fric,T_string] = dynamics_radiation(t,y,yp,p)

    T_gen_cmd = p.T_gen_amplitude * sin(p.w * t);
    T_exc = 0;

    [err,P,T_gen,T_fric,T_string] = dynamics(T_exc,T_gen_cmd,t,y,yp,p);

end

% dynamics for impedance controlled power generation
function [err,P,T_gen,T_fric,T_string] = dynamics_power(t,y,yp,p)

    % states
    th = y(1,:);
    th_dot = y(2,:);

    % effective gear ratio
    alpha = 0; % fixme - should be a function of theta
    GR_eff = p.GR * cos(alpha);

    T_gen_cmd = -(p.Kp * th + p.Bp * th_dot) / GR_eff; % fixme

    % hydro excitaton torques on flap
    T_exc = p.Fh * sin(p.w * t);

    [err,P,T_gen,T_fric,T_string] = dynamics(T_exc,T_gen_cmd,t,y,yp,p);

end

% shared dynamics function - equation of motion
function [err,P,T_gen,T_fric,T_string] = dynamics(T_exc,T_gen_cmd,t,y,yp,p)
    % states
    th = y(1,:);
    th_dot = y(2,:);

    th_dot_next = yp(1,:);
    th_ddot = yp(2,:);

    % effective gear ratio
    alpha = 0; % fixme - should be a function of theta
    GR_eff = p.GR * cos(alpha);

    % hydro and interial torques
    T_hydro = p.Kh * th + p.Bh * th_dot;
    T_inertia = p.I * th_ddot;

    % string torque on drivetrain
    T_string = (T_inertia + T_hydro - T_exc) / GR_eff;
    if p.dof == 5 && p.string_spring
        T_string(T_string < 0) = 0;
    end

    % generator torque limiting
    T_gen = min(max(T_gen_cmd/p.tau_max_Nm, -1), 1) * p.tau_max_Nm;
    motor_rpm = GR_eff * th_dot * 60/(2*pi);
    T_gen(abs(motor_rpm) > p.motor_max_rpm) = 0;

    % friction
```

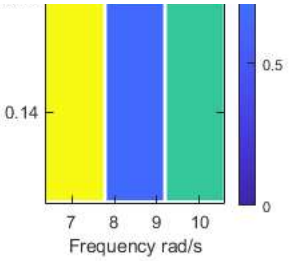
```
    zero_speed_idxs = ismembertol(th_dot,0);
    net_trq_without_fric = T_gen - T_string + p.T_spring;
    T_static = min(p.T_s,abs(net_trq_without_fric)) .* sign(net_trq_without_fric);
    T_dynamic = p.T_d * sign(th_dot) + p.b * GR_eff * th_dot;
    T_fric = zeros(size(T_gen));
    T_fric(zero_speed_idxs) = T_static(zero_speed_idxs);
    T_fric(~zero_speed_idxs) = T_dynamic(~zero_speed_idxs);

    % unbalanced torque on drivetrain
    T_unb = p.Ig * GR_eff * th_ddot - T_gen + T_fric + T_string - p.T_spring;

    err = [T_unb; th_dot - th_dot_next];

    % power
    P = T_gen .* th_dot * GR_eff;
end
```

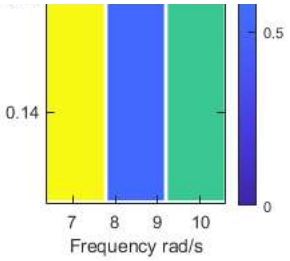
```
Running RM3
```

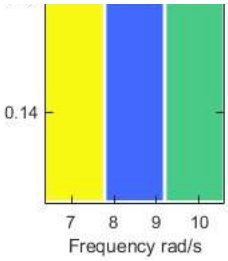

Max Motor Torque (Nm), powered, RM5
GR=1
GR=3
GR=5
GR=7
H - regular wave height (m)
Frequency rad/s

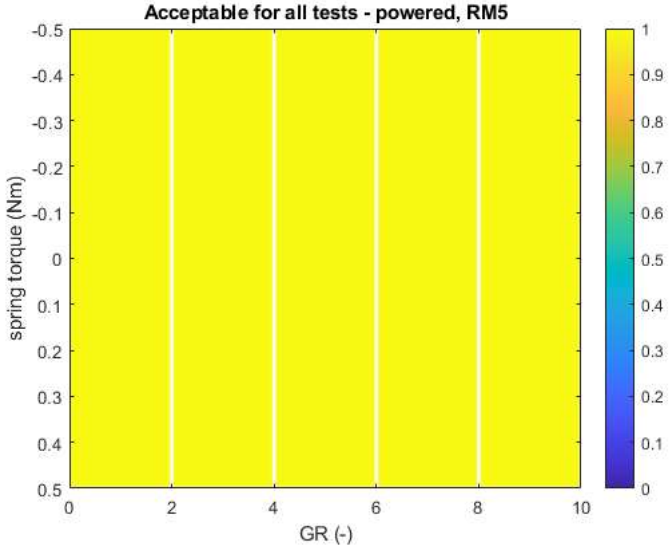

WEC amplitude, powered, RM5
GR=1
GR=3
GR=5
GR=7
H - regular wave height (m)
Frequency rad/s

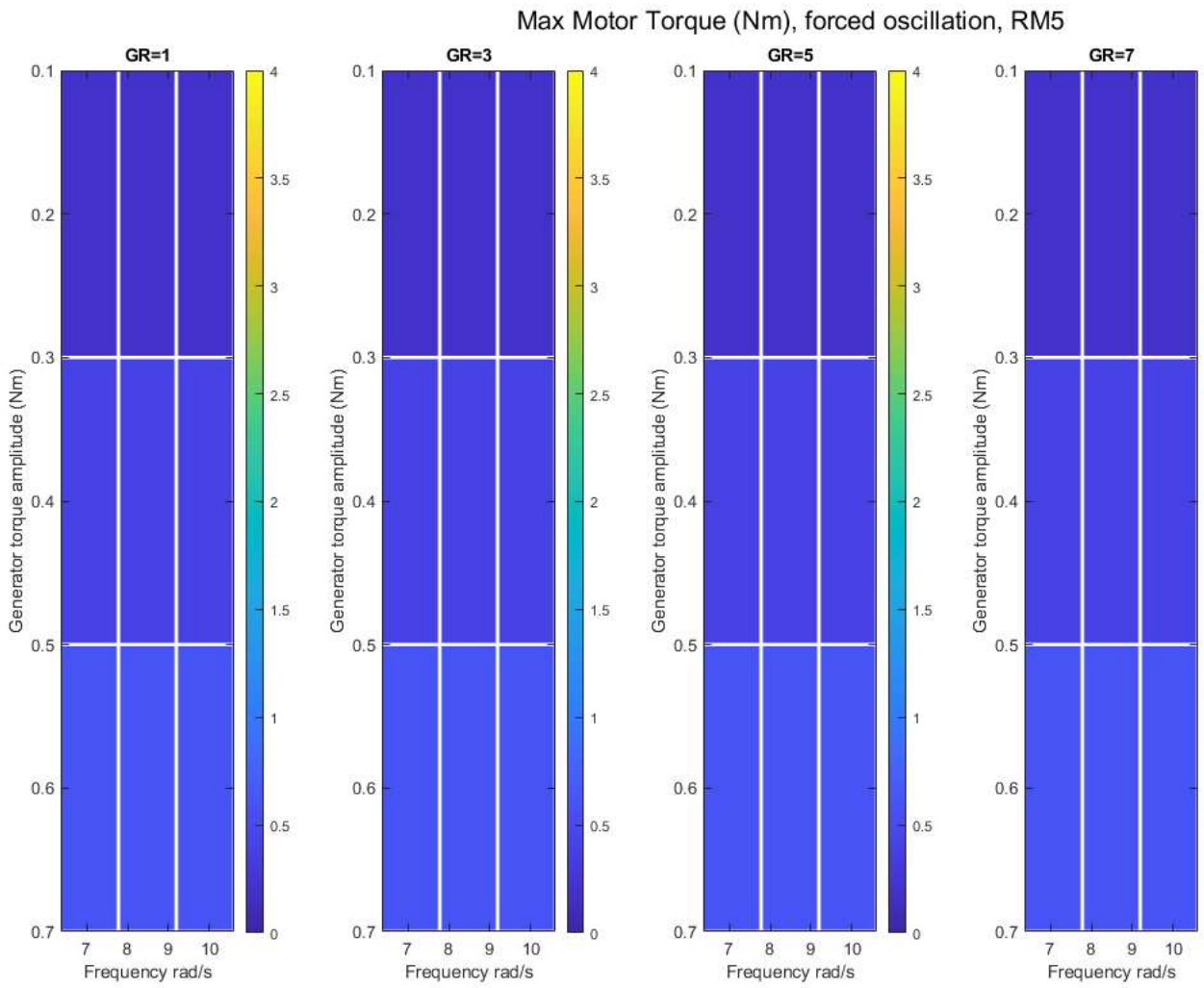

0.14
0.5
0
7
8
9
10
Frequency rad/s

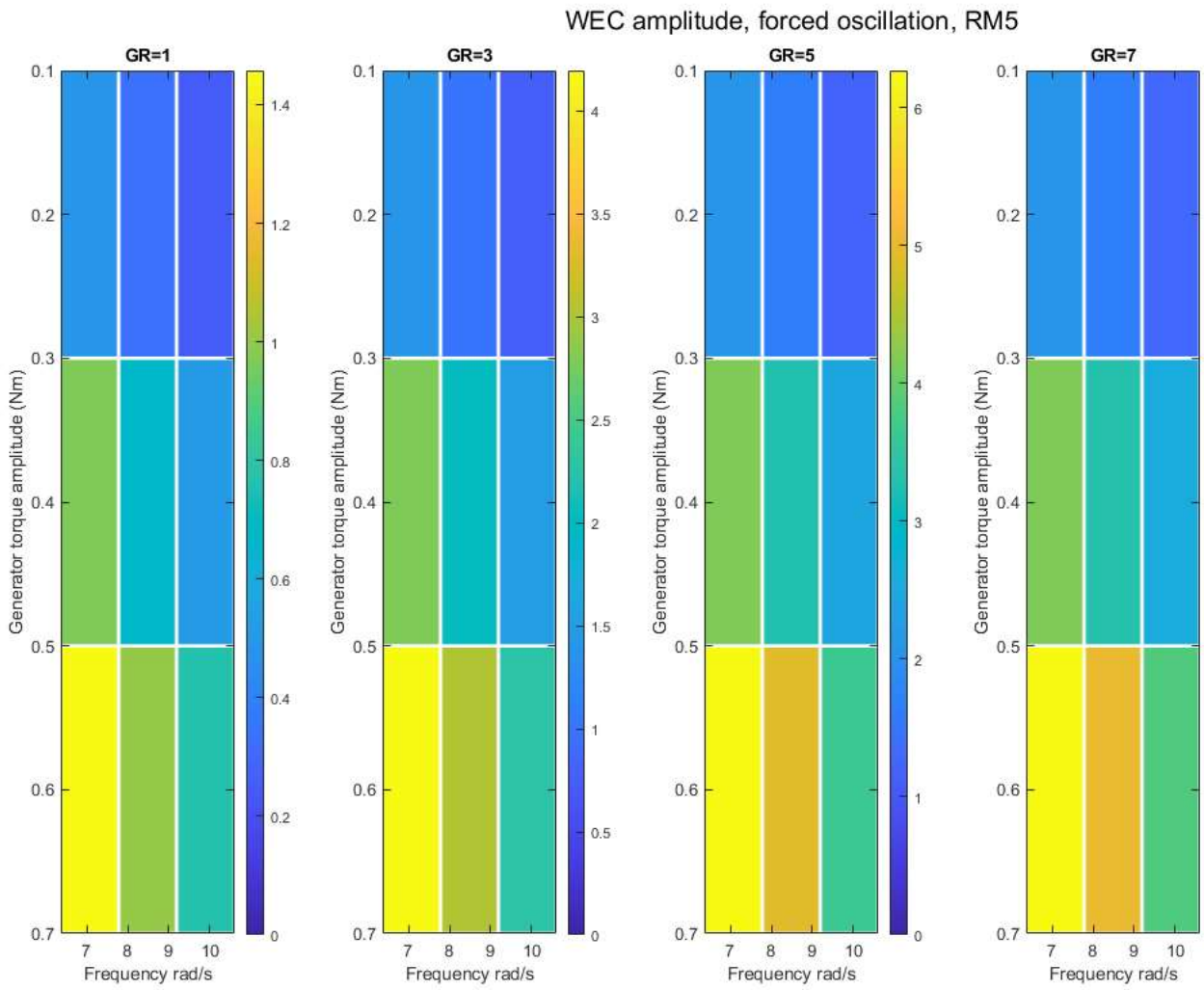

0.14
0.5
0
7
8
9
10
Frequency rad/s

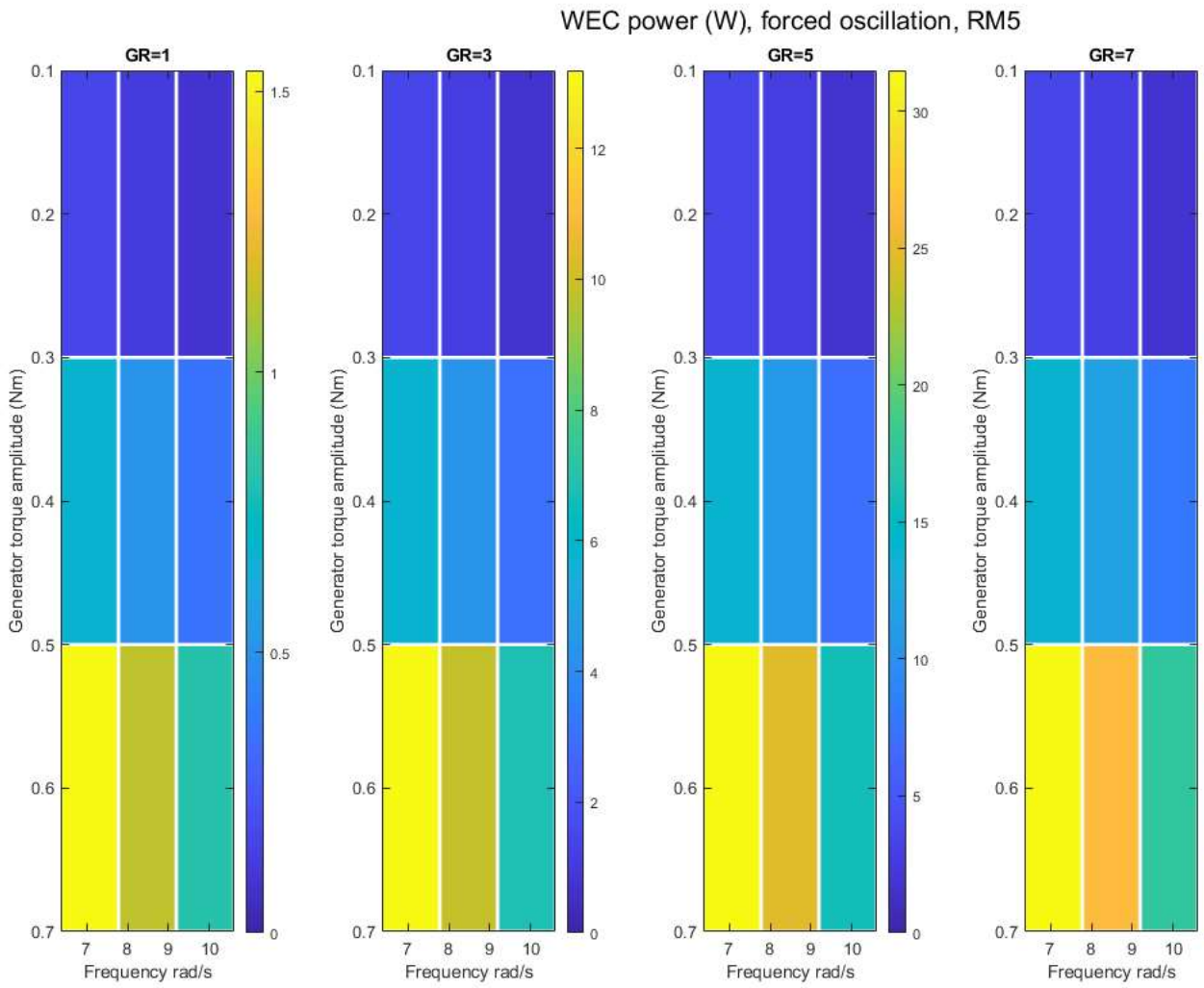

0.14
0.5
0
7
8
9
10
Frequency rad/s

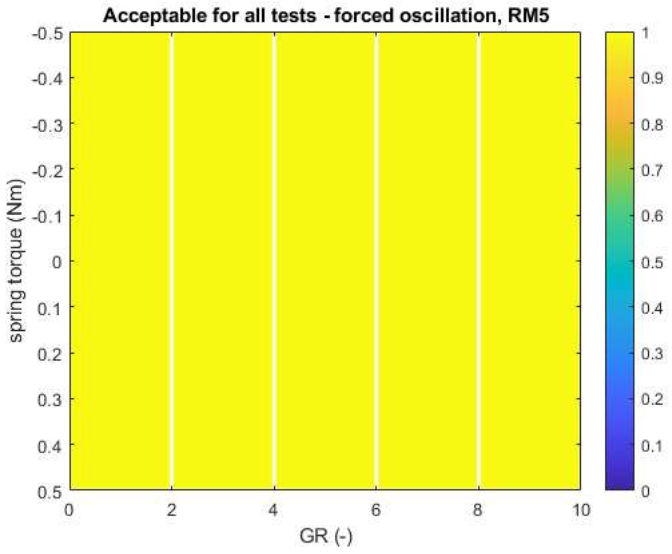

0.14
7
8
9
10
Frequency rad/s

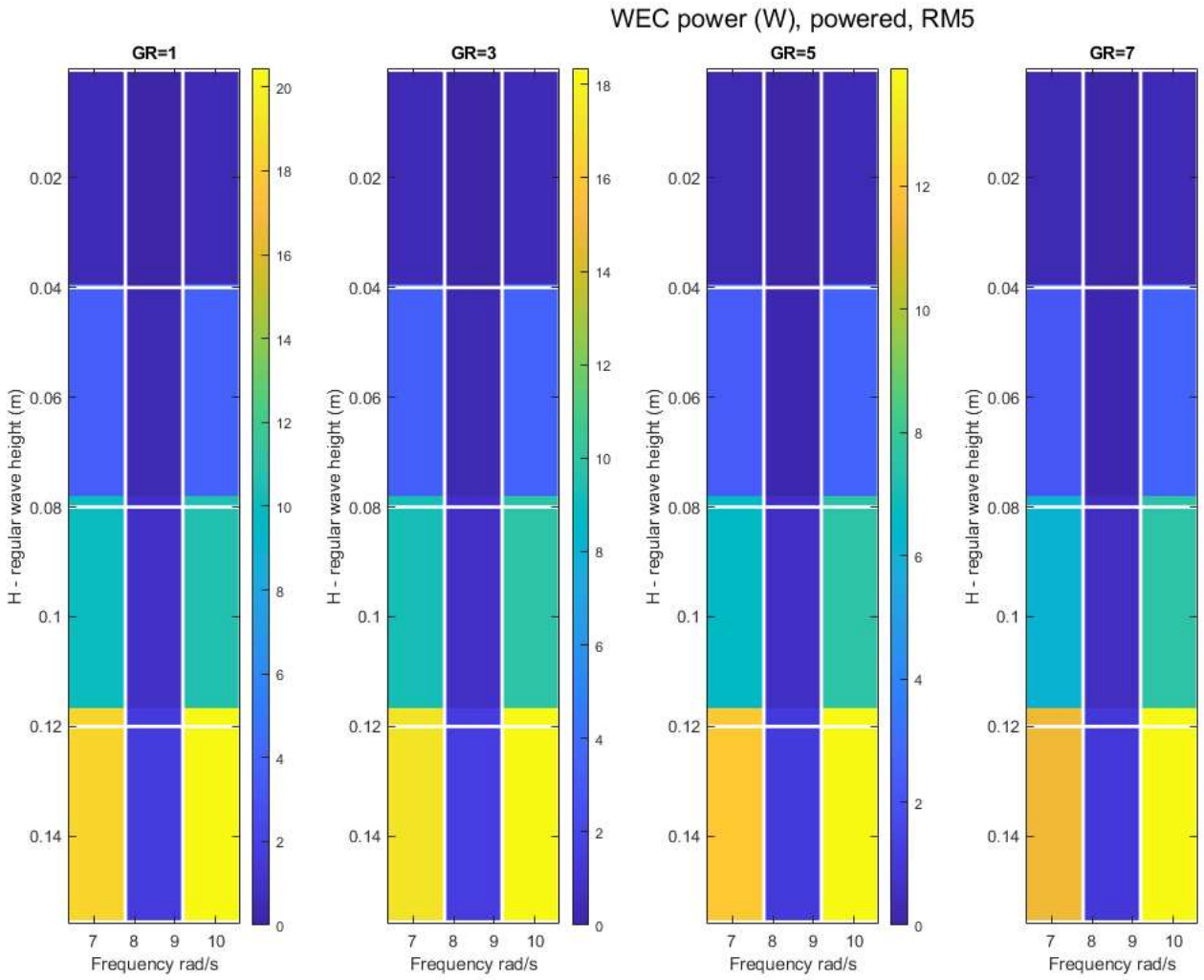

Acceptable for all tests - powered, RM5
spring torque (Nm)
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
0
2
4
6
8
10
GR (-)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

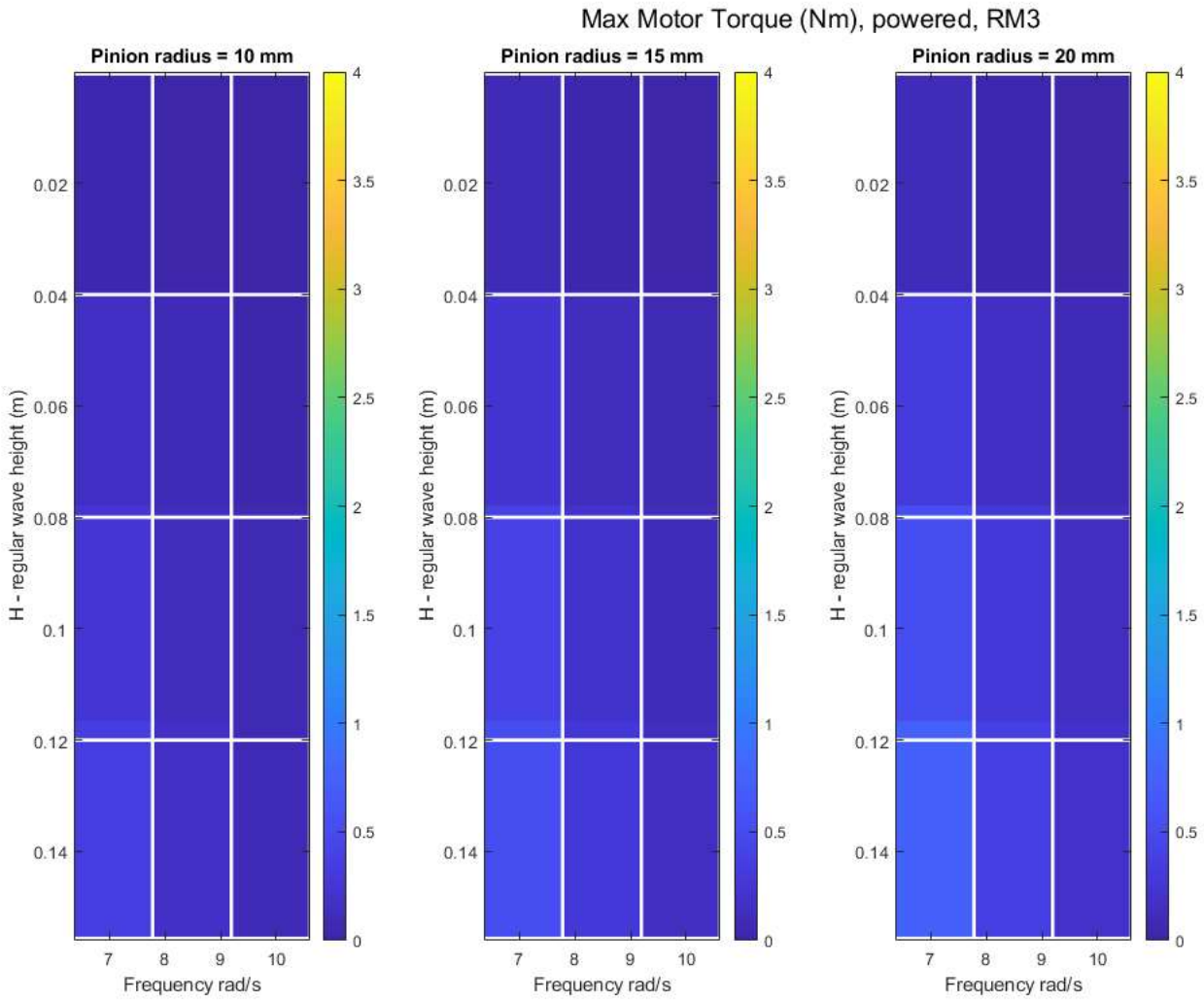

Max Motor Torque (Nm), forced oscillation, RM5
GR=1
GR=3
GR=5
GR=7
Generator torque amplitude (Nm)
Frequency rad/s
0.1
0.2
0.3
0.4
0.5
0.6
0.7
7
8
9
10
0
0.5
1
1.5
2
2.5
3
3.5
4

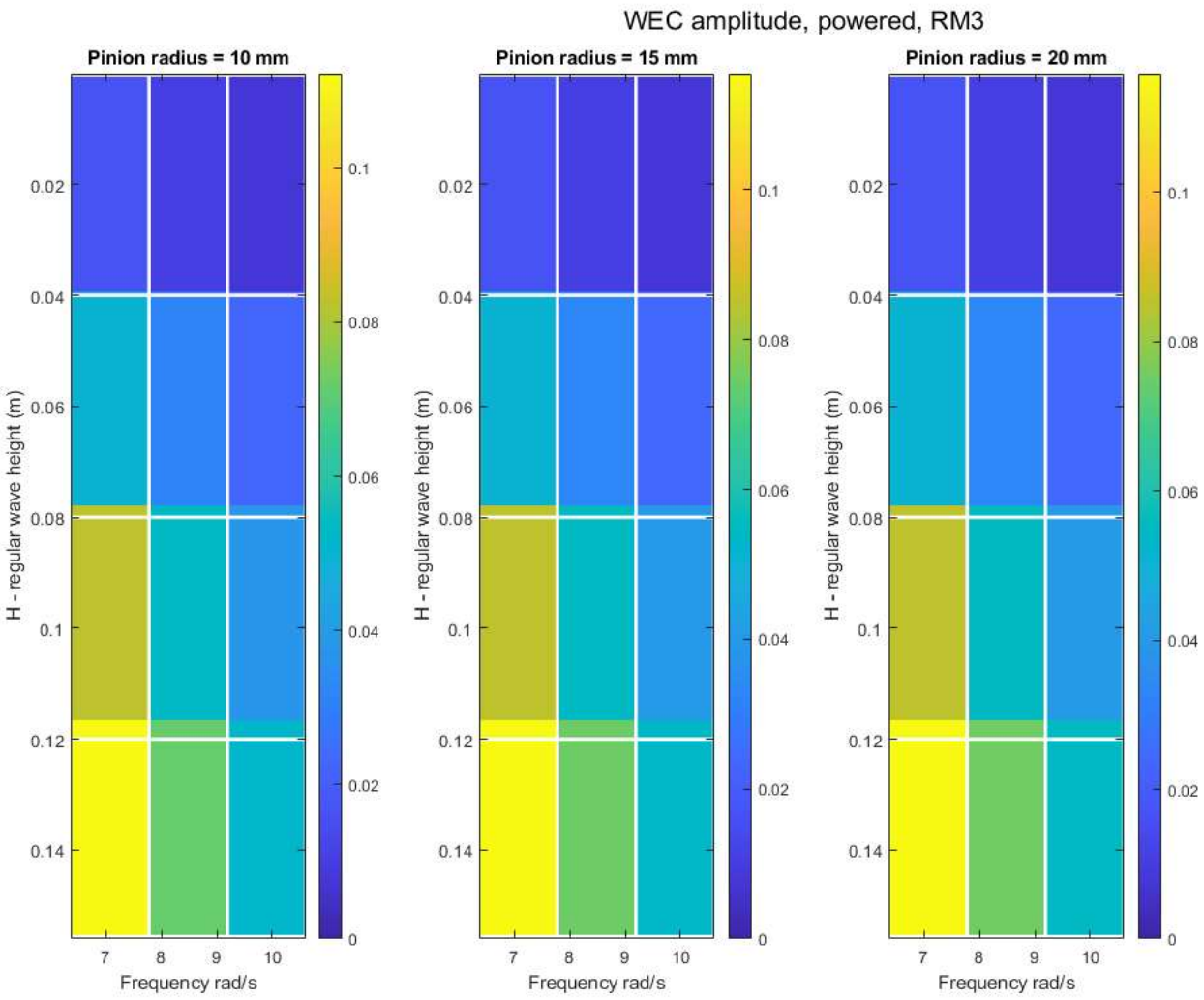

WEC amplitude, forced oscillation, RM5
GR=1
GR=3
GR=5
GR=7
Generator torque amplitude (Nm)
Frequency rad/s
0.1
0.2
0.3
0.4
0.5
0.6
0.7
7
8
9
10
1.4
1.2
1
0.8
0.6
0.4
0.2
0
4
3.5
3
2.5
2
1.5
1
0.5
6
5
4
3
2
1

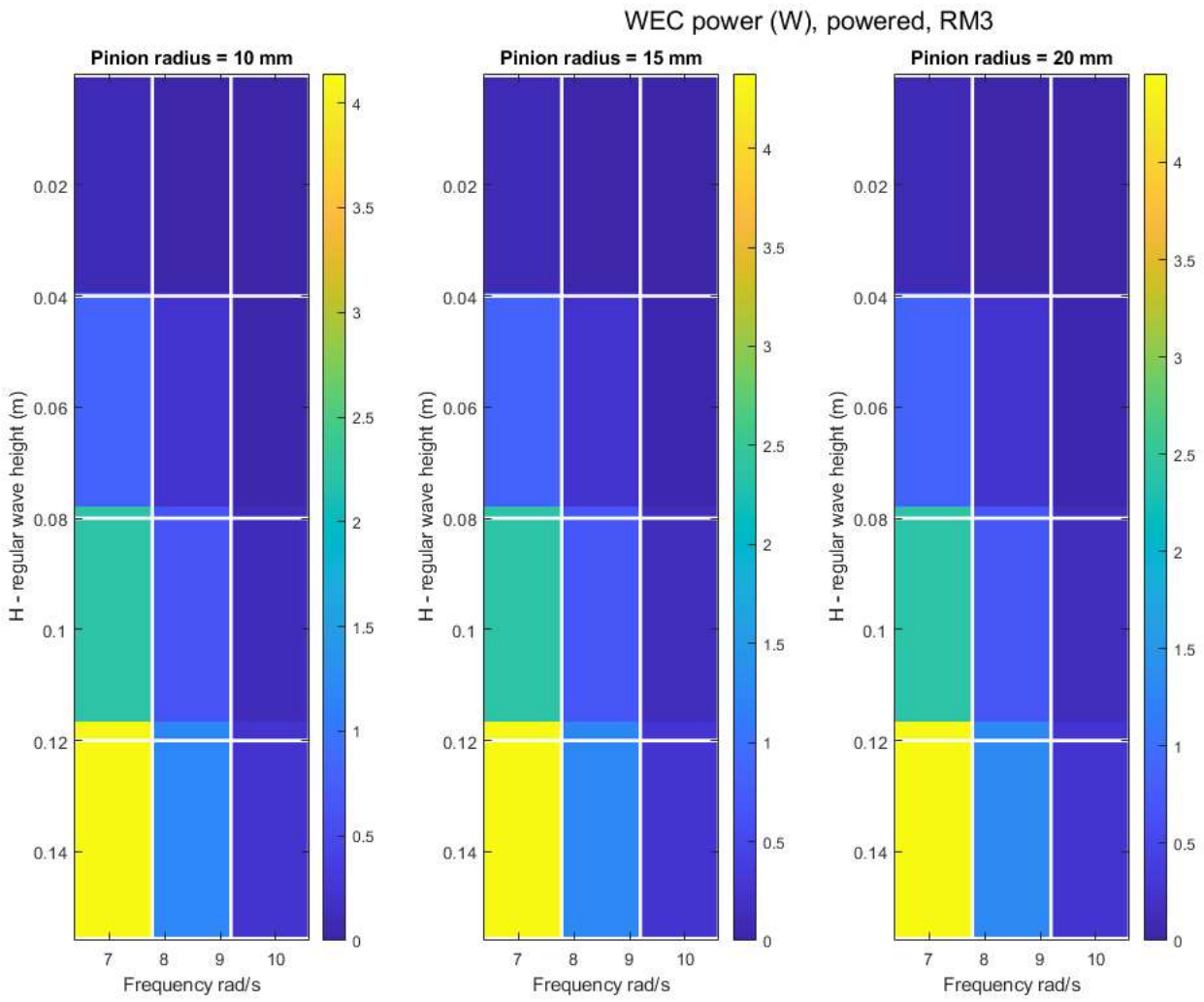

WEC power (W), forced oscillation, RM5
GR=1
GR=3
GR=5
GR=7
Generator torque amplitude (Nm)
Frequency rad/s

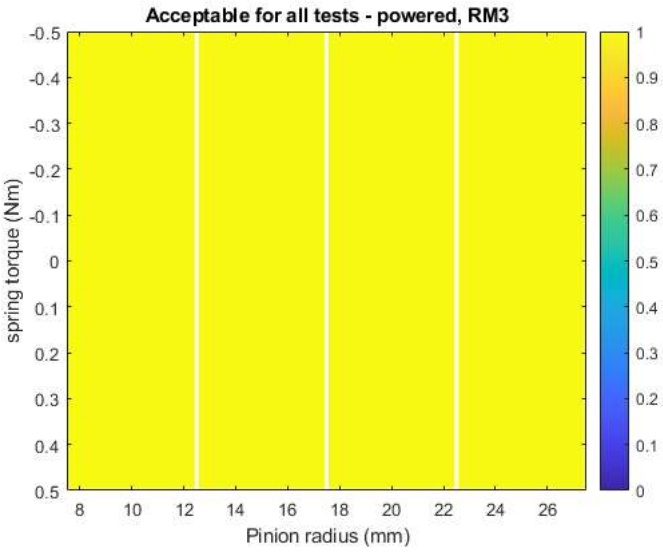

Acceptable for all tests - forced oscillation, RM5
spring torque (Nm)
GR (-)

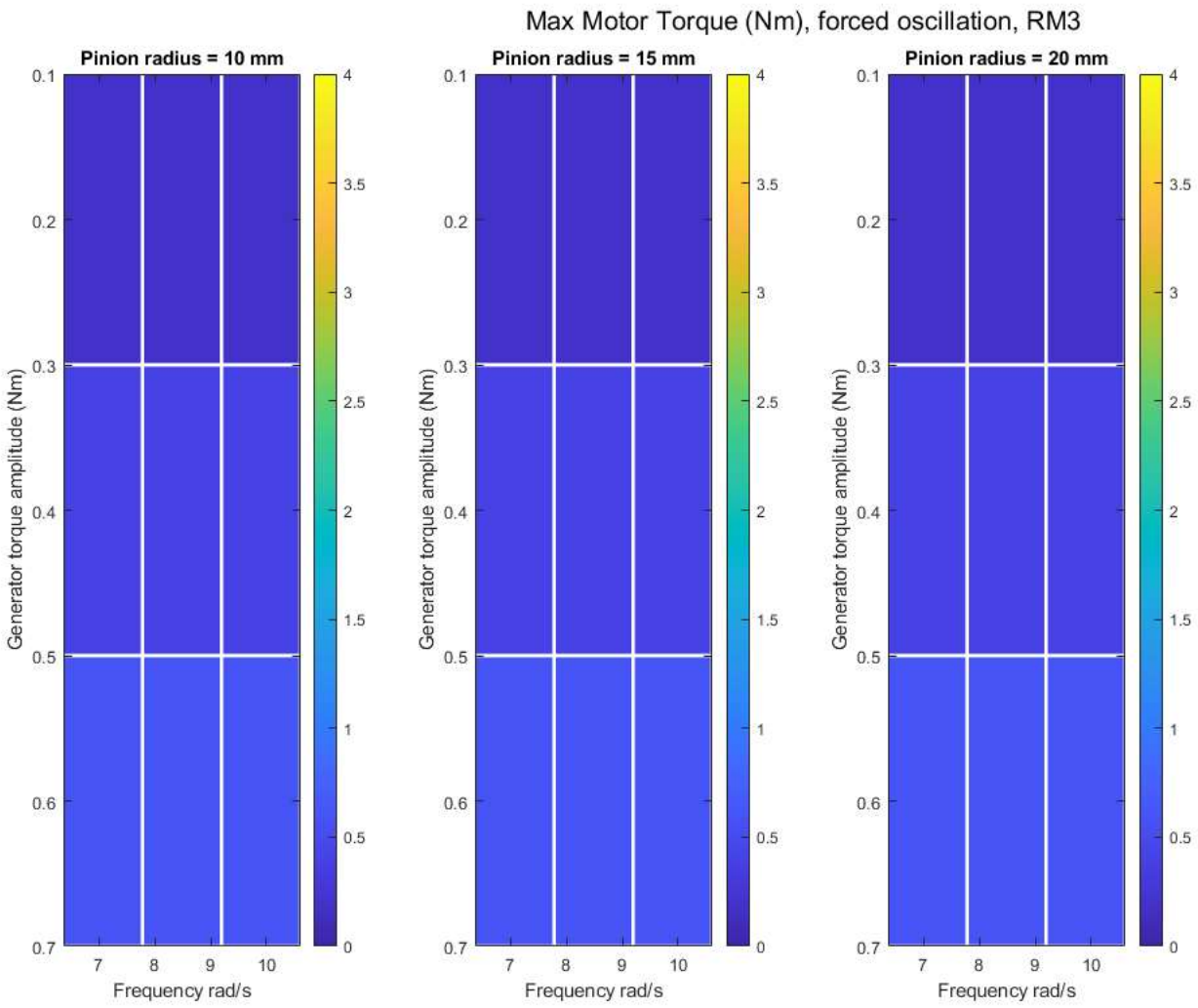

WEC power (W), powered, RM5
GR=1
GR=3
GR=5
GR=7
H - regular wave height (m)
Frequency rad/s
0.02
0.04
0.06
0.08
0.1
0.12
0.14
7
8
9
10

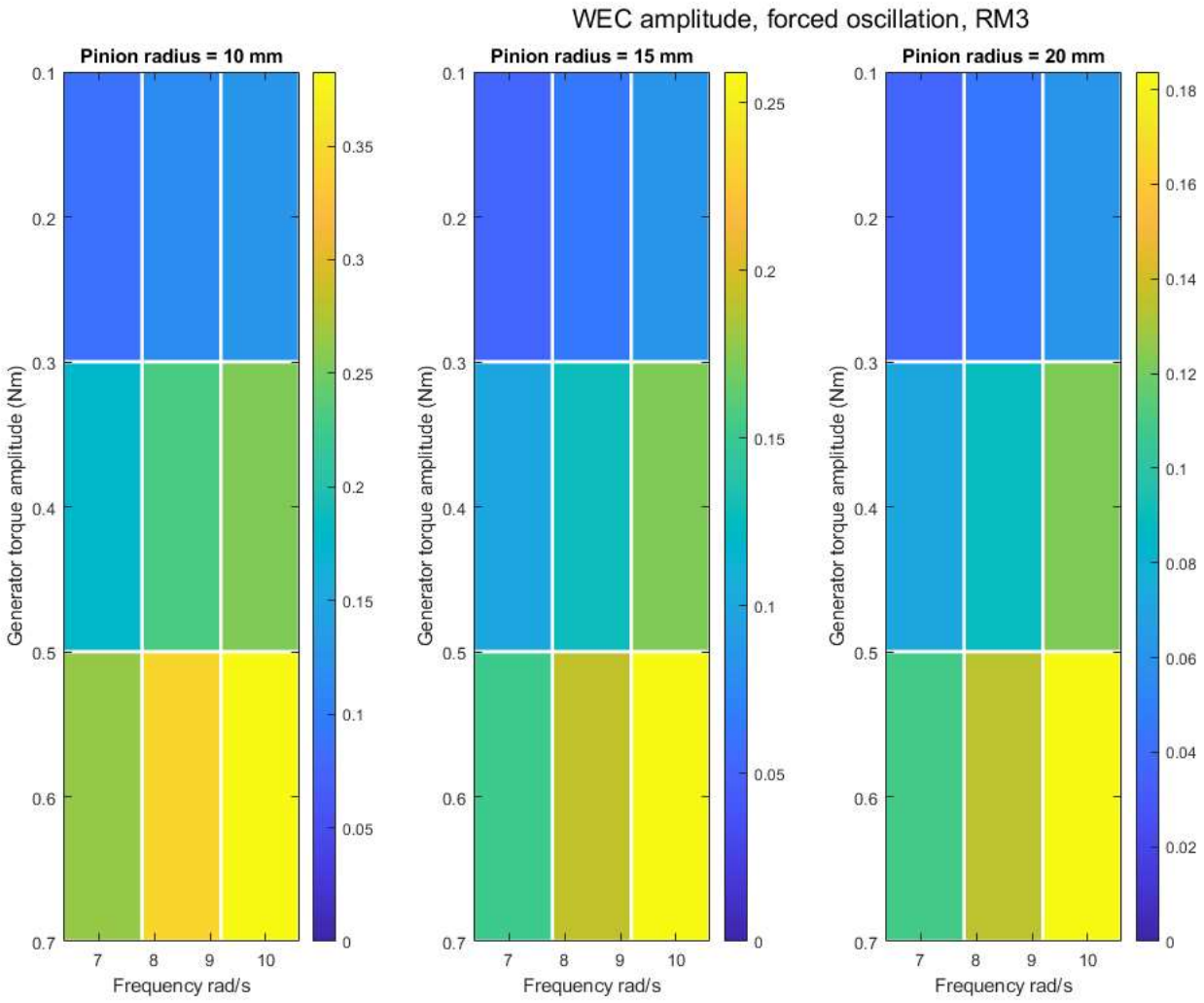

Max Motor Torque (Nm), powered, RM3
Pinion radius = 10 mm
Pinion radius = 15 mm
Pinion radius = 20 mm
H - regular wave height (m)
Frequency rad/s
0.02
0.04
0.06
0.08
0.1
0.12
0.14
7
8
9
10
0
0.5
1
1.5
2
2.5
3
3.5
4

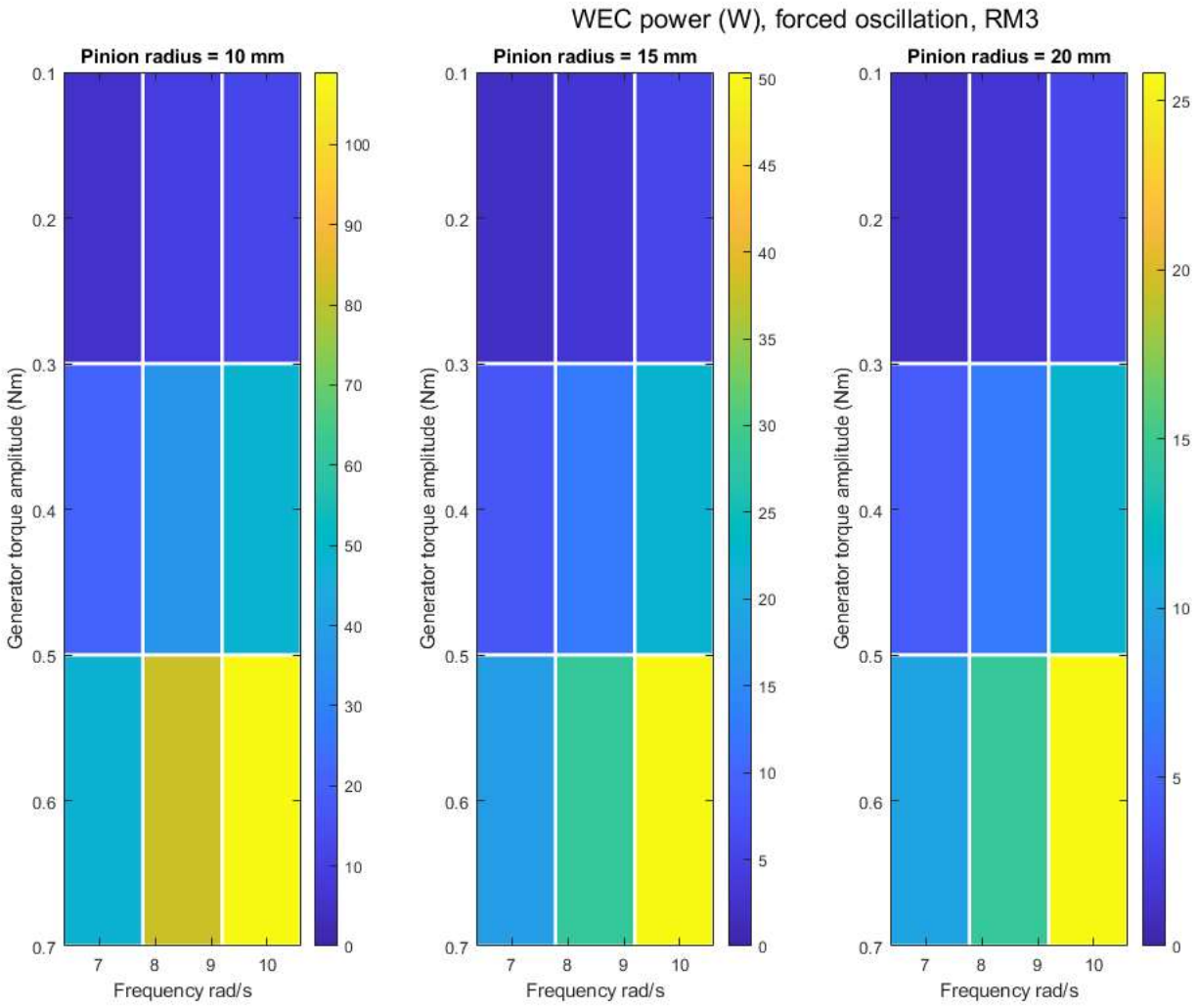

WEC amplitude, powered, RM3
Pinion radius = 10 mm
Pinion radius = 15 mm
Pinion radius = 20 mm
H - regular wave height (m)
Frequency rad/s
0.02
0.04
0.06
0.08
0.1
0.12
0.14
7
8
9
10
0
0.02
0.04
0.06
0.08
0.1

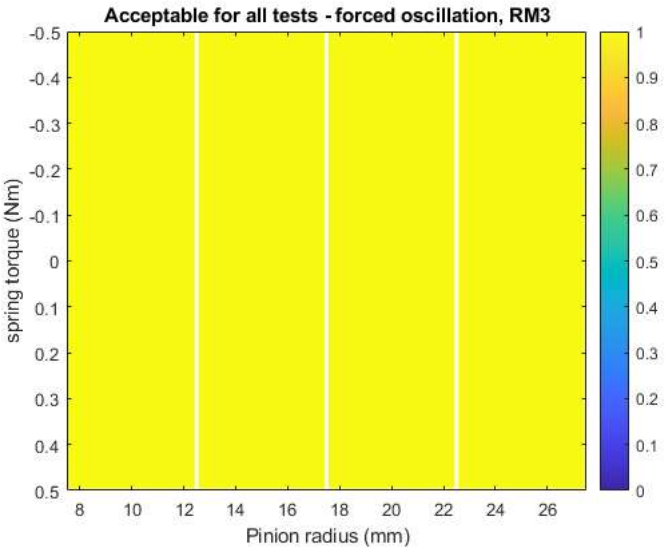

WEC power (W), powered, RM3
Pinion radius = 10 mm
Pinion radius = 15 mm
Pinion radius = 20 mm
H - regular wave height (m)
Frequency rad/s
0.02
0.04
0.06
0.08
0.1
0.12
0.14
7
8
9
10
0
0.5
1
1.5
2
2.5
3
3.5
4


Acceptable for all tests - powered, RM3
spring torque (Nm)
Pinion radius (mm)
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
8
10
12
14
16
18
20
22
24
26
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Max Motor Torque (Nm), forced oscillation, RM3
Pinion radius = 10 mm
Pinion radius = 15 mm
Pinion radius = 20 mm
Generator torque amplitude (Nm)
Frequency rad/s
0.1
0.2
0.3
0.4
0.5
0.6
0.7
7
8
9
10
0
0.5
1
1.5
2
2.5
3
3.5
4


WEC amplitude, forced oscillation, RM3
Pinion radius = 10 mm
Pinion radius = 15 mm
Pinion radius = 20 mm
Generator torque amplitude (Nm)
Frequency rad/s
0.1
0.2
0.3
0.4
0.5
0.6
0.7
7
8
9
10
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.25
0.2
0.15
0.1
0.05
0
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0





Published with MATLAB® R2022a