AUTORESYS Report

WIND TURBINE MODEL AND SIMULATION

Authors:

Lawal Ibrahim Okikiola

Supervisor:

Professor Horst Schulte

February 7th , 2025

DREAM European Master

Contents

List of Figures

List of Tables

Abstract

This project presents an integrated model of a wind turbine, encompassing its aerodynamic components, drivetrain, blade and tower dynamics, and control mechanisms for its three operational regions. The primary objective is to develop a comprehensive understanding of the interplay between different turbine subsystems and their collective impact on overall performance and efficiency. In the aerodynamic analysis, the focus is on the interactions between the wind and turbine blades, employing advanced computational methods to predict forces and moments accurately. The drivetrain section explores the transmission of mechanical energy from the rotor to the generator, highlighting the importance of gear and bearing dynamics. The blade and tower dynamics are investigated to understand the structural responses to wind loads and rotational forces, critical for ensuring structural integrity and longevity. The control aspect of the turbine is meticulously addressed, with particular emphasis on the three distinct operational regions: below-rated wind speed, near-rated wind speed, and above-rated wind speed. Each region poses unique control challenges, ranging from maximizing energy capture to protecting the turbine from excessive loads. By integrating these diverse aspects into a unified model, the project aims to provide a holistic view of wind turbine operation, facilitating the design and development of more efficient, reliable, and sustainable wind energy systems. This comprehensive approach not only enhances the understanding of individual components but also illuminates the complex interactions within the turbine system, paving the way for innovative solutions in wind energy technology.

1 Mechanical structure dynamics: Drive train

The drivetrain of a wind turbine refers to the system of components responsible for transferring mechanical power from the rotating blades to the generator, ultimately converting wind energy into electrical energy. Figure 1 shows main components of a typical drivetrain.

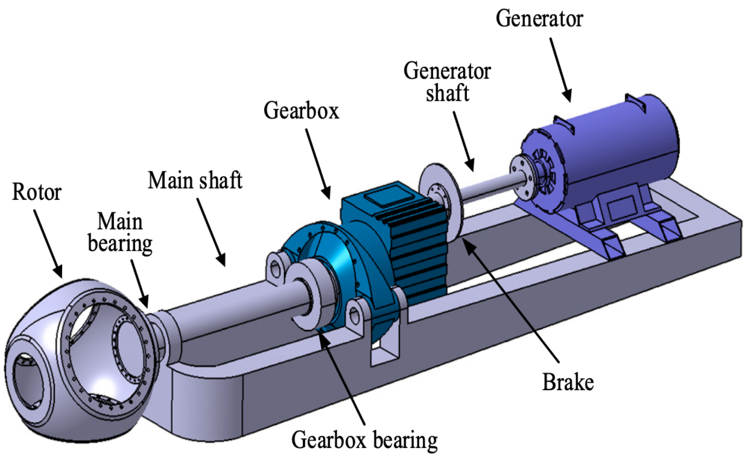

Figure 1: A typical drivetrain and its component [1]

The primary objective of the drivetrain is to optimize power transmission efficiency while accommodating varying wind conditions. There are different types of wind turbine drivetrains, with the most common being the gearbox-driven system and the direct-drive system. In a gearbox-driven system, the rotational speed of the blades is increased by a gearbox before reaching the generator, while a direct-drive system directly connects the low-speed rotor to the generator, eliminating the need for a gearbox. Each drivetrain type has its advantages and challenges, influencing the design and performance of wind turbines in terms of reliability, maintenance, and overall efficiency. In this project we will consider a gearbox-driven model.

In the depicted schematic, denoted as Figure 2, our drive train configuration is illustrated. The input to the model for a two-mass system is defined as torque T_r which is obtained by the aerodynamic system and the generator reaction torque T_g. The output is the changes in the rotor speed ω_r and generator speed ω_g.

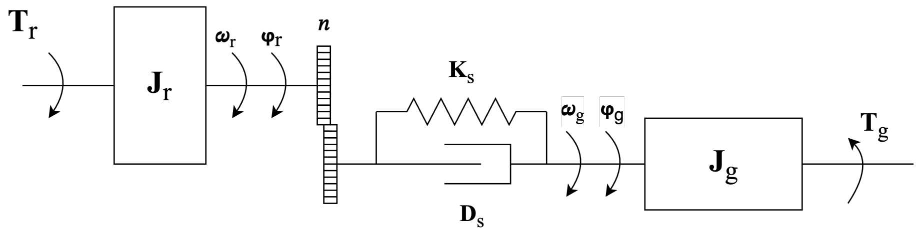

Figure 2: Drivetrain schematic

The rotational spring torque is given in equation 1 where K_s is stiffness constant .

$$T_f = -K_s(\varphi_1 - \varphi_2) \tag{1}$$

The rotational damping torque is given in equation 2 where D_s is damping constant.

$$T_d = -D_s(\dot{\varphi}_1 - \dot{\varphi}_2) \tag{2}$$

The Dynamics of the shaft can be described by [2]:

$$\dot{\varphi} = \omega_r \eta - \omega_g = \dot{\varphi}_r \eta - \dot{\varphi}_g \tag{3}$$

Also implemented in simulation as :

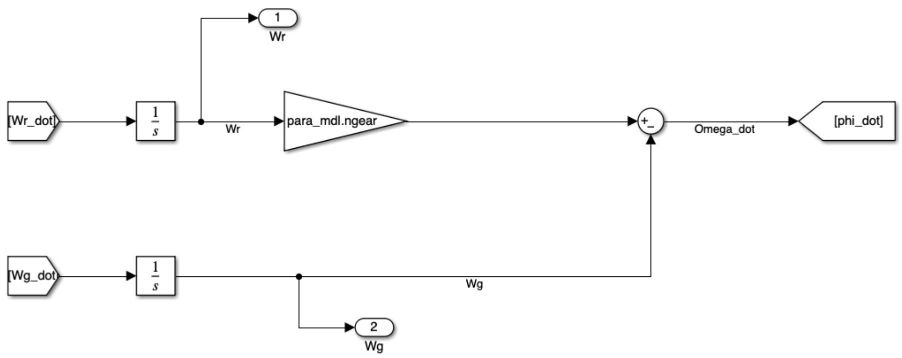

Figure 3: Drivetrain : Shaft

Final drivetrain dynamics is as follow [2]:

$$\dot{\omega}_r = \frac{1}{J_r}(T_r - (K_s\varphi - D_s\dot{\varphi})\eta) \tag{4}$$

Figure 4 indicates implementation of equation 4 in simulation:

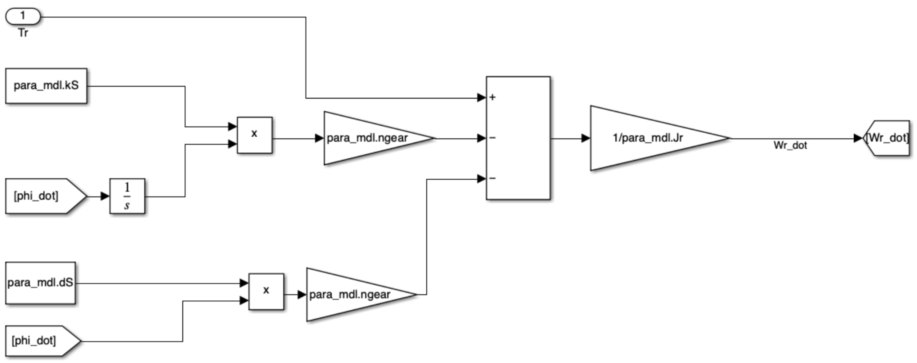

Figure 4: Drivetrain : Angular velocity $\dot{\omega}_r$

$$\dot{\omega}_g = \frac{1}{J_g}(-T_e - K_s\varphi - D_s\dot{\varphi}) \tag{5}$$

Figure 5 indicates implementation of equation 5 in simulation:

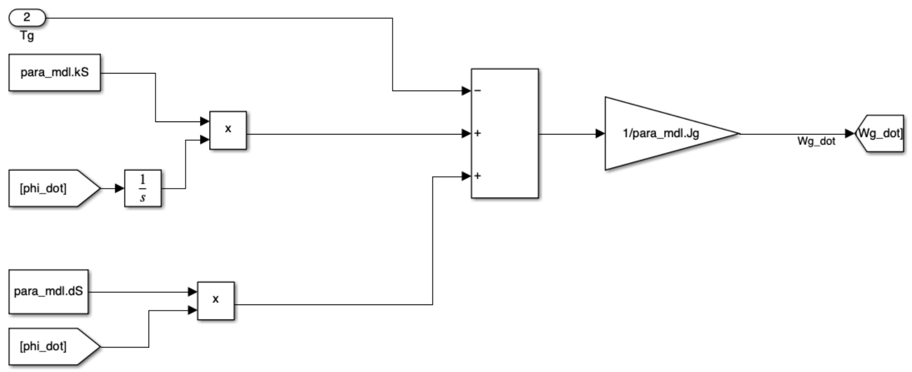

Figure 5: Drivetrain : Angular velocity $\dot{\omega}_g$

Figure 6 indicates rotor and generator speeds (W_r and W_g) at 10 m/s wind speed. We can observe gearbox transformation clearly.

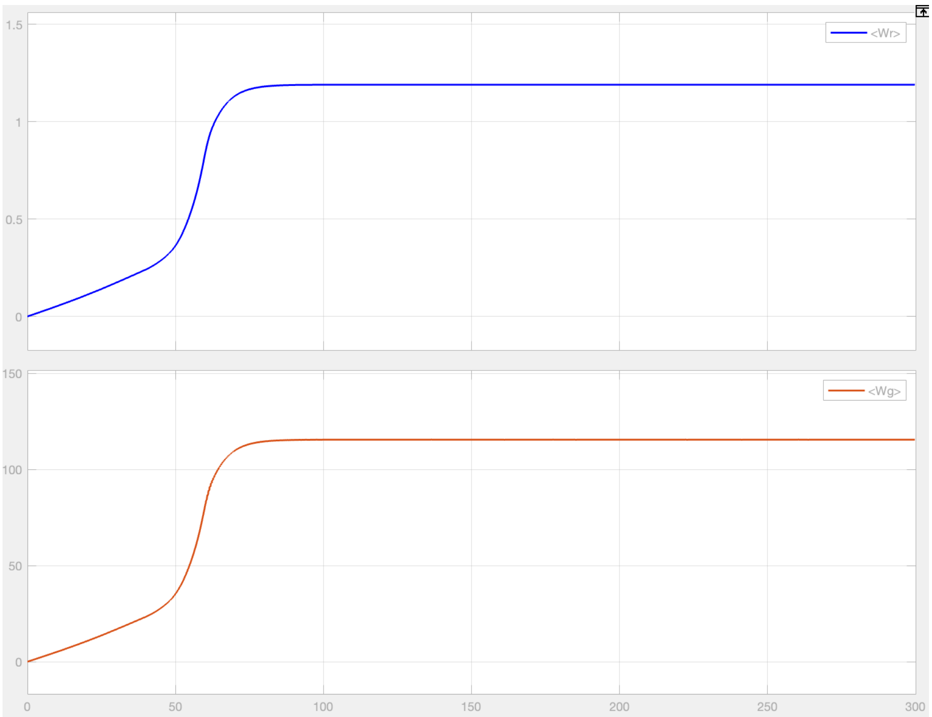

Figure 6: Drivetrain : W_r and W_g at 10 m/s wind speed

2 Aerodynamic Model

This part of the model involves the in-depth analysis of rotor torque and thrust force in wind turbines, employing the coefficients of torque (C_Q) and thrust (C_T) as functions of the tip speed ratio (λ) and blade pitch angle (β). The model aims to establish a robust computational model that accurately predicts the performance of wind turbine rotors under varying operational conditions, thereby enhancing the efficiency and reliability of wind energy generation.

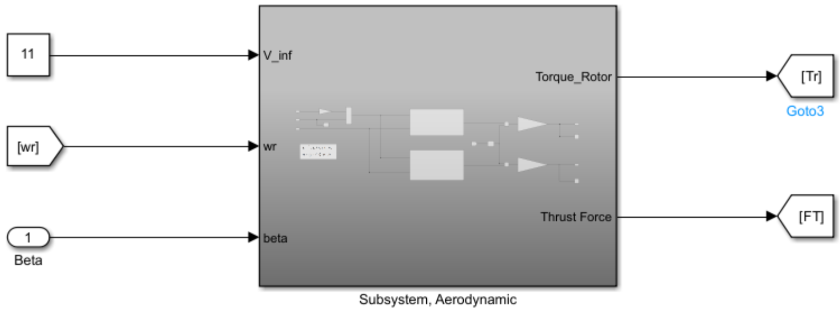

Figure 7: Aerodynamic subsystem on simulink

2.1 CQ and CT curves for Torque and Thrust-force Calculation

The C_x-λ curves are two parametric functions of the tip speed ratio (λ) and blade pitch angle (β) [ref.] which are used in the calculation of rotor torque T_r and thrust force F_T as seen in the equations below [3]

$$T_r = \frac{1}{2}\rho R^3 V_\infty^2 C_Q(\lambda, \beta) \quad (6)$$

$$F_T = \frac{1}{2}\rho R^2 V_\infty^2 C_T(\lambda, \beta) \quad (7)$$

where

- *Radius* (R): The radius of the wind turbine rotor.
- *Density* (ρ): The air density.
- *Infinity Velocity* (V_∞): The upstream wind velocity.

- *Coefficient of Torque* (C_Q): A dimensionless coefficient related to rotor torque.
- *Coefficient of Torque* (C_T): A dimensionless coefficient related to thrust force.
- *Tip Speed Ratio* (λ): The tip speed ratio of the rotor.
- *Blade Pitch Angle* (β): The angle at which the turbine blades are pitched.

These parameters play crucial roles in our analysis and are defined as above.

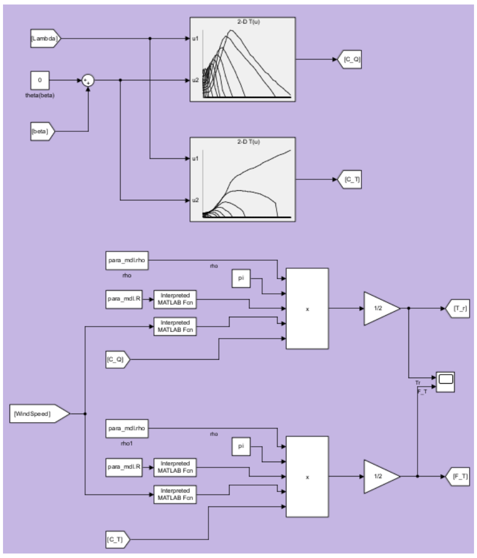

Figure 8: Aerodynamic implemented on simulink

Fig 9 shows the results of rotor torque and thrust force for a simulation of $t = 200\,\text{s}$.

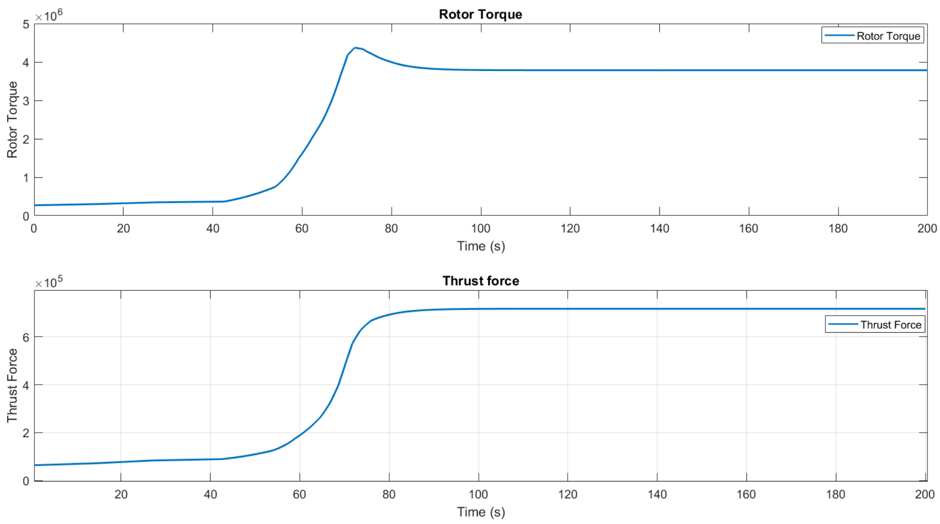

Figure 9: Results of rotor torque and thrust force

3 Mechanical structure dynamics of turbine tower and blades

3.1 Equations used in the modelling

Equations of Motion with Dissipation using Lagrangian Formulation:

The dynamic behavior of the wind turbine tower and blades is modeled through a rigorous mathematical framework based on the Lagrangian equation of the second order. This approach provides a systematic way to derive the equations of motion, accounting for both the kinetic and potential energy of the system [3].

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_i}\right) - \frac{\partial L}{\partial q_i} = f_i - \frac{\partial P_D}{\partial q_i}$$

In the context of the Euler-Lagrange equation, the symbols hold specific meanings: f_i represents the generalized coordinates, Q_i denotes the generalized forces, and P_d stands for the dissipation function.

Here, the Lagrangian function L is defined by the difference between kinetic energy E_k and potential energy E_p:

$$L = E_k - E_p$$

Generalized coordinates and Forces:

$$Q_i = \begin{pmatrix} y_T \\ y_B \\ \theta_r \\ \theta_g \end{pmatrix} f_i = \begin{pmatrix} F_T \\ F_T \\ T_r \\ -T_g \end{pmatrix}$$

Kinetic Energy:

$$E_k = \frac{1}{2} m_T \dot{y}_T^2 + \frac{1}{2} N m_B (\dot{y}_{B,a})^2 + \frac{1}{2} J_r \dot{\theta}_r^2 + \frac{1}{2} J_g \dot{\theta}_g^2$$

Potential Energy:

$$E_p = \frac{1}{2} k_T y_T^2 + \frac{1}{2} N k_B (y_{B,a} - y_T)^2 + \frac{1}{2} k_s (\theta_r n_g - \theta_g)^2$$

Dissipation Function:

$$P_d = \frac{1}{2} d_T \dot{y}_T^2 + \frac{1}{2} N d_B (\dot{y}_{B,a} - \dot{y}_T)^2 + \frac{1}{2} d_s (\dot{\theta}_r n_g - \dot{\theta}_g)^2$$

where, $y_{B,a}$ denotes the blade tip deflection related to absolute coordinates in the inertial system $y_{B,a} = y_B + y_T$ and the $\omega_r = \dot{\theta}_r$ $\omega_g = \dot{\theta}_g$

By applying the Lagrange equation with respect to the coordinates q_i and generalized forces f_i we derive the motion equation for the 4-degree-of-freedom (4-DOF) mechanical model.
To model the mechanical structure of the blades and the tower, a two-degree-of-freedom mechanical model is employed.
After substituting these equations and applying the Lagrangian equation we get the following equations:

$$(m_T + Nm_B)\ddot{y_T} + Nm_B\ddot{y_B} + d_T\dot{y_t} + k_Ty_T = F_T$$

$$Nm_B\ddot{y_T} + Nm_B\ddot{y_B} + Nk_By_B + Nd_B\dot{y_B} = F_T$$

$$J_r\ddot{\theta}_r - d_sn_g\dot{\theta}_g + d_sn_g^2\dot{\theta}_r + k_s^2n_g\theta_r - k_sn_g\theta_g = T_r$$

$$J_g\ddot{\theta}_g - d_sn_g\dot{\theta}_r - k_sn_g\theta_r + k_s\theta_g + d_s\dot{\theta}_g = -T_g$$

To make further calculation and simplify the process of the blades and tower motion equation, we aim to express this dynamic system in state-space form with a given matrix form:

$$M\ddot{q} + D\dot{q} + Kq = Qu_m$$

with the input vector $u_m = (F_t, T_r, T_g)^T$ we get the mass, damping matrices:

$$M = \begin{bmatrix} m_T + Nm_B & Nm_B & 0 & 0 \\ Nm_B & Nm_B & 0 & 0 \\ 0 & 0 & J_r & 0 \\ 0 & 0 & 0 & J_g \end{bmatrix} D = \begin{bmatrix} d_T & 0 & 0 & 0 \\ 0 & Nd_B & 0 & 0 \\ 0 & 0 & d_Sn_g^2 & -d_Sn_g \\ 0 & 0 & -d_Sn_g & d_S \end{bmatrix}$$

And with the stiffness and input distribution matrix

$$K = \begin{bmatrix} k_T & 0 & 0 & 0 \\ 0 & Nk_B & 0 & 0 \\ 0 & 0 & k_sn_g^2 & -k_sn_g \\ 0 & 0 & -k_sn_g & k_s \end{bmatrix} Q = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

3.2 Model in the Simulink

Using the equations we got and the parameters provided by professor, we get the following simulation model.

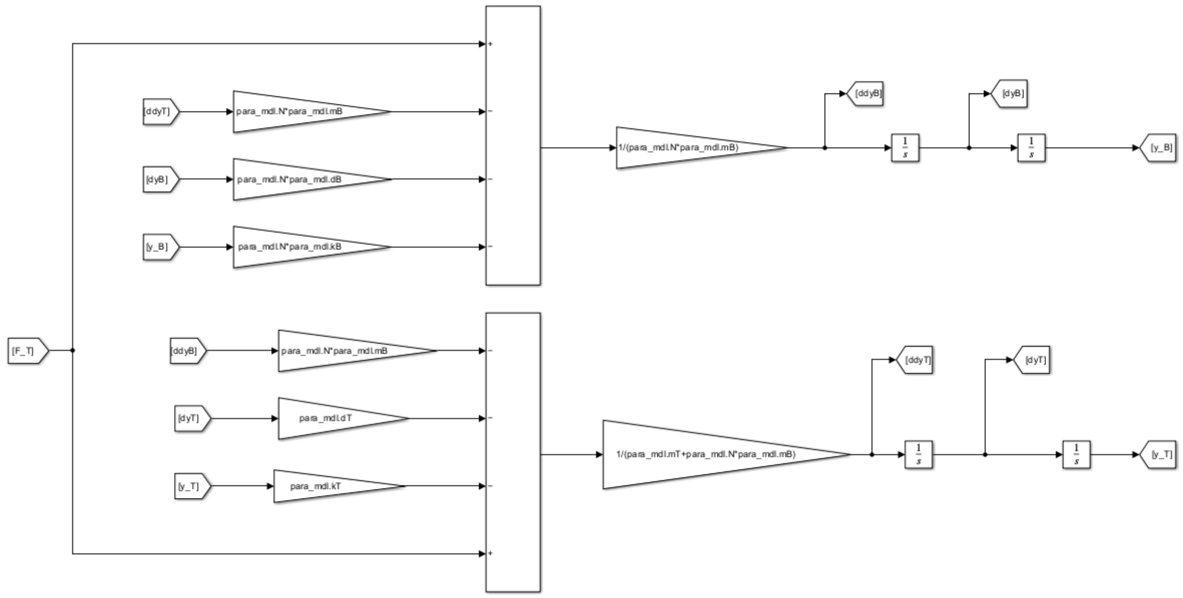

Figure 10: Matlab simulation model

After running the model with a wind speed equal to 11m/s, we get the following scope of motion of the wind turbine tower and the blades.

3.3 Model in the Simulink

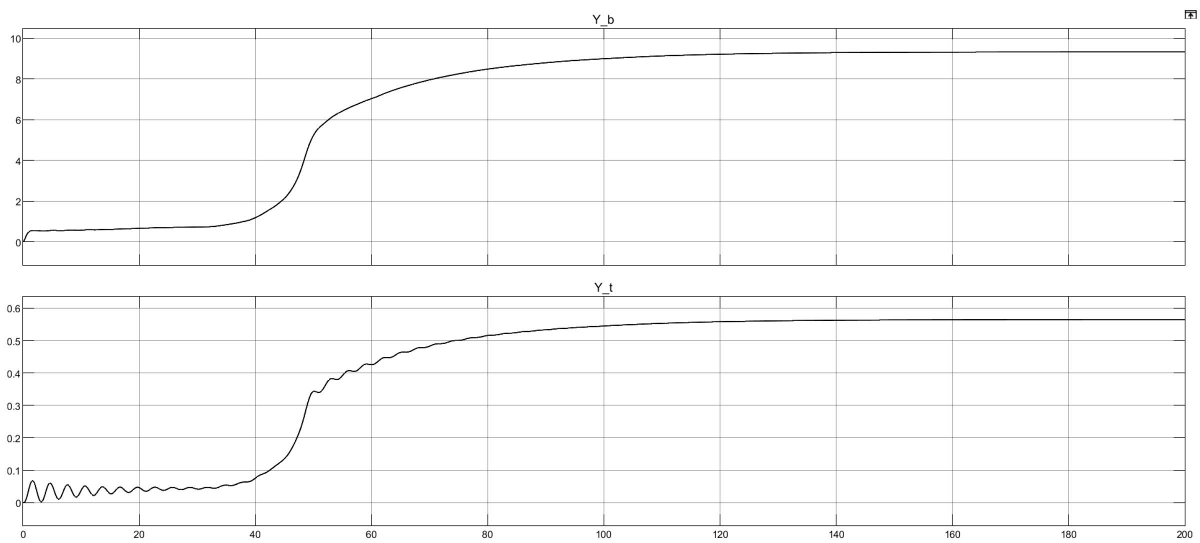

Figure 11: Simulation results

We can see from the scope that the tower is undergoing some vibrations.

4 Control Structure

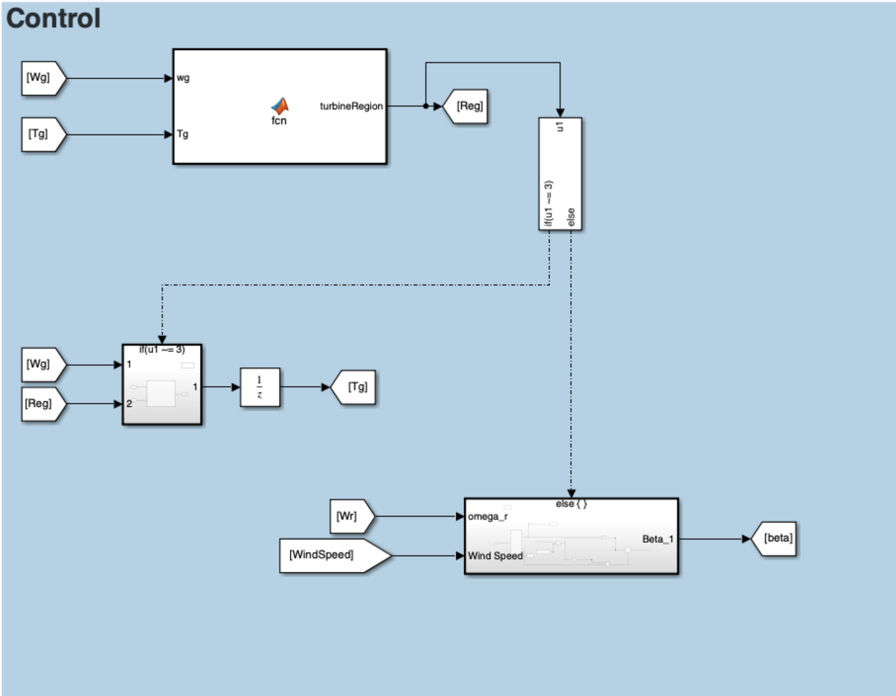

Figure 12: Control structure

The first crucial step involves the creation of a MATLAB function block within Simulink responsible for discerning the current control region. This decision-making process is grounded in the analysis of two pivotal parameters: generator speed (W_g) and generator torque (T_g). Leveraging insights from lecture notes (Fig 13), the function incorporates threshold values specific to each control region. The output of this function, ranging from 1 to 3, serves as a key determinant for subsequent control strategies.

Control Region Thresholds

The determination of thresholds for each control region is paramount, as they dynamically adapt to the varying generator speed and torque conditions. By encapsulating this knowledge in a dedicated function, the system remains responsive to real-time changes, ensuring accurate identification of the current operational state.

Power Optimization vs. Power Limiting

Following the precise identification of the control region, the control system employs an "if" block to bifurcate into distinct branches. This branching is designed to accommodate the two primary operational states: power optimization (region 1-2.5) and power limiting (region 3).

Power Optimizing Area (Region 1-2.5)

In this region, the control system is tailored to optimize power output. Algorithms and controllers are strategically implemented to maximize energy extraction from the wind while maintaining system stability.

Power Limiting Area (Region 3)

As the turbine enters Region 3, the focus shifts towards power limitation to prevent undesirable operating conditions. Specialized controllers are activated to curtail power production, ensuring the turbine operates within safe and sustainable limits.

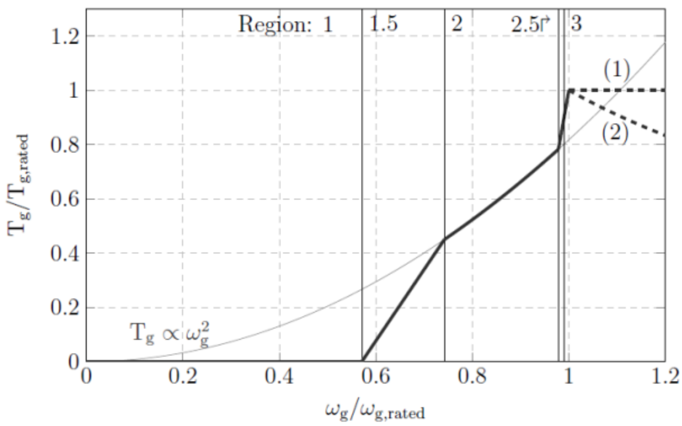

Figure 13: Operation regions related to generator speed and torque[3]

Code implementation

Figure 13 provides us all necessary variables to decide control region, so first we define de regional limits (Fig 14).

```
% Define the thresholds for the conditions
wg_rated = 1.229087000000000e+02;
wg_R1_max = wg_rated * 0.6;
wg_R1_5_max = wg_rated * 0.75;
wg_R2_max = wg_rated * 0.98;
wg_R2_5_max = wg_rated * 1.02;
dwg = 2;
Tg_max = 4.309385747306741e+04;
```

Figure 14: Control variables

and then we design the decision algorithm using a switch-case structure shown in 15.

```
switch prev_region
    case 1 % Region_1
        if wg > wg_R1_max
            turbineRegion = 1.5; % Region_1.5
        else
            turbineRegion = 1; % Region_1
        end
    case 2 % Region_1.5
        if wg > wg_R1_5_max
            turbineRegion = 2; % Region_2
        elseif wg <= wg_R1_max
            turbineRegion = 1; % Region_1
        else
            turbineRegion = 1.5; % Region_1.5
        end
    case 3 % Region_2
        if wg > wg_R2_max && Tg > Tg_max
            turbineRegion = 2.5; % Region_2.5
        elseif wg <= wg_R1_5_max
            turbineRegion = 1.5; % Region_1.5
        else
            turbineRegion = 2; % Region_2
        end
    case 4 % Region_2.5
        if wg > wg_R2_5_max && Tg > Tg_max
            turbineRegion = 3; % Region_3
        elseif wg <= (wg_R2_max - dwg)
            turbineRegion = 2; % Region_2
        else
            turbineRegion = 2.5; % Region_2.5
        end
    case 5 % Region_3
        if wg <= wg_R2_5_max || Tg <= Tg_max
            turbineRegion = 2.5; % Region_2.5
        else
            turbineRegion = 3; % Region_3
        end
end
```

Figure 15: Decision Algorithm

4.1 Control of region 2

In the context of wind turbine control, power optimization in Region 2, where the wind speed is variable and less than the rated speed $(v < v_rated)$, poses unique challenges. In the pursuit of optimizing power in Region 2, a control strategy has been formulated under the assumption that the system operates under steady-state conditions.

$$P_g = P_r = T_r\omega_r$$

with condition $\lambda = \lambda_{opt}$ we get

$$T_g\omega_g = \frac{1}{2}pR^3V^2\omega_r\frac{C_P}{\lambda}\pi$$

using the $\omega_g = n_{gr}$ and extension the equation by multiplying $\frac{\omega_r^2}{\omega_r^2}\ \frac{R^2}{R^2}$

$$T_g = \frac{1}{2}pR^5\omega_g^2\frac{C_{P,max}}{\lambda_opt^3n_g^3}\pi = k_{opt}\omega_g^2$$

So the control low of the region two will be:

$$T_g = k_{opt}\omega_g^2$$

4.2 Control of regions 1, 1.5, 2.5

After trying out the same low for controlling the regions 1, 1.5 and 2.5 as the region 2, we received not pretty stable results, that's why change of the method was applied.

By using the slope intercept form, we got the linear change equations for the regions 1, 1.5 and 2.5 using the curve of the operating regions of the wind turbine and the slope intercept formula.

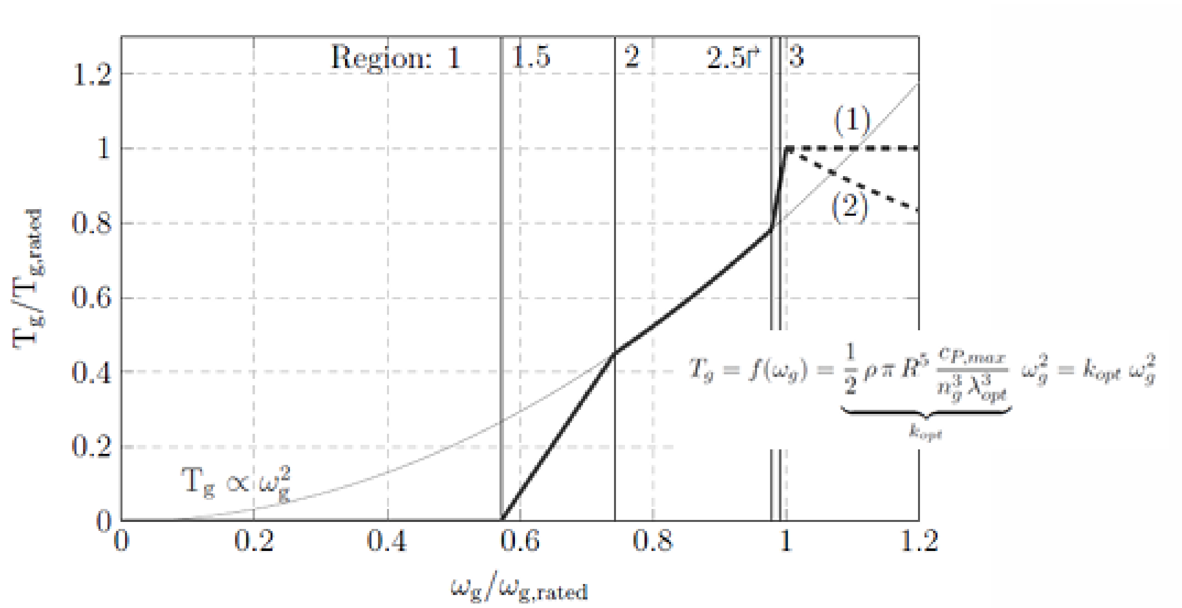

Figure 16: Operation region curve [3]

After dividing each region into the linear slopes and putting the values of the coordinates in the slope intercept form, we get the control low for the regions 1, 1.5, 2.5.

$$y = mx + b$$

Here is an example of the control low obtained for the region 2:

$$0 = \frac{0.45}{0.15}0.6 + b$$

$$b = -1.8$$

$$where y = \frac{T_g}{T_{grat}} and x = \frac{\omega_r}{\omega_{rat}}$$

After substituting the values in the equation, we get the following control low for these regions:

```
function Tg = calculateTg(wg, region)
    % Calculation for Tg based on operating region

    wg_rated = 1.229087000000000e+02;
    Tg_max = 4.309385747306741e+04;
    k_Opt = 2.418610624349642;
    %k_Opt = 0.5 * 1.2250 * pi * 63^5 * (0.4876/(97^3*7.5^3)) * wg^2;

    switch region
        case 1
            Tg = 0; % Region 1
        case 1.5
            %Tg = (3*(wg/wg_rated) - 1.8)*Tg_max; % Region 1.5
            Tg = k_Opt * wg^2;
        case 2
            Tg = k_Opt * wg^2; % Region 2
        case 2.5
            %Tg = (5.5*(wg/wg_rated) - 4.61)*Tg_max; % Region 2.5
            Tg = k_Opt * wg^2;
        otherwise
            Tg = 0; % Default value for other regions
    end
end
```

Figure 17: Matlab function code for the control low of the regions

To check if the control law is working properly, we check putting the first value of the given winds speeds, which is 11,26.

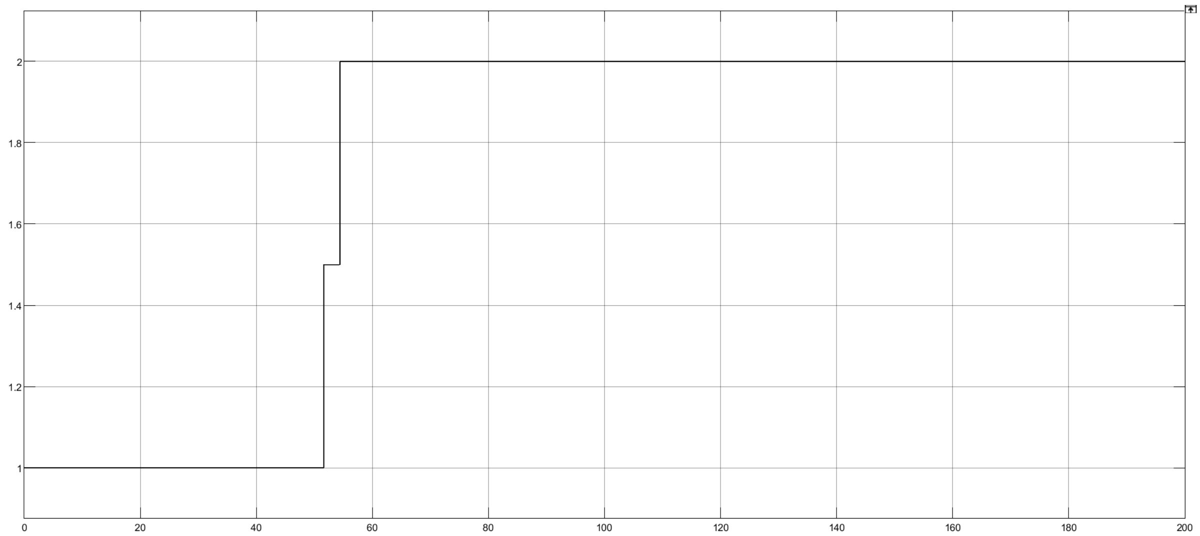

Figure 18: Matlab control region switching function scope

We see, after 30-40 sec, the wind speed is reaching the region 2 because of the inertia and the control of the zone 2 is on.

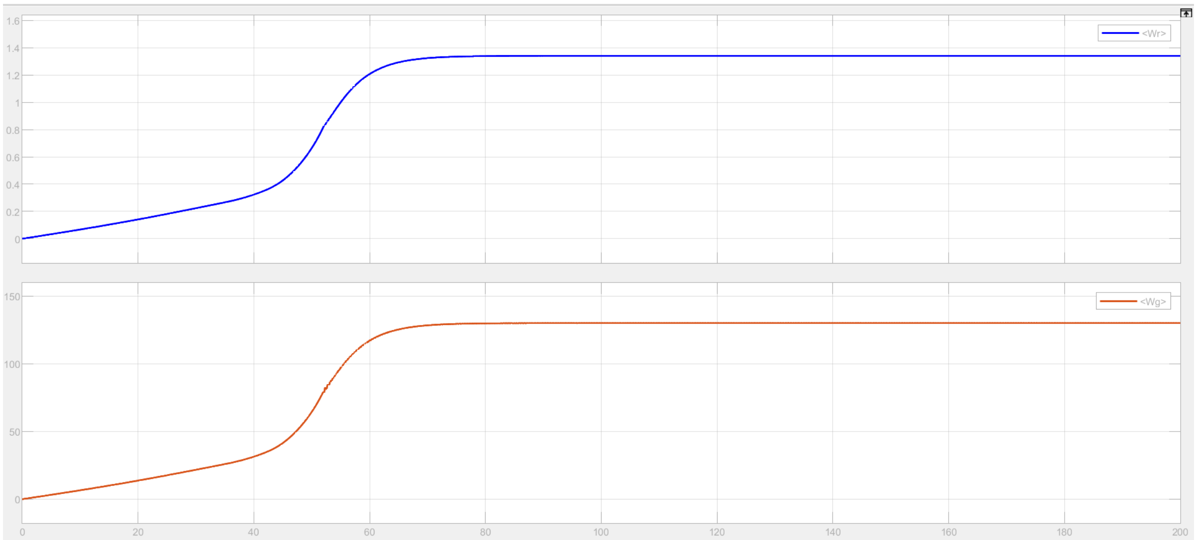

Figure 19: Scope of the rotor and generator speeds

The scope is showing that the rotor and the generator speeds are reaching the optimal value withing 70 seconds and stabilizing after, because of the power optimisation by the controller, which is proving the accuracy of the control low and model.

4.3 Control of region 3

In region 3, we have to hold Wr constant while controlling the pitch angle. Fig 24 shows the control scheme

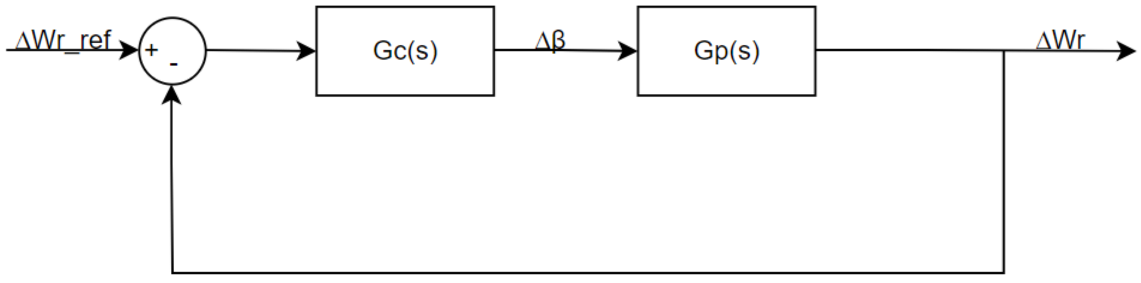

Figure 20: control scheme

Eqn 8 and 9 give the plant and controller transfer functions respectively.

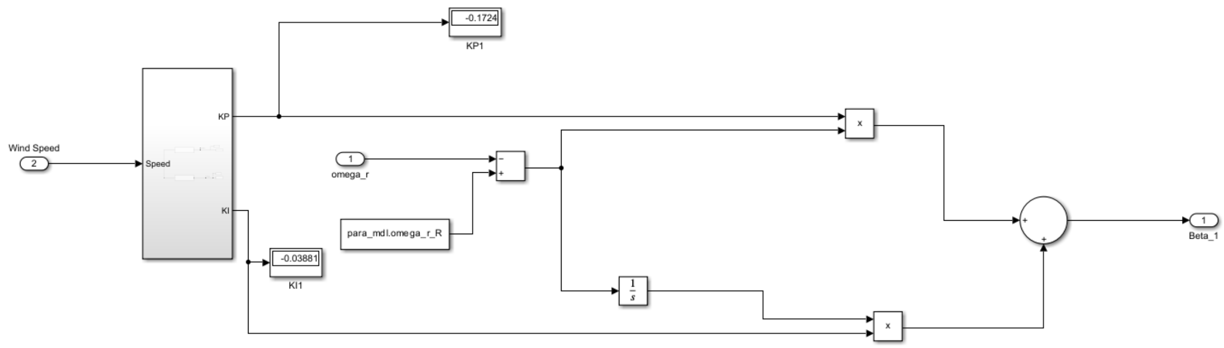

Figure 21: PI control implemented on simulink

$$G_p(s) = \frac{\frac{K\beta}{J}}{s - \frac{K\omega_r}{J}} \tag{8}$$

$$G_c(s) = K_p\left(s + \frac{K_i}{K_p}\right) \tag{9}$$

We use a zero in the controller to cancel the pole in the plant (pole-zero cancellation) by setting $a = \frac{K_p}{K_i}$.

The resulting open loop and closed loop transfer function are given by eqns 10 and 11.

$$G_o(s) = \frac{K_p \cdot b}{s} \tag{10}$$

$$G_{cl}(s) = \frac{K_p \cdot b}{s + K_p \cdot b} \tag{11}$$

Comparing eqn 8 with a standard first-order equation $G(s) = \frac{a}{s+a}$.

$$K_p \cdot b_i = \frac{1}{\tau_{\text{ref}}} \tag{12}$$

$$K_p = \frac{1}{\tau_{\text{ref}} \cdot b} \tag{13}$$

Similarly Ki values are obtained as shown in eqn 15

$$Ki = ai \cdot Kp \tag{14}$$

$$Ki = \frac{ai}{\tau_{\text{ref}} \cdot bi} \tag{15}$$

```
% Create the transfer function model
% G = tf (num,den)
tau_ref = 4;
a = -k_omega/J;
b = k_beta/J;
k_P = 1./(b.*tau_ref);
k_I = a./(b.*tau_ref);
%-----------------------------------------
```

Figure 22: implementation of Kp and Ki on matlab script

After running the matlab code in appendix A, we obtained the following parameters and coefficients of the wind turbine for the different wind scenarios.

Table 1: Calculated coefficients for the turbine model

Wind Speed (m/s)	K_ωr	K_β	K_P	K_I
11.26	-2632796.895	-8601592.48	-1.272571736	-0.076520624
11.5378	-2802502.786	-10259178.72	-1.066960989	-0.068292572
11.8156	-2979797.641	-12380250.6	-0.884161704	-0.060172402
12.0933	-3162925.954	-14559229.45	-0.751835357	-0.054311356
12.3711	-3351027.419	-16716930.6	-0.654793858	-0.050114275
12.6489	-3546148.92	-18857330	-0.580471545	-0.047012871
12.9267	-3744365.735	-20943061.46	-0.522662052	-0.044696972
13.2044	-3948182.006	-22999735.16	-0.475924762	-0.042915516
13.4822	-4156608.065	-25022627.48	-0.437449804	-0.041528493
13.76	-4369510.603	-27013842.83	-0.405204974	-0.040437699
14.26	-4762814.134	-30518483.33	-0.358672591	-0.039015816
15.4533	-5741191.759	-38455805.97	-0.284642155	-0.037323309
16.6467	-6753526.56	-45756090.88	-0.239228117	-0.036899604
17.84	-7788290.044	-52400515.95	-0.208893811	-0.037157506
19.0333	-8828449.525	-58327976.34	-0.187665408	-0.037839687
20.2267	-9856844.167	-63487307.88	-0.172414674	-0.038814231
21.42	-10884845.86	-67904316.63	-0.161199523	-0.040074204
22.6133	-11896219.4	-71548318.77	-0.152989527	-0.041567082
23.8067	-12895449.15	-74447779.17	-0.147031162	-0.043303673
25	-13901608.88	-76660350.05	-0.142787549	-0.045335068

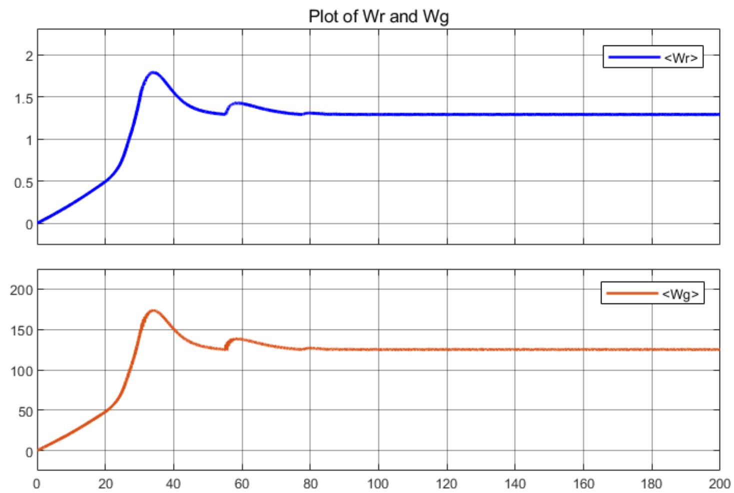

Figure 23: plot of W_r and W_g for wind speed of $20.2267 m/s$

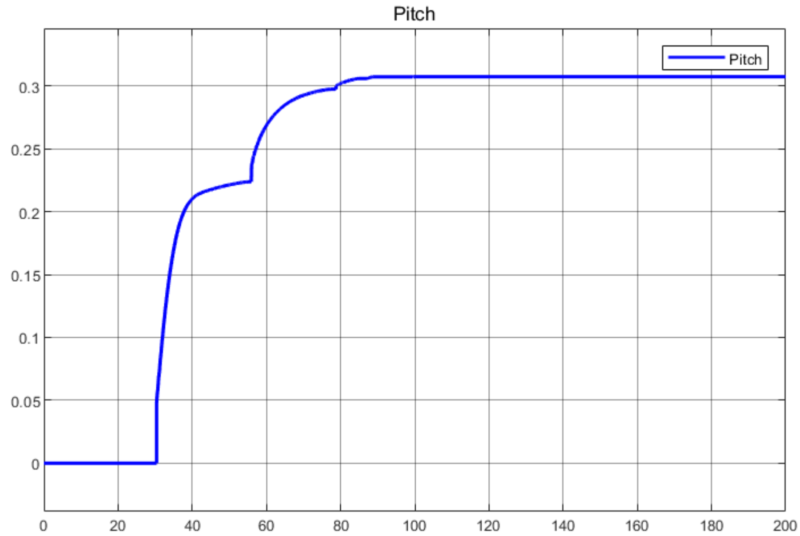

Figure 24: plot of pitch for wind speed of $20.2267 m/s$

It can be seen from fig 23 , that for a wind speed in region 3, the pitch control stabilizes W_r in steady state. As the wind speed exceeds the rated speed, the pitch control system adjusts the blades' angle (pitches them) to reduce their aerodynamic efficiency. This action reduces the rotor's speed and hence the power captured from the wind. The blades are turned away from the wind to reduce the amount of lift generated by the blade's airfoil. The pitch control in Region 3 is a critical aspect of wind turbine design, ensuring optimal operation, safety, and longevity of the turbine, while also meeting the demands of the power grid.

A Appendix A - MATLAB Code

```
%num = (1:1:20)';
% Define the constants and parameters
c1  = 0.005;
c2 = 1.53;
c3 = 0.5;
c4 = 0.18;
c5 = 121;
c6 = 27.9;
c7 = 198;
c8 = 2.36;
c9 = 5.74;
c10 = 11.35;
c11 = 16.1;
c12 = 201;
rho = 1.225; % air density in kg/m^3
R = 63; % rotor radius in m
J = para_mdl.Jr + para_mdl.ngear^2 * para_mdl.Jg;
% New values for beta and v
beta = [0.0000, 0.0306, 0.0560, 0.0772, 0.0953, 0.1113, 0.1254, 0.1382, 0.1499, 0.1607,.
0.1783, 0.2134, 0.2421, 0.2669, 0.2890, 0.3091, 0.3279, 0.3456, 0.3625, 0.3789]';
v = [11.2600, 11.5378, 11.8156, 12.0933, 12.3711, 12.6489, 12.9267, 13.2044, 13.4822,...
13.7600, 14.2600, 15.4533, 16.6467, 17.8400, 19.0333, 20.2267, 21.4200, 22.6133, 23.8067
25.0000]';

omega_r = 1.2671; % rated rotor speed in rad/s

% Define the auxiliary functions
lambda_i = @(lambda,beta) 1./(lambda+0.08*beta)-0.035./(c11+c12*beta.^3);
f1 = @(lambda) c4./lambda;
f2 = @(lambda,beta) (c5*lambda_i(lambda,beta))-c6*beta-c7*beta.^c8-c9;
f3 = @(lambda,beta) exp(-c10*lambda_i(lambda,beta));
```

```
% Define the analytical CQ function and its derivatives
dlambda_i_dlambda = @(lambda,beta) -1./(lambda+0.08*beta).^2;

cQ_tilde = @(lambda,beta) c1*(1+c2*(beta+c3).^2)+f1(lambda).*f2(lambda,beta).*f3(lambda,
cQ = @(lambda,beta) cQ_tilde(lambda,beta).*(1+sign(cQ_tilde(lambda,beta)))./2;
dcQ_dlambda = @(lambda,beta) -c4./lambda.^2.*f2(lambda,beta).*f3(lambda,beta)+
f1(lambda).*c5.*dlambda_i_dlambda(lambda,beta)).*f3(lambda,beta)+f1(lambda).
*f2(lambda,beta).*(-c10.*exp(-c10*lambda_i(lambda,beta)).*dlambda_i_dlambda(lambda,beta)
dcQ_dbeta = @(lambda,beta) f1(lambda).*(c5.*dlambda_i_dbeta(lambda,beta)
-c6-c7*c8*beta.^(c8-1)).*f3(lambda,beta)+f1(lambda).*f2(lambda,beta).
*(-c10.*exp(-c10*lambda_i(lambda,beta)).*dlambda_i_dbeta(lambda,beta))
+0.5*c1*c2*(beta+c3).^(-0.5);

% Define the derivatives of lambda_i function
dlambda_i_dlambda = @(lambda,beta) -1./(lambda+0.08*beta).^2;
dlambda_i_dbeta = @(lambda,beta) -0.08./(lambda+0.08*beta).^2+
(3*0.035*c12*beta.^2)./(c11+c12*beta.^3).^2;

% Define the lambda function as a function of wind speed and rotor speed
lambda = @(V,omega_r) omega_r*R./V;

% Calculate the three formulas for each wind speed
k_omega = zeros(length(v),1); % initialize the vector for k_omega
k_beta = zeros(length(v),1); % initialize the vector for k_beta
k_V = zeros(length(v),1); % initialize the vector for k_V
for i = 1:length(v)
    k_omega(i) = 0.5*rho*v(i)^2*pi*R^3(R/v(i)*dcQ_dlambda(lambda(v(i),omega_r),beta(i));
    k_beta(i) = 0.5*rho*v(i)^2*pi*R^3*dcQ_dbeta(lambda(v(i),omega_r),beta(i));
    k_V(i) = 0.5*rho*pi*R^3*(2*v(i)*cQ(lambda(v(i),omega_r),beta(i))-
    omega_r*R*dcQ_dlambda(lambda(v(i),omega_r),beta(i)));
end

% Create a table with the results
```

```
%T = table(v, k_omega, k_beta, k_V, 'VariableNames', {'WindSpeed', 'k_omega', 'k_beta',

% Write the table to an Excel file
%writetable(T, 'output.xlsx');
% Define the numerator and denominator coefficients
% num = [(-8.6/J) 0]; % s
% den = [0 1 -(-2.63/J)]; % s^2 + 3s + 2

% Create the transfer function model
% G = tf (num,den)
tau_ref = 4;
a = -k_omega/J;
b = k_beta/J;
k_P = 1./(b.*tau_ref);
k_I = a./(b.*tau_ref);
%-----------------------------------------

index_kP = find(k_P == out.KP1.Data);
index_ki = find(k_I == out.KI1.Data);
index_kbeta = find(k_beta == out.k_beta1.Data);
index_komega = find(k_omega == out.k_omega1.Data);

fprintf('Index in k_P: %d\n', index_kP);
fprintf('Index in k_i: %d\n', index_ki);
fprintf('Index in k_beta: %d\n', index_kbeta);
fprintf('Index in k_omega: %d\n', index_komega);
% Create a table with the results
T = table(v, k_omega, k_beta, k_V,k_P, k_I,  'VariableNames', {'WindSpeed', 'k_omega', '

%Write the table to an Excel file
writetable(T, 'output1.xlsx');
```

Bibliography

[1] P. Qian, X. Ma, and D. Zhang, "Estimating health condition of the wind turbine drivetrain system," *Energies*, vol. 10, no. 10, 2017, ISSN: 1996-1073. DOI: 10.3390/en10101583. [Online]. Available: https://www.mdpi.com/1996-1073/10/10/1583.

[2] L. Arturo Soriano, W. Yu, and J. d. J. Rubio, "Modeling and control of wind turbine," *Mathematical Problems in Engineering*, vol. 2013, p. 982 597, Aug. 2013, ISSN: 1024-123X. DOI: 10.1155/2013/982597. [Online]. Available: https://doi.org/10.1155/2013/982597.

[3] *Va3 automation in renewable energy system*, https://moodle.htw-berlin.de/course/view.php?id=45854, Last accessed on 1/28/2024 at 12:20am, 2024.