HAWC2 Course

Lesson 3: Control interface

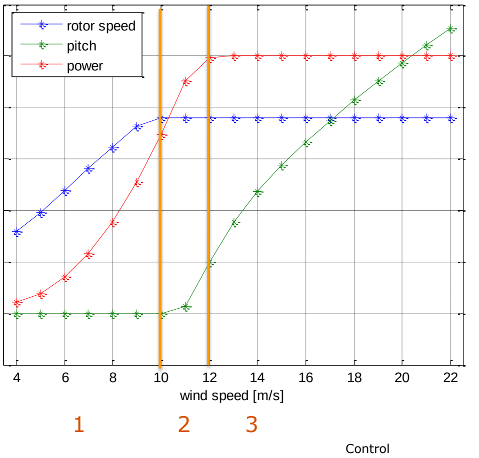
Risø DTU
National Laboratory for Sustainable Energy

Pitch control with variable speed

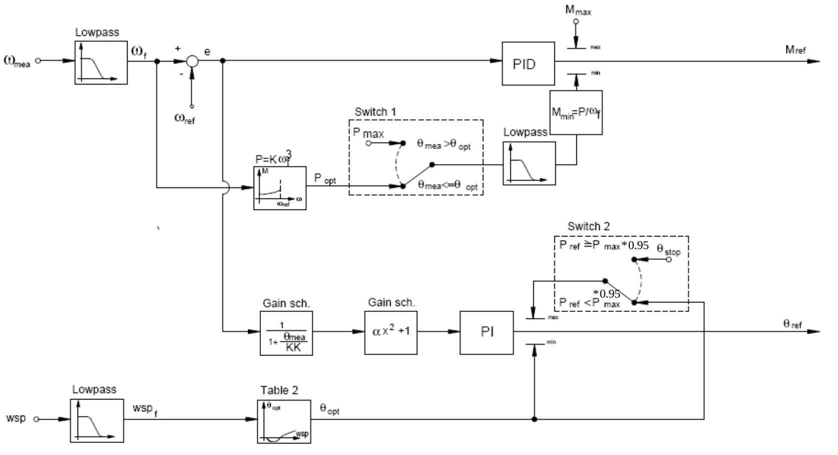
Objectives: Optimize power production below rated wsp, limit power and loads at high wsp.

Region

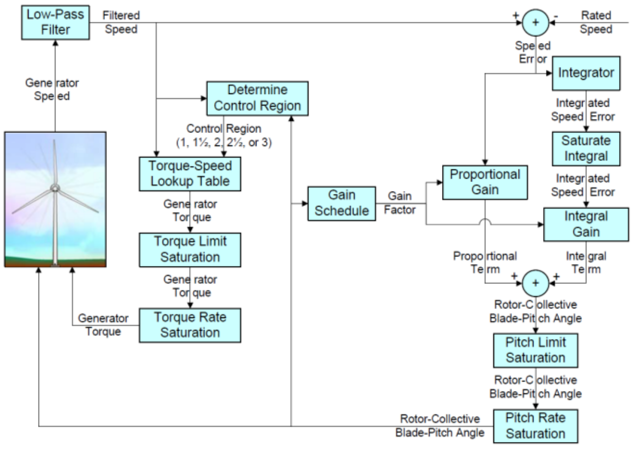
1. Variable speed, opt. power tracking.
2. Constant speed.
3. Power limitation



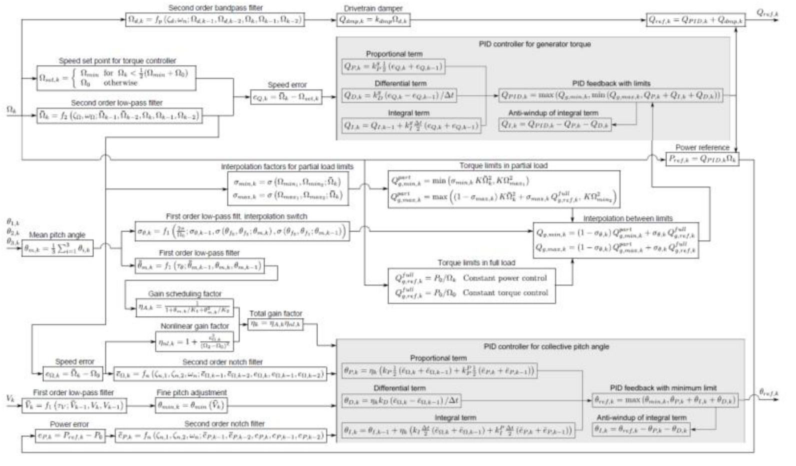
Control diagram
Old Risø controller



Control diagram
NREL 5MW Controller

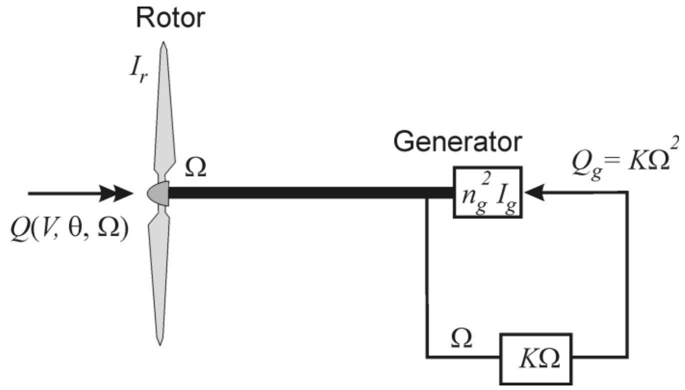


Control Diagram
DTU Wind Energy Basic Controller



Region 1: Optimal Cp tracking (1)

Strategy:	Keep pitch angle at optimum and use generator torque to obtain optimal tip speed ratio in a torque equilibrium



Region 1: Optimal Cp tracking (2)

Optimal aerodynamic power and torque

$$P_{\text{aero}} = \frac{1}{2}\rho A V^3 C_p(\theta_{\text{opt}}, \lambda_{\text{opt}}) \quad \Rightarrow \quad Q_{\text{aero}} = \underbrace{\frac{\frac{1}{2}\rho A R^3 C_p(\theta_{\text{opt}}, \lambda_{\text{opt}})}{\lambda_{\text{opt}}^3}}_{K_{\text{no loss}}} \Omega^2$$

Power equilibrium with losses described by efficiency η

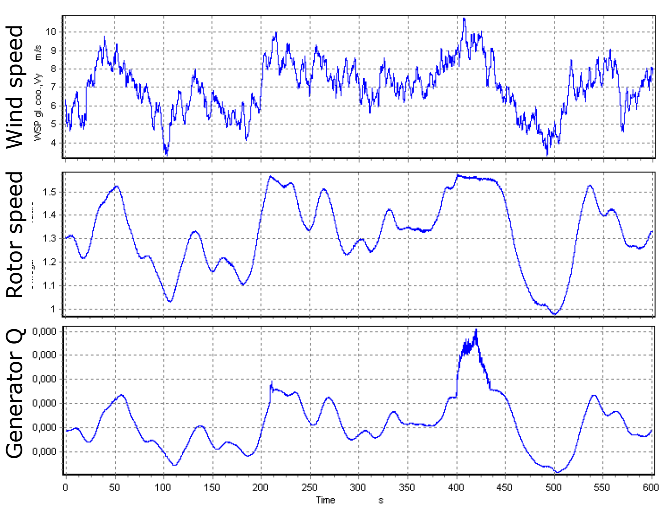
$$P_{\text{elec}} = Q_g \Omega = P_{\text{aero}} - P_{\text{loss}} = \eta\, P_{\text{aero}} = \eta\, K_{\text{no loss}}\, \Omega^3$$

Torque equilibrium

$$Q_g = \eta\, K_{\text{no loss}}\, \Omega^2 = K\, \Omega^2 \text{ where } K = \eta\, \frac{\frac{1}{2}\rho A R^3 C_p(\theta_{\text{opt}}, \lambda_{\text{opt}})}{\lambda_{\text{opt}}^3}$$

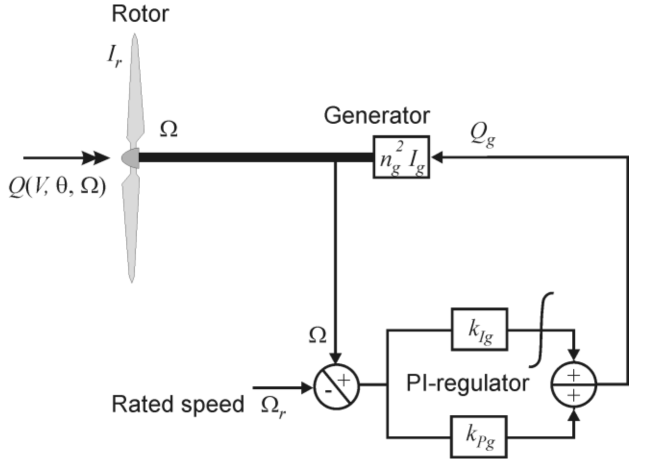
Region 1: Optimal Cp tracking (3)

Example:



Region 2: Speed regulation

Strategy:	Keep pitch angle at optimum and use generator torque to regulate rotor speed to the rated speed



Region 2: Speed regulation (2)

Nonlinear equation of drivetrain motion

$$\left(I_r + n_g^2 I_g\right)\,\dot{\Omega} = Q(V,\Omega,\theta) - \tfrac{1}{\eta} Q_g$$

PI feedback from speed error to generator torque

$$\begin{aligned}\Delta Q_g &= k_{Pg}\left(\Omega - \Omega_r\right) + k_{Ig}\int_0^t \left(\Omega - \Omega_r\right) dt \\ &= k_{Pg}\,\dot{\phi} + k_{Ig}\,\phi \text{ , where } \dot{\phi} = \frac{d\phi}{dt} = \Omega - \Omega_r\end{aligned}$$

Linear closed-loop equation for small drivetrain motion about $\Omega = \Omega_r + \dot{\phi}$

$$\underbrace{\left(I_r + n_g^2 I_g\right)}_{\text{mass}}\,\ddot{\phi} + \underbrace{\left(\frac{1}{\eta} k_{Pg} - \cancelto{0}{\left.\frac{\partial Q}{\partial \Omega}\right|_0}\right)}_{\text{damping}}\dot{\phi} + \underbrace{\frac{1}{\eta} k_{Ig}}_{\text{stiffness}}\,\phi = 0$$

Region 2: Speed regulation (3)

Linear closed-loop equation for small drivetrain motion about $\Omega = \Omega_r + \dot{\phi}$

$$\underbrace{\left(I_r + n_g^2 I_g\right)}_{\text{mass}} \ddot{\phi} + \underbrace{\frac{1}{\eta} k_{Pg}}_{\text{damping}} \dot{\phi} + \underbrace{\frac{1}{\eta} k_{Ig}}_{\text{stiffness}} \phi = 0$$

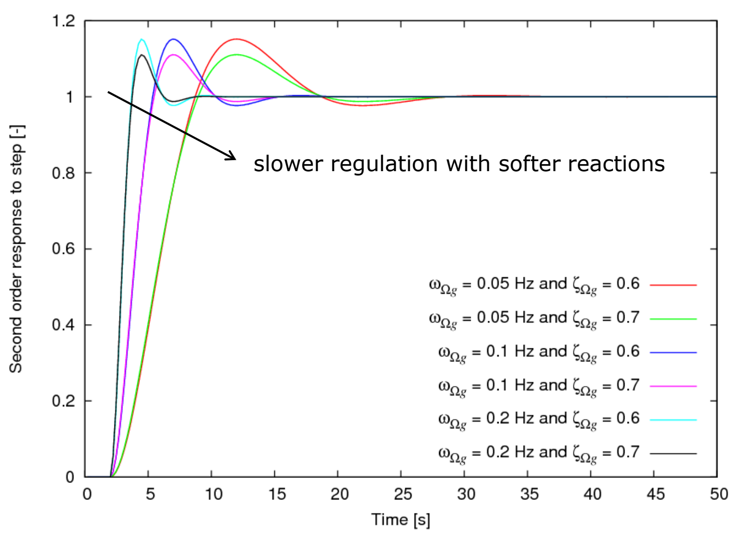
Tuning by placement of closed-loop drivetrain poles/eigenvalues

$$\begin{aligned} k_{Pg} &= 2\,\eta\,\zeta_{\Omega g}\omega_{\Omega g}\left(I_r + {n_g}^2 I_g\right) \\ k_{Ig} &= \eta\,\omega_{\Omega g}^2\left(I_r + n_g^2 I_g\right) \end{aligned} \quad \Rightarrow \quad \lambda = -\zeta_{\Omega g}\omega_{\Omega g} \pm i\omega_{\Omega g}\sqrt{1-\zeta_{\Omega g}^2}$$

Rule of thumb:

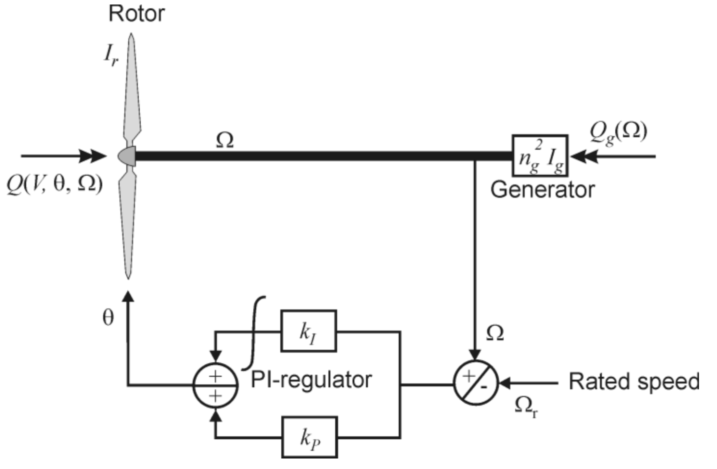
$\omega_{\Omega g} = 0.1$ Hz and $\zeta_{\Omega g} \in [0.6 : 0.7]$

Region 2: Speed regulation (4)



Region 3: Power and speed regulation

Strategy:	Regulate rotor speed by pitching and use generator torque to regulate power (either as constant power, or constant torque)



Region 3: Power and speed regulation (2)

Nonlinear equation of drivetrain motion

$$\left(I_r + n_g^2 I_g\right)\,\dot{\Omega} = Q(V,\Omega,\theta) - \tfrac{1}{\eta} Q_g(\Omega)$$

Generator model

$$Q_g(\Omega) = \begin{cases} \frac{P_r}{\Omega_r} & \text{DFIG with ideal constant torque control} \\ \frac{P_r}{\Omega} & \text{DFIG with ideal constant power control} \end{cases}$$

PI feedback from speed error to pitch angle

$$\theta = k_P\,(\Omega - \Omega_r) + k_I \int_0^t (\Omega - \Omega_r)\,dt$$
$$= k_P\,\dot{\phi} + k_I\,\phi\ , \text{ where } \dot{\phi} = \frac{d\phi}{dt} = \Omega - \Omega_r$$

Region 3: Power and speed regulation (3)

Linear closed-loop equation for small drivetrain motion about $\Omega = \Omega_r + \dot{\phi}$

$$\underbrace{\left(I_r + n_g^2 I_g\right)}_{\text{mass}} \ddot{\phi} + \underbrace{\left(\frac{1}{\eta}\frac{\partial Q_g}{\partial \Omega}\Big|_0 - \cancelto{0}{\frac{\partial Q}{\partial \Omega}\Big|_0} - \frac{\partial Q}{\partial \theta}\Big|_0 k_P\right)}_{\text{damping}} \dot{\phi} - \underbrace{\frac{\partial Q}{\partial \theta}\Big|_0 k_I}_{\text{stiffness}} \phi = 0$$

Tuning by placement of closed-loop drivetrain pole $\lambda = -\zeta_\Omega \omega_\Omega \pm i\omega_\Omega \sqrt{1-\zeta_\Omega^2}$

$$k_P = \frac{2\,\zeta_\Omega \omega_\Omega \left(I_r + {n_g}^2 I_g\right) - \frac{1}{\eta}\frac{\partial Q_g}{\partial \Omega}\Big|_0}{-\frac{\partial Q}{\partial \theta}\Big|_0} \quad \text{where} \quad \frac{\partial Q_g}{\partial \Omega}\Big|_0 = \begin{cases} 0 & \text{const. torque} \\ -\frac{P_r}{\Omega_r^2} & \text{const. power} \end{cases}$$

$$k_I = \frac{\omega_\Omega^2 \left(I_r + n_g^2 I_g\right)}{-\frac{\partial Q}{\partial \theta}\Big|_0}$$

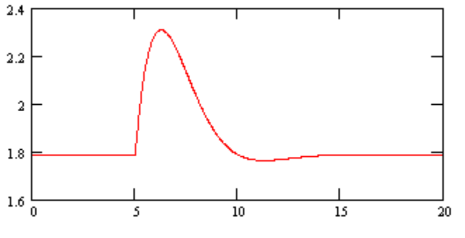
Rule of thumb:

$\omega_\Omega = 0.1$ Hz and $\zeta_\Omega \in [0.6 : 0.7]$

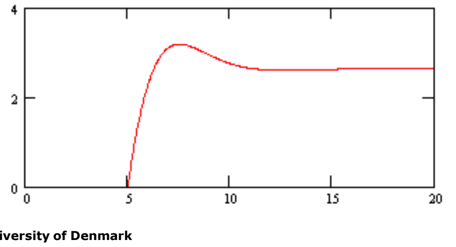
Region 3: Power and speed regulation (4)

Step change in external torque

Rotor speed

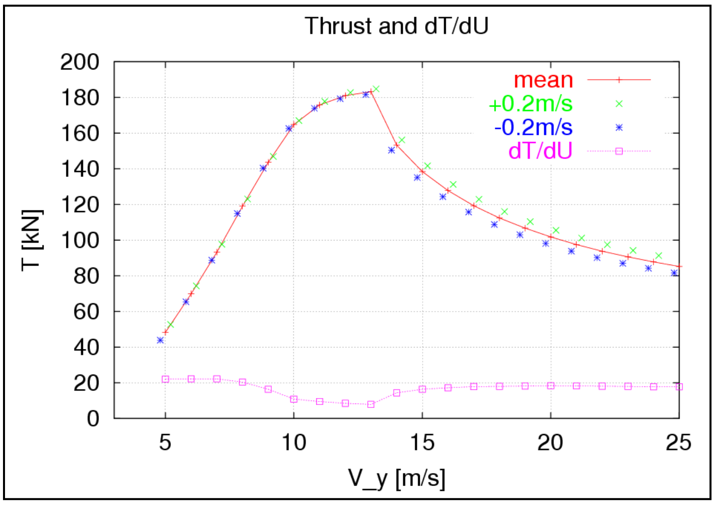


Pitch angle

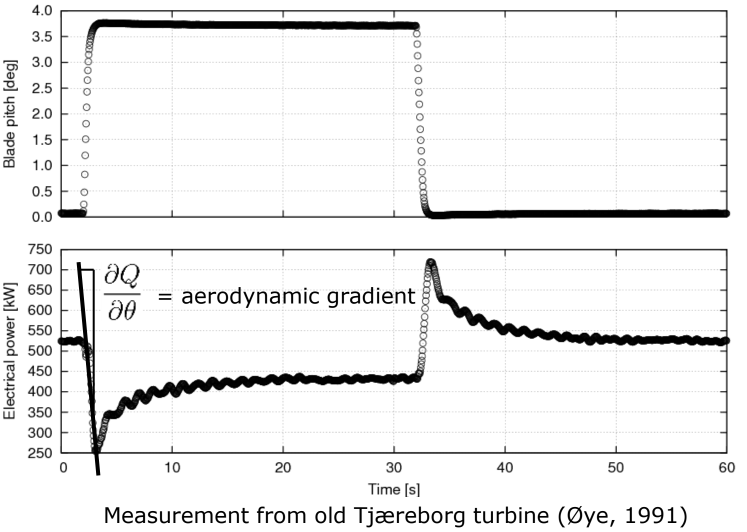


Region 3: Power and speed regulation (5)

Why is the control frequency so important?



Region 3: Power and speed regulation (6)



Measurement from old Tjæreborg turbine (Øye, 1991)

Region 3: Power and speed regulation (7)

Aerodynamic gradient assuming constant induced velocities (frozen wake)

$$\frac{\partial Q}{\partial \theta} = \frac{1}{2}\rho B \int_0^R c(r)U(r)^2 \left(C'_L(r)\sin\varphi(r) - C'_D(r)\cos\varphi(r)\right) r dr$$

where

$$C'_L(r) = \left.\frac{dC_L}{d\alpha}\right|_{\alpha=\alpha(r)}$$

$$C'_D(r) = \left.\frac{dC_D}{d\alpha}\right|_{\alpha=\alpha(r)}$$

$c(r)$ = chord distribution

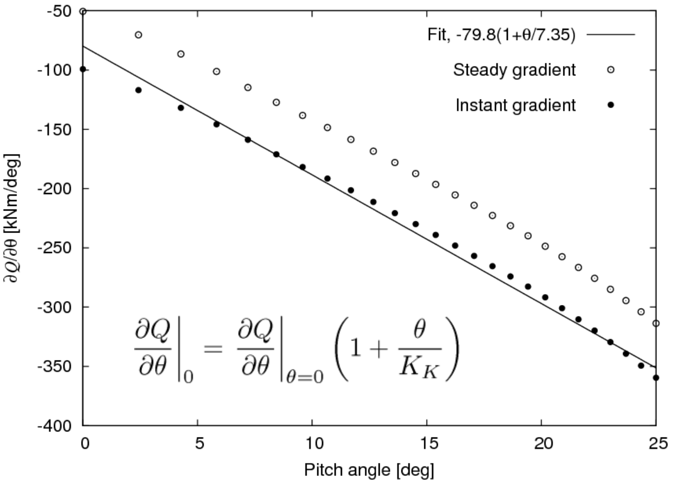
$U(r)$ = steady state relative inflow speed

$\phi(r)$ = steady state inflow angle

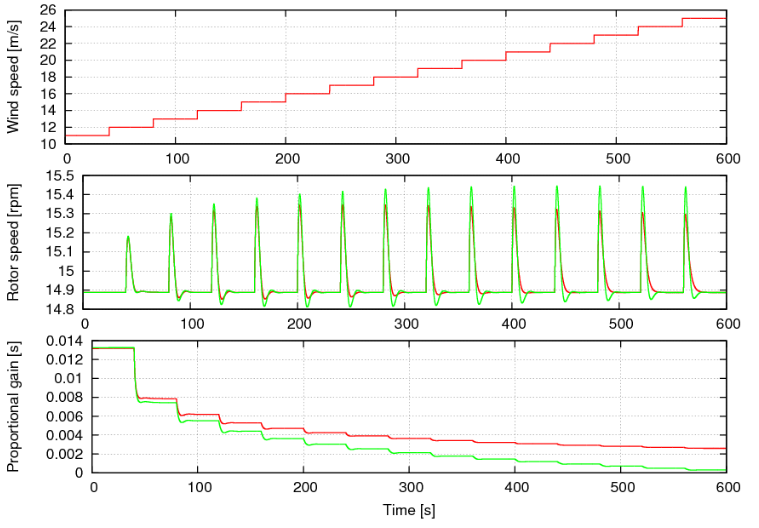
$\alpha(r)$ = steady state angle of attack

Region 3: Power and speed regulation (8)

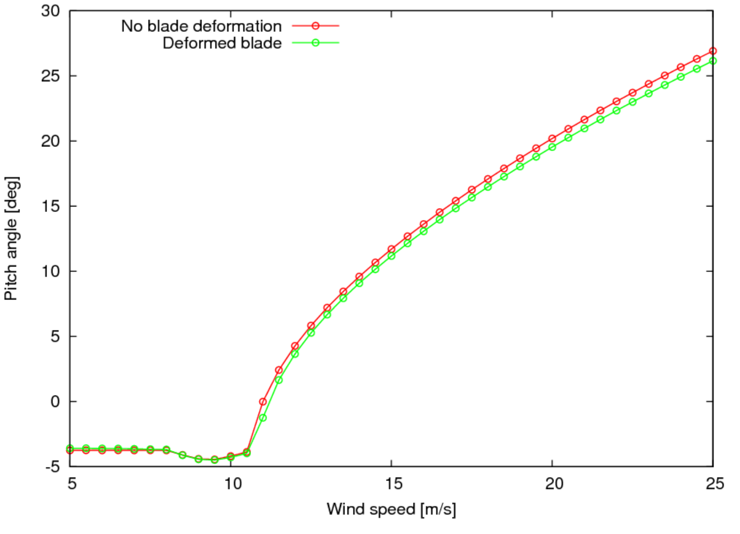
Gain scheduling of aerodynamic gradient



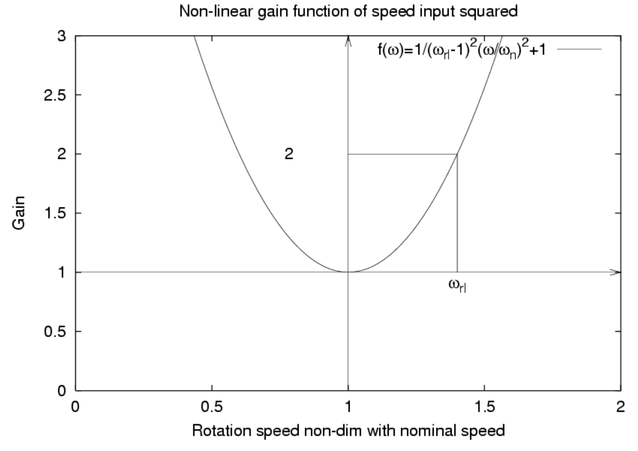
Region 3: Power and speed regulation (9)



Effect of blade deformations



Non-linear gain



Gain function used with special low frequent towers (floating turbines etc.)

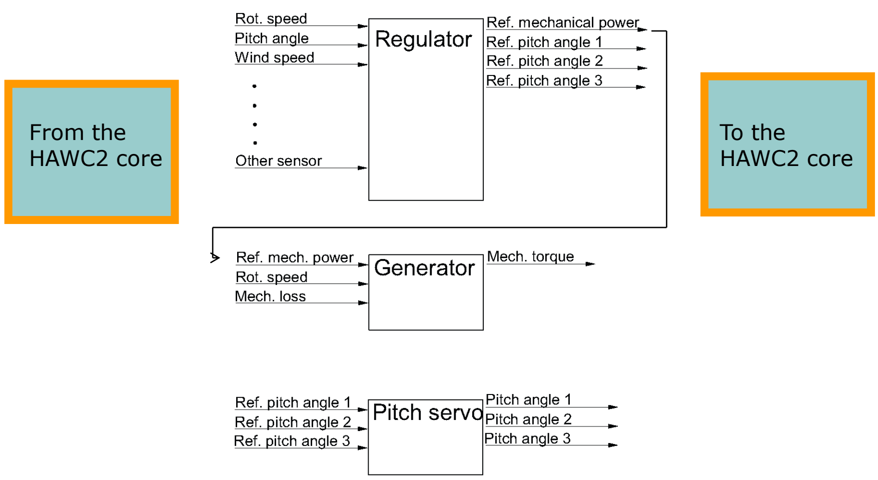
Parameter is non-dim rotor speed where gain equals 2.0

Tuning of Basic DTU Wind Energy Controller with HAWCStab2

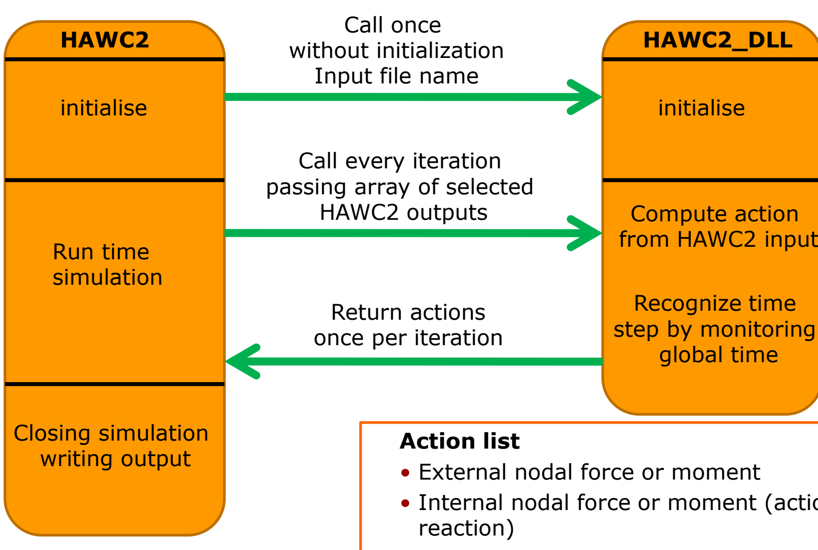
```
begin hawcstab2 ;
  begin ground_fixed_substructure ;
    main_body tower ;
    main_body towertop ;
  end ground_fixed_substructure ;
  begin rotating_axissym_substructure ;
    main_body shaft ;
  end rotating_axissym_substructure ;
  begin rotating_threebladed_substructure ;
    main_body hub1 ;
    main_body blade1 ;
end rotating_threebladed_substructure ;
…
```

```
…
  begin operational_data ;
    windspeed 4.000 25.000 22 ; cut-in, cut-out, points
    genspeed 6.900 12.100 ;
    gearratio 1.000 ;
    minpitch 0.000 ;
    opt_lambda 7.500 ;
    maxpow 5296.610 ;
    prvs_turbine 1 ;
    include_torsiondeform 0 ;
  end operational_data ;
  begin controller_tuning ;
    partial_load 0.050 0.700; fn [hz], zeta [-]
    full_load 0.100 0.700 ; fn [hz], zeta [-]
    gain_scheduling 1 ; 1 linear, 2 quadratic
  end controller_tuning ;
end hawcstab2 ;
```

Control through external DLL's

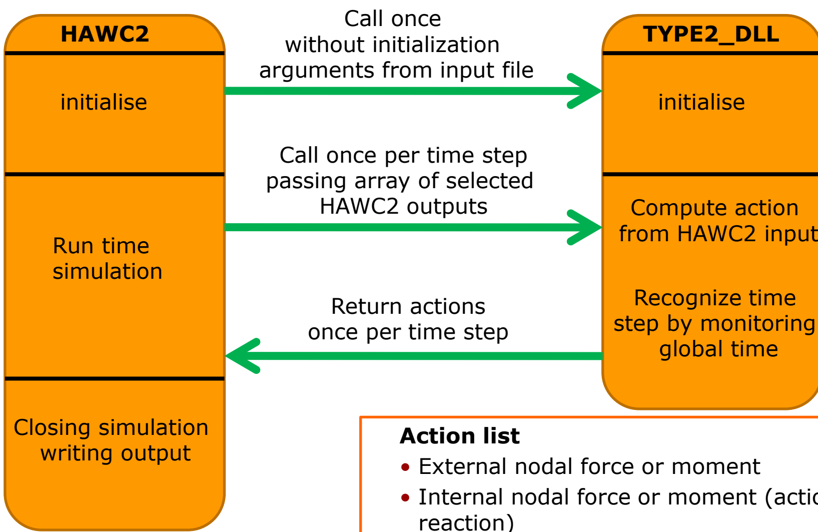


HAWC2_DLL interface



Action list
- External nodal force or moment
- Internal nodal force or moment (action and reaction)
- Bearing angle
- Flap angle

TYPE2_DLL interface



Action list
- External nodal force or moment
- Internal nodal force or moment (action and reaction)
- Bearing angle
- Flap angle

Coupling of control to HAWC2

```
begin type2_dll;
   name risoe_controller ;
   filename  ./control/risoe_controller.dll ;
   dll_subroutine_init init_regulation ;
   dll_subroutine_update update_regulation ;
   arraysizes_init  50 1 ;
   arraysizes_update  8 100 ;
  begin init ;
      ; Overall parameters
     constant   1 5000.0  ; Rated power [kW]
     constant   2   0.7233  ; Minimum rotor speed [rad/s]
     constant   3    1.2671  ; Rated rotor speed [rad/s]
     constant   4  4.598e6  ; Maximum allowable generator torque [Nm]
     constant   5 -1.1    ; Minimum pitch angle, theta_min [deg],
     . . .
     constant  42  0.5  ; PitNonLin1 [rad/s]
  end init ;
;
   begin output ;
     general time ; [s]
     constraint bearing1 shaft_rot 1 only 2 ; Drivetrain speed [rad/s]
     constraint bearing2 pitch1 1 only 1; [rad]
     constraint bearing2 pitch2 1 only 1; [rad]
     constraint bearing2 pitch3 1 only 1; [rad]
     wind free_wind 1 0.0 0.0 -90.0 ; global coords at hub height
    general constant 0.0      ; Pitch rate from external system [rad/s]
   end output;
 end type2_dll;
```

Coupling of simple generator

```
 begin type2_dll;
    name generator_servo ;
    filename  ./control/generator_servo.dll ;
    dll_subroutine_init init_generator_servo ;
    dll_subroutine_update update_generator_servo ;
    arraysizes_init  6 1 ;
    arraysizes_update  3 6 ;
   begin init ;
      constant 1  20.0     ; Frequency of genertor 2nd order control model [Hz]
      constant 2  0.9     ; Damping ratio of genertor 2nd order control model [-]
      constant 3 4.598e6   ; Maximum allowable LSS torque (pull-out torque) [Nm]
      constant 4 0.944     ; Generator efficiency [-]
      constant 5 1.0      ; Gearratio [-]
    end init ;
;    ;
    begin output;
      general time                         ;   Time [s]
      dll inpvec 1 1                       ;   Electrical torque reference [Nm]
      constraint bearing1 shaft_rot 1 only 2;   Generator LSS speed [rad/s]
    end output;
;;
    begin actions;
      mbdy moment_int shaft 1 -3 shaft towertop 2 ;   Generator LSS torque [Nm]
    end actions;
  end type2_dll;
```


Coupling of simple pitch servo

```
 begin type2_dll;
    name servo_with_limits ;
    filename  ./control/servo_with_limits.dll ;
    dll_subroutine_init init_servo_with_limits ;
    dll_subroutine_update update_servo_with_limits ;
    arraysizes_init  7 1 ;
    arraysizes_update  4 9 ;
   begin init ;
      constant 1  3     ; 1: Number of blades [-]
      constant 2  1.0   ; 2: Filter frequency [Hz]
      constant 3  0.7   ; 3: Filter damping ratio [-]
      constant 4 8.0   ; 4: Max. pitch speed [deg/s]
      constant 5  15.0   ; 5: Max. pitch acceleration [deg/s^2]
      constant 6  -5.0   ; 6: Min. pitch angle [deg]
      constant 7 90.0   ; 7: Max. pitch angle [deg]
   end init ;
   begin output;
      general time       ;  1: Time                        [s]
      dll inpvec 1 2     ;  2: Pitch1 demand angle          [rad]
      dll inpvec 1 3     ;  3: Pitch2 demand angle          [rad]
      dll inpvec 1 4     ;  4: Pitch3 demand angle          [rad]
   end output;
   begin actions;
      constraint bearing2 angle pitch1 ; Angle pitch1 bearing    [rad]
      constraint bearing2 angle pitch2 ; Angle pitch2 bearing    [rad]
      constraint bearing2 angle pitch3 ; Angle pitch3 bearing    [rad]
   end actions;
 end type2_dll;
```

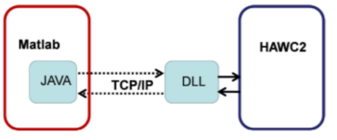
Other dll possibilities

type2_dll

```
  begin type2_dll;
    name  hss_convert ;
    filename  ./hss_convert.dll ;
    dll_subroutine_init 'Initialize' ;
    dll_subroutine_update 'Sensors' ;
    arraysizes_init  3 1 ;
    arraysizes_update  2 2 ;
    begin init ;
      constant 1 2.0 ; number of used sensors
      constant 2 112.43 ; gearbox ratio
      constant 3 112.43 ; gearbox ratio
    end init ;
;
    begin output ;
      constraint bearing1 shaft_rot 2 only 2 ; rotor
   speed in rpm
      constraint bearing1 shaft_rot 3 only 2 ; rotor
   speed in rad/s
    end output ;
;
    begin actions;
;     rotor_speed in rpm*gear_ratio
;     rotor_speed in rad/s*gear_ratio
    end actions;
  end type2_dll;
```

tcp/ip communication with e.g. Matlab or Matlab/Simulink

```
begin hawc_dll;
  filename  ./tcpip/TCPserver.dll ;
  dll_subroutine tcplink_delay ;
  init_string 1139 ;
  arraysizes  60 60 ;
begin output;
  continue_in_file ./htc/tcplink_sensors.htc ;
end output;
;
begin actions;
end actions;
end hawc_dll ;
```



Exercise

- Run the file structure_aero_control.htc 600s
- Run the file structure_aero_control_turb.htc 600s