electrical power and an anemometer (noisy, single-point) wind speed. Many other types of sensors are available commercially and could conceivably be incorporated in a wind turbine if there were a compelling application.

5.2 Control Algorithms

Figure 5.4 is a hybrid state-space/block diagram of a wind turbine employing feedback control.[5] The turbine dynamics are represented by the states $\mathbf{x}_w$ and state function $\mathbf{w}$, acted upon by environmental inputs $\mathbf{e}$, such as wind and waves, and control inputs $\mathbf{u}$. The response of the system is characterized by selected outputs $\mathbf{y}_w$. These outputs are measured by sensors which have some internal dynamics, including filtering of the signals, represented by the states $\mathbf{x}_s$ and state function $\mathbf{s}$. The measurements are not perfect and are subject to noise, represented by $\mathbf{m}$. The filtered measurements $\mathbf{y}_s$ are fed into a controller, together with operator inputs $\mathbf{p}$.[6]

The control design task is to specify what goes into the 'Controller' box: the states $\mathbf{x}_c$, state function $\mathbf{c}$, and output function $\mathbf{u}$. In principle, *this can be just about anything*; for instance, the controller may include an embedded model of arbitrary complexity, in practice limited only by the capabilities of the control hardware. The various branches of control theory are attempts at finding 'recipes' which guarantee that the closed-loop system behaviour has certain desirable properties, such as stability, good performance according to some selected metrics and robustness against external disturbances and inaccuracies in the plant model $\mathbf{w}$.

It is possible to fulfil the engineering requirements of a wind turbine controller with a variety of control architectures: there is no single correct solution. At the same time,

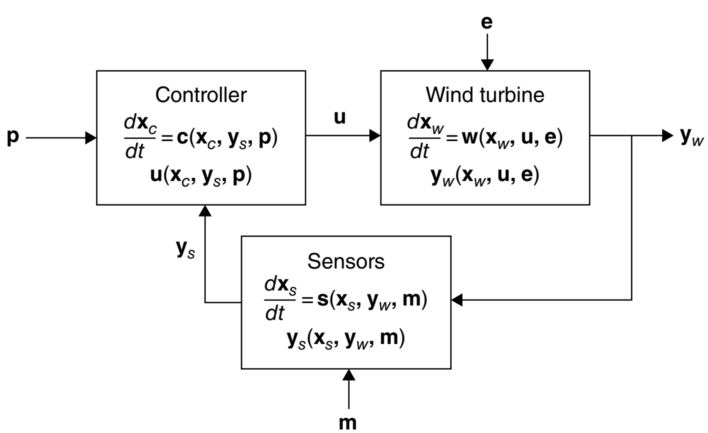

Figure 5.4 A high-level block diagram that could represent a wind turbine or wind power plant employing feedback control.

5 The same system could be represented by a single, unified state space, or an alternate block diagram representation. This particular form was chosen in order to emphasize the aspects of sensor measurements and control.

6 Do not worry about memorizing this choice of terminology, it is essentially unused outside of this section.

established recipes for control design must be treated with caution and validated extensively on a high order model of the turbine or plant.

5.2.1 Overview of Algorithms

A review of the literature on wind turbine control will reveal that there are many algorithms that are capable of fulfilling the control objectives to a satisfactory degree.

5.2.1.1 Single-input, Single-output Controls

The simplest controller for a pitch-regulated, variable speed wind turbine consists of a look-up table for the generator torque, as a function of rotor speed, and a PI controller for collective blade pitch, which becomes active when the rotor speed hits its rated value. The sole sensor input required for normal operation is a measurement of the shaft speed. This controller can fulfil the primary objective of power production and rotor speed limitation.

Even the most simple blade pitch PI controller is complicated by the need to schedule the gains as a function of the wind speed. This is not trivial and to be done properly requires a detailed aeroelastic model.[7] The resulting controller is not robust, in the sense that it may give a significantly different performance if implemented on a model other than that with which it was tuned (Merz, 2016b).

Additional single-input, single-output pathways are added in order to fulfil specific secondary objectives. Each objective is assigned either to the generator torque or the blade pitch action; thus, tower fore–aft damping is provided by blade pitch, while side-to-side damping is provided by generator torque. Frequency filters are used to target specific modes or functions with each pathway, and adjust the phase of the control action relative to the sensor inputs. The required sensor inputs depend upon which secondary objectives are implemented. For example, driveshaft torsional damping can be implemented with the basic measurement of rotor speed, whereas tower damping requires some additional measure of the tower motions. Active power commands can be implemented by scaling the nominal torque–speed relationship of the generator, according to a PI control on the measured electrical power (Jeong *et al.*, 2014).

If a controller of this sort is asked to perform several objectives simultaneously, then the tuning of the gains and filters can become a challenge. The reason is that the different control loops interact, through the response of the wind turbine. That being said, it is shown in Section 5.5 that, through a judicious selection of filters, the control functions can be tuned *almost* independently.[8]

5.2.1.2 Advanced Controls

Alternate PI-type control strategies have been proposed. For instance, a controller might take advantage of a measured or estimated wind speed (Østergaard *et al.*, 2007)

7 The reason for this is that the flexibility of the blades and tower influence the relationship between a perturbation in the blade pitch and the resulting aerodynamic torque on the rotor. See Section 5.4.

8 Although, tuning in a simulation environment and real life are not always the same thing (Fleming *et al.*, 2013).

to supplement the rotor speed input. Such additional measurements can be used to obtain improvements in energy production or load reduction (Bottasso *et al.*, 2013).

The principle alternative to control systems constructed of single-input, single-output pathways is the class of multiple-input, multiple-output 'modern' control systems. These construct the set of control commands (blade pitch, generator torque) based upon the set of sensor measurements (rotor speed, electrical power, accelerations etc.) in a way that can explicitly account for the system dynamics and interactions between the variables.

Multiple-input, multiple-output control systems are typically formulated in state-space and based around a model of the plant that is embedded within the controller. A distinction is made between *linear parameter-varying* models – those which are formulated using linear theory and scheduled on the basis of one or more slowly-varying parameters – and truly nonlinear models.

In the fully nonlinear case, one approach is to solve the optimal control problem explicitly, at frequent intervals, using the embedded model and forecasted inputs. This is known as *model predictive control* and requires the definition of a suitable cost function (Spencer *et al.*, 2013; Schlipf *et al.*, 2013; and many others). Another method is *sliding-mode control*, where a dominant control action – in the sense that it dominates the system dynamics and disturbances – is used to force the nonlinear system onto a trajectory with known, desirable, typically linear dynamics. The dominant control action employs an approximation to a rapid on-off switching behaviour, though techniques are available to relax the severity of the switching (Munteanu *et al.*, 2008; Beltran *et al.*, 2009). Other control algorithms may also be designed based directly upon Lyapunov proofs of stability, though this may require some mathematical ingenuity. *Passivity* theory, ensuring that the closed-loop system dissipates energy, is one way to develop nonlinear controllers that are stable in the Lyapunov sense (Sørensen *et al.*, 2014).

From an engineering standpoint, linear systems are preferable to nonlinear systems. If the system is linear and time-invariant, then recipes are available for an optimal *linear quadratic Gaussian* controller. This takes the form of a *Kalman filter* with a *linear quadratic regulator*, as sketched in Figure 5.5, ignoring operator inputs $\mathbf{p}$. A Kalman filter is based on an embedded model, where an estimate of the states is obtained by driving the error between the measured and computed outputs, $\mathbf{y}_s - \mathbf{y}_c$, to zero. The observer gain matrix $\mathbf{K}_o$ is computed so as to minimize the mean-squared error in the state estimate, and the control gain matrix $\mathbf{K}_c$ is computed so as to minimize the time integral of a quadratic function of the states and control actions (Kumar and Stol, 2009; Camblong *et al.*, 2014). Alternatively, these gain matrices can be specified so as to place the observer and turbine system poles – that is, the modal frequency and damping properties of the system response – as desired. Depending on the number of actuators, it is also possible to specify portions of the mode shapes, the eigenvectors, in addition to the poles (Stevens and Lewis, 2003).

There are many variations on the basic linear quadratic Gaussian approach. For instance, an integral effect, $\dot{\mathbf{x}}_e = \mathbf{r} - \mathbf{y}_s$, with $\mathbf{r}$ some reference, may be included as *state augmentation* in order to provide tracking capability. Another possibility is to use additional states in the model of Figure 5.5 to account for the dynamics of the disturbances $\mathbf{e}$ and $\mathbf{m}$. This leads to a category of *disturbance accommodating control.* (Stol and Balas, 2003; Wright, 2004)

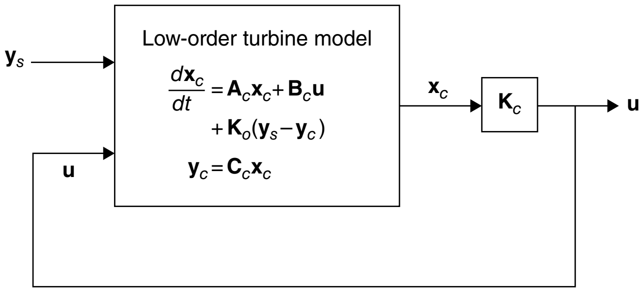

Figure 5.5 A controller consisting of a linear state observer (a typically low-order embedded model of the wind turbine) and state-based control law *u = K_c × _c*.

There may be some question as to how the quadratic cost function of states and control actions should be weighted. Recipes are available for computing these weights in terms of common-sense goals, such as minimizing the root mean square of the disturbance-output transfer function ($\mathcal{H}_2$ *optimal control*), or minimizing the highest disturbance-output gain at any frequency ($\mathcal{H}_\infty$ *optimal control*).[9] It is possible to add additional frequency-dependent weighting functions in order to attain a desired input-output transfer function, by the process of *loop shaping*. For instance, one may wish to suppress a particular set of resonant modes (Fleming *et al.*, 2013).

A wind turbine is fundamentally a nonlinear system, and although linear theory provides useful results, in a practical application the nonlinearity must be dealt with. One way to do so, while retaining the form of linear theory, is to formulate the equations as a *linear parameter-varying* system, generically as either

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}(\mathbf{x},\mathbf{u},t)\mathbf{x} + \mathbf{B}(\mathbf{x},\mathbf{u},t)\mathbf{u} \tag{5.2}$$

or

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{f}(\mathbf{x},\mathbf{u},t) \tag{5.3}$$

where, in the latter case, the nonlinearity has been moved into **f**; and **u** is here the complete set of inputs to the system, not only the controls.[10] A controller in the form of Equation 5.2 is said to be *gain-scheduled*. If the parameter-varying aspect is formulated as a discrete set of equations ($\mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$ and so on), with some form of interpolation used to obtain the net solution, then this is sometimes referred to as a *fuzzy* representation.[11] The magnitude of the nonlinearity does not need to be small; but it can be intuitively

9 Although the goals are common-sense and can be stated simply in plain language, the mathematical formulation and solution of the problem is not trivial.

10 Equation 5.2 is the more useful form in general, whereas Equation 5.3 is beneficial once we have chosen an operating point, and wish to investigate the degree of nonlinearity in its vicinity.

11 (Jelavic *et al.*, 2007; Pan and Ma, 2013). Applications of fuzzy logic techniques to wind turbine control have thus far been superfluous, in the sense that the terminology and theorems of fuzzy logic are not really put to good use. A common-sense formulation – solving the linearized equations at discrete points and interpolating on the basis of selected parameters which represent the operating characteristics of the system – is covered in jargon (fuzzified and defuzzified).

appreciated that, if Equations 5.2 or 5.3 are to be useful for a large nonlinearity, then its time scale must be slow in comparison with that of the linear dynamics. (This is indeed often the case for wind turbines.)

It is possible to design control algorithms that adapt themselves online, during operation, in order to incrementally improve performance towards some optimum. Such *adaptive, evolutionary or intelligent control* algorithms do not necessarily require an embedded model like Figure 5.5.[12]

All of the techniques mentioned above have been suggested for application to the control of wind turbines. For engineering applications, we favour methods based on linear theory; although ultimately (as with any real-life system) the true optimum is nonlinear.

5.2.2 Realization of a Controller for a 10-MW Wind Turbine

In order to demonstrate the control of a large offshore wind turbine, and study the closed-loop dynamics, we need a controller. Obtaining a suitable one is not entirely straightforward. Industrial wind turbine controllers are proprietary and closely guarded; whereas the baseline controllers for publically-available reference wind turbines, such as those from NREL (Jonkman *et al.*, 2009) and DTU (Hansen and Henriksen, 2013), do not include all of the functions required to fulfil the control objectives of Section 5.1.

A complete control architecture can be pieced together from functions that have been described in the literature. We can get most of the way with van der Hooft *et al.* (2003) and van Engelen *et al.* (2011), whose controllers perform all the functions save power command tracking. The DTU Basic Wind Energy controller (Hansen and Hendriksen, 2013) includes rotor speed and driveshaft damping functions, and is tailored to the DTU-designed rotor used on the ORT. Power command tracking functions can be obtained from Ela *et al.* (2014) and Jeong *et al.* (2014). Individual blade pitch control is described by Bossanyi (2003), among others.

A basic control strategy assigns each control function to either the blade pitch or the generator torque. One single-input, single-output control path is designed for each function. Then the gains and filter parameters are tuned such that an acceptable overall performance is obtained. It will be shown that it is possible to implement the functions such that they are largely independent, with only a minor amount of coupling between paths. This allows each function to be tuned in sequence, without having to iterate. However, such a controller with independent paths is limiting. If the goal is the best possible performance, then a model-based control algorithm should be selected.

The control functions of a wind turbine are arranged hierarchically. At the highest level is the supervisory control, which will not be considered here; it is assumed that the wind turbine is operating normally. The next level of control is split into two separate paths, one for the generator and another for the blade pitch. The generator handles control of the electrical power, as well as damping of the drivetrain and the tower

12 Model-free adaptive controllers have been proposed for wind turbines; but since a good model of the turbine dynamics can be constructed, why not put this to use? Adaptive control seems best suited as an augmentation to a model-based controller, in order to increase the robustness to uncertainties in the model. (Kusiak *et al.*, 2010)

side-to-side motion. Region II rotor speed control and tower fore–aft damping are handled by collective blade pitch, while additional rotor load rejection is provided by individual blade pitch.

If, at each stage, one attempts to select the simplest implementation, the result is the control architecture shown in Figures 5.6, 5.7 and 5.8. This is composed of filters, integrators and gains, the filter types being listed in Table 5.1. The architecture has been sketched without explicitly showing gain scheduling, rate limits or integrator anti-windup. Not all the pathways are effective simultaneously; path selection, and transition between paths, can be accomplished by scaling the gains. Filter time constants may also be scheduled.

Generator torque control is shown in Figures 5.6 and 5.7. The first stage, Figure 5.6, is an active power controller. In Region I, this operates in the maximum power-point tracking mode, where the power command is set as a cubic function of the measured speed. Alternatively, a speed tracking mode is active when operating at the minimum rotor speed. Here a PI controller drives the speed error to zero. Only one of these two pathways is active at a given time. The transition between modes may be scheduled as a function of the estimated windspeed, or blade pitch angle. There is also the option of bypassing the power control and providing a power command $\hat{P}_e$ directly. This occurs in Region II, where the power command is simply held constant at the rated power. A supervisory-level power command can also override the other power control functions.

A tower side-to-side damping function is achieved based on a measurement of nacelle acceleration. Side-to-side damping of the tower may be implemented here, as a power command, or alternatively at the level of the generator torque/current controller of Figure 5.7. It is shown in Section 5.6 that it makes sense to implement tower damping as a power command, provided that the active power control responds quickly, relative to the tower resonant frequency.

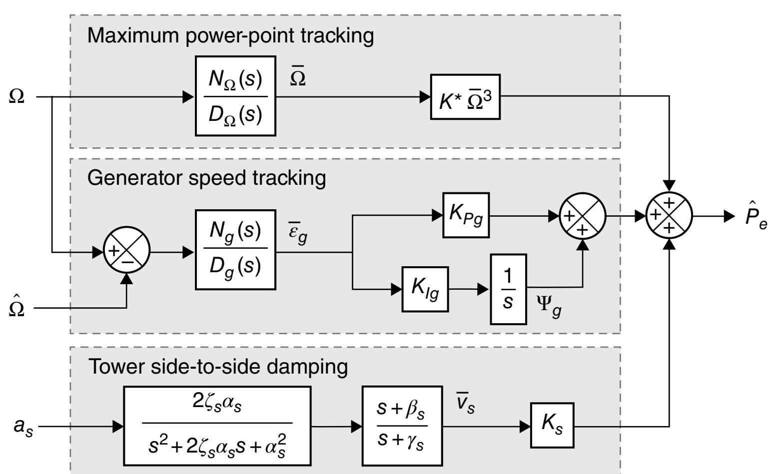

Figure 5.6 The active power set-point stage. Maximum power-point tracking is active in Region I, and generator speed tracking is active when operating at the minimum rotor speed cut-off.

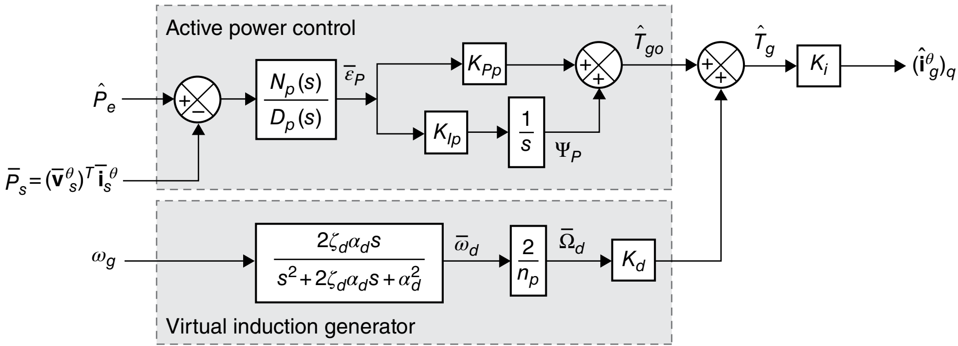

Figure 5.7 The generator torque/current controller. This consists of a function that controls the generator in order to provide a commanded active power at the network-side terminals of the turbine's transformer, and a virtual induction generator for damping the driveshaft. Active damping of tower side-to-side resonance may also be implemented at this level of control.

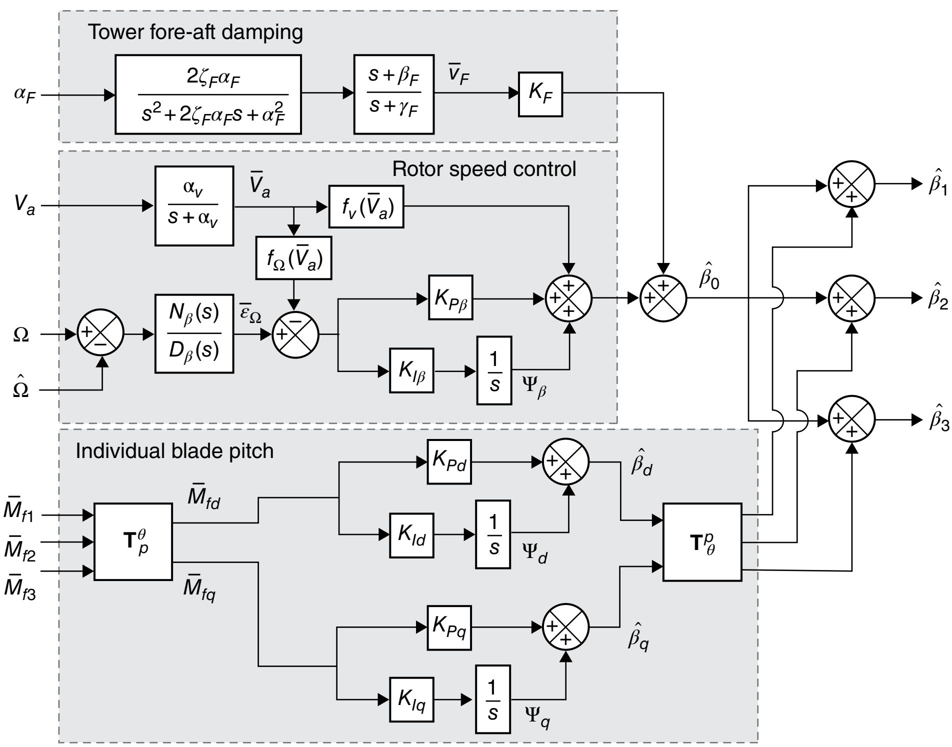

Figure 5.8 The blade pitch controller. Not all the pathways shown in the rotor speed control function are active simultaneously. Either the speed reference is fed in externally, it is a function of the estimated windspeed (through f_Ω), or the gains are set to zero. In the latter case the blade pitch may be prescribed directly as a function of the estimated windspeed.

Table 5.1 Filters employed in the controller.

Filter type	Formulation
Lowpass	$\frac{\alpha}{s+\alpha}$
Bandpass	$\frac{2\zeta\alpha s}{s^2+2\zeta\alpha s+\alpha^2}$
(Bandpass plus integrator)	$\frac{2\zeta\alpha}{s^2+2\zeta\alpha s+\alpha^2}$
Notch	$\frac{s^2+2\zeta_1\alpha s+\alpha^2}{s^2+2\zeta_2\alpha s+\alpha^2}$
Phase shift/lead-lag	$\frac{s+\beta}{s+\gamma}$

The second stage of generator control is the torque/current controller shown in Figure 5.7. This accepts a power command as input and provides a current command to the power converter.[13] A virtual induction generator also interfaces at this stage. This damps driveshaft and blade edgewise modes that would otherwise have poor stability.

Figure 5.8 shows the architecture of the blade pitch controller. The basic rotor speed control function is the path that leads from the rotor speed error $\Omega-\widehat{\Omega}$ through to the collective blade pitch command $\hat{\beta}_0$. Under certain operating conditions this is augmented by an estimate of the windspeed, or a derived quantity like the tip-speed ratio. Specifically, when operating at the minimum speed cut-off in low winds, the function f_V is used to directly schedule the blade pitch; and under certain types of operator power command tracking, $f\Omega$ sets the target rotor speed based on an estimate of the wind speed. Tower fore–aft damping can be provided through control of blade pitch. Individual blade pitch is used to counter wind shear, as well as asymmetric turbulence and wake effects. Flapwise moments at the blade root are measured and transformed into the nonrotating nacelle nod (d) and yaw (q) axes, and independent controllers drive the moments about each of these axes to zero.

At this point, we have intentionally not gone into the details of each control function. The functions are so simple in structure that not much enlightenment is gained by looking at the controller in isolation.[14] What is interesting is how the feedback control influences the wind turbine dynamics. This is explored in Sections 5.4 to 5.6, after first sketching out a dynamic model of an offshore wind turbine.

13 What happens with the current command after it is handed off to the converter control is described in Section 5.3.3. It can safely be thought of as a torque command.

14 However, the reader who is unfamiliar with dynamic filters is encouraged to construct magnitude and phase plots of the individual control functions, in the frequency domain, picking some example tuning parameters. This can be accomplished in just a few lines of code, using a program like MATLAB, which can handle complex variables.