PREDICTIVE CONTROL FRTN15 2012

Some Reading Advice[1] to "Predictive and Adaptive Control", v. Aug 2011

Chapter 1

In Chapter 1 you don't need to understand any details (these are sometimes left out).

The second section in 1.1 seems to include some incomplete or hard to understand sentences. Just skip them.

In (1.1) the z-transform is used. It is described in more detail in Ch2 and Appendix A.1-A.5

In figure (1.1) the lower signal is only known part of the time (otherwise the problem to reproduce u_k would be trivial for the reciever.) These symbols, sometimes called "pilot symbols", are fixed and known by both transmitter and receiver. They dont convey any information, so they are overhead that one wants to minimize when one designs the system.

Skip (1.14) and the sentence before it, if you don't know what a "describing function" is.

In Fig 1.8 it is hard to see what the process parameters and controller parameters are. Don't bother. The figure shows that one runs some parameter estimation for some time, and then connects the adaptive controller. The new controller has much better control of the output y, with only a modest in crease in control variance.

Fig 1.9 seems not to be referenced in the text. The controller is of a form we will study in Ch5. It focuses on minimizing the variance of the output y.

In (1.24) the signal r_k should not be mixed with the control parameter r_1. Unfortunate, but common notation.

The Smith predictor (that we also studies in our basic course) described in 1.5.1 is a special case of the calcuation in 1.4 with $G_m = (1-z^d)G_0$. It looks like (1.38) gives a very nice result and design method, saying that one can neglect the time delay in the controller design and design the controller G_R so the system without timedelay, see (1.36), gives good performance. The true closed loop system would then be (1.38), i.e. the the good system, but delayed. There is however a problem for unstable systems. It can be seen that the control signal will become unbounded if G_0 is unstable.

In Section 1.6 dont try to understand the details.

In (1.16) variables x and y in the first plot are the two elements in u.

Chapter 2

Skip (2.65)-(2.66), since it gives a useless recursion. The right hand side describes terms that have not been calculated at previuos time instances.

(2.67)-(2.71) is actually the transforms corresponding to (2.73)-(2.74).

The order of matrices in (2.81) is not correct in the multivariable case. But dont bother we will focus on scalar systems in most of the course.

[1]I have not included a full list of all typos. If you find any major errors I have missed, please mail to me /BoB

Chapter 3

We will try to keep the needed theory for stochastic signals to a minimum. If you havent seen the contents of this chapter before, dont worry too much. If you can follow the calculations in Sec 3.4 or Example 8 you should be fine. It is possible to relate the spectrum defined as in (3.32) (the z-transform of the covariance function 3.28) to the spectrum you would get if you just did an FFT of the sequence y. So if the spectrum has a large magnitude the intution is that "there is usually a lot of energy in the stochastic signal at that frequency". You can skip Sec 3.9 for now. We will see if we need it later.

There is a dotted line in Fig 3.2 lower left figure which is almost invisible in my book. Look on the lecture slides instead where I do this example in matlab. One of the lines in Fig 3.3. looks strange. And it would have been better to use logarithmic scale. The point of the plot is to show that a=0.8 has more energy at low frequencies whereas a=0.2 is almost flat over all frequency (a=0 would give a flat spectrum as you see from 3.49)

In (3.72) last equation the last term should be $Q_{ve}(A^{-\tau-1})^T C^T$ and in (3.74) it should be $(A^{-\tau-1})^T$ instead of $(A^{-\tau+1})^T$. A simple way to see this is to change τ to $-\tau$ in the equations for positive τ, since we should have $C_{yy}(-\tau) = C_{yy}^T(\tau)$.

Chapter 4

Start by reading the Appendix pp 70-79! In the chapter v and w denotes the same thing. Skip the Kalman filter interpretation for now. Also skip sections 4.3-4.6. (The point with Sec 4.6 is to solve a linear equation system in n^2 steps instead of n^3 that normal Gaussian elimination would give. This was very useful in the early computer days, more rarely so today. If you ever want to use it, I think (4.87)-(4.91) needs to be checked. I dont agree with all the indices there).

In the appendix I dont think (4.129) corresponds well with figure 4.10.

Chapter 5

This chapter is about three things: Polynomial based design in general, optimal d-step ahead prediction, and its use in minimum (output) variance control. It is a very condensed version and it is not so easy to follow. It might be a good idea to also read the polynomial design handout from the course in real time systems (pp 76-82). It was handed out during the lecture.

In (5.13) one should not draw the conclusion that $B_m = BT$ and $A_m = RA + SB$. As described in the lecture there are often some pole-zero cancellations going on, which is the tough part to explain when one describes this design method. Here some explanations are given later on p99, and it might be good idea to read that at the same time as p85. The design example on p86 is difficult to follow, for different reasons. I think you can skip it and look on the one I presented at the lecture instead. Skip section 5.2.3 for now.

The formulas for the prediction $\hat{y}_{k+d|k}$ (the notation means that we want to predict the value of $y(k+d)$ given measurements up to time k) in Sec 5.3 are beautiful and useful, so this should be known. I found the calculations in Sec 5.3 repeated several times in the book (i.e. Sec 10.4, A.6, B.7) so you should have many opportunities to learn it :-) The calculations that need to be done are also very simple, it is only a polynomial division required. It is hence very fast to implement in practice. You can skip the upper half of p92, we will cover it later.

On p. 93 it is no restrictions to assume deg(A(z))=deg(C(z)) (one can add zero coefficients and/or re-index the noise sequence w) so this is often done. In (5.91) the last coefficient should of course be a b (I would like to put it equal to $b_{n-d}z^{n-d}$) and in (5.92) the coefficients should be c_1 to c_n. I have never liked minimum variance control myself, since it often requires large input signals and often gives a nonrobust controller design. A big warning sign should have been raised early in 5.4 saying that the closed loop system will only be stable if $B(z)$ is Schur (i.e. has all roots inside the unit circle, i.e. the system is minimum phase).

I dont understand the point of (5.111), skip it. In (5.113) it should say $\hat{y}_{k+2|k}$.

The split (5.128) is not unique. Often one therefore also requires e.g. B^+ to be monic. On p100, the motivation for why $deg(P) \geq 2deg(A)-1$ is strange. Moreover, this condition is sometimes not needed. For some P of lower order, the design equations are solvable, giving a causal controller (for instance $P = A$ and the controller $S = 0, R = 1$ can be used if we are happy with the open loop system). What one can say is that the equation $RA + SB = P$ is solvable for **any** P satisfying this degree condition.

Sec 5.3.3. seems to be the same as 5.2.2.

You can skip Sec 5.5.5 if you read the handout and understand how to introduce integral action in the controller.

Skip 5.6 for now, and only scan the two appendices.

Chapter 6

This chapter gives an overview over linear optimal control using a quadratic cost function, thus LQ control.

An LQ controller is a state feedback controller, where the feedback law is not determined by e.g. pole placement, but by minimization of a quadratic cost function. In the cost function as defined in equation (6.2), the aim is to punish both deviations of the state from the origin and too large control signals.

In section 6.3, the solution of the minimization problem in equation (6.2) is derived. The solution is a state-feedback law that you can see in equations (6.24) to (6.27) and also Table 6.1. The state-feedback gain L depends on the weight matrices that were chosen for the quadratic cost function in equation (6.2). In this case, the state-feedback gain is variant in time.

This solution is derived using the principle of optimality and dynamic programming. The principle of optimiality states that an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must be optimal with respect to the state resulting from the first decision. In dynamic programming the optimal solution is obtained backwards in time. First, an optimal policy is constructed for the last stage, i.e. final time N. Then, using the solution for time N, the optimal policy for time N-1 is determined, see eq. (6.9). This proceeds sequentially, solving all subproblems of a given length, i.e. a certain time step, by using the solution of one length shorter, i.e. one time step into the future, see eq. (6.19). This procedure called dynamic programming was introduced by Bellman.

The minimum of equation (6.11) is reached for $u_{N-1} + L_{N-1}x_{N-1} = 0$. From this results equations (6.15) to (6.18). Similar for equations (6.24) to (6.27).

In eq. (6.28), the Riccati equation (6.27) is rewritten to be used as a Lyapunov function. Using Lyapunov Theory, eq. (6.34) to (6.38) prove that the closed-loop system is stable. There is an appendix about Lyapunov theory, if you want more details.

The case when the feedback gain L as solution to the cost function (6.2) is time-invariant is given in equations (6.39) to (6.41). Using Lyapunov theory, also here it can be proven that the resulting closed-loop system is stable.

Equations (6.40)-(6.41) normally have many solutions (L, S). The one we want satisfies $S \geq 0$ and is such that $A - BL$ is stable.

The description in the chapter is sketchy and many mathematical details and technical conditions are left out. For a deeper study, there are special courses just about LQG.

Chapter 7

One dont remember all these formulas in this chapter. But the main idea is simple and worth remembering: If you want to estimate a variable x and you can measure a correlated variable y (assume both have zero mean) then the optimal estimate is given by $\hat{x} = C_{xy}C_{yy}^{-1}y$ (see 7.3-7.9).

In section (7.2.2) the formulas are only correct if $Q_{ve} = 0$. Table 7.1 gives $\hat{x}_{k|k-1}$ also for $Q_{ve} \neq 0$, but the Kalman filter with direct term $\hat{x}_{k|k}$ is more complicated if $Q_{ve} \neq 0$. Also in chapter 8 we will assume $Q_{ve} = 0$.

You can scan skip sections 7.3-7.5.

Chapter 8

Now we will put together the two previous chapters. So we add noise to the control problem and use the Kalman filters just derived. In section 8.3 it seems to be assumed that $y = x$. You can skip all this and go directly to section 8.4.

In (8.47) there should be a $\hat{x}_{k-1}$ instead of x_{k-1}, the same in (8.50). The ideas behind (8.47) is that $\tilde{x}_{k-1}$ is uncorrelated to $L\hat{x}_{k-1}$ (otherwise $\hat{x}_{k-1}$ would not be an optimal estimator). The same calculation is valid with and without direct term in the estimator.

The main design paramaters in LQG are the Q_1, Q_{12}, Q_2 weights and the covariance matrices Q_v and Q_e. It is usually rather easy to tune a controller using these parameters (easier than pole placement). The

Chapter 9

Equation (9.17) does not follow as a consequence of causality, as stated. There might be lower degree solutions that are causal. However, if (9.17) is satisfied one is guaranteed existence of a causal controller for any choice of coefficients of A^C. Many of the plots are hard to see, sorry for that. In Fig 9.5 the conclusion to draw is that the parameters converges almost instantly. In (9.46) one should proably also normalize so $G_m(1) = 1$.

In (9.53) the polynomial B_1^m satisfies $B^m = B^- B_1^m$.

In figure 155 it looks like the implementation cancels the full B polynomial. But the detailed analysis shows that only B^+ will be part of the closed loop polynomial.

In 9.4.1 it could be better explained that b_0 is unknown, but that this is not a problem since it is cancelled when calculating the control signal later (with a special choice of B_m and T).

In the MRAC algorithm it is not necessary to choose $degA_0 = d_0 - 1$. But if $degA_0 \geq d_0 - 1$ one is guaranteed a causal controller exists.

Skip the appendix.

Chapter 10

In this chapter we will add a noise model $C(q)e$. In (10.23) the e_{k+m} should be w_{k+m}. Section 10.4 you have seen before. (10.42) is not correct, it should say $A(z)F(z) + G(z) = z^{d-1}C(z)$. The point with Fig 10.4 is that the adaptive controller (MVAC) works just as well as the ideal controller (MV). The claims in 10.4.1 can be proved, but you do not need to think about how.

I dont agree with the statement on p.181 that the sensitivity function usually resembles a bandpass filter.

Chapter 11

You can skip this chapter, except scanning Example 27, which gives a typical application of Lyapunov theory based design of an adaptive controller. Here also some physical intuition about the energy functions for the robot is used to find a Lyapunov candidate function.

Chapter 12

Unfortunately, this chapter is hard to read because of the bad language. What I would like you to understand is 1) the idea of ILC described in Sec 12.1, 2) why the Q filter is introduced, 3) how (12.12) can be used to understand ILC convergence and 4) how to tune L and Q. Besides this, it is enough to study the lecture slides.

In (12.7) it should be $\omega h \in [-\pi, \pi]$.

The following can help if you decide to read Ch 12: It is not obvious that (12.7) gives ILC stability. What is meant with "stability" is that $e_k(t) \to 0$ for all t when $k \to \infty$, i.e. when we run the ILC iteration the entire error function goes to zero. From (12.6) and (12.7) it follows that $E_k(i\omega) \to 0$ for each frequency, and then one can use Parseval's formula to translate this to the time domain.

In 12.1.1 the formula for y_k assumes that it is the same transfer function T_c from y_d and u. The analysis of the more general situation with two different transfer functions follows the same lines, se (12.13). I dont know why the formula for J_{k+1} occurs on the middle of p 213, it is probably a cut-and paste error.

It turns out that the δ in (12.15) doesnt occur in the solution in (12.21-23), it is instead implicitly related to the choice of λ. A large λ will force $u_{k+1} - u_k$ to be small. The work envolved getting an algorithm of the form in 12.3.2 is much larger than the heuristic ILC algorithms. It can however pay off in some situations.

I think the heurisitc ILC algorithm in example 12.4 is rather strangely tuned. With a sample rate of 1 millisecond it seems wrong to use a filter of the form $L = kq$, i.e. look on the error only 1 millisecond in the future. A better solution would be to look roughly 20-50 milliseconds in the future, judging from the time constant in the plots. I also think figure 12.7 indicates some questionable high-frequency behavior in the initial 20 milliseconds or so (even though it is hard to see in the plots).

In 12.5 "bidimensional" means the same as "two-dimensional". In figure 12.16 it is hard to see the improvement, but please notice that the scales are different in the different plots.

In (12.36) b is the same as h, the liquid depth.

Chapter 13

Chapter 13 is rather brief on MPC, and you should therefore also scan the MPC Manual, which contains good descriptions of the MPC method and describes a small tool package to implement MPC. The notation is slightly different, but I hope this will not be a large problem. There are more ambitious MPC development platforms available, all major control equipment manufacturers have their own version, but this small tool is good enough for our purposes. MPC is rapidly becoming widely used in industry, some people say it will fill the role the PID has had for such a long time, being rather easy to tune and having better performance than pure PID. Typically a model is needed though, which can involve quite some work, so I am not so sure.

Notice that it is important that the resulting optimization problem turns out to be on a very nice form (Convex Quadratic Programming). This means that good algorithms with guaranteed convergence exist. Putting general optimization software with no guaranteed convergence and risk of getting stuck in local minima is not as popular.

I wouldn't call (13.1) "user parameters". In (13.5) and (13.8) there are some errors in the C and D matrices. The equations are obtained by assuming a state space model for the input noise d (A_d, B_d etc) and another for the output noise w (A_m, B_m etc)and putting everything into one big equation. I am not sure I got it right on the lecture slides either :-(

In (13.17) u_k^r denotes desired levels on input signals (if such are known). In 13.2.4 and (13.24) it is assumed $u_k^r = 0$.

You can skip reading 13.4.

App A

Some background on z-transform, and some repeated text from elsewhere in the book Scan it quickly.

App B

Some definitions in the beginning. Then much overlap with material in Ch3, Ch7 and AppA. Scan it quickly.

App C

You should know most of C.1., and C.2-C.4 is also useful, but we will only use a small part in the course. If there are some details you dont know, it proably dont matter. Definitly skip C.6-C.7.

App D

This is further studied in the course Nonlinear Control. I think it is enough reading definitions 51-54 and sections D.4-D.6. Skip D.5.1. On p 337, "singular point" means the same as "equilibrium point". In theorem 19 (D.34) one should change D to $D - \{0\}$.

App E

App F

App G

Skip

App H

Skip

App I

Skip

App J

Skip