19.4 Feedback control and data assimilation

The primary hypotheses that we relied on in the previous sections was on the linearization of the dynamics, quadratic cost, and the nature of the application of the control variable. We can in a sense, generalize LQR to the nonlinear setting if we only suppose that the control variable is introduced as a feedback term, i.e. we augment the dynamical system

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}), \tag{19.16}$$

with the control mechanism

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}) + B\mathbf{u}, \tag{19.17}$$

where B is a linear operator, and $\mathbf{u}$ is the selected control variable which is chosen to optimize some given cost functional, potentially under the relevant constraints. An even more specific setting is realized if the desired cost functional is specified such that the desired outcome is to drive the system $\mathbf{x}(t)$ to a given dynamical state $\tilde{\mathbf{x}}(t)$, i.e. we want to force the dynamics $\mathbf{x}(t) \to \tilde{\mathbf{x}}(t)$ as $t \to \infty$. We will avoid the details for now, but we suffice it to say that a solution to this problem is to specify the control so that the modified system becomes

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}) - K(\mathbf{x} - \tilde{\mathbf{x}}), \tag{19.18}$$

where the matrix K can be selected to push (often referred to as 'nudging') the solution toward the desired state $\tilde{\mathbf{x}}(t)$ along different directions in the phase space.

To be a bit more specific about this, consider the problem where you have the linear evolution equation (with potential control $\mathbf{u}(t)$)

$$\mathbf{x}' = A\mathbf{x} + B\mathbf{u},$$

where the observable part of the state is given by:

$$\mathbf{y} = C\mathbf{x},$$

that is to say that you can only see a part of the solution. The linear operator C is usually called the 'observation' operator, and $C\mathbf{x}$ refers to the part of the solution that can be seen or observed. Suppose hypothetically that your goal is to identify the control $\mathbf{u}(t)$ that will drive the full state $\mathbf{x}(t)$ to the origin, that is you want $\mathbf{x}(t) \to 0$. Nudging the solution toward $\tilde{\mathbf{x}}$ would only be a change of variables (so long as the system is linear). Unfortunately because you can't observe the full state the best you can come up with is to minimize the following cost functional

$$J[\mathbf{u}] = \frac{1}{2}\int_0^\infty \left[|C\mathbf{x}|^2 + \alpha|\mathbf{u}|^2\right] dt.$$

This is an LQR setup with optimal solution

$$\tilde{\mathbf{u}} = -\frac{1}{\alpha}B^T P\mathbf{x},$$

where

$$PA + A^T P + C^2 - \frac{1}{\alpha}PBB^T P = 0.$$

If we let $K = \frac{1}{\alpha}B^T P$ then this Riccatti equation becomes

$$K^2 - 2AB^{-T}K - \frac{1}{\alpha}C^2 = 0.$$

When we solve for this matrix K then the optimal solution will obey:

$$\mathbf{x}' = A\mathbf{x} - \frac{1}{\alpha}BK\mathbf{x},$$

where we choose the matrix B so that the eigenvalues of $A - BK$ have real part strictly less than zero indicating that the solution $\mathbf{x}(t) \to 0$ (so long as that is what we wanted to achieve). Not only is this closely related to LQR, but it looks an awful lot like a Kalman filter (no coincidence actually).

19.4.1 Data assimilation

We will now move beyond the linear setting and focus on something that is a bit more interesting, but will not be able to make any of the following completely rigorous. For starters, suppose that we somehow know exactly how weather evolves (as if), and that the weather system is described precisely by the dynamical system

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0, \tag{19.19}$$

where $\mathbf{x}_0$ is unknown. Basically we are saying that we have perfectly figured out all the physics that governs the weather, but we can't measure every variable that goes into weather prediction perfectly, so we don't know how to initialize our system. Instead we couple this equation with an operator denoting the observable part of the state:

$$\mathbf{y}(t) = H(\mathbf{x}(t)). \tag{19.20}$$

We suppose that H is known, i.e. we at least know how the observable data is computed from the full state. This is referred to as the data assimilation problem from here on out.

Our goal is to find a feedback control term that will 'nudge' the solution $\mathbf{x}(t)$ as close to the full true solution as possible. That is we suppose that $\mathbf{x}(t)$ is the exact solution to (19.19), and we are restricted to simulating the system

$$\tilde{\mathbf{x}}' = \mathbf{f}(\tilde{\mathbf{x}}) + B\mathbf{u}, \quad \tilde{\mathbf{x}}(0) = \tilde{\mathbf{x}}_0,$$

where we can choose B and $\mathbf{u}$ to get $\tilde{\mathbf{x}}$ to get close to $\mathbf{x}$. We can't minimize the difference between $\tilde{\mathbf{x}}$ and $\mathbf{x}$ however because $\mathbf{x}(t)$ isn't entirely available. Instead we may consider the optimal control problem:

$$\min_{\mathbf{u}} \frac{1}{2}\int_0^\infty \left[(\mathbf{y} - H(\tilde{\mathbf{x}}))^T (\mathbf{y} - H(\tilde{\mathbf{x}})) + \mathbf{u}^T R\mathbf{u}\right]^2 dt, \tag{19.21}$$

subject to

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{y}(t) = H(\mathbf{x}(t)), \tag{19.22}$$

$$\tilde{\mathbf{x}}' = \mathbf{f}(\tilde{\mathbf{x}}) + B\mathbf{u}, \quad \tilde{\mathbf{x}}(0) = \tilde{\mathbf{x}}_0. \tag{19.23}$$

Remark 19.4.1. In reality there is of course noise in both the model $\mathbf{f}(\mathbf{x})$ and the observational operator H so that in reality we would want to minimize both the difference between the observed state, and the predicted (simulated) state. In continuous variables this is a bit complicated, but in the fully linear case then the optimal solution is the Kalman filter, and in the nonlinear setting, the extended Kalman filter is a reasonable approximation of the optimal solution.

Rather than focus on the truly optimal solution, we will now turn our attention to a simplified type of feedback control mechanism for this setting that is not optimal in the sense we have been using, but is nevertheless extremely useful. From our experience with the Kalman filter and LQR, we may suppose that a reasonable solution to the data assimilation problem is to set

$$\tilde{\mathbf{x}}' = \mathbf{f}(\tilde{\mathbf{x}}) - \mu(H(\tilde{\mathbf{x}}) - \mathbf{y}), \tag{19.24}$$

where

$$\mathbf{y} = H(\mathbf{x})$$

is the observable part of the true solution $\mathbf{x}(t)$ which satisfies

$$\mathbf{x}' = \mathbf{f}(\mathbf{x}).$$

Example 19.4.2. To consider an interesting example, we will turn to the canonical and well-loved Lorenz equations:

$$\begin{aligned} x' &= \sigma(y - x), \\ y' &= x(\rho - z) - y, \\ z' &= xy - \beta z, \end{aligned}$$

where we will assume that $\rho = 28$, $\sigma = 10$ and $\beta = 8/3$. Suppose that the observation operator H allows us to only see $x(t)$ and $z(t)$ that is our 'nudged' system is given by:

$$\begin{aligned} \tilde{x}' &= \sigma(\tilde{y} - \tilde{x}) - \mu(\tilde{x} - x), \\ \tilde{y}' &= \tilde{x}(\rho - \tilde{z}) - \tilde{y}, \\ \tilde{z}' &= \tilde{x}\tilde{y} - \beta\tilde{z} - \mu(\tilde{z} - z), \end{aligned}$$

where $x(t), y(t)$, and $z(t)$ are assumed to come from the 'true' system. The real question then becomes, what happens to the state as $t \to \infty$? Do we recover the true solution (even $y(t)$) even though we only observe part of it? If we do recover the entire state, what values of μ are required? Do these change as the parameters (ρ, σ and β) change?

```python
import numpy as np
from matplotlib import pyplot as plt
from scipy.integrate import odeint, solve_ivp

def lorenz_nudge(t,vals,rho,sigma,beta,mu):
    # A solver for the 'nudged' data assimilation Lorenz '63 system

    #break out the dependent variables
    x = vals[0]
    y = vals[1]
    z = vals[2]
    #the 'p' refers to the assimilated variables of the same name
    xp = vals[3]
    yp = vals[4]
    zp = vals[5]

    #the right hand side of the original ODE
    dx = sigma*(y-x)
    dy = x*(rho-z)-y
    dz = x*y-beta*z

    #the right hand side of the nudged or data assimilated system
    dxp = sigma*(yp-xp) - mu*(xp-x)
    dyp = xp*(rho-zp)-yp
    dzp = xp*yp-beta*zp - mu*(zp-z)

    return np.array([dx,dy,dz,dxp,dyp,dzp])


#Now to compute the solution of the nudged system and compare against the 'truth'
rho=28
sigma=10
beta=8/3
mu=10
time = [0,100]
init_vals = [0.1, 0, 0,-0.1,1,1]

results = solve_ivp(lorenz_nudge, time, init_vals, args=(rho,sigma,beta,mu), dense_output=True)
```

Algorithm 19.3: Example code that simulates the 'true' solution of the Lorenz '63 model as well as a nudged solution where only x and z are observable.

19.11. Modify (19.21) to a finite time interval $[0, t_f]$, then use the standard methods we have used before in the Calculus of Variations to determine what the optimal control solution should be when coupled with the evolution equations given in (19.22) and (19.23).

19.12. Write up a data assimilation simulation code in scipy for the nonlinear pendulum where the observation operator H is reduced to either the angle of the pendulum $\phi(t)$ or its velocity $\phi'(t)$. Will the system truly synchroninze for either of these options? If so, what values of the nudging parameter μ do you need to specify? Use $g = 9.8$ as the gravitational constant. Initialize the true and data assimilated systems with different initial conditions. Does synchronization depend on the initial conditions?

Notes