MATH4406 (Control Theory)
Unit 1: Introduction to "Control"

Prepared by Yoni Nazarathy, August 11, 2014

Unit Outline

- The many faces of "Control Theory"
- A bit of jargon, history and applications
- Focus on inherently deterministic systems
- Briefly diving into:
  - The PID Controller
  - Linear Control Theory
  - Model Predictive Control

Units 2,3,4,5,6,7 – Inherently stochastic systems (Markov Decision Processes) - starting in third hour of today

Unit 8 – Open loop deterministic continuous optimal control (calculus of variations) – also taught in MATH3404 (called at UQ: optimization theory)

Unit 9 – Closure

Assessment (for units 1, 8, 9) : Only through course summary

What is "Control Theory"

Hard to find a precise definition. So here are some terms very strongly related:

- Dynamical systems and systems theory
- Linear vs. non-linear systems
- Inherently deterministic vs. inherently stochastic systems
- Open Loop vs. Closed Loop (Feedback)
- SISO vs. MIMO
- Trajectory tracking and/or Disturbance rejection
- Stabilization vs. fine tuning vs. optimization
- Adaptive control and learning
- Partial and/or noise observations
- Chemical systems, Electrical systems, Mechanical systems, Biological systems, Telecommunications, Robotics and Artificial Intelligence, Systems in logistics...

Who does "Control Theory"

- Engineering control - descriptive tools - simple SISO systems (e.g. PID controllers)
- Engineering control - supervisory control (sensors, displays, status, etc....)
- "Advanced engineering control" (state space) - planes, helicopters, etc... etc...
- Control theory research:
  1. Electrical, chemical, mechanical engineers
  2. Applied mathematics
  3. IEEE Control Systems Society (http://www.ieeecss.org)
  4. AUCC - Australian Control Conference (http://www.aucc.org.au/AUCC2014/)
  5. Operations Research, Applied Probability etc...
  6. In recent years: Optimization, real-time computation and control have converged

Inherently deterministic systems

A system (plant)

- *state*: $x \in \mathbb{R}^n$
- *input*: $u \in \mathbb{R}^m$
- *output*: $y \in \mathbb{R}^k$

The basic system model:

$$\begin{aligned} \dot{x}(t) &= f\big(x(t), u(t), t\big), \\ y(t) &= h\big(x(t), u(t), t\big). \end{aligned}$$

Or in *discrete time*,

$$\begin{aligned} x(n+1) &= f\big(x(n), u(n), n\big), \\ y(n) &= h\big(x(n), u(n), n\big). \end{aligned}$$

Open loop control

$$\begin{aligned} \dot{x}(t) &= f\big(x(t), u(t), t\big), \\ y(t) &= h\big(x(t), u(t), t\big). \end{aligned}$$

Choose a "good" input $u(t)$ from onset (say based on $x(0)$).
$y(t)$ is not relevant.

Open loop *optimal control* deals with choosing $u(t)$ as to,

$$\min \quad J\Big(\{x(t), u(t),\ 0 \leq t \leq T_{\text{max}}\}\Big),$$

subject to constraints on x and u.

E.g. Make a "thrust plan" for a spaceship

Closed loop (feedback) control

$$
\begin{aligned}
\dot{x}(t) &= f\big(x(t), u(t), t\big), \\
y(t) &= h\big(x(t), u(t), t\big).
\end{aligned}
$$

Choose (design) a controller (control law), $g(\cdot)$:

$$u(t) = g\big(y(t), t\big).$$

The design may be "optimal" w.r.t. some performance measure, or may be "sensible"

Combination of open loop and closed loop

Consider a jet flight from Brisbane to Los Angeles (approx $13:45$)

$$\dot{x}(t) = f\big(x(t), u(t), t\big)$$

Dynamical system modeling: What are possibilities for x? For u? Why is the problem time dependent?

Open loop methods can be used to set the best $u(\cdot)$. Call this a *reference* control. This also yields a *reference state*. We now have, $x_r(t)$ and $u_r(t)$ and we know that, if the reference is achieved then the output is,

$$y_r(t) = h\big(x_r(t), u_r(t), t\big).$$

We can now try to find a controller $g(\cdot)$,

$$u(t) = g\big(y(t), t, x_r, u_r\big),$$

such that $||x(t) - x_r(t)||$ "behaves well".

Combination of open loop and closed loop, cont.

The *trajectory tracking* problem of having $||x(t) - x_r(t)||$ "behave well" is essentially called the *regularization problem*. "Behaving well" stands for:

- Ideally 0
- Very important: $\lim_{t \to \infty} ||x(t) - x_r(t)|| = 0$ (stability)
- Also important: How $||x(t) - x_r(t)||$ is regulated to 0. E.g. fast/slow. E.g. with many osculations or not

Many (perhaps most) controllers are "regulators" and are not even associated with a trajectory, instead they try to regulate the system at a set point. E.g. cruise control in cars, control the rotational speed of a hard-disk ...

Optimal Trajectory Plans

Calculus of Variations and Pontryagin's Minimum Principle

Calculus of variations is a classic subject dealing with finding optimal functions. E.g., find a function, $u(\cdot)$ such that, $u(a) = a_u$, $u(b) = b_u$ and the following integral is kept minimal:

$$\int_a^b \sqrt{\frac{1 + u'(t)^2}{u'(t)}} dt.$$

Question: What is the best $u(\cdot)$ if we remove the denominator?

Pontryagin's Minimum Principle is a result allowing to find $u(\cdot)$ in the presence of constraints, generalising some results of the calculus of variations, yet suited to optimal control.

Stability

Lyapunov Stability Theory

Here we "leave" control for a minute and look at the *autonomous system*:

$$\dot{x}(t) = f\big(x(t)\big).$$

Note that this system may be the system after a regulating control law is applied.

An equilibrium point is a point x_0, such that $f(x_0) = 0$.

Lyapunov's stability results provide methods for verifying if equilibrium points are stable (in one of several senses).

One way is to linearize $f(\cdot)$ and see if the point is locally stable.

Another (stronger) way is to find a so called *Lyapunov function* a.k.a. *summarizing function* or *energy function*.

Back to a bird-eye's view on the field

Key Developers of (Highly Used) Ideas and Methods

Up to 60's:

- E. J. Routh
- A. M. Lyapunov
- H. Nyquist
- W. R. Evans
- R. Bellman
- L. S. Pontryagin
- R. E. Kalman

- The developers of Model Predictive Control (MPC) - Still active today

Implementations

- Mechanical
- Analog
- Digital

The Basic Feedback Loop

- plant
- controller
- actuators
- sensors

Book: "Feedback and Control for Everyone", Pedro Albertos, Iven Mareels (2010). Avail on-line at UQ library.

Some LTI, SISO, Plant Examples

Car Driving Straight (Newton's law: $F = ma$)

- F - Force
- m - Mass
- a - Acceleration

Assume:

- Rotational inertia of the wheels is negligible.
- Friction retarding the motion of the car is proportional to the car's speed with constant β (in practice it may be proportional to speed squared). If $x(t)$ is location:

$$u(t) - \beta \dot{x}(t) = m\ddot{x}(t)$$

Set $y(t) = \dot{x}(t)$,

$$\dot{y}(t) + \frac{\beta}{m} y(t) = \frac{1}{m} u(t).$$

$$H(s) = \frac{Y(s)}{U(s)} = \frac{m^{-1}}{s + \beta m^{-1}}.$$

Pendulum

Assume:

- $\theta(t)$ is angle relative to hanging down position.
- Mass m, Length ℓ, Gravity g
- Torque with direction of θ, $u(t)$

$$u(t) - mg\ell \sin\theta(t) = m\ell^2\ddot{\theta}(t)$$

This is non-linear, yet for $\theta = 0$, $\sin\theta \approx \theta$ So we get,

$$\ddot{y}(t) + \frac{g}{\ell}y(t) = \frac{1}{m\ell^2}u(t)$$

set $\omega_n = \sqrt{g/\ell}$,

$$H(s) = \frac{Y(s)}{U(s)} = \frac{m^{-1}\ell^{-2}}{s^2 + \omega_n^2}.$$

Stability criteria

Given, $H_c(s)$ or $H(s) = \frac{N(s)}{D(s)}$ the standard way to check for stability is to solve,

$$D(s) = 0,$$

and see all solutions are in the LHP.

Routh-Hurwitz is an alternative (today still good for analytic purposes).

Another approach is Nyquist's Stability Criterion.

Profiling the Step Response

Specification for type-1 regulators with respect to "change of reference point":

$$r(t) = \mathbf{1}(t).$$

If (controlled system) is BIBO then, $\lim_{t\to\infty} y(t) = 1$. But how does it get there?

- **Rise time** - The time it takes the system to reach the "vicinity" of the new point: $t_r = \inf\{t : y(t) = 0.9\}$.
- **settling time** - The time it takes the transients to "decay": $t_s = \inf\{t : |y(\tau) - 1| \leq 0.01,\ \forall \tau > t\}$.
- **overshoot** - The maximum amount the system overshoots its final value divided by its final value. If it exists: $M_p = \max\{y(t)\}$.
- **peak time** - The time it takes to reach the maximum overshoot, $t_p = \inf\{t : y(t) = M_p\}$.

There is often a tradeoff between low M_p and low t_r.

Basic Engineering Control (LTI, input-output, SISO)

Signals and Systems

- Essentials of "Signals and Systems" (no state-space yet)
- Convolutions, integral transforms (Laplace, Fourier, Z...) and generalized functions (δ)

Consider a system with plant transfer function, $H(s)$, and negative feedback controller, $G(s)$. The transfer function of the whole system is:

$$Q(s) = \frac{H(s)}{1 + G(s)H(s)}.$$

Elements of Classic Engineering Control

The main object is,

$$Q(s) = \frac{H(s)}{1 + G(s)H(s)}.$$

- The “objectives” in controller (regulator) design
- PID Controllers (Proportional, Integral, Derivative)
- The Nyquist Stability Criterion

Goals in Designing G

- **Stability**:$H_c(s)$ should be stable system
- **Regulation** (for $R(s) = 0$): $E(s)$ small. Properties of the disturbances $W(s)$ and $V(s)$ can be taken into consideration
- **Tracking** (for $R(s) \neq 0$): For "desired" references, $R(s)$, the error $E(s)$ should be "small"
- **Robustness**: Model error of the plant, e.g. the plant $G'(s) = G(s)\big(1 + \delta(s)\big)$ should still be controlled well.
- **Simplicity**: Often a three parameter PID controller (or even simpler) "does the job"
- **Practicality**: Staying within dynamic limits, not using too many components, etc...

Controlling for Stability and Robustness

In case $H(s)$ is not stable. A first goal in designing $G(s)$ is to achieve stability of

$$H_c(s) := \frac{H(s)}{1 + H(s)G(s)}.$$

It is further important from the view point of disturbances and robustness to have good *stability margins*. Common are:

- *gain margin*
- *phase margin*

The PID (Proportional – Integral – Derivative) Controller

Parameterized by k_P, k_I and k_D

Controller Transfer Function:

$$G(s) = k_P + k_I \frac{1}{s} + k_D s$$

Closed Loop System Transfer Function:

$$H_c(s) = \frac{H(s)}{1 + (k_P + k_I \frac{1}{s} + k_D s) H(s)}$$

The "tunining game": Given $H(\cdot)$ set the three coefficients such that $H_c(\cdot)$ is as you wish.

Linear State Space Systems

Linear State-Space Systems and Control

The main object of study is,

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t).\end{aligned}$$

Together with the discrete time analog.

- When can such system be well controlled (e.g. stabilized)?
- Linear state feedback controllers are of the form $u(t) = Ky(t)$ for some matrix K
- Is there a way to estimate x based on y? The Luenberger observer
- The Kalman decomposition

This was a major area of study in the course, two years ago.

Systems with Stochastic Noise

The Kalman filter deals with the following type of model:

$$\begin{aligned} x(n+1) &= Ax(n) + Bu(n) + \xi_1(n), \\ y(n) &= Cx(n) + Du(n) + \xi_2(n), \end{aligned}$$

where ξ_i are *noise processes*, typically taken to be Gaussian.

The Kalman filter is a way to optimally reconstruct x based on y. It is a key ingredient in many tracking, navigation and signal processing applications.

Related is the Linear Quadratic Gaussian Regulator (LQG) which generalises the LQR by incorporating noise processes as in the Kalman filter.

Controllability and Observability

These are regularity properties:

The system is *controllable* if there exists a $u(\cdot)$ that can drive the state anywhere desired in finite time. This is known to be equivalent to a rank property of A and B.

The system is *observable* if given the observation $y(\cdot)$ over some finite time, it is possible to reconstruct the initial state, $x(0)$.

State Feedback Controller

With $u(t) = Kx(t)$, we get,

$$\dot{x}(t) = (A + BK)x(t).$$

If system is controllable, there exists K that sets eigenvalues of $A + BK$ as we wish.

Luenberger Observer

Make a "mock system" based on input, u and observed output, y.

$$\begin{aligned}\dot{\hat{x}}(t) &= A\hat{x}(t) + Bu(t) - K\big(C\hat{x}(t) + Du(t) - y(t)\big) \\ &= (A - KC)\hat{x}(t) + \text{stuff}\end{aligned}$$

Look at $e(t) = x(t) - \hat{x}(t)$. We get,

$$\dot{e}(t) = (A - KC)e(t).$$

If system is observable there exists K that sets eigenvalues of $A - KC$ as we wish.

Optimal Control in Linear State Space Systems

LQR

- The Linear Quadratic Regulator (LQR) problem deals with fully observed linear dynamics: $\dot{x}(t) = Ax(t) + Bu(t)$, and can be posed as follows:

$$\min_{u(\cdot)} \; x(T)'Q_f x(T) + \int_0^T x(t)'Qx(t) + u(t)'Ru(t)'dt$$

- It is beautiful that the solution is of the form $u(t) = Kx(t)$ (linear state feedback) where the matrix K can be found by what is called a Ricatti equation

Dynamic Programming can be used for LQR

Bellman's principle of optimality can be briefly described as follows:

If x-y-w-z is the "optimal" path from x to z then y-w-z is the optimal path from y to z.

This works for systems of the form $\dot{x}(t) = f\big(x(t), u(t)\big)$ and a cost function:

$$J\big(x(t_0), t_0\big) = \min_u \int_{t_0}^{T} C\big(x(s), u(s)\big)\, ds + D\big(x(T)\big).$$

Bellman's optimality principle implies that the optimal cost satisfies:

$$J^*\big(x(t), t\big) = \min_u C\big(x(t+dt), u(t+dt)\big)\, dt + J^*\big(x(t+dt), t+dt\big).$$

This leads to the Hamilton-Jacobi-Bellman PDE.

The “L” part of LQR

The same old **linear** dynamics:

$$\dot{x}(t) = Ax(t) + Bu(t), \quad \text{or} \quad x(k+1) = Ax(k) + Bu(k),$$

$$A \in \mathbb{R}^{n\times n},\ B \in \mathbb{R}^{n\times m},\ x(0) = x_0.$$

Assume,

$$\text{rank}\Big(con(A,B) := \big[B, AB, A^2B, \ldots, A^{n-1}B\big]\Big) \quad = \quad n,$$

i.e., the system is controllable (reachable)

So in the continuous time case, we can drive the state from x_0 to any state in any finite time, T. For the discrete time case it can be done in at most n steps.

The "Q" part of LQR

A cost structure:

$$J(u) = \int_0^T \Big(x(t)' \, Q \, x(t) + u(t)' \, R \, u(t) \Big) dt + x(T)' \, Q_f \, x(T),$$

or,

$$J(u) = \sum_{k=0}^{N-1} \Big(x(k)' \, Q \, x(k) + u(k)' \, R \, u(k) \Big) + x(N)' \, Q_f \, x(N).$$

The time horizon, T or N, can be finite or infinite.

The cost matrices satisfy,

$$Q = Q' \geq 0, \quad Q_f = Q_f' > 0, \quad R = R' > 0.$$

The "R" part of LQR

Since cost structure is,

$$J(u) = \int_0^T \Big(x(t)'\, Q\, x(t) + u(t)'\, R\, u(t) \Big) dt + x(T)'\, Q_f\, x(T),$$

or similar for discrete time, we see that we are trying to find a control $u(t), t \in [0, T]$ that will "regulate" the system "at 0". The payment is "quadratic" for both "state" and "control effort".

Typical choices for Q (or Q_f) are,

$$Q = \mathbf{1}\mathbf{1}' \quad \text{or} \quad Q = I \quad \text{or} \quad Q = \text{diag}(q_i),$$

where $q_i \geq 0$.

A typical choice for R is $R = \text{diag}(r_i)$, with $r_i > 0$.

The time horizon, T or N is often taken as ∞.

The LQR Success Story

It turns out that the optimal control is a linear state feedback control law. In the continuous time case,

$$u(t) = \big(-R^{-1}B'P(t)\big)x(t),$$

where the $n \times n$ matrix $P(t)$ is the solution of a Riccati differential equation.

In the discrete time case,

$$u(k) = \Big(-\big(R + B'P(k+1)B\big)^{-1}B'P(k+1)A\Big)x(k),$$

where the $n \times n$ matrix $P(k)$ is the solution of a Riccati difference equation.

Further if $T = \infty$ (or $N = \infty$) the terms $P(t)$ (or $P(k)$) are replaced by a constant matrix that is a solution of associated Riccati algebraic equations (different versions for discrete and continuous time).

The Riccati Equation - Continuous Time

This is the Riccati matrix differential equation used to find the state feedback control law of continuous time LQR. Solve it for $\{P(t),\ t \in [0, T]\}$

$$-\dot{P}(t) = A'P(t) + P(t)A - P(t)BR^{-1}B'P(t) + Q, \quad P(T) = Q_f.$$

Observe that it is specified "backward in time".

If $T = \infty$ the steady state solution P of the Riccati differential equation replaces $P(t)$ in the optimal control law. This P is the unique positive definite solution of the algebraic Riccati equation,

$$0 = A'P + PA - PBR^{-1}B'P + Q.$$

The optimal control is:

$$u(t) = \big(-R^{-1}B'P(t)\big)x(t) \quad \text{or} \quad u(t) = \big(-R^{-1}B'P\big)x(t).$$

The Riccati Equation - Discrete Time

This is the Riccati matrix- difference equation. Solve it for $\{P(k),\ k \in \{0, \dots, N\}\}$

$$P(k) = Q + A'P(k+1)A - A'P(k+1)B\big(R + B'P(k+1)B\big)^{-1}B'P(k+1)A,$$
$$P(N) = Q_f.$$

If $N = \infty$ the steady state solution P replaces $P(k)$. This P is the unique positive define solution found by the algebraic Riccati equation,

$$P = Q + A'PA - A'PB(R + B'PB)^{-1}B'PA.$$

The optimal control is:

$$u(k) = \Big(-\big(R+B'P(k+1)B\big)^{-1}B'P(k+1)A\Big)x(k), \text{ or } \quad u(k) = \Big(-\big(R+B'PB\big)^{-1}B'PA\Big)x(k).$$

LQR in MATLAB

Very simple:

$$[K, S, e] = lqr(SYS, Q, R, N)$$

N is an additional type of cost term,

$$2x(t)'Nu(t).$$

The return values:

$-K$ is the state feedback gain matrix.

S is the solution of the algebraic Riccati equation

e are the resulting closed loop eigenvalues (i.e. the eigenvalues of $A - BK$).

In practice this is often the preferred way of deriving an initial controller before making finer refinements (based on simulations and tests).

Model Predictive Control

MPC Overview

Model Predictive Control (MPC), also called "receding horizon control", works as follows:
For a plant modeled as, $x(k+1) = f\big(x(k), u(k)\big)$ an input,

$$u_{\cdot|k} = \Big(u(k|k), u(k+1|k), \dots, u(k+N-1|k)\Big)$$

is determined at each time slot k based on $x(k)$. The input is selected as to minimize predicted costs over the "planning horizon" $k, k+1, \dots, k+N$. Here N is the length of the planning horizon. Once $u_{|k}$ is determined, the control $u(k|k)$ is applied and at time $k+1$ the process is repeated.

For the calculation made during time k, denote the "predicted state" (due to $u_{\cdot|k}$) by $x(k+1|k), x(k+2|k), \dots, x(k+N|k)$. Observe that in general if $N < \infty$, $u(k+1|k) \neq u(k+1|k+1)$ even though (without disturbances/noise),

$$x(k+1) = f\Big(x(k), u(k|k)\Big) = x(k+1|k).$$

MPC Notes

Model Predictive Control (MPC) is a sub-optimal control method that "makes sense". If you think about it, this is in a sense how we (individuals) sometimes make decisions.

It originated from the chemical process control industry in the 80's. There each time step is in the order of a few hours. With the advent of cheap fast computers - it is now often the method of choice for real-time controllers also (e.g. time step every 10 milliseconds). The challenge is to solve the optimization problem for $u_{\cdot|k}$ quickly.

It is not always the case that increasing the time horizon "N" is better.

The stability of systems controlled by MPC is in generality not trivial.

Linear Quadratic MPC

Model:

$$x(k+1) = Ax(k) + Bu(k).$$

Cost:

$$J(u) = \sum_{k=0}^{N-1} x(k)' \, Q \, x(k) + u'(k) \, R \, u(k) + x'(N) \, Q_f \, x(N).$$

so far... exactly LQR. But... add constraints:

$$F \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \leq b.$$

In practice there are often hard constraints on the state and the control. Hence linear quadratic MPC is in practice very useful.

Formulation as a Quadratic Program (QP)

At time k (taken to be 0 for simplicity), given a measured (or estimated) state $x(k)$ we need to solve,

$$min_{u(0),u(1),\ldots,u(N-1)} \sum_{k=0}^{N-1} x(k)' \, Q \, x(k) + u'(k) \, R \, u(k) + x'(N) \, Q_f \, x(N)$$

s.t. $x(k+1) = Ax(k) + Bu(k)$ and,

$$F \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} \leq b.$$

How can we pose this as an optimization problem (over finite vectors) just in the mN variables $u(0), \ldots, u(N-1)$?

Formulation as a Quadratic Program (QP)

Use

$$\begin{bmatrix} x(1) \\ x(2) \\ \vdots \\ x(N) \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x(0) + \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & & \vdots \\ \vdots & & \ddots & \\ A^{N-1}B & & \cdots & B \end{bmatrix} \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N-1) \end{bmatrix}$$

This converts the optimization problem of the MPC controller to one that simply depends on the mN dimensional vector $u(0), \ldots, u(N-1)$.

The general form of a quadratic program (QP) is:

$$\begin{aligned} & min_z z' \tilde{Q} z + \tilde{P} z, \\ s.t. \qquad & \tilde{F} z \leq \tilde{b}. \end{aligned}$$

With a bit of (tedious rearranging) the MPC controller can then be presented as a convex QP in mN decision variables. QPs where $\tilde{Q} > 0$ have a unique solution and are quite efficiently solvable!!!

The Closed Loop System is Non-Linear

MPC generates a “feedback” control law $u(k) = g\big(x(k)\big)$, where the function $g(\cdot)$ is implicitly defined by the unique solution of the QP. The resulting controlled system,

$$x(k+1) = Ax(k) + Bg\big(x(k)\big),$$

is in general non-linear (it is linear if there are no-constraints because then the problem is simply LQR).

Stability of MPC

A system controlled by MPC is generally not guaranteed to be stable.

It is thus important to see how to “modify” the optimization problem in the controller so that the resulting system is stable.

One such method based on adding an “end-point constraint” that forces the optimized $u_{\cdot|k}$ to drive the predicted system to state 0.

Our proof is for linear-quadratic MPC, yet this type of result exists for general MPC applied to non-linear systems.

Generalizations of the “end-point constraint method” also exist.

Other paradigms

Non-linear (hybrid) Control

Hybrid dynamical systems have a continuous component $x(t)$ evolving in Euclidean space but also a discrete component $m(t)$ evolving on discrete set. Informally, for a given "mode" $m(t) = m$, $x(t)$ evolves according to the standard (say linear) dynamics that we know driven by A_m that depends on the mode:

$$\dot{x} = A_m x.$$

Then at the first time instance at which $x(t)$ reaches one of several sets, say $\mathcal{G}_{m'}$, the mode changes to m' and hence the dynamics change to,

$$\dot{x} = A_{m'} x.$$

E.g. a standard thermostat....

Here also, the same control questions exists (and have been partly answered): Stability, Controllability, Observability, Feedback control, State Estimation and optimal control.

Adaptive Control

Here the story (a very common one in practice) is the fact that the exact values of the plant parameters, say (A, B) are not known. Hence the parameters need to be estimated while the system is controlled (as opposed to off-line). In fact, some adaptive control techniques do **not** try to estimate the parameters, but simply try to control the system in an adaptive manner matching desired output to observed output and calibrating the control law on the go.

The theory is quite well developed, yet is advanced since typically **linear** plants controlled by adaptive controllers yield a **non-linear** systems.

Robust Control

Here the story is somewhat similar to adaptive control – there is plant uncertainty. Yet as opposed to developing a controller that tries to learn the plant, a controller is designed for the "worst case".

E.g. take an (A, B, C, D) system and assume that the actual A is $A + \delta G$ where G is some other matrix and δ is a scalar that is not too big.

A main theme is then to design a controller that ensures certain behavior (e.g. stability, optimality etc...) for a given range of δ.

Supervisory Control

This field uses a complete different set of tools: Computer science and discrete mathematics. The idea is to control discrete event systems with complicated (yet typically finite) state spaces. Think for example of a complicated photo-copier machine.

There are certain scientific questions dealing with state-reduction, computability and equivalent systems.

Moving onto inherently stochastic systems

Control of inherently stochastic systems

The system

$$\dot{x} = Ax + \xi_x,$$

is inherently deterministic (e.g. A is modeled from Newton's laws) yet is subject to random disturbances.

Other systems arising in telecommunications, population models and logistics are well modeled as inherently stochastic systems (Markov Chains).

The field of Markov Decision Processes deals with finding optimal feedback laws for such systems – yet the problem is often with computation (curse of dimensionality).

Approximate dynamic programming for such systems is currently a hot research topic. Another related topic is stability analysis of such systems.

Control of stochastic queueing networks