Introduction to Tracking Filters

Stanford University
April 28, 2009

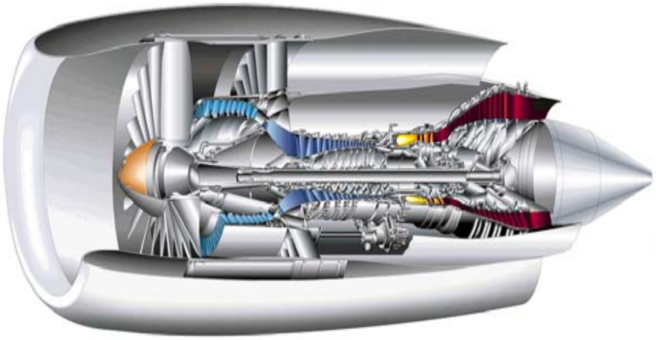
Shreeder Adibhatla, Ph.D., P.E.
Principal Engineer
Controls Center of Excellence
GE Aviation

imagination at work

Overview

imagination at work

Jet Engines 101



Jet engine has inlet, fan, optional booster, compressor, HP turbine, LP turbine, and nozzle

LP turbine drives the fan, and is referred to as the "LP spool" with rotor speed N1

HP turbine drives the compressor, and is referred to as the "HP spool" with rotor speed N2

Typical high-fidelity gas-path performance models of engines are 1D models

Modeling is performed by executing each module, inlet to exhaust

Component characteristics are captured using "maps"

Iteration schemes are used to enforce mass and energy balance

Models usually represent a nominal or average new engine

Models have adjustments or modifiers to tweak component characteristics

Fan, booster, compressor, HPT, and LPT usually have two modifiers each, for flow and efficiency

Can also have other modifiers to correct for cooling flows, pressure losses, discharge coeffs., etc.

These modifiers are used to match a model to the actual engine

The Need for PHM

PHM = Prognostics and Health Management

- From reactive to predictive
- From component-level diagnostics to system and fleet level management
- From time-based to condition-based maintenance

What are the benefits of PHM?

- Improved availability (readiness)
- Fewer delays & cancellations, aborted takeoff, and in-flight shut downs
- Better fault detection and isolation to reduce line maintenance costs
- Higher time-on-wing
- Lower repair costs

Reduce Cost of Ownership or Life Cycle Cost

Tracking Filter Design

Problem Statement

Given Sensor Information:

- Actuator positions
- Inlet sensors and aircraft parameters
- Engine sensors (speeds, temperatures, pressures)

Estimate Parameters:

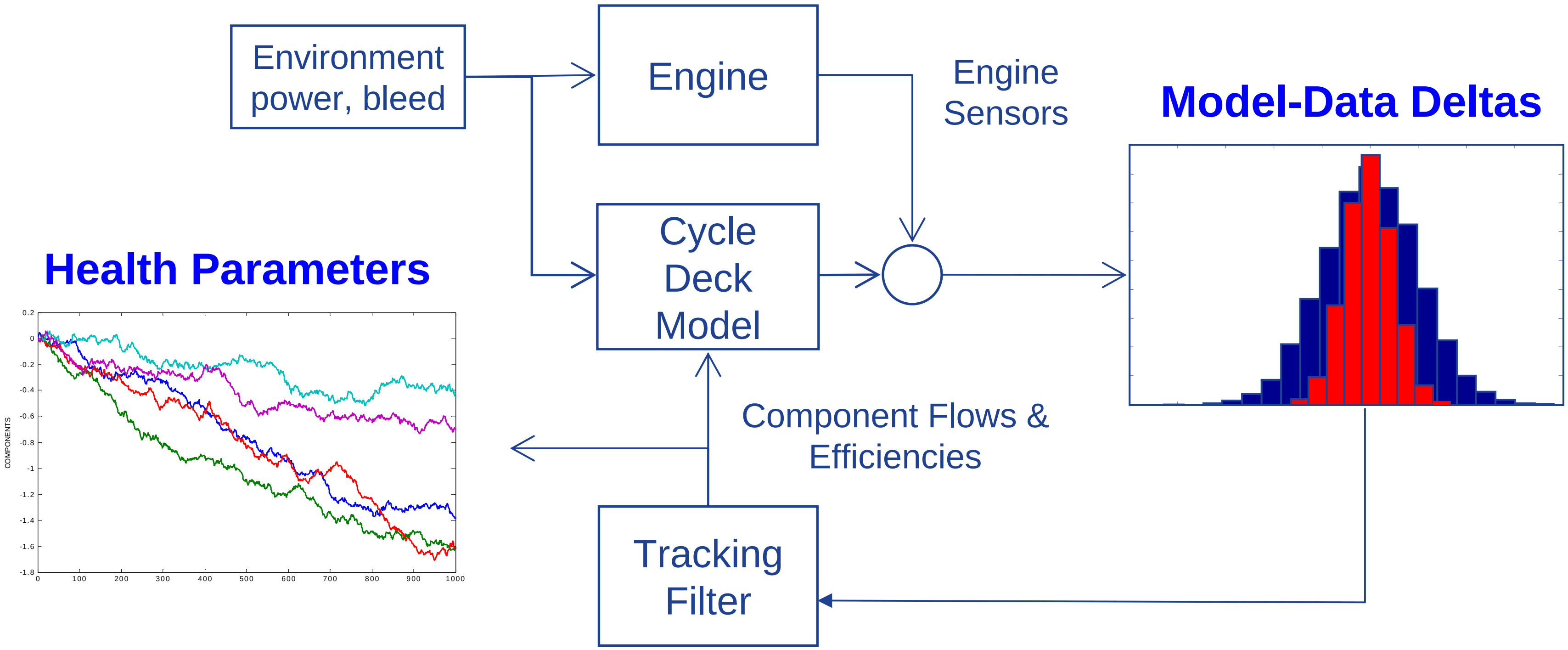
- Unmeasurable parameters (for model-based control)
- Sensed values (for use as virtual sensors)
- Actuator and sensor biases (for control and fault detection)
- Engine health parameters (for diagnostics)

In the Presence of Uncertainties:

- Engine-to-engine variations
- Deterioration
- Sensor biases and lags

Tracking Filter Concept – Parameter Estimation

Wrap an estimator around the engine model to personalize it to each engine



Basics

Sensor = f(Engine Health, Operating Point)

Nonlinear Model:

$$y = f(x(t), u(t), p) \quad [x = states, u = inputs, \; p = parameters]$$

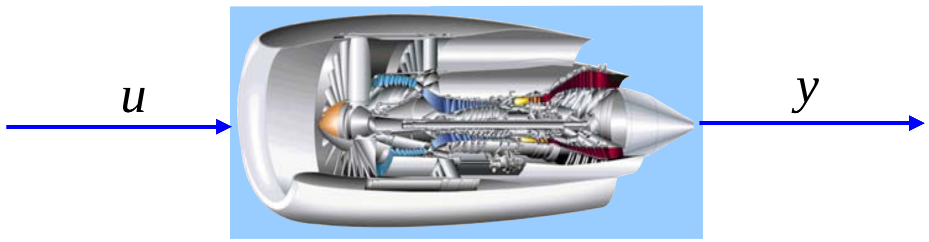
Linear Model:

$$\begin{aligned} \dot{x} &= Ax + Bu + Ep \\ y &= Cx + Du + Fp \end{aligned}$$

Observer

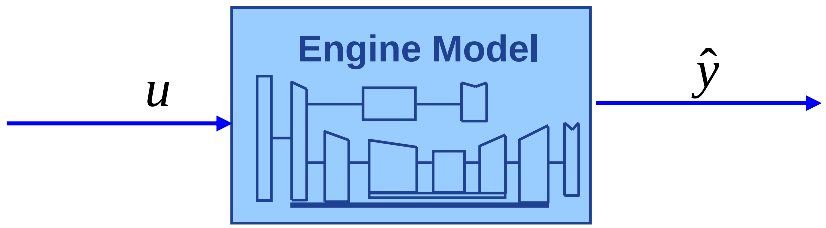
Plant

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



Open – Loop Observer ("Model")

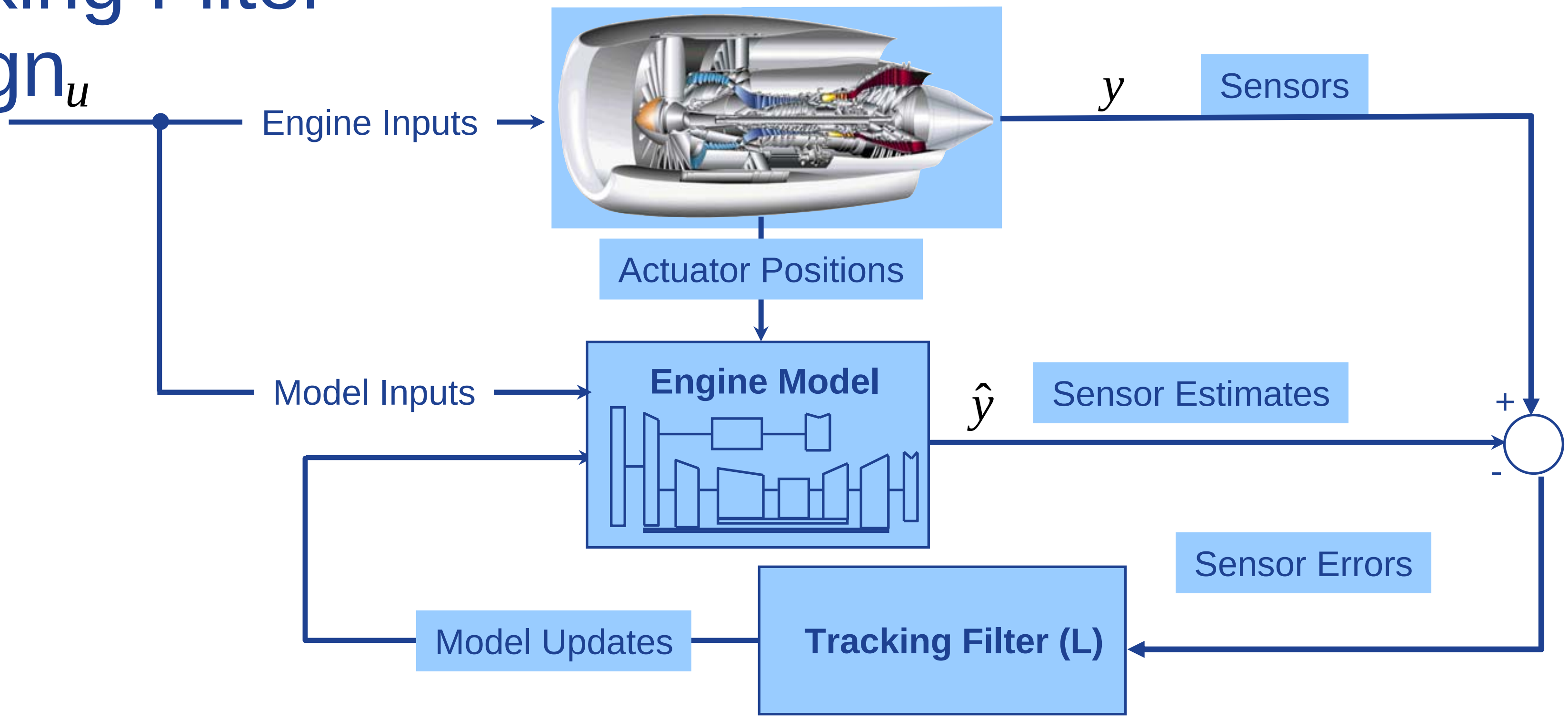
$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu \\ \hat{y} &= C\hat{x} + Du \end{aligned}$$



Closed – Loop Observer (Model + Output Tracking)

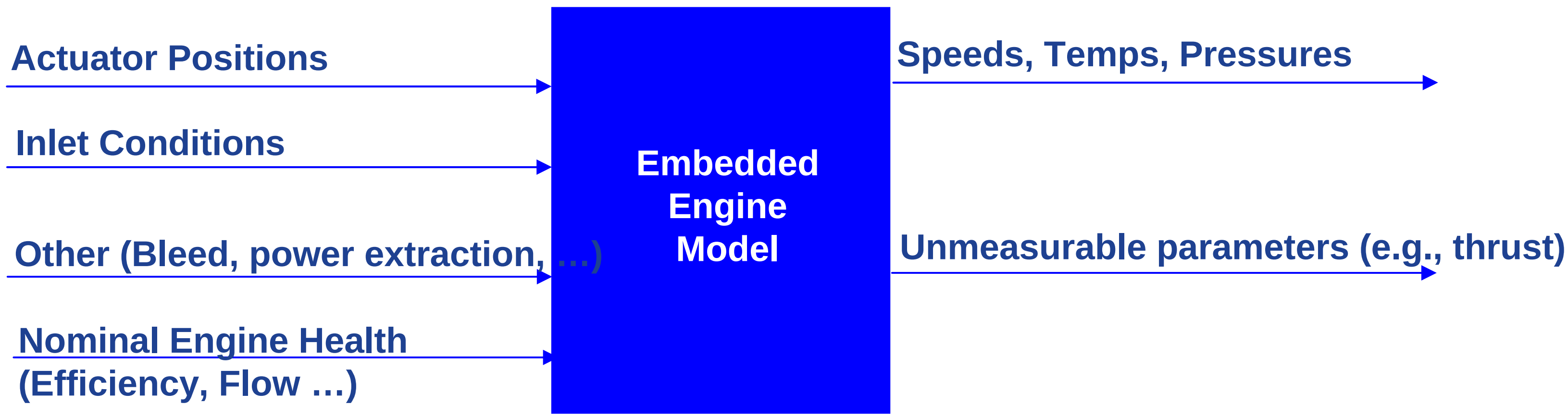
$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + L(y - \hat{y}) \\ \hat{y} &= C\hat{x} + Du \end{aligned}$$

Tracking Filter Design

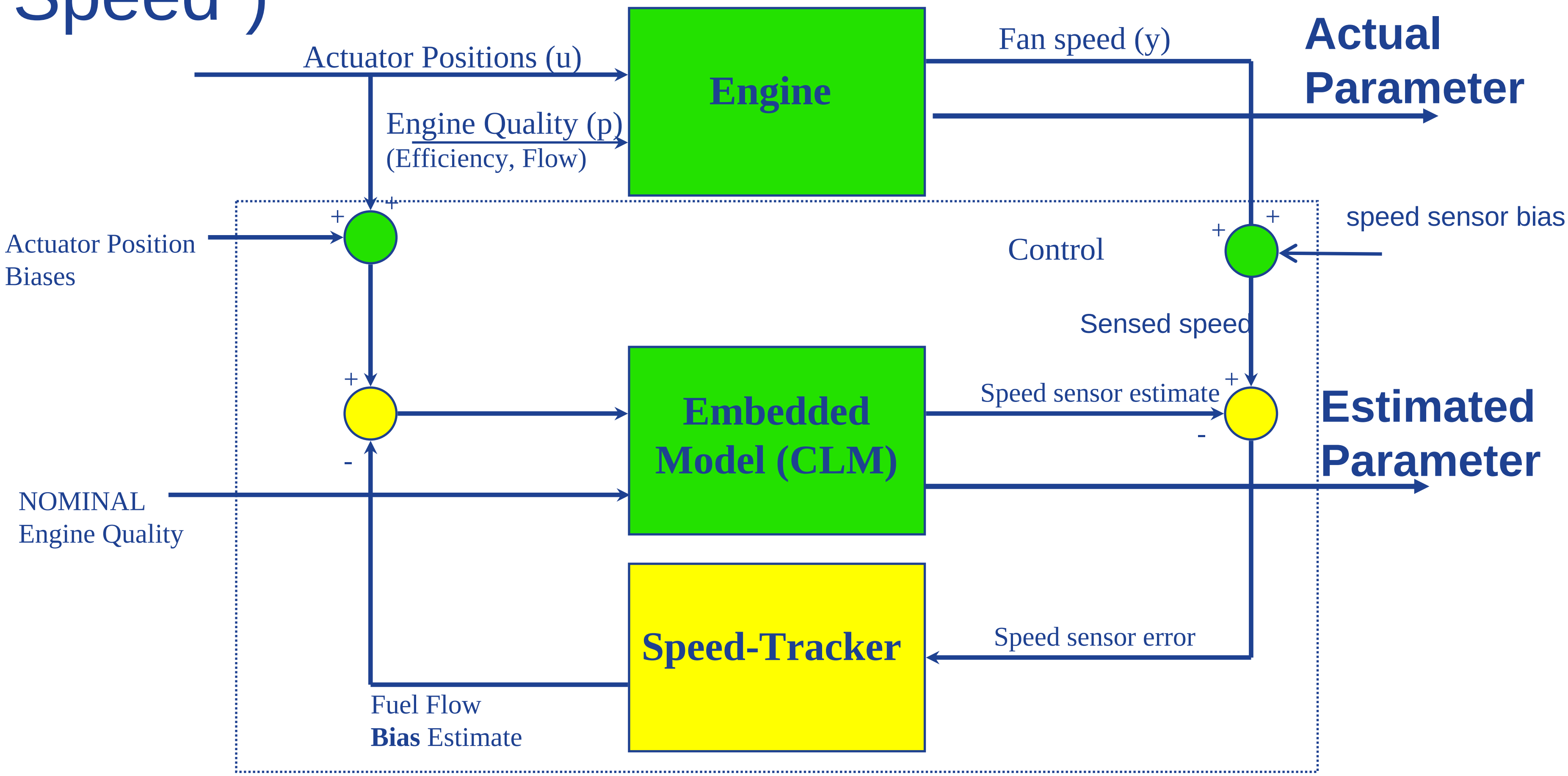


- **Provides a means of tuning model to match engine**
- **Accounts for**
  - Engine-to-engine variations
  - Deterioration
  - Sensor biases
- **Types of tracking filters**
  - Classical observer
  - Inverse Jacobian tracking filter
  - Least-squares tracker
  - Kalman Filter (KF) [Optimal Observer] ← Check out Wikipedia! Or Gelb
- **Designed so that sensor failures reduce effectiveness gradually**

No Tracking ("Run to Fuel Flow")



Fan-Speed Tracker: 1x1 TF ("Run to Fan Speed")



"Obvious" Extension:
2x2 TF that tracks fan & core speed by tweaking fuel flow and a health parameter

“Inverse Jacobian” Tracking Filter

Special case of observer gain “L”

Re-cast problem with parameters as inputs rather than states

Linear Model:

$$\begin{aligned} \dot{x} &= Ax + Bu + Ep \\ y &= Cx + Du + Fp \end{aligned}$$

Steady-State (DC Gain):

$$y = G_1 u + G_2 p$$

[G = Gain Matrix or Jacobian Matrix]

Since we want to know: “what value of dp will lead to dy=0?” we solve for:

$$dp = (G_2)^{-1} dy = L dy$$

$$\Rightarrow L = (G_2)^{-1}$$

imagination at work

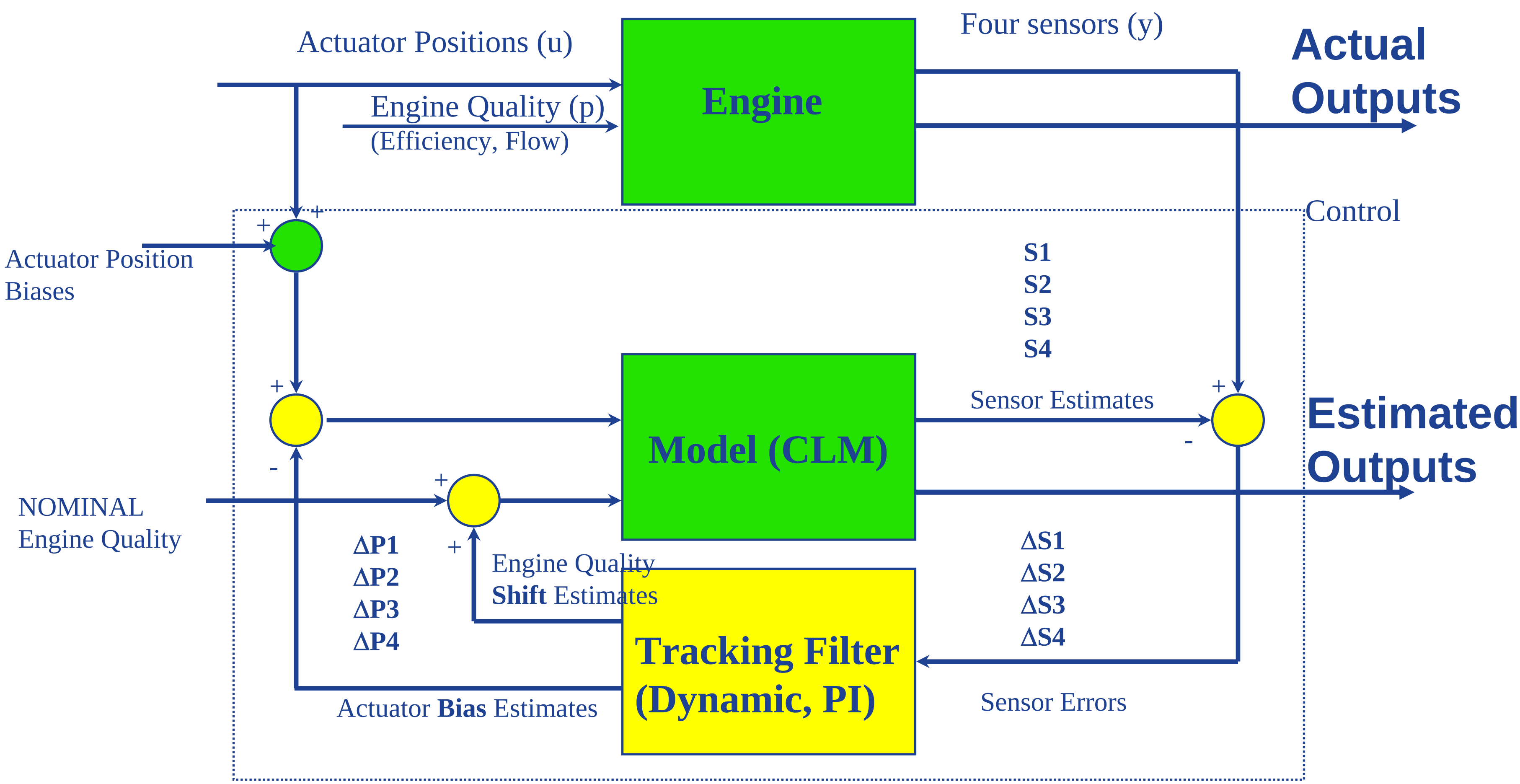
Simple Least Squares Tracking Filter

Problem:
Want to solve for 15 unknowns (typical)
Using 7 sensors (typical)

Solution 1 (nxn TF):
- Choose 7 of the most important parameters, and solve
  $dp = (G_2)^{-1} dy$
  Shortcoming: Lumping many effects into few

Solution 2 (Non-square TF):
- Solve $dp = pinv(G_2) dy$
  Shortcoming: Solution not physically meaningful
- Could use *weighted* least-squares

4x4 Dynamic TF



4x4 PI tracking filter designed using regular MIMO control design tools

imagination at work

Benchmark Problem
Design of a Turbine-Engine Health Estimator

imagination at work

Linear Model

$$\dot{x} = Ax + Bu$$
$$y = Cx + Du$$

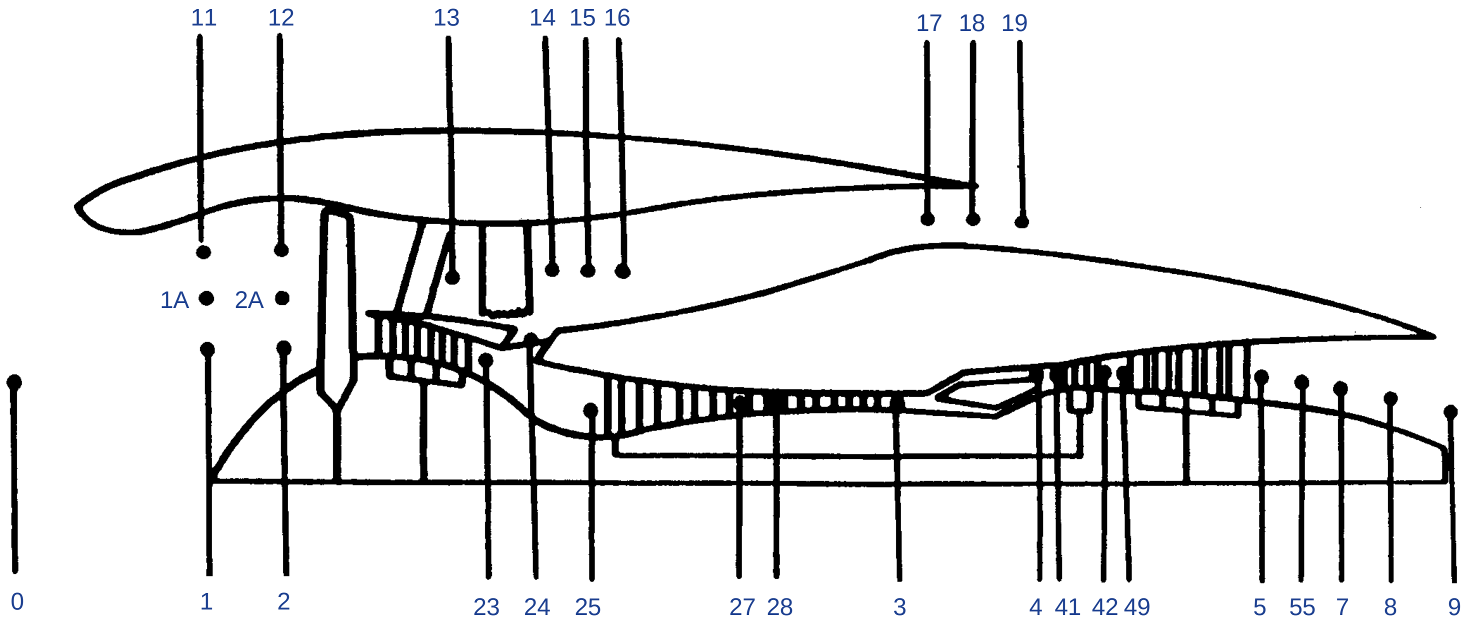
x: States (20)
- 2 speeds
- 7 metal temperatures
- 11 health parameters (efficiency, flow capacity, and cooling flow modifiers)

Reduced (9) state model, fully observable: 2 speeds, 7 health parameters

u: Inputs (4)
- Fuel flow
- Three variable geometries
- No inlet sensors included, since data is for a single operating condition

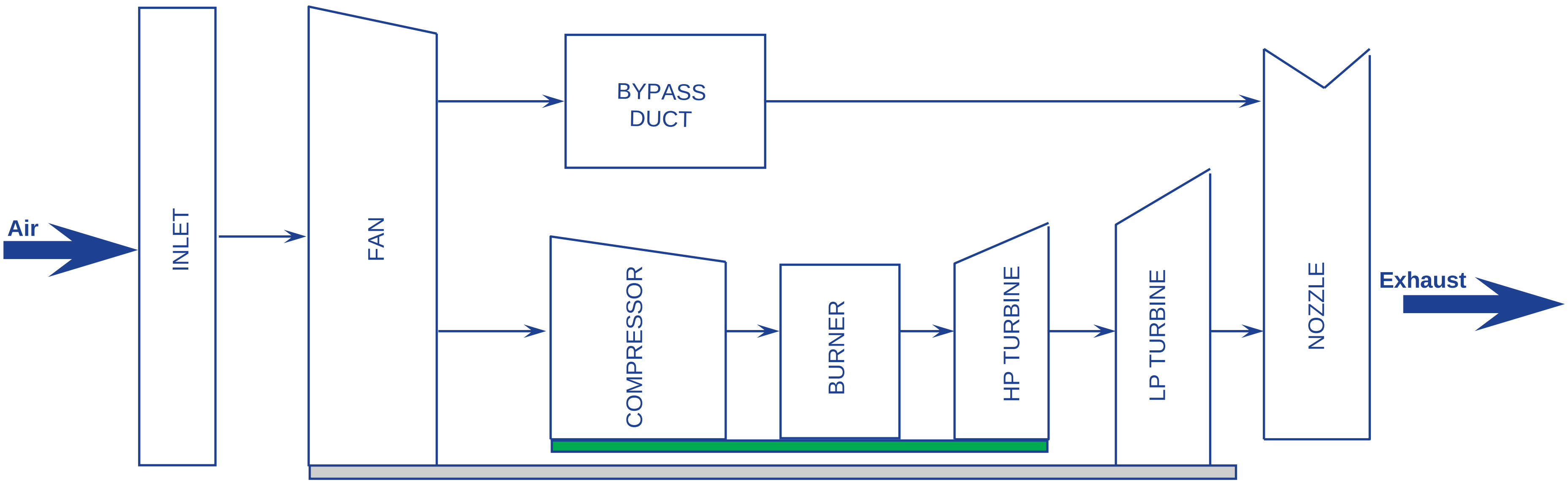
y: Outputs (7) – Speeds, pressures, temperatures

Engine Station Diagram



0	AMBIENT
1	INLET DISCHARGE AT FAN HUB
1A	AVERAGE AT INLET/ENGINE INTERFACE
11	INLET DISCHARGE AT FAN TIP
12	FAN INLET AT TIP (INCLUDES ACOUSTIC TREATMENT)
13	FAN DISCHARGE
14	BYPASS DUCT INLET BEFORE BOOSTER BLEED VALVE
15	BYPASS DUCT INLET - INCLUDES BOOSTER BLEED FLOW
16	BYPASS DUCT - CUSTOMER BLEED
17	BYPASS STREAM AT NOZZLE INLET
18	BYPASS NOZZLE THROAT
19	BYPASS STREAM AT NOZZLE EXIT
2	FAN INLET AT HUB
2A	FLOW WEIGHTED AVERAGE AT FAN FRONT FACE (INCLUDES ACOUSTIC TREATMENT)
23	BOOSTER DISCHARGE
24	BOOSTER BLEED VALVE
25	COMPRESSOR INLET
27	HP COMPRESSOR 8TH STAGE BLEED PORT
28	HP COMPRESSOR 11TH STAGE BLEED PORT
3	COMPRESSOR DISCHARGE
4	HP TURBINE FIRST STAGE NOZZLE INLET
41	HP TURBINE ROTOR INLET
42	HP TURBINE DISCHARGE
49	LP TURBINE INLET
5	LP TURBINE EXIT
55	INTERFACE PLANE (LP TURBINE REAR FRAME EXIT)
7	CORE STREAM AT NOZZLE INLET
8	CORE NOZZLE THROAT
9	CORE STREAM AT NOZZLE EXIT

Component Level Model (CLM)



- CLM is a hi-fidelity, real-time, nonlinear, physics-based model
- Performs mass and energy balance
- Includes component characterizations

Creating the Linear Model

Continuous time linear model created from calculated partial derivatives:

$$\dot{X}(t) = AX(t) + BU(t)$$

$$Y(t) = CX(t) + DU(t)$$

Discrete time linear model can be created using c2d in Matlab, useful for implementing and testing estimator designs

$$X(k+1) = A_d X(k) + B_d U(k)$$

$$Y(k) = C_d X(k) + D_d U(k)$$

Linear model is extracted from nonlinear model using perturbations

Things You Can Do With This Linear Model

- Use arbitrary input values (steps, ramps, etc.) or use provided "flight profile" inputs
- Add sensor white noise:

  Vk=diag(sigma_y)*randn(size(ZS_2));

  ZS = ZS_2 + Vk;
- Simulate a "deteriorated" engine:

  HP = 0.001*[0,4,6,-12,-6,-5,-11,-14,-9,-1,-15] (full model)

  HP = 0.001*[6,-12,-6,-11,-14,-9,-1] (reduced state model)
- Simulate a "damaged" engine:

  HP = 0.001*[0,-10,0,0,0,-10,0] (reduced state model)
- Simulate a biased sensor:

  S1B=0.98; ZS(:,1)=ZS(:,1)*S1B;
- Test different estimation algorithms with different design parameters

imagination at work