State-Space Identification, Noisy Data

6.1: Stochastic identification via subspace methods

- Of the methods we studied last chapter, the subspace algorithms look to be the most promising.
- However, as presented so far, these methods are not capable of correctly identifying systems having disturbance or sensor noise.
- We relax these restrictions in this chapter in two steps.
  - We look first at identifying a purely stochastic system;
  - Then, at identifying a combined deterministic–stochastic system.

Stochastic identification via subspace methods

- Recall the model assumed by the subspace methods

$$x[k+1] = Ax[k] + Bu[k] + w[k]$$
$$y[k] = Cx[k] + Du[k] + v[k].$$

- The deterministic system-ID methods assumed $w[k] \equiv 0$, $v[k] \equiv 0$.
- Here, we pursue the purely stochastic problem, with $u[k] \equiv 0$ instead.

PROBLEM: Given s measurements of the output $y[k] \in \mathbb{R}^p$, generated by an unknown stochastic system of order n,

$$x^s[k+1] = Ax^s[k] + w[k]$$

$$y[k] = Cx^s[k] + v[k],$$

and $w[k]$ and $v[k]$ are zero-mean stationary white processes with

$$\mathbb{E}\left[\begin{bmatrix} w[i] \\ v[i] \end{bmatrix}\begin{bmatrix} w^T[j] & v^T[j] \end{bmatrix}\right] = \begin{bmatrix} Q & S \\ S^T & R \end{bmatrix}\delta_{ij} \geq 0,$$

determine the order n of the unknown system, and the system matrices A and C up to a similarity transformation, and $Q \in \mathbb{R}^{n\times n}$, $S \in \mathbb{R}^{n\times p}$, and $R \in \mathbb{R}^{p\times p}$ so that the second-order statistics of the output of the model and of the given output sequence are equal.

- Note the new notation $x^s[k]$, where the "s" stands for "stochastic".
  - The combined deterministic–stochastic problem will rely on $x^d[k]$ and $x^s[k]$ separately, so we must distinguish between them.
- We will assume that the the stochastic process is stationary with zero mean $\mathbb{E}[x^s[k]] = 0$, which implies that the state covariance is constant: $\mathbb{E}[x^s[k](x^s[k])^T] \triangleq \Sigma^s$.
- This also requires that A be a stable matrix, otherwise the impact of process noise will grow without bound.
- Note: There are multiple equivalent representations of stochastic systems treated in VODM; here, we look at the "forward model" only.
- For this model, we can develop a block-matrix notation

$$Y_f = \mathcal{O}_i X_f^s + \underbrace{\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ C & 0 & 0 & \cdots & 0 \\ CA & C & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{i-2} & CA^{i-3} & CA^{i-4} & \cdots & 0 \end{bmatrix}}_{\Psi_i^w} N_f^w + N_f^v$$

$$= \mathcal{O}_i X_f^s + Y_f^s.$$

- The goal is to show that an orthogonal projection $\xi = Y_f / \boldsymbol{Y_p}$ will project away the $\Psi_i^w N_f^w$ and N_f^v contributions, leaving $\xi = \mathcal{O}_i \widehat{X}$, where $\widehat{X}$ contains the Kalman-filter estimates of the states.

ROADMAP: It's easy to get bogged down in the details, so here's a roadmap of where we'll be going in the next pages:

1. Deriving statistical relationships and defining notation to be used;
2. Deriving noisy-measurements based Kalman-filter state estimates;
3. Seeing how to use projections to recover the Kalman-filter states;
4. Using the Kalman-filter states to recover the system matrices A, C, Q, R, and S.

6.2: Step 1: Some statistical relationships

- We need to define some correlation matrices, which will be used in the development of the stochastic method.
- First, we assume that $w[k]$ and $v[k]$ are zero-mean white-noise vector sequences, independent of $x^s[k]$.
  - This gives us that $\mathbb{E}[x^s[k]v^T[k]] = 0$ and $\mathbb{E}[x^s[k]w^T[k]] = 0$.
- Next, we find the "Lyapunov equation" for the state covariance matrix

$$\begin{aligned}\Sigma^s &= \mathbb{E}[x^s[k+1](x^s[k+1])^T] \\ &= \mathbb{E}[(Ax^s[k] + w[k])(Ax^s[k] + w[k])^T] \\ &= A\mathbb{E}[x^s[k](x^s[k])^T]A^T + \mathbb{E}[w[k]w^T[k]] \\ \Sigma^s &= A\Sigma^s A^T + Q \text{ in steady state.}\end{aligned}$$

  - If you know A and Q, you can solve this discrete-time Lyapunov equation for the steady-state covariance matrix Σ^s (cf. App. A).
- We define G to be the correlation between the future state and present output

$$\begin{aligned}G &\triangleq \mathbb{E}[x^s[k+1]y^T[k]] \\ &= \mathbb{E}[(Ax^s[k] + w[k])(Cx^s[k] + v[k])^T] \\ &= A\mathbb{E}[x^s[k](x^s[k])^T]C^T + \mathbb{E}[w[k]v^T[k]] \\ &= A\Sigma^s C^T + S.\end{aligned}$$

- We define the family of output covariance matrices as

$$\Lambda_i \triangleq \mathbb{E}[y[k+i]y^T[k]].$$

- Note that this is a covariance matrix in addition to being a correlation matrix because $\mathbb{E}[y[k]] = 0$.

- We find for Λ_0,

$$\begin{aligned}\Lambda_0 &= \mathbb{E}[y[k]y^T[k]] \\ &= \mathbb{E}[(Cx^s[k] + v[k])(Cx^s[k] + v[k])^T] \\ &= C\,\mathbb{E}[x^s[k](x^s[k])^T]C^T + \mathbb{E}[v[k]v^T[k]] \\ &= C\Sigma^s C^T + R.\end{aligned}$$

- We can follow similar steps to find Λ_i for $i > 0$

$$\begin{aligned}\Lambda_i &= \mathbb{E}[y[k+i]y^T[k]] \\ &= \mathbb{E}[(Cx^s[k+i] + v[k+i])(Cx^s[k] + v[k])^T] \\ &= \mathbb{E}\left[\left(CA^i x^s[k] + C\begin{bmatrix} A^{i-1}, & \cdots & A, & I\end{bmatrix}\begin{bmatrix} w[k] \\ \vdots \\ w[k+i-1]\end{bmatrix}\right)(Cx^s[k]+v[k])^T\right] \\ &\qquad + \mathbb{E}[v[k+i](Cx^s[k] + v[k])^T] \\ &= CA^i \underbrace{\mathbb{E}[x^s[k](x^s[k])^T]}_{\Sigma^s} C^T + CA^{i-1}\underbrace{\mathbb{E}[w[k]v^T[k]]}_{S} \\ &= CA^{i-1}(A\Sigma^s C^T + S) = CA^{i-1}G.\end{aligned}$$

- Similarly, we can find that $\Lambda_{-i} = G^T(A^{i-1})^T C^T$.

- These last observations indicate that the output covariances can be considered as Markov parameters of the deterministic linear-time-invariant system $\{A, G, C, \Lambda_0\}$.

- This is an important observation that plays a major role in the derivation of stochastic subspace-identification algorithms.

Notation

- As with the deterministic algorithm, we retain the notations $Y_{0|2i-1}$, $Y_{0|i-1}$, $Y_{i|2i-1}$, Y_p, Y_f, Y_p^+, Y_p^- and so forth.
- Recall that we also defined $\Im_i^d = \begin{bmatrix} A^{i-1}B, & \cdots & A^2B, & AB, & B \end{bmatrix}$, where $\Im_i^d$ was the reversed extended controllability matrix.
- We now define $\Im_i^c = \begin{bmatrix} A^{i-1}G, & \cdots & A^2G, & AG, & G \end{bmatrix}$, where $\Im_i^c$ is the <u>reversed extended stochastic controllability matrix</u>.
  - The superscript "c" stands for "covariance".
- We assume that the pair $\{A, Q^{1/2}\}$ is controllable, implying that all dynamical modes of the system are excited by the process noise.
- The block Toeplitz matrices C_i and L_i are constructed from the output covariance matrices as

$$C_i \triangleq \begin{bmatrix} \Lambda_i & \Lambda_{i-1} & \cdots & \Lambda_2 & \Lambda_1 \\ \Lambda_{i+1} & \Lambda_i & \cdots & \Lambda_3 & \Lambda_2 \\ \Lambda_{i+2} & \Lambda_{i+1} & \cdots & \Lambda_4 & \Lambda_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Lambda_{2i-1} & \Lambda_{2i-2} & \cdots & \Lambda_{i+1} & \Lambda_i \end{bmatrix} \in \mathbb{R}^{pi\times pi}$$

$$L_i \triangleq \begin{bmatrix} \Lambda_0 & \Lambda_{-1} & \cdots & \Lambda_{2-i} & \Lambda_{1-i} \\ \Lambda_1 & \Lambda_0 & \cdots & \Lambda_{3-i} & \Lambda_{2-i} \\ \Lambda_2 & \Lambda_1 & \cdots & \Lambda_{4-i} & \Lambda_{3-i} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Lambda_{i-1} & \Lambda_{i-2} & \cdots & \Lambda_1 & \Lambda_0 \end{bmatrix} \in \mathbb{R}^{pi\times pi}.$$

- We can approximate each Λ_i as

$$\Lambda_i \approx \lim_{N\to\infty} \frac{1}{N}\sum_{k=0}^{N-1} y[k+i]y^T[k] \triangleq \mathbb{E}_N[y[k+i]y^T[k]]$$

$$= \lim_{N\to\infty} \frac{1}{N} Y_{i|i} Y_{0|0}^T \triangleq \Phi_{[Y_{i|i}, Y_{0|0}]}.$$

- Note in passing that we have introduced two new notational ideas:

$$\mathbb{E}_N[\cdot] \triangleq \lim_{N\to\infty} \frac{1}{N}[\cdot]$$

$$\Phi_{[A,B]} \triangleq \mathbb{E}_N[AB^T].$$

- These allow us to extend the geometric tools used in a deterministic framework to a stochastic framework. In particular,

$$A/\boldsymbol{B} = \Phi_{[A,B]} \Phi_{[B,B]}^{\dagger} B.$$

- Using this new notation, we can show

$$C_i = \Phi_{[Y_f, Y_p]} = \frac{1}{N} Y_f Y_p^T$$

$$L_i = \Phi_{[Y_p, Y_p]} = \frac{1}{N} Y_p Y_p^T$$

$$= \Phi_{[Y_f, Y_f]} = \frac{1}{N} Y_f Y_f^T.$$

6.3: Step 2a: Kalman-filter covariance

- Recall that for deterministic subspace identification, we required the state sequence X_f^d. From that sequence we can recover $\{A, B, C, D\}$.
- For stochastic subspace identification, we also require an estimate of the state sequence, $\widehat{X}_i$. From this sequence, we will recover A and C.
- But, how to get an estimate of a system state $\hat{x}[k]$?
- A Kalman filter is a recursive algorithm for optimally estimating an LTI system's state, given the system's $\{A, B, C, D, Q, R, S\}$ matrices.
- An amazing result of the stochastic subspace system-ID section is that we can recover the Kalman-filter state estimates from measurements of a stochastic system's output data only, Y_p and Y_f, without needing to know its $\{A, C, Q, R, S\}$ matrices!
- It is beyond the scope of this course to go into the intricate details of Kalman filters. They are treated in some depth in ECE5530; they are furthermore the entire subject of ECE5550.
- Here, we present some results without proof.
- The Kalman filter is a recursive state estimator that computes

$$\begin{aligned}
\hat{x}[k] &= A\hat{x}[k-1] + K[k-1](y[k-1] - C\hat{x}[k-1]) \\
K[k-1] &= (G - AP[k-1]C^T)(\Lambda_0 - CP[k-1]C^T)^{-1} \\
P[k] &= AP[k-1]A^T + (G - AP[k-1]C^T) \times \\
&\qquad (\Lambda_0 - CP[k-1]C^T)^{-1}(G - AP[k-1]C^T)^T,
\end{aligned}$$

where $K[k]$ is time-varying "Kalman gain", $P[k]$ is correlation of state estimate $P[k] = \mathbb{E}[\hat{x}[k]\hat{x}^T[k]]$, and both $\hat{x}[0] = 0$ and $P[0] = 0$.

KEY POINT: We are going to show that $\hat{x}[k]$ and $P[k]$ can be written using our notation as

$$\hat{x}[k] = \Im_k^c L_k^{-1} \begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[k-1] \end{bmatrix} \quad \text{and} \quad P[k] = \Im_k^c L_k^{-1} (\Im_k^c)^T.$$

- These two relationships are what make the stochastic subspace identification algorithm work.
- We'll prove them by induction. That is,
  - We'll show that they are true for $k = 1$.
  - Then, if they are true for $k = p$, we'll show that they are also true for $k = p + 1$. This completes the proof.
- For $k = 1$, the Kalman-filter equation is

$$\hat{x}[1] = A \underbrace{\hat{x}[0]}_{0} + K[0](y[0] - C \underbrace{\hat{x}[0]}_{0}) = K[0]y[0]$$

$$K[0] = (G - A \underbrace{P[0]}_{0} C^T)(\Lambda_0 - C \underbrace{P[0]}_{0} C^T)^{-1} = G\Lambda_0^{-1} = \Im_1^c L_1^{-1}.$$

- Therefore, $\hat{x}[1] = \Im_1^c L_1^{-1} y[0]$, which proves the hypothesized state relationship for $k = 1$. Note that we also have

$$\begin{aligned} P[1] = A \underbrace{P[0]}_{0} A^T + (G - A \underbrace{P[0]}_{0} C^T) \times \\ (\Lambda_0 - C \underbrace{P[0]}_{0} C^T)^{-1}(G - A \underbrace{P[0]}_{0} C^T)^T \\ = G\Lambda_0^{-1} G^T = \Im_1^c L_1^{-1} (\Im_1^c)^T, \end{aligned}$$

which proves the hypothesized correlation relationship for $k = 1$.

- Now, we assume that the hypothesized relationships are true for $k = p$. That is, we assume that

$$\hat{x}[p] = \Im_p^c L_p^{-1} \begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[p-1] \end{bmatrix} \quad \text{and} \quad P[p] = \Im_p^c L_p^{-1} (\Im_p^c)^T.$$

- With this assumption, we show that the hypotheses are true for $k = p+1$. We first find $P[p+1]$

$$\begin{aligned} P[p+1] &= \Im_{p+1}^c L_{p+1}^{-1} (\Im_{p+1}^c)^T \\ &= \left[\begin{array}{c:c} A\Im_p^c & G \end{array}\right] \left[\begin{array}{c:c} L_p & (\Im_p^c)^T C^T \\ \hdashline C\Im_p^c & \Lambda_0 \end{array}\right]^{-1} \left[\begin{array}{c} (\Im_p^c)^T A^T \\ \hdashline G^T \end{array}\right]. \end{aligned}$$

- To handle the block matrix inverse, we invoke the block matrix inversion lemma,

$$\left[\begin{array}{c:c} a & b \\ \hdashline c & d \end{array}\right]^{-1} = \left[\begin{array}{c:c} a^{-1} + a^{-1}b(d - ca^{-1}b)^{-1}ca^{-1} & -a^{-1}b(d - ca^{-1}b)^{-1} \\ \hdashline -(d - ca^{-1}b)^{-1}ca^{-1} & (d - ca^{-1}b)^{-1} \end{array}\right],$$

which may be proven by multiplying on the left by $\left[\begin{array}{c:c} a & b \\ \hdashline c & d \end{array}\right]$, and finding the identity.

- For our special case, $a = L_p$, $b = (\Im_p^c)^T C^T$, $c = C\Im_p^c$, and $d = \Lambda_0$.

$$\left[\begin{array}{c:c} L_p & (\Im_p^c)^T C^T \\ \hdashline C\Im_p^c & \Lambda_0 \end{array}\right]^{-1} = \left[\begin{array}{c:c} m_{11} & m_{12} \\ \hdashline m_{21} & m_{22} \end{array}\right] \qquad \text{where}$$

$$\begin{aligned} m_{11} &= L_p^{-1} + L_p^{-1}(\Im_p^c)^T C^T \Delta^{-1} C\Im_p^c L_p^{-1} \\ m_{12} &= -L_p^{-1}(\Im_p^c)^T C^T \Delta^{-1} \end{aligned}$$

$$m_{21} = -\Delta^{-1} C \Im_p^c L_p^{-1}$$

$$m_{22} = \Delta^{-1} = (\Lambda_0 - C \underbrace{\Im_p^c L_p^{-1} (\Im_p^c)^T}_{P[p]} C^T)^{-1} = (\Lambda_0 - CP[p]C^T)^{-1}.$$

- Multiplying out the three matrices gives

$$P[p+1] = \left[\begin{array}{c:c} A\Im_p^c & G \end{array}\right] \left[\begin{array}{c:c} m_{11} & m_{12} \\ \hdashline m_{21} & m_{22} \end{array}\right] \left[\begin{array}{c} (\Im_p^c)^T A^T \\ \hdashline G^T \end{array}\right]$$

$$= A\Im_p^c m_{11} (\Im_p^c)^T A^T + G m_{21} (\Im_p^c)^T A^T + A\Im_p^c m_{12} G^T + G m_{22} G^T.$$

- Treating the terms individually,

$$\begin{aligned} A\Im_p^c m_{11} (\Im_p^c)^T A^T &= A \underbrace{\Im_p^c L_p^{-1} (\Im_p^c)^T}_{P[p]} A^T \\ &\quad + A \underbrace{\Im_p^c L_p^{-1} (\Im_p^c)^T}_{P[p]} C^T \Delta^{-1} C \underbrace{\Im_p^c L_p^{-1} (\Im_p^c)^T}_{P[p]} A^T \\ &= AP[p]A^T + AP[p]C^T \Delta^{-1} CP[p]A^T \\ G m_{21} (\Im_p^c)^T A^T &= -G\Delta^{-1} C \underbrace{\Im_p^c L_p^{-1} (\Im_p^c)^T}_{P[p]} A^T = -G\Delta^{-1} CP[p]A^T \\ A\Im_p^c m_{12} G^T &= -A \underbrace{\Im_p^c L_p^{-1} (\Im_p^c)^T}_{P[p]} C^T \Delta^{-1} G^T = -AP[p]C^T \Delta^{-1} G^T \\ G m_{22} G^T &= G\Delta^{-1} G^T. \end{aligned}$$

- Putting everything together,

$$\begin{aligned} P[p+1] &= AP[p]A^T + (G - AP[p]C^T)\Delta^{-1}(G - AP[p]C^T)^T \\ &= AP[p]A^T + (G - AP[p]C^T)(\Lambda_0 - CP[p]C^T)^{-1}(G - AP[p]C^T)^T. \end{aligned}$$

- This is the matrix recursion that we started with, thus we conclude that it is indeed true that $P[p] = \Im_p^c L_p^{-1} (\Im_p^c)^T$.

6.4: Step 2b: Kalman-filter state

- We now focus on the state relationship.

$$\hat{x}[p+1] = \Im^c_{p+1} L^{-1}_{p+1} \begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[p] \end{bmatrix}$$

$$= \left[\begin{array}{c:c} A\Im^c_p & G \end{array}\right] \left[\begin{array}{c:c} L_p & (\Im^c_p)^T C^T \\ \hdashline C\Im^c_p & \Lambda_0 \end{array}\right]^{-1} \left[\begin{array}{c} y[0] \\ \vdots \\ y[p-1] \\ \hdashline y[p] \end{array}\right]$$

$$= \left[\begin{array}{c:c} A\Im^c_p & G \end{array}\right] \left[\begin{array}{c:c} m_{11} & m_{12} \\ \hdashline m_{21} & m_{22} \end{array}\right] \left[\begin{array}{c} y[0] \\ \vdots \\ y[p-1] \\ \hdashline y[p] \end{array}\right].$$

- Based on this partitioning, we have

$$\hat{x}[p+1] = \left[\begin{array}{c:c} A\Im^c_p m_{11} + G m_{21} & A\Im^c_p m_{12} + G m_{22} \end{array}\right] \left[\begin{array}{c} y[0] \\ \vdots \\ y[p-1] \\ \hdashline y[p] \end{array}\right]$$

$$= \left(A\Im^c_p m_{11} + G m_{21}\right) \begin{bmatrix} y[0] \\ \vdots \\ y[p-1] \end{bmatrix} + \left(A\Im^c_p m_{12} + G m_{22}\right) y[p]$$

$$= A\Im_p^c L_p^{-1} \begin{bmatrix} y[0] \\ \vdots \\ y[p-1] \end{bmatrix}$$

$$+ \left(A\Im_p^c L_p^{-1} \Im_p^c C^T \Delta^{-1} C \Im_p^c L_p^{-1} - G\Delta^{-1} C \Im_p^c L_p^{-1} \right) \begin{bmatrix} y[0] \\ \vdots \\ y[p-1] \end{bmatrix}$$

$$+ \left(G\Delta^{-1} - A\Im_p^c L_p^{-1} (\Im_p^c)^T C^T \Delta^{-1} \right) y[p]$$

$$= [A - (G - A \underbrace{\Im_p^c L_p^{-1} (\Im_p^c)^T}_{P[p]} C^T) \Delta^{-1} C] \underbrace{\Im_p^c L_p^{-1} \begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[p-1] \end{bmatrix}}_{\hat{x}[p]}$$

$$+ (G - A \underbrace{\Im_p^c L_p^{-1} (\Im_p^c)^T}_{P[p]} C^T) \Delta^{-1} y[p]$$

$$= A\hat{x}[p] - \left(G - AP[p]C^T \right) \Delta^{-1} C \hat{x}[p] + \left(G - AP[p]C^T \right) \Delta^{-1} y[p]$$

$$= A\hat{x}[p] + \underbrace{(G - AP[p]C^T)(\Lambda_0 - CP[p]C^T)^{-1}}_{K[p]} (y[p] - C\hat{x}[p])$$

$$= A\hat{x}[p] + K[p](y[p] - C\hat{x}[p]),$$

which proves the relationship.

BOTTOM LINE: We have now shown the very significant result that

$$\hat{x}[k] = \Im_k^c L_k^{-1} \begin{bmatrix} y[0] \\ y[1] \\ \vdots \\ y[k-1] \end{bmatrix}.$$

- We can combine multiple state estimates in a block matrix to get

$$\begin{aligned} \widehat{X}_i &= \begin{bmatrix} \hat{x}[i], & \hat{x}[i+1], & \cdots & \hat{x}[k+N-1] \end{bmatrix} \\ &= \Im_i^c L_i^{-1} Y_p. \end{aligned}$$

- This state sequence is generated by a bank of non-steady-state Kalman filters working in parallel on each of the columns of the block Hankel matrix of past outputs Y_p.

$$\begin{array}{lcl} \widehat{X}_0: & \begin{bmatrix} 0 & \cdots & 0 & \cdots & 0 \end{bmatrix} & \text{Kalman filter} \\ Y_p: & \begin{bmatrix} y[0] & & y[q] & & y[N-1] \\ \vdots & & \vdots & & \vdots \\ y[i-1] \downarrow & & y[i+q-1] \downarrow & & y[i+N-2] \downarrow \end{bmatrix} & \downarrow \\ \widehat{X}_i: & \begin{bmatrix} \hat{x}[i] & \cdots & \hat{x}[i+q] & \cdots & \hat{x}[i+N-1] \end{bmatrix} & \end{array}$$

- Note that the Kalman filters use only *partial* output information. For instance, the $(q+1)$st column of $\widehat{X}_i$ can be written as

$$\hat{x}[i+q] = \Im_i^c L_i^{-1} \begin{bmatrix} y[q] \\ \vdots \\ y[i+q-1] \end{bmatrix},$$

which indicates that the Kalman filter generating the estimate $\hat{x}[i+1]$ uses only i output measurements $y[q] \cdots y[i+q-1]$ instead of the full set of output measurements $y[0] \cdots y[i+q-1]$ as would be expected.

6.5: Step 3: Geometric properties of stochastic systems

- The miracle isn't so much that we can represent the operation of the Kalman filter in such a compact notation (which is pretty miraculous) but that we can moreover determine the Kalman-filter state estimates $\widehat{X}_i$ using only output data via subspace projections.
- We're going to now show that $\xi_i \triangleq Y_f/\boldsymbol{Y_p}$ also satisfies the relationship $\xi_i = \mathcal{O}_i \widehat{X}_i$.
- Therefore, we can recover the Kalman-filter states from $Y_{0|2i-1}$.
- We assume that process noise $w[k]$ and sensor noise $v[k]$ are not identically zero, and that the number of measurements $N \to \infty$.
  - For finite N, the outcomes will be somewhat biased.
- We start the proof with

$$\begin{aligned}\xi_i &\triangleq Y_f/\boldsymbol{Y_p} \\ &= \Phi_{[Y_f,Y_p]}\Phi^{\dagger}_{[Y_p,Y_p]}Y_p \\ &= C_i L_i^{-1} Y_p.\end{aligned}$$

- We then take a closer look at C_i

$$C_i = \begin{bmatrix} \Lambda_i & \Lambda_{i-1} & \cdots & \Lambda_2 & \Lambda_1 \\ \Lambda_{i+1} & \Lambda_i & \cdots & \Lambda_3 & \Lambda_2 \\ \Lambda_{i+2} & \Lambda_{i+1} & \cdots & \Lambda_4 & \Lambda_3 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \Lambda_{2i-1} & \Lambda_{2i-2} & \cdots & \Lambda_{i+1} & \Lambda_i \end{bmatrix}$$

$$= \begin{bmatrix} CA^{i-1}G & CA^{i-2}G & \cdots & CAG & CG \\ CA^{i}G & CA^{i-1}G & \cdots & CA^2G & CAG \\ CA^{i+1}G & CA^{i}G & \cdots & CA^3G & CA^2G \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ CA^{2i-2}G & CA^{2i-3}G & \cdots & CA^{i}G & CA^{i-1}G \end{bmatrix}$$

$$= \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{i-1} \end{bmatrix} \begin{bmatrix} A^{i-1}G, & A^{i-2}G, & \cdots & AG, & G \end{bmatrix}$$

$$= \mathcal{O}_i \Im_i^c.$$

- Therefore, $\xi_i = \mathcal{O}_i \Im_i^c L_i^{-1} Y_p$. But, we already know $\Im_i^c L_i^{-1} Y_p = \widehat{X}_i$, so

$$\xi_i = \mathcal{O}_i \widehat{X}_i.$$

- We proceed to break up ξ_i into its component parts via SVD, much like we did for the deterministic subspace identification problem.
- We first define weighting matrix $W_1 \in \mathbb{R}^{pi \times pi}$ and $W_2 \in \mathbb{R}^{N \times N}$, as before, such that W_1 is full rank and $\text{rank}(Y_p) = \text{rank}(Y_p W_2)$.
- Then, we compute the singular value decomposition

$$W_1 \xi_i W_2 = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix} = U_1 \Sigma_1 V_1^T.$$

- Since W_1 is full rank, and since the rank of $Y_p W_2$ is equal to the rank of Y_p, we find that the rank of $W_1 \xi_i W_2$ is equal to the rank of ξ_i, which in turn is equal to n, the system order.

- The SVD can now be split into two parts:

$$W_1 \mathcal{O}_i = U_1 \Sigma_1^{1/2} T$$

$$\widehat{X}_i W_2 = T^{-1} \Sigma_1^{1/2} V_1^T.$$

- From this, we can extract $\mathcal{O}_i = W_1^{-1} U_1 \Sigma_1^{1/2} T$, and $\widehat{X}_i = \mathcal{O}_i^{\dagger} \xi_i$.
- This overall result indicates that the row space of the future states $\widehat{X}_i$ can be found by orthogonally projecting the row space of the future outputs Y_f onto the row space of the past outputs Y_p.

6.6: Step 4: Computing the system matrices

- We now consider how the system matrices A, C, Q, S, and R may be computed from the data.
- VODM proposes three algorithms; we consider only the third here.
- We already have that $\xi_i = Y_f/\boldsymbol{Y_p} = \mathcal{O}_i \widehat{X}_i$.
- Through similar reasoning, we can show $\xi_{i-1} \triangleq Y_f^-/\boldsymbol{Y_p^+} = \mathcal{O}_{i-1}\widehat{X}_{i+1}$.
- So, we can calculate both $\widehat{X}_i$ and $\widehat{X}_{i+1}$ using only the output data.
- We now form the following set of equations

$$\underbrace{\begin{bmatrix} \widehat{X}_{i+1} \\ Y_{i|i} \end{bmatrix}}_{\text{known}} = \begin{bmatrix} A \\ C \end{bmatrix} \widehat{X}_i + \underbrace{\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}}_{\text{residuals}}.$$

- The Kalman-filter residuals ρ_w and ρ_v are uncorrelated with $\widehat{X}_i$, so we can solve for A and C as

$$\begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} = \begin{bmatrix} \widehat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} \widehat{X}_i^{\dagger}.$$

- The estimate of A and C is asymptotically unbiased.
- We can now solve for

$$\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} = \begin{bmatrix} \widehat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} \widehat{X}_i,$$

and then

$$\begin{bmatrix} \hat{Q}_i & \hat{S}_i \\ \hat{S}_i^T & \hat{R}_i \end{bmatrix} = \mathbb{E}_N \left[\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \begin{bmatrix} \rho_w^T & \rho_v^T \end{bmatrix} \right].$$

- Note that we would really like to compute steady-state Q, R, and S, but $\hat{Q}_i$, $\hat{R}_i$, $\hat{S}_i$ are non-steady-state matrices satisfying

$$P[i+1] = AP[i]A^T + \hat{Q}_i$$

$$G = AP[i]C^T + \hat{S}_i$$

$$\Lambda_0 = CP[i]C^T + \hat{R}_i.$$

- When $N \to \infty$, $\hat{Q}_i \to Q$, $\hat{R}_i \to R$, and $\hat{S}_i \to S$. Otherwise, there is some noise/bias in the estimates.

- However, we are guaranteed, by construction, that

$$\begin{bmatrix} \widehat{Q} & \widehat{S} \\ \widehat{S}^T & \widehat{R} \end{bmatrix} \triangleq \mathbb{E}_N \left[\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \begin{bmatrix} \rho_w^T & \rho_v^T \end{bmatrix} \right] > 0,$$

which is important to guarantee a "positive real" solution, and is not guaranteed by the other two algorithms presented in VODM.

Summary of the stochastic subspace system-identification method

1. Form the data matrix $Y_{0|2i-1}$, and break it up into Y_p, Y_f, Y_p^+, Y_f^-.
2. Compute the projections $\xi_i = Y_f / \boldsymbol{Y_p}$ and $\xi_{i-1} = Y_f^- / \boldsymbol{Y_P^+}$.
3. Calculate the SVD of the weighted projection $W_1 \xi_i W_2 = U \Sigma V^T$.
4. Determine the order by inspecting the singular values in Σ, and partition the SVD accordingly to obtain U_1 and Σ_1.
5. Determine $\mathcal{O}_i = W_1^{-1} U_1 \Sigma_1^{1/2}$ and $\mathcal{O}_{i-1} = \mathcal{O}_i^{\downarrow}$.
6. Determine $\widehat{X}_i = \mathcal{O}_i^{\dagger} \xi_i$ and $\widehat{X}_{i+1} = \mathcal{O}_{i-1}^{\dagger} \xi_{i-1}$.
7. Solve for $\hat{A}$ and $\hat{C}$ via

$$\begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} = \begin{bmatrix} \widehat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} \widehat{X}_i^{\dagger}.$$

8. Solve for the residuals

$$\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} = \begin{bmatrix} \widehat{X}_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} \widehat{X}_i.$$

9. Solve for the noise covariance matrices via

$$\begin{bmatrix} \widehat{Q} & \widehat{S} \\ \widehat{S}^T & \widehat{R} \end{bmatrix} \triangleq \mathbb{E}_N \left[\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \begin{bmatrix} \rho_w^T & \rho_v^T \end{bmatrix} \right].$$

- In the van Overschee toolbox, `sto_pos.m` contains a MATLAB implementation of this algorithm.
  - Note that the toolbox model is slightly different from the text model:

$$x[k+1] = Ax[k] + Bu[k] + Ke[k]$$

$$y[k] = Cx[k] + Du[k] + e[k].$$

  - We can see that $v[k] = e[k]$ and $w[k] = Ke[k]$. Then,

$$R = \mathbb{E}\left[v[k]v^T[k]\right] = \mathbb{E}\left[e[k]e^T[k]\right]$$

$$S = \mathbb{E}\left[w[k]v^T[k]\right] = K\mathbb{E}\left[e[k]e^T[k]\right] = KR$$

$$Q = \mathbb{E}\left[w[k]w^T[k]\right] = K\mathbb{E}\left[e[k]e^T[k]\right]K^T = KRK^T.$$

  - So, if we can find K and R, then we can also compute S and Q.
    - But, notice that the method is slightly less general since it does not allow for uncorrelated $w[k]$ and $v[k]$.
    - I'm not sure why they implement the less general version.

6.7: Combined deterministic–stochastic identification

- In deterministic subspace system identification,
  - We started by recognizing that $Y_f = \mathcal{O}_i X_f^d + \Psi_i U_f$.
  - We used **oblique** projection $\xi = Y_f /_{U_f} \boldsymbol{W_p} = \mathcal{O}_i X_i^d$ to recover the noiseless state sequence and the extended observability matrix;
  - We then used the SVD to separate $\mathcal{O}_i$ from X_i^d.
- In stochastic subspace system identification,
  - We started by recognizing that $Y_f = \mathcal{O}_i X_f^s + \Psi_i^w N_f^w + N_f^v$.
  - We used **orthogonal** projection to compute $\xi_i = Y_f / \boldsymbol{Y_p} = \mathcal{O}_i \widehat{X}_i$ to project out noise and find a Kalman-filter state sequence.
  - We then used the SVD to separate $\mathcal{O}_i$ from $\widehat{X}_i$.
- For combined deterministic–stochastic identification,
  - We start with $Y_f = \mathcal{O}_i (X_f^d + X_f^s) + \Psi_i U_f + \Psi_i^w N_f^w + N_f^v$.
  - We would like to use orthogonal projection to eliminate the noise and find a Kalman-filter state sequence, but what we actually find has a U_f component still embedded in it.
  - Alternately, we would like to use oblique projection to eliminate the U_f component, but then find that we have a state sequence that is not useful in itself.
  - It turns out that we need to do **both** types of projection, combining results, to come up with our system matrices.

ROADMAP: In this section, we will

1. First look at what a matrix of Kalman-filter states looks like;

2. Then look at the orthogonal projection $\mathcal{Z}_i = Y_f / \begin{pmatrix} \boldsymbol{W_p} \\ \boldsymbol{U_f} \end{pmatrix}$, and find that it gives $\mathcal{Z}_i = \mathcal{O}_i \widehat{X}_i + \Psi_i U_f$, which has Kalman-filter states embedded within, but is also influenced by U_f.
3. Then look at the oblique projection $\xi_i = Y_f /_{U_f} \boldsymbol{W_p}$, and find that it gives $\xi_i = \mathcal{O}_i \widetilde{X}_i$, where the states are not so useful, but where the extended observability matrix can be found.
4. Finally, combine the two methods, computing $\mathcal{O}_i$ from the oblique projection, and computing A, B, C, D, Q, R, and S from $\mathcal{O}_i$ and $\mathcal{Z}_i$.

6.8: Step 1a: Kalman-filter covariance

- We state without proof that the non-steady-state Kalman filter estimates $\hat{x}_k$ according to the following equations:

$$\hat{x}[k] = A\hat{x}[k-1]+Bu[k-1]+K[k-1]\,(y[k-1]-C\hat{x}[k-1]-Du[k-1])$$

$$K[k-1] = (G - AP[k-1]C^T)(\Lambda_0 - CP[k-1]C^T)^{-1}$$

$$P[k] = AP[k-1]A^T + (G-AP[k-1]C^T)(\Lambda_0-CP[k-1]C^T)^{-1} \times (G-AP[k-1]C^T)^T.$$

- We desire to show that this set of recursive equations can be written explicitly as

$$\hat{x}[k] = \left[\begin{array}{c:c:c} A^k - \Omega_k\mathcal{O}_k & \Im_k^d - \Omega_k\Psi_k & \Omega_k \end{array}\right] \left[\begin{array}{c} \hat{x}[0] \\ \hdashline u[0] \\ \vdots \\ u[k-1] \\ \hdashline y[0] \\ \vdots \\ y[k-1] \end{array}\right],$$

where

$$\Omega_k \triangleq (\Im_k^c - A^k P[0]\mathcal{O}_k^T)(L_k - \mathcal{O}_k P[0]\mathcal{O}_k^T)^{-1}.$$

- We will also show that the explicit solution to the matrix $P[k]$ equals

$$P[k] = A^k P[0](A^T)^k + \Omega_k(\Im_k^c - A^k P[0]\mathcal{O}_k^T)^T.$$

- Before we prove this, note carefully how the solution depends on the initial state estimate $\hat{x}[0]$ and the initial covariance estimate $P[0]$.

- This is different from the stochastic case (where both were set to zero). Turns out we need to be very careful in how we account for these initial estimates if we are to come up with an unbiased solution.
- We prove these two hypotheses by induction: We show they are true for $k = 1$; then, assume they are true for $k = p$, and prove that they are true for $k = p + 1$ also, completing the proof.
- Starting with $k = 1$, we look at Ω_1, and from it find $\hat{x}[1]$:

$$\Omega_1 = (G - A^1 P[0]C^T)(\Lambda_0 - CP[0]C^T)^{-1} = K[0]$$

$$\begin{aligned}\hat{x}[1] &= (A - \Omega_1 \mathcal{O}_1)\,\hat{x}[0] + \left(\Im_1^d - \Omega_1 \Psi_1\right) u[0] + \Omega_1 y[0] \\ &= (A - K[0]C)\,\hat{x}[0] + (B - K[0]D)\,u[0] + K[0]y[0] \\ &= A\hat{x}[0] + Bu[0] + K[0]\left(y[0] - C\hat{x}[0] - Du[0]\right),\end{aligned}$$

which is exactly the state update equation for $k = 1$.

- We can also show that the expression for $P[1]$ is

$$\begin{aligned}P[1] &= AP[0]A^T + (G - AP[0]C^T)(\Lambda_0 - CP[0]C^T)^{-1}(G - AP[0]C^T)^T \\ &= AP[0]A^T + (\Im_1^c - AP[0]\mathcal{O}_1^T)(L_1 - \mathcal{O}_1 P[0]\mathcal{O}_1^T)^{-1}(\Im_1^c - AP[0]\mathcal{O}_1^T)^T \\ &= AP[0]A^T + \Omega_1(\Im_1^c - AP[0]\mathcal{O}_1^T)^T.\end{aligned}$$

- This completes the proof for $k = 1$.
- We assume that the hypotheses are true for $k = p$, which gives

$$\hat{x}[p] = \left(A^p - \Omega_p \mathcal{O}_p\right)\hat{x}[0] + \left(\Im_p^d - \Omega_p \Psi_p\right)\begin{bmatrix} u[0] \\ \vdots \\ u[p-1] \end{bmatrix} + \Omega_p \begin{bmatrix} y[0] \\ \vdots \\ y[p-1] \end{bmatrix},$$

$$P[p] = A^p P[0](A^T)^p + \Omega_p \left(\Im_p^c - A^p P[0]\mathcal{O}_p^T\right)^T.$$

- We now proceed to show that the hypotheses are true for $k = p + 1$.
- But first, define some notation to clean things up a bit:

$$\alpha_p = (\Im_p^c - A^p P[0]\mathcal{O}_p^T)$$
$$\beta_p = (L_p - \mathcal{O}_p P[0]\mathcal{O}_p^T)$$
$$\gamma_p = (G - A^{p+1} P[0](A^T)^p C^T).$$

- Using this notation, we find that

$$\begin{aligned} P[p+1] &= A^{p+1}P[0](A^T)^{p+1} + \alpha_{p+1}\beta_{p+1}^{-1}\alpha_{p+1}^T \\ &= A^{p+1}P[0](A^T)^{p+1} + \left(\Im_{p+1}^c - A^{p+1}P[0]\mathcal{O}_{p+1}^T\right)\times \\ &\quad \left(L_{p+1} - \mathcal{O}_{p+1}P[0]\mathcal{O}_{p+1}^T\right)^{-1}\left(\Im_{p+1}^c - A^{p+1}P[0]\mathcal{O}_{p+1}^T\right)^T. \end{aligned}$$

- Note that we can write

$$\Im_{p+1}^c = \left[\begin{array}{c:c} A\Im_p^c & G \end{array}\right]$$
$$\mathcal{O}_{p+1}^T = \left[\begin{array}{c} \mathcal{O}_p \\ \hdashline CA^p \end{array}\right]^T = \left[\begin{array}{c:c} \mathcal{O}_p^T & (A^T)^p C^T \end{array}\right]$$
$$L_{p+1} = \left[\begin{array}{c:c} L_p & (\Im_p^c)^T C^T \\ \hdashline C\Im_p^c & \Lambda_0 \end{array}\right].$$

- Substituting these relationships gives

$$\begin{aligned} P[p+1] &= A^{p+1}P[0](A^T)^{p+1} + \alpha_{p+1}\beta_{p+1}^{-1}\alpha_{p+1}^T \\ &= A^{p+1}P[0](A^T)^{p+1} + \\ &\quad \left[\begin{array}{c:c} A\left(\Im_p^c - A^p P[0]\mathcal{O}_p^T\right) & G - A^{p+1}P[0](A^T)^p C^T \end{array}\right]\times \end{aligned}$$

$$\left(\left[\begin{array}{c:c} L_p & (\Im_p^c)^T C^T \\ \hdashline C\Im_p^c & \Lambda_0 \end{array}\right] - \left[\begin{array}{c} \mathcal{O}_p \\ \hdashline CA^p \end{array}\right] P[0] \left[\begin{array}{c:c} \mathcal{O}_p^T & (A^T)^p C^T \end{array}\right]\right)^{-1} \times$$

$$\left[\begin{array}{c} \left((\Im_p^c)^T - \mathcal{O}_p P[0](A^T)^p\right) A^T \\ \hdashline G^T - CA^p P[0](A^T)^{p+1} \end{array}\right]$$

$$= A^{p+1} P[0](A^T)^{p+1} +$$

$$\left[\begin{array}{c:c} A\left(\Im_p^c - A^p P[0]\mathcal{O}_p^T\right) & G - A^{p+1} P[0](A^T)^p C^T \end{array}\right] \times$$

$$\left[\begin{array}{c:c} L_p - \mathcal{O}_p P[0]\mathcal{O}_p^T & (\Im_p^c)^T C^T - \mathcal{O}_p P[0](A^T)^p C^T \\ \hdashline C\Im_p^c - CA^p P[0]\mathcal{O}_p^T & \Lambda_0 - CA^p P[0](A^T)^p C^T \end{array}\right]^{-1} \times$$

$$\left[\begin{array}{c} \left((\Im_p^c)^T - \mathcal{O}_p P[0](A^T)^p\right) A^T \\ \hdashline G^T - CA^p P[0](A^T)^{p+1} \end{array}\right]$$

$$= A^{p+1} P[0](A^T)^{p+1} +$$

$$\left[\begin{array}{c:c} A\alpha_p & \gamma_p \end{array}\right] \left[\begin{array}{c:c} \beta_p & \alpha_p^T C^T \\ \hdashline C\alpha_p & \Lambda_0 - CA^p P[0](A^T)^p C^T \end{array}\right]^{-1} \left[\begin{array}{c} \alpha_p^T A^T \\ \hdashline \gamma_p^T \end{array}\right].$$

- We use the matrix inversion lemma on the middle term

$$\left[\begin{array}{c:c} \beta_p & \alpha_p^T C^T \\ \hdashline C\alpha_p & \Lambda_0 - CA^p P[0](A^T)^p C^T \end{array}\right]^{-1} = \left[\begin{array}{c:c} m_{11} & m_{12} \\ \hdashline m_{21} & m_{22} \end{array}\right]$$

$$m_{11} = \beta_p^{-1} + \beta_p^{-1}\alpha_p^T C^T \Delta^{-1} C\alpha_p \beta_p^{-1}$$

$$m_{12} = -\beta_p^{-1}\alpha_p^T C^T \Delta^{-1}$$

$$m_{21} = -\Delta^{-1} C\alpha_p \beta_p^{-1}$$

$$m_{22} = \Delta^{-1}$$

where

$$\Delta = \Lambda_0 - C\underbrace{\left[A^pP[0](A^T)^p + \alpha_p\beta_p^{-1}\alpha_p^T\right]}_{P[p]}C^T = \Lambda_0 - CP[p]C^T.$$

- Then,

$$\begin{aligned}
P[k+1] = {} & A\underbrace{\left[A^pP[0](A^T)^p + \alpha_p\beta_p^{-1}\alpha_p^T\right]}_{P[p]}A^T + \\
& (\gamma_p - A\alpha_p\beta_p^{-1}\alpha_p^TC^T)\Delta^{-1}(\gamma_p - A\alpha_p\beta_p^{-1}\alpha_p^TC^T)^T \\
= {} & AP[p]A^T + \Big(G - A\underbrace{\left[A^pP[0](A^T)^p + \alpha_p\beta_p^{-1}\alpha_p^T\right]}_{P[p]}C^T\Big)^T \times \\
& \Delta^{-1}\Big(G - A\underbrace{\left[A^pP[0](A^T)^p + \alpha_p\beta_p^{-1}\alpha_p^T\right]}_{P[p]}C^T\Big)^T \\
= {} & AP[p]A^T + (G - AP[p]C^T)(\Lambda_0 - CP[p]C^T)^{-1} \times \\
& (G - AP[p]C^T)^T,
\end{aligned}$$

which is what we were trying to prove.

- Therefore, the $P[k]$ recursion is proven.

6.9: Step 1b: Kalman-filter state

- Before completing the proof of the $\hat{x}[k]$ recursion, we need to see how to write Ω_{p+1}

$$\begin{aligned}\Omega_{p+1} &= \alpha_{p+1}\beta_{p+1}^{-1}\\ &= \left(\Im_{p+1}^{c} - A^{p+1}P[0]\mathcal{O}_{p+1}^{T}\right)\left(L_{p+1} - \mathcal{O}_{p+1}P[0]\mathcal{O}_{p+1}^{T}\right)^{-1}.\end{aligned}$$

- Note that we have already done these calculations as a part of computing $P[p+1]$.

$$\begin{aligned}\Omega_{p+1} &= \left[\begin{array}{c:c} A\alpha_p m_{11} + \gamma_p m_{21} & A\alpha_p m_{12} + \gamma_p m_{22}\end{array}\right]\\ &= \left[\begin{array}{c:c} v_1 & v_2 \end{array}\right]\\ v_1 &= A\alpha_p\beta_p^{-1} + A\alpha_p\beta_p^{-1}\alpha_p^T C^T\Delta^{-1}C\alpha_p\beta_p^{-1} - \gamma_p\Delta^{-1}C\alpha_p\beta_p^{-1}\\ &= A\Omega_p + A\Omega_p\alpha_p^T C^T\Delta^{-1}C\Omega_p - \gamma_p\Delta^{-1}C\Omega_p\\ &= A\Omega_p - (\gamma_p - A\Omega_p\alpha_p^T C^T)\Delta^{-1}C\Omega_p\\ &= A\Omega_p - (G - AP[p]C^T)(\Lambda_0 - CP[p]C^T)^{-1}C\Omega_p\\ &= (A - K[p]C)\Omega_p\\ v_2 &= -A\alpha_p\beta_p^{-1}\alpha_p^T C^T\Delta^{-1} + \gamma_p\Delta^{-1}\\ &= (\gamma_p - A\Omega_p\alpha_p^T C^T)\Delta^{-1}\\ &= (G - AP[p]C^T)(\Lambda_0 - CP[p]C^T)^{-1}\\ &= K[p].\end{aligned}$$

- So, $\Omega_{p+1} = \left[\begin{array}{c:c} (A - K[p]C)\Omega_p & K[p]\end{array}\right]$.
- We now use this result to rewrite $\hat{x}[p+1]$

$$\hat{x}[p+1] = \left(A^{p+1} - \Omega_{p+1}\mathcal{O}_{p+1}\right)\hat{x}[0] + \left(\Im^d_{p+1} - \Omega_{p+1}\Psi_{p+1}\right)\begin{bmatrix} u[0] \\ \vdots \\ u[p] \end{bmatrix}$$

$$+\Omega_{p+1}\begin{bmatrix} y[0] \\ \vdots \\ y[p] \end{bmatrix}$$

$$= \left(A^{p+1} - (A - K[p]C)\Omega_p\mathcal{O}_p - K[p]CA^p\right)\hat{x}[0] +$$

$$\left(A\Im^d_p - (A - K[p]C)\Omega_p\Psi_p - K[p]C\Im^d_p\right)\begin{bmatrix} u[0] \\ \vdots \\ u[p-1] \end{bmatrix}$$

$$+(B - K[p]D)u[p] +$$

$$(A - K[p]C)\,\Omega_p\begin{bmatrix} y[0] \\ \vdots \\ y[p-1] \end{bmatrix} + K[p]y[p]$$

$$= (A - K[p]C)\left((A^p - \Omega_p\mathcal{O}_p)\hat{x}[0] + (\Im^d_p - \Omega_p\Psi_p)\begin{bmatrix} u[0] \\ \vdots \\ u[p-1] \end{bmatrix}\right.$$

$$\left. + \Omega_p\begin{bmatrix} y[0] \\ \vdots \\ y[p-1] \end{bmatrix}\right) + (B - K[p]D)u[p] + K[p]y[p]$$

$$= (A - K[p]C)\hat{x}[p] + (B - K[p]D)u[p] + K[p]y[p]$$

$$= A\hat{x}[p] + Bu[p] + K[p]\left(y[p] - C\hat{x}[p] - Du[p]\right),$$

which completes the proof.

PERSPECTIVE: We have now shown that the time-varying Kalman filter, operating on the data $y[0]\cdots y[k]$ and $u[0]\cdots u[k]$ will produce the state estimates

$$\hat{x}[k] = \left[\begin{array}{c:c:c} A^k - \Omega_k \mathcal{O}_k & \mathfrak{I}_k^d - \Omega_k \Psi_k & \Omega_k \end{array}\right] \left[\begin{array}{c} \hat{x}[0] \\ \hdashline u[0] \\ \vdots \\ u[k-1] \\ \hdashline y[0] \\ \vdots \\ y[k-1] \end{array}\right],$$

where

$$\Omega_k \stackrel{\triangle}{=} (\mathfrak{I}_k^c - A^k P[0] \mathcal{O}_k^T)(L_k - \mathcal{O}_i P[0] \mathcal{O}_k^T)^{-1}.$$

- This is important: We will find this relationship showing up when we consider the orthogonal and oblique projections in the next sections.
- This means that these projections result in sequences of non-steady-state Kalman-filter state estimates, initialized with some initial state $\hat{x}[0]$ and covariance $P[0]$.
- This will become more clear when we organize the state estimates in a block matrix as

$$\begin{aligned} \widehat{X}_i &= \left[\begin{array}{cccc} \hat{x}[i], & \hat{x}[i+1], & \cdots & \hat{x}[i+N-1] \end{array}\right] \\ &= \left[\begin{array}{c:c:c} A^i - \Omega_i \mathcal{O}_i & \mathfrak{I}_i^d - \Omega_i \Psi_i & \Omega_i \end{array}\right] \left[\begin{array}{c} \widehat{X}_0 \\ \hdashline U_p \\ \hdashline Y_p \end{array}\right] \\ &= \left[\begin{array}{c:c} A^i - \Omega_i \mathcal{O}_i & \left[\begin{array}{c:c} \mathfrak{I}_i^d - \Omega_i \Psi_i & \Omega_i \end{array}\right] \end{array}\right] \left[\begin{array}{c} \widehat{X}_0 \\ \hdashline W_p \end{array}\right]. \end{aligned}$$

- This state sequence is generated by a bank of non-steady-state Kalman filters working in parallel on each of the columns of W_p as

$$
\begin{array}{rl}
\widehat{X}_0: & \left[\begin{array}{ccccc} \hat{x}^i[0] & \cdots & \hat{x}^i[q] & \cdots & \hat{x}^i[N-1] \end{array}\right] \quad \text{Kalman filter} \\
W_p: & \left[\begin{array}{c|c|c} u[0] & u[q] & u[N-1] \\ \vdots & \vdots & \vdots \\ u[i-1] & u[i+q-1] & u[i+N-2] \\ \hdashline y[0] & y[q] & y[N-1] \\ \vdots & \vdots & \vdots \\ y[i-1] \downarrow & y[i+q-1] \downarrow & y[i+N-2] \downarrow \end{array}\right] \quad \downarrow \\
\widehat{X}_i: & \left[\begin{array}{ccccc} \hat{x}[i] & \cdots & \hat{x}[i+q] & \cdots & \hat{x}[i+N-1] \end{array}\right],
\end{array}
$$

where $\hat{x}^i[k]$ denotes an "initial" state, as different from the Kalman-filter estimated state $\hat{x}[k]$.

- In what follows, we will encounter different Kalman-filter sequences (in the sense of different initial states $\widehat{X}[0]$ and covariances $P[0]$), so we will denote the Kalman-filter state sequence with initial state $\widehat{X}[0]$ and covariance matrix $P[0]$ as $\widehat{X}_{i_{[\widehat{X}[0],P[0]]}}$.

6.10: Step 2a: Orthogonal projection for combined systems ($\mathcal{A}$, $\mathcal{B}^{\Gamma 1}$)

- Now that we have seen what a Kalman-filter state sequence looks like, we will investigate some projection results, and find that these state sequences show up naturally.
- We start with

$$\begin{aligned} Y_p &= \mathcal{O}_i(X_p^d + X_p^s) + \Psi_i U_p + \Psi_i^w N_p^w + N_p^v \\ &= \mathcal{O}_i X_p^d + \Psi_i U_p + \underbrace{\mathcal{O}_i X_p^s + \Psi_i^w N_p^w + N_p^v}_{Y_p^s} \\ Y_f &= \mathcal{O}_i X_f^d + \Psi_i U_f + Y_f^s \\ X_f^d &= A^i X_p^d + \mathfrak{I}_i^d U_p. \end{aligned}$$

- Over the next pages, our goal is to show that

$$\mathcal{Z}_i \triangleq Y_f / \begin{bmatrix} \boldsymbol{W_p} \\ \boldsymbol{U_f} \end{bmatrix} = \mathcal{O}_i \widehat{X}_i + \Psi_i U_f,$$

where

$$\begin{aligned} \widehat{X}_i &\triangleq \widehat{X}_{i_{[\widehat{X}[0], P[0]]}} \\ \widehat{X}[0] &= S^{xu}(R^{uu})^{-1} \begin{pmatrix} U_p \\ U_f \end{pmatrix} \\ P[0] &= -\left[\Sigma^d - S^{xu}(R^{uu})^{-1}(S^{xu})^T\right], \end{aligned}$$

where

$$\begin{aligned} R^{uu} &\triangleq \Phi_{[U_{0|2i-1}, U_{0|2i-1}]} = \begin{bmatrix} \Phi_{[U_p,U_p]} & \Phi_{[U_p,U_f]} \\ \Phi_{[U_f,U_p]} & \Phi_{[U_f,U_f]} \end{bmatrix} = \begin{bmatrix} R_p^{uu} & R_{pf}^{uu} \\ (R_{pf}^{uu})^T & R_{ff}^{uu} \end{bmatrix} \\ S^{xu} &= \Phi_{[X_p^d, U_{0|2i-1}]} = \begin{bmatrix} \Phi_{[X_p^d,U_p]} & \Phi_{[X_p^d,U_f]} \end{bmatrix} = \begin{bmatrix} S_p^{xu} & S_f^{xu} \end{bmatrix} \\ \Sigma^d &\triangleq \Phi_{[X_p^d, X_p^d]}. \end{aligned}$$

- The importance of this result is that it shows *one* way in which the Kalman-filter state sequence $\widehat{X}_i$ relates to given input-output data.
- The projected matrix $\mathcal{Z}_i$ can indeed be computed from the given data, without knowing the system matrices.
- We assume that the deterministic input $u[k]$ is uncorrelated with the process noise $w[k]$ and the sensor noise $v[k]$.
- We also assume the input is persistently exciting of order $2i$, and that the number of measurements $N \to \infty$.
- The first assumption gives us

$$\begin{aligned}
\mathbb{E}_N[Y_p^s U_p^T] &= \mathbb{E}_N[Y_p^s U_f^T] = 0\\
\mathbb{E}_N[Y_f^s U_p^T] &= \mathbb{E}_N[Y_f^s U_f^T] = 0\\
\mathbb{E}_N[Y_p^s (X_p^d)^T] &= \mathbb{E}_N[Y_p^s (X_f^d)^T] = 0\\
\mathbb{E}_N[Y_f^s (X_p^d)^T] &= \mathbb{E}_N[Y_f^s (X_f^d)^T] = 0.
\end{aligned}$$

- We wish to compute

$$\mathcal{Z}_i = Y_f / \begin{bmatrix} \boldsymbol{U_p} \\ \boldsymbol{U_f} \\ \boldsymbol{Y_p} \end{bmatrix} \triangleq \mathcal{A}\mathcal{B}^{-1} \begin{bmatrix} U_p \\ U_f \\ Y_p \end{bmatrix},$$

where

$$\mathcal{A} = \mathbb{E}_N\left[Y_f \left[\begin{array}{c:c:c} U_p^T & U_f^T & Y_p^T \end{array} \right] \right] = \left[\begin{array}{c:c:c} \mathcal{A}_1 & \mathcal{A}_2 & \mathcal{A}_3 \end{array} \right]$$

$$\mathcal{B} = \mathbb{E}_N\left[\left[\begin{array}{c} U_p \\ U_f \\ \hdashline Y_p \end{array} \right] \left[\begin{array}{cc:c} U_p^T & U_f^T & Y_p^T \end{array} \right] \right] = \left[\begin{array}{c:c} \mathcal{B}_{11} & \mathcal{B}_{12} \\ \hdashline \mathcal{B}_{21} & \mathcal{B}_{22} \end{array} \right].$$

- We now compute $\mathcal{A}_1$, $\mathcal{A}_2$, and $\mathcal{A}_3$,

$$\begin{aligned}
\mathcal{A}_1 &= \mathbb{E}_N[Y_f U_p^T] = \mathbb{E}_N[(\mathcal{O}_i A^i X_p^d + \mathcal{O}_i \Im_i^d U_p + \Psi_i U_f + Y_f^s) U_p^T] \\
&= \mathcal{O}_i A^i S_p^{xu} + \mathcal{O}_i \Im_i^d R_p^{uu} + \Psi_i (R_{pf}^{uu})^T \\
\mathcal{A}_2 &= \mathbb{E}_N[Y_f U_f^T] = \mathbb{E}_N[(\mathcal{O}_i A^i X_p^d + \mathcal{O}_i \Im_i^d U_p + \Psi_i U_f + Y_f^s) U_f^T] \\
&= \mathcal{O}_i A^i S_f^{xu} + \mathcal{O}_i \Im_i^d R_{pf}^{uu} + \Psi_i R_f^{uu} \\
\mathcal{A}_3 &= \mathbb{E}_N[Y_f Y_p^T] \\
&= \mathbb{E}_N[(\mathcal{O}_i A^i X_p^d + \mathcal{O}_i \Im_i^d U_p + \Psi_i U_f + Y_f^s)((X_p^d)^T \mathcal{O}_i^T + U_p^T \Psi_i^T + (Y_p^s)^T)] \\
&= \mathcal{O}_i A^i \Sigma^d \mathcal{O}_i^T + \mathcal{O}_i A^i S_p^{xu} \Psi_i^T + \mathcal{O}_i \Im_i^d (S_p^{xu})^T \mathcal{O}_i^T + \mathcal{O}_i \Im_i^d R_p^{uu} \Psi_i^T \\
&\qquad + \Psi_i (S_f^{xu})^T \mathcal{O}_i^T + \Psi_i (R_{pf}^{uu})^T \Psi_i^T + C_i \\
&= \mathcal{O}_i A^i \Sigma^d \mathcal{O}_i^T + \mathcal{O}_i A^i S_p^{xu} \Psi_i^T + \mathcal{O}_i \Im_i^d (S_p^{xu})^T \mathcal{O}_i^T + \mathcal{O}_i \Im_i^d R_p^{uu} \Psi_i^T \\
&\qquad + \Psi_i (S_f^{xu})^T \mathcal{O}_i^T + \Psi_i (R_{pf}^{uu})^T \Psi_i^T + \mathcal{O}_i \Im_i^c.
\end{aligned}$$

- Similarly, we compute $\mathcal{B}_{11}$, $\mathcal{B}_{12}$, $\mathcal{B}_{21}$, and $\mathcal{B}_{22}$.

$$\begin{aligned}
\mathcal{B}_{11} &= \mathbb{E}_N[U_{0|2i-i} U_{0|2i-1}^T] = R^{uu} \\
\mathcal{B}_{21} &= \mathbb{E}_N\left[Y_p \left[\begin{array}{cc} U_p^T & U_f^T \end{array}\right]\right] = \mathbb{E}_N\left[(\mathcal{O}_i X_p^d + \Psi_i U_p + Y_p^s)\left[\begin{array}{cc} U_p^T & U_f^T \end{array}\right]\right] \\
&= \mathcal{O}_i S^{xu} + \Psi_i \left[\begin{array}{cc} R_p^{uu} & R_{pf}^{uu} \end{array}\right] \\
\mathcal{B}_{12} &= \mathcal{B}_{21}^T \\
\mathcal{B}_{22} &= \mathbb{E}_N[Y_p Y_p^T] \\
&= \mathbb{E}_N[(\mathcal{O}_i X_p^d + \Psi_i U_p + Y_p^s)((X_p^d)^T \mathcal{O}_i^T + U_p^T \Psi_i^T + (Y_p^s)^T)] \\
&= \mathcal{O}_i \Sigma^d \mathcal{O}_i^T + \mathcal{O}_i S_p^{xu} \Psi_i^T + \Psi_i (S_p^{xu})^T \mathcal{O}_i^T + \Psi_i R_p^{uu} \Psi_i^T + L_i.
\end{aligned}$$

- Note that $\mathcal{B}$ is guaranteed to be of full rank due to the persistently exciting input and the non-zero stochastic system.
- To compute $\mathcal{B}^{-1}$, we again make use of the matrix inversion lemma

$$\begin{bmatrix} \mathcal{B}_{11} & \mathcal{B}_{21}^T \\ \mathcal{B}_{21} & \mathcal{B}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}$$

$$\begin{aligned} m_{11} &= \mathcal{B}_{11}^{-1} + \mathcal{B}_{11}^{-1}\mathcal{B}_{21}^T\Delta^{-1}\mathcal{B}_{21}\mathcal{B}_{11}^{-1} \\ m_{12} &= -\mathcal{B}_{11}^{-1}\mathcal{B}_{21}^T\Delta^{-1} \\ m_{21} &= -\Delta^{-1}\mathcal{B}_{21}\mathcal{B}_{11}^{-1} \\ m_{22} &= \Delta^{-1} \qquad \text{where} \quad \Delta = \mathcal{B}_{22} - \mathcal{B}_{21}\mathcal{B}_{11}^{-1}\mathcal{B}_{21}^T. \end{aligned}$$

- In our case, this becomes:

$$\begin{aligned} m_{11} &= (R^{uu})^{-1} + \left(\begin{bmatrix} I \\ 0 \end{bmatrix}\Psi_i^T + (R^{uu})^{-1}(S^{xu})^T\mathcal{O}_i^T\right)\Delta^{-1} \times \\ &\qquad \left(\Psi_i \begin{bmatrix} I & 0 \end{bmatrix} + \mathcal{O}_i S^{xu}(R^{uu})^{-1}\right) \\ m_{12} &= -\left(\begin{bmatrix} I \\ 0 \end{bmatrix}\Psi_i^T + (R^{uu})^{-1}(S^{xu})^T\mathcal{O}_I^T\right)\Delta^{-1} \\ m_{21} &= -\Delta^{-1}\left(\Psi_i \begin{bmatrix} I & 0 \end{bmatrix} + \mathcal{O}_i S^{xu}(R^{uu})^{-1}\right) \\ m_{22} &= \Delta^{-1}, \end{aligned}$$

with

$$\begin{aligned} \Delta = {} & \mathcal{O}_i\Sigma^d\mathcal{O}_i^T + L_i + \mathcal{O}_i S_p^{xu}\Psi_i^T + \Psi_i(S_p^{xu})^T\mathcal{O}_i^T + \Psi_i R_p^{uu}\Psi_i^T \\ & - \left(\mathcal{O}_i S^{xu} + \Psi_i \begin{bmatrix} R_p^{uu} & R_{pf}^{uu} \end{bmatrix}\right)(R^{uu})^{-1}\left((S^{xu})^T\mathcal{O}_i^T + \begin{bmatrix} R_p^{uu} \\ (R_{pf}^{uu})^T \end{bmatrix}\Psi_i^T\right) \end{aligned}$$

$$= L_i - \mathcal{O}_i \underbrace{\left(S^{xu}(R^{uu})^{-1}(S^{xu})^T - \Sigma^d\right)}_{P[0]} \mathcal{O}_i^T.$$

6.11: Step 2b: Orthogonal projection for combined systems...

- Recall that we wish to compute

$$\mathcal{Z}_i = Y_f / \begin{bmatrix} \boldsymbol{U_p} \\ \boldsymbol{U_f} \\ \boldsymbol{Y_p} \end{bmatrix} \triangleq \mathcal{A}\mathcal{B}^{-1} \begin{bmatrix} U_p \\ U_f \\ Y_p \end{bmatrix},$$

and we are now ready to compute $\mathcal{A}\mathcal{B}^{-1}$.

$$\begin{aligned} \mathcal{A}\mathcal{B}^{-1} &= \begin{bmatrix} \begin{bmatrix} \mathcal{A}_1 & \mathcal{A}_2 \end{bmatrix} & \mathcal{A}_3 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \\ &= \begin{bmatrix} \begin{bmatrix} \mathcal{A}_1 & \mathcal{A}_2 \end{bmatrix} m_{11} + \mathcal{A}_3 m_{21}, & \begin{bmatrix} \mathcal{A}_1 & \mathcal{A}_2 \end{bmatrix} m_{12} + \mathcal{A}_3 m_{22} \end{bmatrix} \\ &\triangleq \begin{bmatrix} \begin{bmatrix} L_{U_p} & L_{U_f} \end{bmatrix} & L_{Y_p} \end{bmatrix}. \end{aligned}$$

- We start with $\begin{bmatrix} L_{U_p} & L_{U_f} \end{bmatrix} = \begin{bmatrix} \mathcal{A}_1 & \mathcal{A}_2 \end{bmatrix} m_{11} + \mathcal{A}_3 m_{21}$. Note that we can write

$$\begin{aligned} \begin{bmatrix} \mathcal{A}_1 & \mathcal{A}_2 \end{bmatrix} &= \mathcal{O}_i A^i \begin{bmatrix} S_p^{xu} & S_f^{xu} \end{bmatrix} + \mathcal{O}_i \Im_i^d \begin{bmatrix} R_p^{uu} & R_{pf}^{uu} \end{bmatrix} + \Psi_i \begin{bmatrix} (R_{pf}^{uu})^T & R_f^{uu} \end{bmatrix} \\ &= \mathcal{O}_i A^i S^{xu} + \mathcal{O}_i \Im_i^d \begin{bmatrix} I & 0 \end{bmatrix} R^{uu} + \Psi_i \begin{bmatrix} 0 & I \end{bmatrix} R^{uu} \end{aligned}$$

$$\begin{bmatrix} m_{11} \\ m_{21} \end{bmatrix} = \begin{bmatrix} (R^{uu})^{-1} \\ 0 \end{bmatrix} - \begin{bmatrix} \begin{bmatrix} I \\ 0 \end{bmatrix} \Psi_i^T + (R^{uu})^{-1} (S^{xu})^T \mathcal{O}_i^T \\ -I \end{bmatrix} m_{21}.$$

- Putting these together, we can write

$$\begin{aligned} &\begin{bmatrix} L_{U_p} & L_{U_f} \end{bmatrix} = \\ &\quad \left(\mathcal{O}_i A^i S^{xu} + \mathcal{O}_i \Im_i^d \begin{bmatrix} I & 0 \end{bmatrix} R^{uu} + \Psi_i \begin{bmatrix} 0 & I \end{bmatrix} R^{uu} \right) (R^{uu})^{-1} \\ &\quad - \left(\mathcal{O}_i A^i \begin{bmatrix} S_p^{xu} & S_f^{xu} \end{bmatrix} + \mathcal{O}_i \Im_i^d \begin{bmatrix} R_p^{uu} & R_{pf}^{uu} \end{bmatrix} + \Psi_i \begin{bmatrix} (R_{pf}^{uu})^T & R_f^{uu} \end{bmatrix} \right) \times \end{aligned}$$

$$
\begin{aligned}
&\begin{bmatrix} I \\ 0 \end{bmatrix} \Psi_i^T m_{21} \\
&- \Big(\mathcal{O}_i A^i S^{xu} + \begin{bmatrix} \mathcal{O}_i \Im_i^d & 0 \end{bmatrix} R^{uu} + \begin{bmatrix} 0 & \Psi_i \end{bmatrix} R^{uu} \Big) (R^{uu})^{-1} (S^{xu})^T \mathcal{O}_i^T m_{21} \\
&+ \Big[\mathcal{O}_i A^i \Sigma^d \mathcal{O}_i^T + \mathcal{O}_i A^i S_p^{xu} \Psi_i^T + \mathcal{O}_i \Im_i^d (S_p^{xu})^T \mathcal{O}_i^T + \mathcal{O}_i \Im_i^d R_p^{uu} \Psi_i^T \\
&\quad + \Psi_i (S_f^{xu})^T \mathcal{O}_i^T + \Psi_i (R_{pf}^{uu})^T \Psi_i^T + \mathcal{O}_i \Im_i^c \Big] m_{21} \\
&= \Big(\mathcal{O}_i A^i S^{xu} (R^{uu})^{-1} + \begin{bmatrix} \mathcal{O}_i \Im_i^d, & \Psi_i \end{bmatrix} \Big) \\
&- \Big(\underbrace{\mathcal{O}_i A^i S_p^{xu} \Psi_i^T}_{\alpha} + \underbrace{\mathcal{O}_i \Im_i^d R_p^{uu} \Psi_i^T}_{\beta} + \underbrace{\Psi_i (R_{pf}^{uu})^T \Psi_i^T}_{\gamma} \Big) m_{21} \\
&- \Big(\mathcal{O}_i A^i S^{xu} (R^{uu})^{-1} (S^{xu})^T \mathcal{O}_i^T + \underbrace{\begin{bmatrix} \mathcal{O}_i \Im_i^d, & \Psi_i \end{bmatrix} (S^{xu})^T \mathcal{O}_i^T}_{\delta_{1:2}} \Big) m_{21} \\
&+ \Big[\mathcal{O}_i A^i \Sigma^d \mathcal{O}_i^T + \underbrace{\mathcal{O}_i A^i S_p^{xu} \Psi_i^T}_{\alpha} + \underbrace{\mathcal{O}_i \Im_i^d (S_p^{xu})^T \mathcal{O}_i^T}_{\delta_1} + \underbrace{\mathcal{O}_i \Im_i^d R_p^{uu} \Psi_i^T}_{\beta} \\
&\quad + \underbrace{\Psi_i (S_f^{xu})^T \mathcal{O}_i^T}_{\delta_2} + \underbrace{\Psi_i (R_{pf}^{uu})^T \Psi_i^T}_{\gamma} + \mathcal{O}_i \Im_i^c \Big] m_{21}.
\end{aligned}
$$

- Notice the cancellations α, β, γ, δ_1, and δ_2. We can write the simplified equation as

$$
\begin{bmatrix} L_{U_p} & L_{U_f} \end{bmatrix} = \Big(\mathcal{O}_i A^i S^{xu} (R^{uu})^{-1} + \begin{bmatrix} \mathcal{O}_i \Im_i^d, & \Psi_i \end{bmatrix} \Big)
$$

$$
\begin{aligned}
&+\mathcal{O}_i\left(\Im_i^c - A^i\underbrace{\left(S^{xu}(R^{uu})^{-1}(S^{xu})^T - \Sigma^d\right)}_{P[0]}\mathcal{O}_i^T\right)m_{21}\\
&= \left(\mathcal{O}_i A^i S^{xu}(R^{uu})^{-1} + \begin{bmatrix} \mathcal{O}_i\Im_i^d, & \Psi_i \end{bmatrix}\right)\\
&\quad +\mathcal{O}_i\left(\Im_i^c - A^i P[0]\mathcal{O}_i^T\right)m_{21}\\
&= \left(\mathcal{O}_i A^i S^{xu}(R^{uu})^{-1} + \begin{bmatrix} \mathcal{O}_i\Im_i^d, & \Psi_i \end{bmatrix}\right)\\
&\quad -\mathcal{O}_i\left(\Im_i^c - A^i P[0]\mathcal{O}_i^T\right)\left(L_i - \mathcal{O}_i P[0]\mathcal{O}_i^T\right)^{-1}\times\\
&\qquad \left(\Psi_i\begin{bmatrix} I & 0 \end{bmatrix} + \mathcal{O}_i S^{xu}(R^{uu})^{-1}\right)\\
&= \left(\mathcal{O}_i A^i S^{xu}(R^{uu})^{-1} + \begin{bmatrix} \mathcal{O}_i\Im_i^d, & \Psi_i \end{bmatrix}\right)\\
&\quad -\mathcal{O}_i\Omega_i\left(\Psi_i\begin{bmatrix} I & 0 \end{bmatrix} + \mathcal{O}_i S^{xu}(R^{uu})^{-1}\right)\\
&= \begin{bmatrix} \mathcal{O}_i\left(\Im_i^d - \Omega_i\Psi_i\right), & \Psi_i \end{bmatrix} + \mathcal{O}_i\left(A^i - \Omega_i\mathcal{O}_i\right)S^{xu}(R^{uu})^{-1}.
\end{aligned}
$$

6.12: Step 2c: Orthogonal projection for combined systems. . .

- Whew. Now we proceed to find $L_{Y_p} = \begin{bmatrix} \mathcal{A}_1 & \mathcal{A}_2 \end{bmatrix} m_{12} + \mathcal{A}_3 m_{22}$. First note,

$$\begin{bmatrix} m_{12} \\ m_{22} \end{bmatrix} = -\begin{bmatrix} \begin{bmatrix} \Psi_i \\ 0 \end{bmatrix} + (R^{uu})^{-1}(S^{xu})^T \mathcal{O}_i^T \\ -I \end{bmatrix} m_{22}.$$

- This gives us

$$\begin{aligned}
L_{Y_p} m_{22}^{-1} &= -\left(\mathcal{O}_i A^i S^{xu} + \begin{bmatrix} \mathcal{O}_i \Im_i^d & 0 \end{bmatrix} R^{uu} + \begin{bmatrix} 0 & \Psi_i \end{bmatrix} R^{uu}\right) \times \\
&\qquad \left(\begin{bmatrix} \Psi_i^T \\ 0 \end{bmatrix} + (R^{uu})^{-1}(S^{xu})^T \mathcal{O}_i^T\right) \\
&\qquad + \Big[\mathcal{O}_i A^i \Sigma^d \mathcal{O}_i^T + \mathcal{O}_i A^i S_p^{xu} \Psi_i^T + \mathcal{O}_i \Im_i^d (S_p^{xu})^T \mathcal{O}_i^T + \mathcal{O}_i \Im_i^d R_p^{uu} \Psi_i^T \\
&\qquad + \Psi_i (S_f^{xu})^T \mathcal{O}_i^T + \Psi_i (R_{pf}^{uu})^T \Psi_i^T + \mathcal{O}_i \Im_i^c\Big] \\
&= -\mathcal{O}_i A^i S^{xu} \left(\begin{bmatrix} \Psi_i^T \\ 0 \end{bmatrix} + (R^{uu})^{-1}(S^{xu})^T \mathcal{O}_i^T\right) \\
&\qquad - \begin{bmatrix} \mathcal{O}_i \Im_i^d & 0 \end{bmatrix} R^{uu} \left(\begin{bmatrix} \Psi_i^T \\ 0 \end{bmatrix} + (R^{uu})^{-1}(S^{xu})^T \mathcal{O}_i^T\right) \\
&\qquad - \begin{bmatrix} 0 & \Psi_i \end{bmatrix} R^{uu} \left(\begin{bmatrix} \Psi_i^T \\ 0 \end{bmatrix} + (R^{uu})^{-1}(S^{xu})^T \mathcal{O}_i^T\right) \\
&\qquad + \Big[\mathcal{O}_i A^i \Sigma^d \mathcal{O}_i^T + \mathcal{O}_i A^i S_p^{xu} \Psi_i^T + \mathcal{O}_i \Im_i^d (S_p^{xu})^T \mathcal{O}_i^T + \mathcal{O}_i \Im_i^d R_p^{uu} \Psi_i^T \\
&\qquad + \Psi_i (S_f^{xu})^T \mathcal{O}_i^T + \Psi_i (R_{pf}^{uu})^T \Psi_i^T + \mathcal{O}_i \Im_i^c\Big]
\end{aligned}$$

$$
\begin{aligned}
&= -\mathcal{O}_i A^i S^{xu} \begin{bmatrix} \Psi_i^T \\ 0 \end{bmatrix} - \mathcal{O}_i A^i S^{xu} (R^{uu})^{-1} (S^{xu})^T \mathcal{O}_i^T \\
&\quad - \begin{bmatrix} \mathcal{O}_i \Im_i^d & 0 \end{bmatrix} R^{uu} \begin{bmatrix} \Psi_i^T \\ 0 \end{bmatrix} - \begin{bmatrix} \mathcal{O}_i \Im_i^d & 0 \end{bmatrix} (S^{xu})^T \mathcal{O}_i^T \\
&\quad - \begin{bmatrix} 0 & \Psi_i \end{bmatrix} R^{uu} \begin{bmatrix} \Psi_i^T \\ 0 \end{bmatrix} - \begin{bmatrix} 0 & \Psi_i \end{bmatrix} (S^{xu})^T \mathcal{O}_i^T \\
&\quad + \Bigg[\mathcal{O}_i A^i \Sigma^d \mathcal{O}_i^T + \mathcal{O}_i A^i S_p^{xu} \Psi_i^T + \mathcal{O}_i \Im_i^d (S_p^{xu})^T \mathcal{O}_i^T + \mathcal{O}_i \Im_i^d R_p^{uu} \Psi_i^T \\
&\qquad + \Psi_i (S_f^{xu})^T \mathcal{O}_i^T + \Psi_i (R_{pf}^{uu})^T \Psi_i^T + \mathcal{O}_i \Im_i^c \Bigg] \\
&= -\underbrace{\mathcal{O}_i A^i S_p^{xu} \Psi_i}_{\alpha} - \mathcal{O}_i A^i S^{xu} (R^{uu})^{-1} (S^{xu})^T \mathcal{O}_i^T \\
&\quad - \underbrace{\mathcal{O}_i \Im_i^d R_p^{uu} \Psi_i^T}_{\beta} - \underbrace{\mathcal{O}_i \Im_i^d (S_p^{xu})^T \mathcal{O}_i^T}_{\gamma} \\
&\quad - \underbrace{\Psi_i (R_{pf}^{uu}) \Psi_i^T}_{\delta} - \underbrace{\Psi_i (S_f^{xu})^T \mathcal{O}_i^T}_{\epsilon} \\
&\quad + \Bigg[\mathcal{O}_i A^i \Sigma^d \mathcal{O}_i^T + \underbrace{\mathcal{O}_i A^i S_p^{xu} \Psi_i^T}_{\alpha} + \underbrace{\mathcal{O}_i \Im_i^d (S_p^{xu})^T \mathcal{O}_i^T}_{\gamma} + \underbrace{\mathcal{O}_i \Im_i^d R_p^{uu} \Psi_i^T}_{\beta} \\
&\qquad + \underbrace{\Psi_i (S_f^{xu})^T \mathcal{O}_i^T}_{\epsilon} + \underbrace{\Psi_i (R_{pf}^{uu})^T \Psi_i^T}_{\delta} + \mathcal{O}_i \Im_i^c \Bigg].
\end{aligned}
$$

- Again, note all the cancellations $\alpha \cdots \epsilon$. With these terms eliminated, we find

$$
\begin{aligned}
L_{Y_p} &= \mathcal{O}_i \left(\Im_i^c - A^i \underbrace{S^{xu}(R^{uu})^{-1}(S^{xu})^T - \Sigma^d}_{P[0]} \mathcal{O}_i^T \right) m_{22} \\
&= \mathcal{O}_i \left(\Im_i^c - A^i P[0] \mathcal{O}_i^T \right) \left(L_i - \mathcal{O}_i P[0] \mathcal{O}_i^T \right)^{-1} \\
&= \mathcal{O}_i \Omega_i .
\end{aligned}
$$

- We're nearly there now. All that remains is to put these results together to compute

$$
\begin{aligned}
\mathcal{Z}_i &= Y_f / \begin{pmatrix} \boldsymbol{U_p} \\ \boldsymbol{U_f} \\ \boldsymbol{Y_p} \end{pmatrix} \triangleq \mathcal{A}\mathcal{B}^{-1} \begin{bmatrix} U_p \\ U_f \\ Y_p \end{bmatrix} \\
&= \begin{bmatrix} \begin{bmatrix} L_{U_p} & L_{U_f} \end{bmatrix} & L_{Y_p} \end{bmatrix} \begin{bmatrix} \begin{bmatrix} U_p \\ U_f \end{bmatrix} \\ Y_p \end{bmatrix} \\
&= \left(\begin{bmatrix} \mathcal{O}_i \left(\Im_i^d - \Omega_i \Psi_i \right), & \Psi_i \end{bmatrix} + \mathcal{O}_i \left(A^i - \Omega_i \mathcal{O}_i \right) S^{xu}(R^{uu})^{-1} \right) \begin{bmatrix} U_p \\ U_f \end{bmatrix} \\
&\quad + \mathcal{O}_i \Omega_i Y_p \\
&= \mathcal{O}_i \left(\left(\Im_i^d - \Omega_i \Psi_i \right) U_p + \left(A^i - \Omega_i \mathcal{O}_i \right) \underbrace{S^{xu}(R^{uu})^{-1} \begin{bmatrix} U_p \\ U_f \end{bmatrix}}_{\widehat{X}[0]} + \Omega_i Y_p \right) \\
&\quad + \Psi_i U_f .
\end{aligned}
$$

- Recall that we showed earlier that the non-steady-state Kalman-filter state sequence can be expressed as

$$
\widehat{X}_i = \begin{bmatrix} A^i - \Omega_i \mathcal{O}_i, & \begin{bmatrix} \Im_i^d - \Omega_i \Psi_i, & \Omega_i \end{bmatrix} \end{bmatrix} \begin{bmatrix} \widehat{X}_0 \\ W_p \end{bmatrix}.
$$

- This allows us to rewrite our expression for $\mathcal{Z}_i$ as

$$\boxed{\mathcal{Z}_i = Y_f / \begin{bmatrix} \boldsymbol{W_p} \\ \boldsymbol{U_f} \end{bmatrix} = \mathcal{O}_i \widehat{X}_i + \Psi_i U_f,}$$

where $\widehat{X}[0] = S^{xu}(R^{uu})^{-1} \begin{bmatrix} U_p \\ U_f \end{bmatrix}$, and $P[0] = S^{xu}(R^{uu})^{-1}(S^{xu})^T - \Sigma^d$, which, if you still recall, is what we were trying to prove.

- Note that we can write $\widehat{X}[0]$ differently:

$$\begin{aligned} \widehat{X}[0] &= S^{xu}(R^{uu})^{-1} \begin{bmatrix} U_p \\ U_f \end{bmatrix} \\ &= \Phi_{[X_p^d, U_{0|2i-1}]} \Phi^{\dagger}_{[U_{0|2i-1}]} \begin{bmatrix} U_p \\ U_f \end{bmatrix} \\ &= X_p^d / \begin{bmatrix} \boldsymbol{U_p} \\ \boldsymbol{U_f} \end{bmatrix}. \end{aligned}$$

- The "real" initial state is $X_p^d + X_p^s$, but we have no way of predicting X_p^s, so this projection sets it to zero.
- This projection uses the $\widehat{X}[0]$ that is the best estimate of X_p^d lying in the row space of past and future inputs.

6.13: Step 3: Oblique projections for combined systems

- As mentioned before, we will need to be able to compute both orthogonal and oblique projections to arrive at our combined algorithm.
- Here, we consider the oblique projection $\xi_i \triangleq Y_f /_{U_f} \boldsymbol{W_p}$.
- We will show that $\xi_i = \mathcal{O}_i \widetilde{X}_i$, where $\widetilde{X}_i$ is the output of a non-steady-state Kalman filter with initial conditions

$$\widehat{X}[0] = X_p^d /_{U_f} \boldsymbol{U_p}$$

$$P[0] = S^{xu}(R^{uu})^{-1}(S^{xu})^T - \Sigma^d.$$

- Note that the initial covariance of this filter is the same as for the orthogonal projection case, but the initial states are different. Therefore, the state sequence $\widetilde{X}_i$ will be different from the state sequence $\widehat{X}_i$. This is a subtle but important point.
- Fortunately, most of the "heavy lifting" has already been done. We take advantage of our prior work.
- First, we establish a relationship between orthogonal and oblique projections

$$X_p^d / \begin{bmatrix} \boldsymbol{U_p} \\ \boldsymbol{U_f} \end{bmatrix} = \mathbb{E}_N \left[X_p^d \begin{bmatrix} U_p^T & U_f^T \end{bmatrix} \begin{bmatrix} U_p U_p^T & U_p U_f^T \\ \hline U_f U_p^T & U_f U_p^T \end{bmatrix}^{\dagger} \begin{bmatrix} U_p \\ \hline U_f \end{bmatrix} \right]$$

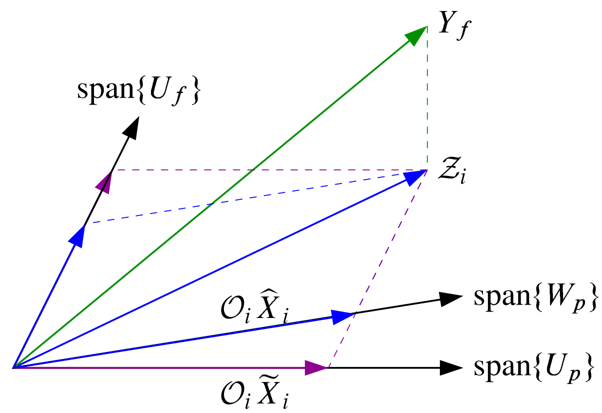
$$= \mathbb{E}_N \left[X_p^d \begin{bmatrix} U_p^T & U_f^T \end{bmatrix} \begin{bmatrix} U_p U_p^T & U_p U_f^T \\ \hline U_f U_p^T & U_f U_p^T \end{bmatrix}^{\dagger} \begin{bmatrix} U_p \\ \hline 0 \end{bmatrix} \right]$$

$$+ \mathbb{E}_N \left[X_p^d \begin{bmatrix} U_p^T & U_f^T \end{bmatrix} \begin{bmatrix} U_p U_p^T & U_p U_f^T \\ \hline U_f U_p^T & U_f U_p^T \end{bmatrix}^{\dagger} \begin{bmatrix} 0 \\ \hline U_f \end{bmatrix} \right]$$

$$= X_p^d/_{U_f}\boldsymbol{U_p} + X_p^d/_{U_p}\boldsymbol{U_f},$$

where we have used a relationship from VODM chapter 1 for the oblique projection that is different from what we have previously used.

- Therefore, if we can break up our prior relationship for $\widehat{X}[0] = X_p^d/\begin{bmatrix}\boldsymbol{U_p}\\ \boldsymbol{U_f}\end{bmatrix}$ into parts that are linear in U_p and U_f, we can then determine the desired projection.
- The figure shows the relationships.



- The orthogonal projection creates $\mathcal{Z}_i$.
- But, $\widehat{X}_i$ has components related to U_f, so the oblique projection of Y_f along the row space of U_f onto the row space of U_p will result in a different $\mathcal{O}_i\widetilde{X}_i$.
- To find $\widetilde{X}_i$, we start with the expression that we have previously proven is true for a non-steady-state Kalman-filter state sequence

$$\widehat{X}_i = \left[\begin{array}{c:c} A^i - \Omega_i\mathcal{O}_i & \left[\begin{array}{c:c}\mathfrak{J}_i^d - \Omega_i\Psi_i & \Omega_i\end{array}\right]\end{array}\right]\left[\begin{array}{c}\widehat{X}_0\\ \hdashline W_p\end{array}\right]$$

$$= \left(A^i - \Omega_i\mathcal{O}_i\right)X_p^d/\begin{bmatrix}\boldsymbol{U_p}\\ \boldsymbol{U_f}\end{bmatrix} + \left[\begin{array}{c:c}\mathfrak{J}_i^d - \Omega_i\Psi_i & \Omega_i\end{array}\right]W_p.$$

- We substitute this result into $\mathcal{Z}_i = \mathcal{O}_i\widehat{X}_i + \Psi_i U_f$

$$\mathcal{Z}_i = \mathcal{O}_i\left(A^i - \Omega_i\mathcal{O}_i\right)X_p^d/\begin{bmatrix}\boldsymbol{U_p}\\ \boldsymbol{U_f}\end{bmatrix}$$

$$+ \mathcal{O}_i\left[\begin{array}{cc}\mathfrak{J}_i^d - \Omega_i\Psi_i, & \Omega_i\end{array}\right]W_p + \Psi_i U_f.$$

$$= \mathcal{O}_i\left(A^i - \Omega_i\mathcal{O}_i\right)\left(X_p^d/_{U_f}\boldsymbol{U_p} + X_p^d/_{U_p}\boldsymbol{U_f}\right)$$

$$+\,\mathcal{O}_i\left[\ \Im_i^d - \Omega_i\Psi_i \ \vdots\ \Omega_i\ \right]W_p + \Psi_i U_f.$$

- We drop those terms that are linear in U_f to give the oblique projection

$$\begin{aligned}\xi_i &= \mathcal{O}_i\left(A^i - \Omega_i\mathcal{O}_i\right)\left(X_p^d/_{U_f}\boldsymbol{U_p}\right) + \left[\ \Im_i^d - \Omega_i\Psi_i \ \vdots\ \Omega_i\ \right]W_p \\ &= \mathcal{O}_i\widetilde{X}_i,\end{aligned}$$

  if we define

$$\widetilde{X}_i = \left[\ A^i - \Omega_i\mathcal{O}_i \ \vdots\ \left[\ \Im_i^d - \Omega_i\Psi_i \ \vdots\ \Omega_i\ \right]\right]\left[\begin{array}{c} X_p^d/_{U_f}\boldsymbol{U_p} \\ \hdashline W_p \end{array}\right],$$

  which satisfies the equation for a non-steady-state Kalman-filter state sequence with

$$\widehat{X}[0] = X_p^d/_{U_f}\boldsymbol{U_p},$$

  which is what we were trying to prove.

- Because $\xi_i = \mathcal{O}_i\widetilde{X}_i$, it has rank n, so we can determine the system order from the SVD of $W_1\xi_i W_2$.
  - When $W_1 = I$ and $W_2 = I$, we get the N4SID method;
  - When $W_1 = I$ and $W_2 = \Pi_{U_f^\perp}$, we get the MOESP method;
  - When $W_1 = \Phi^{-1/2}_{[Y_f/\boldsymbol{U_f^?},Y_f/\boldsymbol{U_f^?}]}$, and $W_2 = \Pi_{U_f^\perp}$, we get the CVA method.
  - It is not presently known which of these methods is “best.”

- We can determine the extended observability matrix $\mathcal{O}_i = W_1^{-1}U_1\Sigma_1^{1/2}T$, and the state sequence $\widetilde{X}_i = \mathcal{O}_i^\dagger\xi_i$.

6.14: Step 4a: Computing the system matrices $\hat{A}$, $\hat{C}$

- There does not appear to be consensus regarding how to compute the system matrices for a combined deterministic–stochastic system.
- Generally, we find n and $\mathcal{O}_i$ from the SVD of $W_1 \xi_i W_2$.
- We might consider finding a state sequence $\widetilde{X}_i$ from $\xi_i = Y_f /_{U_f} \boldsymbol{W_p}$ and a second state sequence $\widetilde{X}_{i+1}$ from $\xi_{i+1} = Y_f^- /_{U_f^-} \boldsymbol{W_p^+}$, but it turns out that these have different initial conditions:
  - Initial state for $\widetilde{X}_i$ is $X_p^d /_{U_f} \boldsymbol{U_p}$;
  - Initial state for $\widetilde{X}_{i+1}$ is $X_p^d /_{U_f^-} \boldsymbol{U_p^+}$.
- Because of these different initial conditions, the two resulting state sequences are incompatible. That is,

$$\begin{bmatrix} \widetilde{X}_{i+1} \\ Y_{i|i} \end{bmatrix} \neq \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \widetilde{X}_i \\ U_{i|i} \end{bmatrix} + \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}.$$

- Even so, as the amount of data collected $N \to \infty$, or if the system is purely deterministic, or if the deterministic input is white noise, then this inequality becomes an equality, and can be used as the basis for a subspace system-identification algorithm (algorithm 2 in VODM).
- We will not assume that here. So, we must work a little harder.
- We consider the orthogonal projection computed by shifting the border between "past" and "future" down by one block row in the data matrices.
- We can show that the orthogonal projection becomes

$$\mathcal{Z}_{i+1} = Y_f^- / \begin{bmatrix} \boldsymbol{W_p^+} \\ \boldsymbol{U_f^-} \end{bmatrix}$$

$$= \mathcal{O}_{i-1}\widehat{X}_{i+1} + \Psi_{i-1}U_f^-,$$

where the initial state estimates for $\mathcal{Z}_i$ and $\mathcal{Z}_{i+1}$ are the same, as are the initial covariance.

- Therefore, $\mathcal{Z}_i$ and $\mathcal{Z}_{i+1}$ are compatible in the sense talked about above. So,

$$\begin{aligned}\widehat{X}_{i+1} &= A\widehat{X}_i + BU_{i|i} + K_i(Y_{i|i} - C\widehat{X}_i - DU_{i|i})\\ Y_{i|i} &= C\widehat{X}_i + DU_{i|i} + (Y_{i|i} - C\widehat{X}_i - DU_{i|i}).\end{aligned}$$

- Because the Kalman-filter innovations $(Y_{i|i} - C\widehat{X}_i - DU_{i|i})$ are uncorrelated with the states $\widehat{X}_i$, the past inputs and outputs W_p, and the future inputs U_f, we have

$$\begin{bmatrix}\widehat{X}_{i+1}\\ Y_{i|i}\end{bmatrix} = \begin{bmatrix}A & B\\ C & D\end{bmatrix}\begin{bmatrix}\widehat{X}_i\\ U_{i|i}\end{bmatrix} + \begin{bmatrix}\rho_w\\ \rho_v\end{bmatrix},$$

where the row spaces of ρ_w and ρ_v are orthogonal to the row space of $\widehat{X}_i$, W_p, and U_f.

- If we were able to compute $\widehat{X}_{i+1}$ and $\widehat{X}_i$, then we could use this expression to find the system matrices.
- Unfortunately, we are not. The orthogonal projection does not return the state sequence, but the direct sum $\mathcal{Z}_i = \mathcal{O}_i\widehat{X}_i + \Psi_i U_f$.
- But, we can write

$$\begin{aligned}\widehat{X}_i &= \mathcal{O}_i^{\dagger}\left(\mathcal{Z}_i - \Psi_i U_f\right)\\ \widehat{X}_{i+1} &= \mathcal{O}_{i-1}^{\dagger}\left(\mathcal{Z}_{i+1} - \Psi_{i-1}U_f^-\right).\end{aligned}$$

- The only unknowns in this relationship are Ψ_i and Ψ_{i-1} because the other terms can be found by projections.

- So, we can write

$$\begin{bmatrix} \mathcal{O}_{i-1}^{\dagger}\mathcal{Z}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} A \\ C \end{bmatrix} \mathcal{O}_i^{\dagger}\mathcal{Z}_i + \underbrace{\begin{bmatrix} \begin{bmatrix} B, & \mathcal{O}_{i-1}^{\dagger}\Psi_{i-1} \end{bmatrix} - A\mathcal{O}_i^{\dagger}\Psi_i \\ \begin{bmatrix} D, & 0 \end{bmatrix} - C\mathcal{O}_i^{\dagger}\Psi_i \end{bmatrix}}_{\mathcal{K}} U_f + \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}.$$

- We solve this equation in a least-squares sense for $\hat{A}$, $\hat{C}$, and $\widehat{\mathcal{K}}$.
- Once $\hat{A}$ and $\hat{C}$ are known, the matrices B and D appear linearly in $\widehat{\mathcal{K}}$. They are a little cumbersome to solve for, but we will see one method in the sequel.
- Then, when A, B, C, and D are known, we compute

$$\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} = \begin{bmatrix} \widehat{X}_{i+1} - \hat{A}\widehat{X}_i - \hat{B}U_{i|i} \\ Y_{i|i} - \hat{C}\widehat{X}_i - \hat{D}U_{i|i} \end{bmatrix},$$

and compute the covariance matrices

$$\begin{bmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{bmatrix} = \mathbb{E}_N\left[\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}\begin{bmatrix} \rho_w^T & \rho_v^T \end{bmatrix}\right].$$

- In the van Overschee toolbox, `com_alt.m` contains a MATLAB implementation of this algorithm, using one method to find B and D.

6.15: Step 4b: Computing the system matrices $\hat{B}$, $\hat{D}$

- Let's now consider briefly the problem of finding $\hat{B}$ and $\hat{D}$

$$\hat{B},\hat{D} = \arg\min_{B,D} \left\| \underbrace{\begin{bmatrix} \mathcal{O}_{i-1}^{\dagger}\mathcal{Z}_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} \mathcal{O}_i^{\dagger}\mathcal{Z}_i}_{\text{known}} - \widehat{\mathcal{K}}(B,D) \underbrace{U_f}_{\text{known}} \right\|_F^2 .$$

  - The subscript "F" refers to a "Frobenius norm".
    $\|A\|_F = \sqrt{\text{trace}(A^T A)} = \sqrt{\Sigma_i \sigma_i}$.

- For ease of notation, define

$$\mathcal{P} \triangleq \begin{bmatrix} \mathcal{O}_{i-1}^{\dagger}\mathcal{Z}_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} \mathcal{O}_i^{\dagger}\mathcal{Z}_i$$

$$\mathcal{Q} \triangleq U_f = \begin{bmatrix} \mathcal{Q}_1 \\ \mathcal{Q}_2 \\ \vdots \\ \mathcal{Q}_i \end{bmatrix},$$

  where the $\mathcal{Q}_k$ are block row vectors.

- Then, we can write $\hat{B},\hat{D} = \arg\min_{B,D} \left\| \mathcal{P} - \widehat{\mathcal{K}}(B,D)\mathcal{Q} \right\|_F^2$.

- We now investigate the structure of $\widehat{\mathcal{K}}(B,D)$

$$\widehat{\mathcal{K}} = \begin{bmatrix} \begin{bmatrix} B, & \mathcal{O}_{i-1}^{\dagger}\Psi_{i-1} \end{bmatrix} - \hat{A}\mathcal{O}_i^{\dagger}\Psi_i \\ \begin{bmatrix} D, & 0 \end{bmatrix} - \hat{C}\mathcal{O}_i^{\dagger}\Psi_i \end{bmatrix}$$

$$= \begin{bmatrix} B & 0 \\ D & 0 \end{bmatrix} + \begin{bmatrix} 0 & \mathcal{O}_{i-1}^{\dagger} \\ 0 & 0 \end{bmatrix} \Psi_{i-1} - \begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} \mathcal{O}_i^{\dagger}\Psi_i .$$

- Note that Ψ_{i-1} and Ψ_i are both functions of B and D, but we can compute

$$\mathcal{L} \triangleq \begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} \mathcal{O}_i^{\dagger} = \begin{bmatrix} \mathcal{L}_{1|1} & \mathcal{L}_{1|2} & \cdots & \mathcal{L}_{1|i} \\ \mathcal{L}_{2|1} & \mathcal{L}_{2|2} & \cdots & \mathcal{L}_{2|i} \end{bmatrix}$$

$$\mathcal{M} \triangleq \mathcal{O}_{i-1}^{\dagger} = \begin{bmatrix} \mathcal{M}_1 & \mathcal{M}_2 & \cdots & \mathcal{M}_{i-1} \end{bmatrix}$$

using the available data from the projections. Further, we can break $\widehat{\mathcal{K}}$ into blocks

$$\widehat{\mathcal{K}} = \begin{bmatrix} \mathcal{K}_{1|1} & \mathcal{K}_{1|2} & \cdots & \mathcal{K}_{1|i} \\ \mathcal{K}_{2|1} & \mathcal{K}_{2|2} & \cdots & \mathcal{K}_{2|i} \end{bmatrix}$$

and write

$$\begin{bmatrix} \mathcal{K}_{1|1} \\ \vdots \\ \mathcal{K}_{1|i} \\ \mathcal{K}_{2|1} \\ \vdots \\ \mathcal{K}_{2|i} \end{bmatrix} = \mathcal{N} \begin{bmatrix} D \\ B \end{bmatrix},$$

where

$$\mathcal{N} = \left[\begin{array}{ccccc} -\mathcal{L}_{1|1} & \mathcal{M}_1 - \mathcal{L}_{1|2} & \cdots & \mathcal{M}_{i-2} - \mathcal{L}_{1|i-1} & \mathcal{M}_{i-1} - \mathcal{L}_{1|i} \\ \mathcal{M}_1 - \mathcal{L}_{1|2} & \mathcal{M}_2 - \mathcal{L}_{1|3} & \cdots & \mathcal{M}_{i-1} - \mathcal{L}_{1|i} & 0 \\ \mathcal{M}_2 - \mathcal{L}_{1|3} & \mathcal{M}_3 - \mathcal{L}_{1|4} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathcal{M}_{i-1} - \mathcal{L}_{1|i} & 0 & \cdots & 0 & 0 \\ \hdashline I - \mathcal{L}_{2|1} & -\mathcal{L}_{2|2} & \cdots & -\mathcal{L}_{2|i-1} & -\mathcal{L}_{2|i} \\ -\mathcal{L}_{2|2} & -\mathcal{L}_{2|3} & \cdots & -\mathcal{L}_{2|i} & 0 \\ -\mathcal{L}_{2|3} & -\mathcal{L}_{2|4} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\mathcal{L}_{2|i} & 0 & \cdots & 0 & 0 \end{array}\right] \times$$

$$\begin{bmatrix} I & 0 \\ 0 & \mathcal{O}_{i-1} \end{bmatrix}.$$

- We could stop right now by solving this rather enormous set of linear equations using least squares.
- VODM proposes an alternative, which can be solved more efficiently. Define

$$\mathcal{N}_1 = \begin{bmatrix} -\mathcal{L}_{1|1} & \cdots & \mathcal{M}_{i-1} - \mathcal{L}_{1|i} \\ I - \mathcal{L}_{2|1} & \cdots & -\mathcal{L}_{2|i} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \mathcal{O}_{i-1} \end{bmatrix}$$

$$\mathcal{N}_2 = \begin{bmatrix} \mathcal{M}_1 - \mathcal{L}_{1|2} & \cdots & \mathcal{M}_{i-1} - \mathcal{L}_{1|i} & 0 \\ -\mathcal{L}_{2|2} & \cdots & -\mathcal{L}_{2|i} & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \mathcal{O}_{i-1} \end{bmatrix}$$

$$\vdots$$

$$\mathcal{N}_i = \begin{bmatrix} \mathcal{M}_{i-1} - \mathcal{L}_{1|i} & \cdots & 0 & 0 \\ -\mathcal{L}_{2|i} & \cdots & 0 & 0 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \mathcal{O}_{i-1} \end{bmatrix},$$

where each matrix has one block row from the top half of $\mathcal{N}$ and one block row from the bottom half of $\mathcal{N}$.

- We can now write more compactly

$$\begin{bmatrix} \mathcal{K}_{1|k} \\ \mathcal{K}_{2|k} \end{bmatrix} = \mathcal{N}_k \begin{bmatrix} D \\ B \end{bmatrix},$$

and

$$B, D = \arg\min_{B,D} \left\| \mathcal{P} - \sum_{k=1}^{i} \mathcal{N}_k \begin{bmatrix} D \\ B \end{bmatrix} \mathcal{Q}_k \right\|_F^2.$$

- The apparent problem is that the only unknown terms, which we would like to find, are sandwiched in between some known quantities.
- To turn this into a more useful expression, we make use of the identity

$$\text{vec}(AXB) = (B^T \otimes A)\text{vec}(X),$$

where $\text{vec}(\text{X})$ stacks the columns into a vector, and $\otimes$ is the Kronecker product.

- In MATLAB, $\text{vec}(X)$ is computed as `X(:)`.

- We now rewrite our goal

$$B, D = \arg\min_{B,D} \left\| \text{vec}(\mathcal{P}) - \left(\sum_{k=1}^{i} \mathcal{Q}_k^T \otimes \mathcal{N}_k \right) \text{vec}\left(\begin{bmatrix} D \\ B \end{bmatrix} \right) \right\|_F^2,$$

which may be solved using least squares

$$\text{vec}\left(\begin{bmatrix} D \\ B \end{bmatrix} \right) = \left(\sum_{k=1}^{i} \mathcal{Q}_k^T \otimes \mathcal{N}_k \right)^{\dagger} \text{vec}(\mathcal{P}).$$

- This still requires quite a large matrix inversion, but VODM chapter 6 shows how to solve the relationships using an LQ factorization, which makes for a robust algorithm.

Summary of the combined deterministic–stochastic solution

- We now summarize the combined deterministic–stochastic section by presenting an algorithm that is industrially robust.

1. Calculate the projections $\xi_i = Y_f /_{U_f} \boldsymbol{W_p}$, $\mathcal{Z}_i = Y_f / \begin{bmatrix} \boldsymbol{W_p} \\ \boldsymbol{U_f} \end{bmatrix}$, and $\mathcal{Z}_{i+1} = Y_f^- / \begin{bmatrix} \boldsymbol{W_p^+} \\ \boldsymbol{U_f^-} \end{bmatrix}$.

2. Calculate the SVD of the weighted oblique projection

$$\xi_i \Pi_{U_f^\perp} = U \Sigma V^T.$$

Note that this has assumed the weighting matrices for the MOESP method. Other weighting matrices W_1 and W_2 could be used.

3. Determine the order n of the system by inspecting the singular values in Σ, partitioning appropriately to obtain U_1 and Σ_1.
4. Determine $\mathcal{O}_i = U_1 \Sigma_1^{1/2}$ and $\mathcal{O}_{i-1} = \mathcal{O}_i^{\downarrow}$.
5. Solve the set of linear equations for $\hat{A}$, $\hat{C}$, and $\widehat{\mathcal{K}}$

$$\begin{bmatrix} \mathcal{O}_{i-1}^{\dagger} \mathcal{Z}_{i+1} \\ Y_{i|i} \end{bmatrix} = \begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} \mathcal{O}_i^{\dagger} \mathcal{Z}_i + \widehat{\mathcal{K}} U_f + \begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix}.$$

   Recompute $\mathcal{O}_i$ and $\mathcal{O}_{i-1}$ from $\hat{A}$ and $\hat{C}$.
6. Solve for $\hat{B}$ and $\hat{D}$ from

$$\hat{B}, \hat{D} = \arg\min_{B,D} \left\| \begin{bmatrix} \mathcal{O}_{i-1}^{\dagger} \mathcal{Z}_{i+1} \\ Y_{i|i} \end{bmatrix} - \begin{bmatrix} \hat{A} \\ \hat{C} \end{bmatrix} \mathcal{O}_i^{\dagger} \mathcal{Z}_i - \widehat{\mathcal{K}}(B, D) U_f \right\|_F^2.$$

7. Finally, determine

$$\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} = \begin{bmatrix} \widehat{X}_{i+1} - \hat{A}\widehat{X}_i - \hat{B} U_{i|i} \\ Y_{i|i} - \hat{C}\widehat{X}_i - \hat{D} U_{i|i} \end{bmatrix},$$

   and compute the covariance matrices

$$\begin{bmatrix} \hat{Q} & \hat{S} \\ \hat{S}^T & \hat{R} \end{bmatrix} = \mathbb{E}_N \left[\begin{bmatrix} \rho_w \\ \rho_v \end{bmatrix} \begin{bmatrix} \rho_w^T & \rho_v^T \end{bmatrix} \right].$$

- In the van Overschee toolbox, `subid.m` contains a MATLAB implementation of this algorithm.

Where from here

- Wow. It's been quite a journey to get to this point.
  - Not exactly hard, but certainly intense, involving huge attention to details/book-keeping in the linear algebra.

- In the end, we have found robust methods for determining system models for deterministic, stochastic, and combined deterministic–stochastic systems.
- There are some very good features of these methods:
  - Few user inputs required;
  - “Simpler” calculations (no local minima);
  - They easily handle MIMO systems.
- Problems with the state space methods is that there are few *knobs*
  - Can get a good model, but how about a great one?
- Suggest that you use the state-space methods as a starting point for the Box–Jenkins (PEM) optimizations.
- From here, we move on to look at a few more “advanced” topics.
- These will probably seem like a relief after all this linear algebra...