Let us assume the system is <u>almost</u> uncontrollable or <u>almost</u> unobservable.

$\Rightarrow$ Σ_H must be <u>almost</u> singular.

However, these are singular values:

$$\Sigma_H = \begin{bmatrix} \sigma_1 & & & \emptyset \\ & \sigma_2 & & \\ & & \ddots & \\ \emptyset & & & \sigma_n \end{bmatrix}$$

very small

$\Rightarrow$ This means that the n^{th} state is <u>almost</u> uncontrollable and/or unobservable, and we can throw it out without much of a change in the input/output behavior.

$$\Sigma_H = \left[\begin{array}{c:c} \diagdown & \phi \\ \hdashline \phi & \diagdown \end{array}\right] \begin{array}{l} \}\, r \\ \}\, n-r \end{array}$$

significant modes

insignificant modes

$$\Rightarrow \quad \underline{\dot{\xi}} = \begin{array}{r} r\{ \\ n-r\{ \end{array} \overbrace{\left[\begin{array}{c:c} A_{11} & A_{12} \\ \hdashline A_{21} & A_{22} \end{array}\right]}^{\quad r \qquad\quad n-r} \cdot \underline{\xi} + \begin{array}{r} r\{ \\ n-r\{ \end{array} \overbrace{\left[\begin{array}{c} B_1 \\ \hdashline B_2 \end{array}\right]}^{m} \underline{u}$$

$$\qquad\qquad A_{bal} \qquad\qquad\qquad\qquad B_{bal}$$

$$\underline{y} = p\{ \underbrace{\left[\begin{array}{c:c} C_1 & C_2 \end{array}\right]}_{\quad r \qquad\quad n-r} \underline{\xi} + \left[\; D \;\right] \underline{u}$$

$$C_{bal}$$

has <u>very</u> similar behavior to:

$$\left| \begin{array}{l} \underline{\dot{\xi}}_{red} = A_{11} \cdot \underline{\xi}_{red} + B_1 \cdot \underline{u} \\ \underline{y} = C_1 \cdot \underline{\xi}_{red} + D \cdot \underline{u} \end{array} \right|$$

$$\underline{\xi} = \left[\begin{array}{c} \underline{\xi}_{red} \\ \hdashline \underline{\xi}_{elim} \end{array}\right] \begin{array}{l} \}\, r \\ \}\, n-r \end{array}$$

Assume:

$$\dot{\underline{x}} = A\underline{x} + B\underline{u}$$

$$\text{Rank}(G_c) < n$$

$\Rightarrow$ The system is truly uncontrollable!

$\Rightarrow$ Need "input decoupling."
The algorithm of ECE 501 requires Q_c and Q_c^{-1}

$\Longrightarrow$ FOR THE BIRDS!!!

Recipe:
Look at the system:

$$\dot{\underline{x}} = A\underline{x} + \left[B \,\vdots\, B_{aux}\right] \cdot \left[\begin{array}{c}\underline{u} \\ \hline \underline{u}_{aux}\end{array}\right]$$

e.g. $B_{aux} = 100 * \varepsilon * \text{rand}(n,1) * \|B\|$

We calculate G_c again for the system with the augmented input vector. If

$$\mathrm{Rank}(G_c) < n$$

we increase the gain of B_{aux}, until

$$\mathrm{Rank}(G_c) = n .$$

Yet, since the elements of B_{aux} are still small, the system is still <u>almost</u> uncontrollable.

Now, we can use Balreal. We eliminate the almost uncontrollable state and get:

$$\dot{\underline{x}}_{red} = A_{11} \cdot \underline{x}_{red} + \left[B_1 \,\vdots\, B_2 \right] \cdot \begin{bmatrix} \underline{u} \\ \hline \underline{u}_{aux} \end{bmatrix}$$

Because the elements of B_{aux} were small, the system output does not change much for <u>any</u> power-limited input signal $\underline{u}_{aux}$. Since the I/O behavior is invariant to similarity transformations:

$\Rightarrow$ The elements of B_2 must still be small, in spite of the similarity transformation. Thus, we simply throw B_2 away:

$$\dot{\underline{x}}_{red} = A_{11} \cdot \underline{x}_{red} + B_1 \cdot \underline{u}$$

Similarly, we augment the C matrix by an additional row of small elements if Rank$(G_0) < n$.

Assume that the original system was <u>unstable</u>.

If any of the unstable modes is unobservable or uncontrollable (or poorly observable/controllable), the system is garbage.

Otherwise, I create a linear output feedback around the original system (using either the stable poleplacement or lqr algorithms) without worrying about performance too much. The purpose is simply stabilization. Once this has happened, we can use ballancing to throw out uncontrollable and/or unobservable modes of the already controlled system. Then, we build a controller around this for performance improvement.

Given the discrete system:

$$\underline{x}_{k+1} = F \cdot \underline{x}_k + G \cdot \underline{u}_k \; ; \; \underline{x}(\emptyset) = \underline{x}_0$$

This system is controllable if the controllability Gramian:

$$G_c = \sum_{i=\emptyset}^{\infty} F^i \cdot G \cdot G^* \cdot (F^*)^i$$

is non-singular.

$$G_c = GG^* + FGG^*F^* + F^2GG^*(F^*)^2 + \ldots$$

$$\Rightarrow F \cdot G_c \cdot F^* = FGG^*F^* + F^2GG^*(F^*)^2 + \ldots$$

$$\Rightarrow F \cdot G_c \cdot F^* - G_c = -GG^*$$

$$\Rightarrow \boxed{F \cdot G_c \cdot F^* - G_c + GG^* = \emptyset}$$

is a discrete Lyapunov equation. The above derivation holds as long as the infinite series converge, i.e., as long as $\|F\| < 1$, i.e., the system is stable.

$$B = U \cdot \Sigma \cdot V^* \qquad \left(\begin{array}{l}\text{singular}\\ \text{value}\\ \text{decomposition}\end{array}\right)$$

$$\Rightarrow F \cdot G_c \cdot F^* - G_c + U \cdot \Sigma^2 \cdot U^* = \emptyset$$

$$\Rightarrow U^* F G_c F^* U - U^* G_c U + \Sigma^2 = \emptyset$$

$$\Rightarrow \underbrace{U^* F U}_{\hat{F}} \cdot \underbrace{U^* G_c U}_{\hat{G}_c} \cdot \underbrace{U^* F^* U}_{\hat{F}^*} - \underbrace{U^* G_c U}_{\hat{G}_c} + \Sigma^2 = \emptyset$$

$$\Rightarrow \boxed{\hat{F} \cdot \hat{G}_c \cdot \hat{F}^* - \hat{G}_c + \Sigma^2 = \emptyset}$$

is again a discrete Lyapunov equation.

```
function [Gc] = dgram(F,G)
   [u,s,v] = svd(G);
   Fhat = u'*F*u;
   Gchat = dlyap(Fhat, s*s');
   Gc = u*Gchat*u';
return
```

is the Matlab implementation.

G_c is uniquely determined by the discrete Lyapunov equation. G_c is always Hermitian:

$$(F \cdot G_c \cdot F^* - G_c + G \cdot G^*)^* = \phi$$

$$\Rightarrow F \cdot G_c^* \cdot F^* - G_c^* + G \cdot G^* = \phi$$

$$\Longrightarrow \boxed{G_c^* \equiv G_c}$$

The observability Gramian:

$$G_o = \sum_{i=\phi}^{\infty} (F^*)^i \cdot H^* \cdot H \cdot F^i$$

can be computed using the Lyapunov equation:

$$\underline{\underline{F^* \cdot G_o \cdot F - G_o + H^* \cdot H = \phi}} \;//$$

In Matlab:

```
Gc = dgram (F, G)
Go = dgram (F', H')
```

Similarity transformations:

$$\underline{\xi} = T \cdot \underline{x}$$

$$\left|\begin{array}{l} \tilde{F} = T \cdot F / T \\ \tilde{G} = T \cdot G \\ \tilde{H} = H / T \\ \tilde{I} = I \end{array}\right|$$

$$\tilde{F} \cdot \tilde{G}_c \cdot \tilde{F}^* - \tilde{G}_c + \tilde{G} \cdot \tilde{G}^* = \emptyset$$

$$\Rightarrow T \cdot F \cdot T^{-1} \cdot \tilde{G}_c \cdot (T^{-1})^* \cdot F^* \cdot T^* - \tilde{G}_c + T \cdot G \cdot G^* \cdot T^* = \emptyset$$

$$\Rightarrow F \cdot \underbrace{T^{-1} \cdot \tilde{G}_c \cdot (T^{-1})^*}_{G_c} \cdot F^* - \underbrace{T^{-1} \tilde{G}_c (T^{-1})^*}_{G_c} + G \cdot G^* = \emptyset$$

$$\Rightarrow G_c = T^{-1} \tilde{G}_c \cdot (T^{-1})^*$$

$$\Rightarrow \underline{\underline{\tilde{G}_c = T \cdot G_c \cdot T^*}}$$

$$\tilde{F}^*\tilde{G}_0\tilde{F} - \tilde{G}_0 + \tilde{H}^*\tilde{H} = \emptyset$$

$$\Rightarrow (T^{-1})^* F^* T^* \cdot \tilde{G}_0 \cdot T F T^{-1} - \tilde{G}_0 + (T^{-1})^* H^* H T^{-1} = \emptyset$$

$$\Rightarrow F^* \underbrace{T^* \cdot \tilde{G}_0 \cdot T}_{G_0} F - \underbrace{T^* \tilde{G}_0 T}_{G_0} + H^* H = \emptyset$$

$$\Rightarrow G_0 = T^* \tilde{G}_0 T$$

$$\Rightarrow \tilde{G}_0 = (T^{-1})^* \cdot G_0 \cdot (T^{-1})$$

Let : $G_c = U_c \cdot \Sigma_c^2 \cdot U_c^*$

$$\Rightarrow F \cdot G_c \cdot F^* - G_c + G \cdot G^* = \emptyset$$

$$\Rightarrow F \cdot U_c \cdot \Sigma_c^2 \cdot U_c^* \cdot F^* - U_c \Sigma_c^2 U_c^* + G G^* = \emptyset$$

$$\Rightarrow (U_c^* F U_c) \Sigma_c^2 (U_c^* F^* U_c) - \Sigma_c^2 + U_c^* G G^* U_c = \emptyset$$

$$\Rightarrow \boxed{T = U_c^*}$$

$$\Rightarrow \bar{G}_c = \Sigma_c^2$$

Similarly:

$$G_o = U_o \cdot \Sigma_o^2 \cdot U_o^*$$

$$\Rightarrow \boxed{T = U_o^*}$$

$$\Rightarrow \bar{G}_o = \Sigma_o^2$$

$$G_c = U_c \cdot \Sigma_c^2 \cdot U_c^*$$

$$G_o = U_o \cdot \Sigma_o^2 \cdot U_o^*$$

$$\Rightarrow G_o G_c = (U_o \cdot \Sigma_o)\underbrace{(\Sigma_o \cdot U_o^* \cdot U_c \cdot \Sigma_c)}_{H} \cdot (\Sigma_c \cdot U_c^*)$$

$$H = U_H \cdot \Sigma_H^2 \cdot V_H^*$$

$$T = \Sigma_H^{-1} \cdot U_H^* \cdot \Sigma_o \cdot U_o^*$$

Since:
$$\hat{G}_c = T \cdot G_c \cdot T^*$$
$$\hat{G}_o = (T^{-1})^* \cdot G_o \cdot (T^{-1})$$

is still true

$$\Rightarrow \hat{G}_c = \hat{G}_o = \Sigma_H^2$$

In Matlab:

```
function [Fbal, Gbal, Hbal, Mode, T] =
      dbalreal (F, G, H)
Gc = dgram (F, G);
Go = dgram (F', H');
[Uc, Sc, Vc] = svd (Gc);
[Uo, So, Vo] = svd (Go);
Sc = diag(diag(Sc)^0.5);
So = diag(diag(So)^0.5);
H = So * Uo' * Uc * Sc;
[UH, SH, VH] = svd (H);
T = diag(diag(SH)^0.5) \ UH' * So * Uo';
Fbal = T*F/T;
Gbal = T*G;
Hbal = H/T;
Mode = diag(SH);
return
```