MFSS: A MATLAB Toolbox for Mixed-Frequency State-Space Models

Scott Brave[1], Andrew Butters[2] & **David Kelley**[3]

[1]Federal Reserve Bank of Chicago
[2]Indiana University
[3]Federal Reserve Bank of New York

October 1, 2025

Problem: Time series data are observed at different frequencies.

Simplistic approach: Aggregate to lowest frequency available.

This approach: State space models that allow for observations at the frequency of each observation and a latent state at a higher frequency.

Why?

- Simplification of modeling assumptions
- High-frequency view of the world
- Timely forecasts without a secondary “plug” model

How? A MATLAB package (**MFSS**) that handles the complexity for you. For details, see “A Practitioner’s Guide and MATLAB Toolbox for Mixed Frequency State Space Models.”

MFSS provides a unified way to model time series of differing frequencies.

State space models can be somewhat complicated. **MFSS** aims to make mixed-frequency analysis and forecasting as easy as possible.

Mixed-frequency data workflow

- Collect your data
  - Each row represents an observation at the base (or high) frequency
  - Lower frequency observations are placed at the end of the period
- Specify a model for how the (latent) state evolves
- Augment the model to match the observed data
- Estimate model parameters
- Extract high-frequency estimates of the state or forecasts of interest

State Space Models

With observed data vector y_t and latent state α_t, the system is a markov chain:

$$y_t = Z_t \alpha_t + d_t + \beta_t x_t + \varepsilon_t \qquad \varepsilon_t \sim \mathcal{N}(0, H_t) \qquad (1)$$

$$\alpha_t = T_t \alpha_{t-1} + c_t + \gamma_t w_t + R_t \eta_t \qquad \eta_t \sim \mathcal{N}(0, Q_t) \qquad (2)$$

We will be splitting up the data vector y_t into those variables that are observed at the base frequency (y_t^b) and those that are observed at a lower frequency (y_t^l).

Low-Frequency Observations

We define an *aggregation function* that moves from a high-frequency (latent) value to a low-frequency observation:

$$y_t^l = \mathcal{A}(y_t^h, y_{t-1}^h, \ldots, y_{t-n+1}^h)$$

We define an *accumulator*, ζ_t as a varaible that can be recursively defined and will (eventually) match the low-frequency quantity we need.

We can then "augment" the system from earlier to get the system that is separated by observation frequency:

$$\begin{bmatrix} y_t^b \\ y_t^l \end{bmatrix} = \begin{bmatrix} Z_t^b & 0 \\ 0 & Z^l \end{bmatrix} \begin{bmatrix} \alpha_t \\ \zeta_t \end{bmatrix} + \begin{bmatrix} d_t^b \\ d_t^l \end{bmatrix} + \begin{bmatrix} \beta_t^b & 0 \\ 0 & \beta^l \end{bmatrix} \begin{bmatrix} x_t^b \\ x_t^l \end{bmatrix} + \begin{bmatrix} \varepsilon_t^b \\ \varepsilon_t^l \end{bmatrix} \tag{3}$$

$$\begin{bmatrix} \alpha_t \\ \zeta_t \end{bmatrix} = \begin{bmatrix} T_t & 0 \\ A_t^T & A_t^\zeta \end{bmatrix} \begin{bmatrix} \alpha_{t-1} \\ \zeta_{t-1} \end{bmatrix} + \begin{bmatrix} c_t \\ A_t^c \end{bmatrix} + \begin{bmatrix} \gamma_t \\ A_t^\gamma \end{bmatrix} w_t + \begin{bmatrix} R_t \\ A_t^R \end{bmatrix} \eta_t. \tag{4}$$

where $\{A_t^\zeta, A_t^T, A_t^c, A_t^\gamma, A_t^R\}$ are functions of the system matrices.

Aggregation Types and Specification

There are three different types of accumulators.

Sampling type	Level	Δ_1	$\Delta_{H\geq 1}$
Point-in-time	None	None	Sum
Sum	Sum	Sum	Sum
Average	Average	Average	Triangle Average

Each is defined by a *calendar* and *horizon*:

- Sum: calendar is zeros until the period of observation.
- Average: calendar starts at one each low-frequency period and increases each base-period until observation
- Triangle average: like an average, but with a *horizon* that measures level averaging period

MFSS Simplifies Specification

Specifying the model parameters for state space model with mixed-frequency data can be hard.

We prefer to specify the model as if everything were at the base frequency then augment the system to match the observed data.

Also included in **MFSS**:

- Semi-automatic accumulator specification for regular data
- Fast (mex) implementation of the Kalman filter
- Parameter estimation
- "Structural" specification using the Symbolic Math Toolbox
- Parameter restrictions (simple and structural)
- State decompositions

Example: A simple dynamic factor model

of log changes of GDP (quarterly), payroll employment, personal income less transfers, industrial production, real manufacturing & trade sales.

```
Z = [1; nan(4,1)];
d = [nan; zeros(4,1)];
H = diag(nan(5,1));
T = nan;
Q = nan;
ssE = StateSpaceEstimation(Z, H, T, Q, 'd', d);

accum = Accumulator.GenerateRegular(y, ...
    {'avg', ' ', ' ', ' ', ' '}, [3 1 1 1 1]);
ssEA = accum.augmentStateSpaceEstimation(ssE);
ssML = ssEA.estimate(y);
alphaHat = ssML.smooth(y);
```

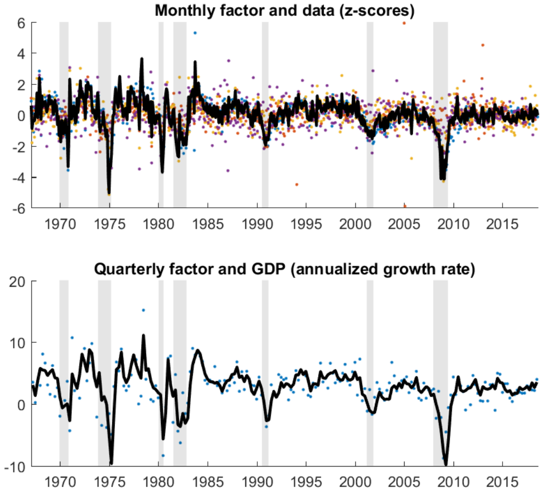
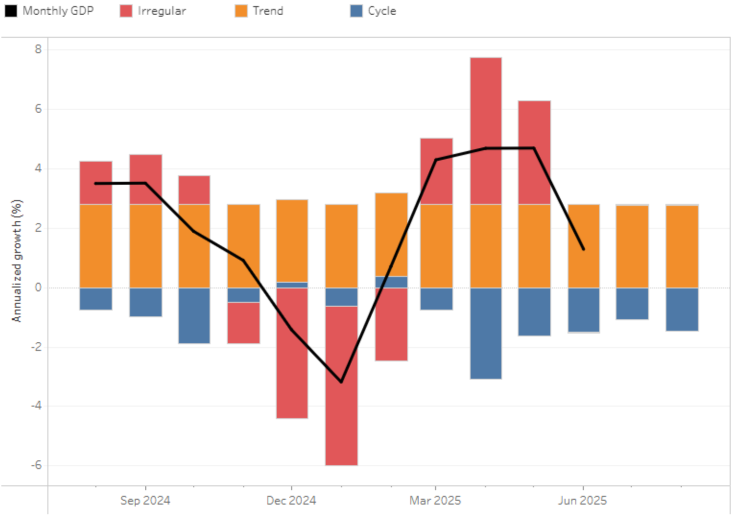


Figure: Mixed-frequency dynamic factor model

Example: BBKI Indexes and GDP Growth Decomposition



Conclusion

MFSS provides a unified way to model time series of differing frequencies and make mixed-frequency analysis and forecasting as easy as possible.

For more details on mixed-frequency state space modeling, see “A Practitioner’s Guide and MATLAB Toolbox for Mixed Frequency State Space Models.”

Code is available at `github.com/davidakelley/MFSS`.