MIDAS Matlab Toolbox*

Eric Ghysels†

First Draft: December 2009

This Draft: August 10, 2017

Version 2.2

*The author benefited from funding by the Federal Reserve Bank of New York through the Resident Scholar Program. I am extremely indebted and grateful to Riccardo Colacito, Xiafei Hu, Jurij Plazzi, Hang Qian, Arthur Sinko, Wasin Siwasarit, Michael Sockin and Bumjean Sohn. They have written parts of the code. Questions, comments and bug reports can be sent to `matlabist@gmail.com`.

†Department of Finance - Kenan-Flagler Business School and Department of Economics, University of North Carolina, McColl Building, Chapel Hill, NC 27599. `e-mail:` *eghysels@unc.edu*.

1 Introduction

Econometric models involving data sampled at different frequencies are of general interest. This Matlab Toolbox covers MIDAS Regression, GARCH-MIDAS, DCC-MIDAS and MIDAS quantile regression models. The former is a framework put forward in recent work by Ghysels, Santa-Clara, and Valkanov (2002), Ghysels, Santa-Clara, and Valkanov (2006) and Andreou, Ghysels, and Kourtellos (2010) using so called MIDAS, meaning Mi(xed) Da(ta) S(ampling), regressions. Several recent surveys on the topic of MIDAS are worth mentioning at the outset. They are: Andreou, Ghysels, and Kourtellos (2011) who review more extensively some of the material summarized in this document, Armesto, Engemann, and Owyang (2010) who provide a very simple introduction to MIDAS regressions and finally Ghysels and Valkanov (2012) who discuss volatility models and mixed data sampling.

The original work on MIDAS focused on volatility predictions, see e.g. Alper, Fendoglu, and Saltoglu (2008), Chen and Ghysels (2011), Engle, Ghysels, and Sohn (2013), Brown and Ferreira (2003), Chen, Ghysels, and Wang (2014), Chen, Ghysels, and Wang (2011), Clements, Galvão, and Kim (2008), Corsi (2009), Forsberg and Ghysels (2006), Ghysels, Santa-Clara, and Valkanov (2005), Ghysels, Santa-Clara, and Valkanov (2006), Ghysels and Sinko (2006), Ghysels and Sinko (2011), Ghysels, Rubia, and Valkanov (2008), León, Nave, and Rubio (2007), among others.

Recent work has used the regressions in the context of improving quarterly macro forecasts with monthly data (see e.g. Armesto, Hernandez-Murillo, Owyang, and Piger (2009), Clements and Galvão (2009), Clements and Galvão (2008), Frale and Monteforte (2011), Kuzin, Marcellino, and Schumacher (2011b), Monteforte and Moretti (2013), Marcellino and Schumacher (2010), Schumacher and Breitung (2008)), or improving quarterly and monthly macroeconomic predictions with daily financial data (see e.g. Andreou, Ghysels, and Kourtellos (2013a), Ghysels and Wright (2009), Hamilton (2008)).

Econometric analysis of MIDAS regressions appears in Ghysels, Sinko, and Valkanov (2006), Andreou, Ghysels, and Kourtellos (2010), Bai, Ghysels, and Wright (2013), Kvedaras and Račkauskas (2010), Rodriguez and Puggioni (2010), Wohlrabe (2009), among others.

MIDAS regression can also be viewed as a reduced form representation of the linear projection that emerges from a state space model approach - by reduced form we mean that the MIDAS regression does not require the specification of a full state space system of equations. Bai, Ghysels, and Wright (2013) show that in some cases the MIDAS regression is an exact representation of the Kalman filter, in other cases it involves approximation errors that are typically small. The Kalman filter, while clearly optimal as far as linear projections goes, has several disadvantages (1) it is more prone to specification errors as a full system of measurement and state equations is required and as a consequence (2) requires a lot more parameters, which in turn results in (3) computational complexities that often limit the scope of applications. In contrast, MIDAS regressions - combined with forecast combination schemes if large data sets are involved (see Andreou, Ghysels, and Kourtellos (2013a)) are computationally easy to implement and more prone to specification errors.

Mixed frequency data issues are not confined to regression models and in the new Version 2.1 we have added code handling GARCH-MIDAS and DCC-MIDAS models. Engle, Ghysels, and Sohn (2013) revisit modeling the economic sources of volatility. They consider a component model and suggest several new component model specifications with direct links to economic activity. Practically speaking, the research pursued is inspired by (1) Engle and Rangel (2008) who introduce a Spline-GARCH model where the daily equity volatility is a product of a slowly varying deterministic component and a mean reverting unit GARCH and (2) the use of MIDAS approach to link macroeconomic variables to the long term component of volatility. Hence, the new class of models is called GARCH-MIDAS, since it uses a mean reverting unit *daily* GARCH process, similar to Engle and Rangel (2008), and a MIDAS polynomial which applies to *monthly, quarterly,* or *bi-annual* macroeconomic or financial variables. Having introduced the GARCH-MIDAS model that allows us to extract two components of volatility, one pertaining to short term fluctuations, the other pertaining to a long run component, we are ready to revisit the relationship between stock market volatility and economic activity and volatility. The first specification we consider uses exclusively financial series. The GARCH component is based on daily (squared) returns, whereas the long term component is based on realized volatilities computed over a monthly, quarterly or bi-annual basis. The GARCH-MIDAS model also allows us to examine directly the macro-volatility links. Indeed, one can estimate GARCH-MIDAS models where macroeconomic variables enter directly the specification of the long term component. The fact that the macroeconomic series are sampled at a different frequency is not an obstacle, again due to the advantages of the MIDAS weighting scheme.

In addition, dynamic correlation models featuring mixed data sampling schemes based on MIDAS have been used by Colacito, Engle, and Ghysels (2011) and Baele, Bekaert, and

Inghelbrecht (2010). The so called DCC-MIDAS model is a multivariate extension to the GARCH-MIDAS model with dynamic correlations. The DCC-MIDAS model decomposes the conditional covariance matrix into the variances and the correlation matrix, with a two-step model specification and estimation strategy. In the first step, conditional variances are estimated by the univariate GARCH-MIDAS models. In the second step, observations are deflated by the estimated means and conditional variances, and the standardized residuals are thus constructed. The standardized residuals have a correlation matrix with GARCH-MIDAS-like dynamics.

Finally, in Version 2.1 we also included MIDAS quantile regressions used in a number of recent studies, including Ghysels, Plazzi, and Valkanov (2016).

2 Intro to MIDAS regressions

For illustrative purpose we start with a combination of two sampling frequencies, respectively high and low. In terms of notation, $t = 1, \ldots, T$ indexes the low frequency time unit, and m is the number of times the higher sampling frequency appears in the same basic time unit (assumed fixed for simplicity). For example, for quarterly GDP growth and monthly indicators as explanatory variables, $m = 3$. The low frequency variable will be denoted by y_t^L, whereas a generic high frequency series will be denoted by $x_{t-j/m}^H$ where $t - j/m$ is the j^{th} (past) high frequency period with $j = 0, \ldots$. For a quarter/month mixture one has x_t^H, $x_{t-1/3}^H$, $x_{t-2/3}^H$ as the last, second to last and first months of quarter t. Obviously, through some (linear?) aggregation scheme, such as flow or stock sampling, we can always construct a low frequency series x_t^L. We will simply assume that $x_t^L = \sum_{i=1}^m a_i x_{t+i/m}^H$ (see Lütkepohl (2012) or Stock and Watson (2002, Appendix) for further discussion of temporal aggregation issues).

MIDAS regressions are essentially tightly parameterized, reduced form regressions that involve processes sampled at different frequencies. The response to the higher-frequency explanatory variable is modeled using highly parsimonious distributed lag polynomials, to prevent the proliferation of parameters that might otherwise result, as well as the issues related to lag-order selection.

2.1 DL-MIDAS regressions

The basic single high frequency regressor MIDAS model for h-step-ahead (low frequency) forecasting, with high frequency data available up to x_t^H is given by:

$$y_{t+h}^L = a_h + b_h C(L^{1/m};\theta_h)x_t^H + \varepsilon_{t+h}^L \tag{2.1}$$

where $C(L^{1/m};\theta) = \sum_{i=0}^N c(i;\theta)L^{i/m}$, and $C(1;\theta) = \sum_{j=0}^N c(j;\theta) = 1$.

The parameterization of the lagged coefficients of $c(k;\theta)$ in a parsimonious way is one of the key MIDAS features. Various specifications for $C(L^{1/m};\theta)$ will be discussed later in subsection 2.3. Note that the MIDAS regression will either require nonlinear least squares (NLS), see Ghysels, Santa-Clara, and Valkanov (2004) and Andreou, Ghysels, and Kourtellos (2010) for more discussion, or so called estimation via profiling, see Ghysels and Qiang (2016), where the latter involves simple linear regression techniques with θ preset taking values on grid.

Suppose now, we want to predict the first out-of-sample (low frequency) observation, namely considering equation (2.1) with $h = 1$:

$$\hat{y}_{T+1|T}^L = \hat{a}_{1,T} + \hat{b}_{1,T} C(L^{1/m};\hat{\theta}_{1,T})x_T^H \tag{2.2}$$

where the MIDAS regression model parameters are estimated over the sample ending at T. Nowcasting, or MIDAS with leads as coined by Andreou, Ghysels, and Kourtellos (2013b), involving equation (2.2) can also obtained. For example, with i/m additional observations the horizon h shrinks to $h - i/m$, and the above equation becomes:

$$\hat{y}_{T+h|T+i/m}^L = \hat{a}_{h-i/m,T} + b_{h-i/m,T} C(L^{1/m};\theta_{h-i/m,T})x_{t_L+i/m}^H$$

where we note that all the parameters are horizon specific. This brings us to the topic of the next subsection.

2.2 Some comments about multi-step horizon forecasts

The topic of mixing different sampling frequencies also emerges even when time series are available at the same frequency, but one is interested in multi-period forecasting. Take the example of an annual forecast with quarterly data. The first approach is to estimate a model with past annual data, and hence collapse the original multi-period setting into a single step forecast. The second approach is to estimate a quarterly forecasting model and then iterate forward the forecasts to a multi-period annual prediction. The forecasting literature refers to the first approach as *direct* and the second as *iterated.* (Marcellino, Stock, and Watson (2006)). Traditionally, the comparison has been made between direct and iterated forecasting, see e.g. Findley (1983), Findley (1985), Lin and Granger (1994), Clements and Hendry (1996), Bhansali (1999), and Chevillon and Hendry (2005). Multi-period forecasts can also be constructed using a mixed-data sampling approach. A MIDAS model can use past quarterly data to produce directly multi-period forecasts. The MIDAS approach can be viewed as a middle ground between the direct and the iterated approaches. Namely, one preserves the past high frequency data, to directly produce multi-period forecasts

2.3 Parameterizations the MIDAS polynomial weights

Various other parsimonious polynomial specifications $C(L^{1/m};\theta)$ have been considered, including (1) beta polynomial, (2) Almon lag polynomial specifications, (3) step functions, among others. Ghysels, Sinko, and Valkanov (2006) provide a detailed discussion.

1. U-MIDAS (unrestricted MIDAS polynomial) approach suggested by Foroni, Marcellino, and Schumacher (2015) - where one estimates the individual coefficients unconstrained and therefore one can use a simple regression program. The U-MIDAS approach was shown to work for small values of m. The prime example is quarterly/monthly mixtures. U-MIDAS is a special case of MIDAS with step functions discussed below.

2. Normalized beta probability density function, unrestricted (u) and restricted (r) cases with non-zero and zero last lag. Please note that for specifications with a small number of MIDAS lags the zero-last-lag assumption may generate significant bias in

the weighting scheme.

$$c_i^{u,nz} = c(i; \theta = [\theta_1, \theta_2, \theta_3]) = \frac{x_i^{\theta_1 - 1}(1 - x_i)^{\theta_2 - 1}}{\sum_{i=1}^{N} x_i^{\theta_1 - 1}(1 - x_i)^{\theta_2 - 1}} + \theta_3 \tag{2.3}$$

$$c_i^{r,nz} = c(i; \theta = [1, \theta_2, \theta_3]) \tag{2.4}$$

$$c_i^{u,z} = c(i; \theta = [\theta_1, \theta_2, 0]) \tag{2.5}$$

$$c_i^{r,z} = c(i; \theta = [1, \theta_2, 0]) \tag{2.6}$$

where $x_i = i/(N+1)$.[1]

3. Normalized exponential Almon lag polynomial

$$c_i^{u} = c(i; \theta = [\theta_1, \theta_2]) = \frac{e^{\theta_1 i + \theta_2 i^2}}{\sum_{i=1}^{N} e^{\theta_1 i + \theta_2 i^2}} \tag{2.7}$$

$$c_i^{r} = c(i; \theta = [\theta_1, 0]) \tag{2.8}$$

4. Almon lag polynomial specification of order P (not normalized, i.e. sum of individual weights is not equal to 1 and $b_h c_i(\theta)$ is specified as

$$b_h c(i; \theta = [\theta_0, \ldots, \theta_P]) = \sum_{p=0}^{P} \theta_p i^p \tag{2.9}$$

Note that this can also be written in matrix form:

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ \vdots \\ c_N \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2^2 & \cdots & 2^P \\ 1 & 3 & 3^2 & \cdots & 3^P \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & N & N^2 & \cdots & N^P \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_P \end{bmatrix} \tag{2.10}$$

Therefore the use of Almon lags in MIDAS models can be achieved via OLS estimation with properly transformed high frequency data regressors using the matrix representation appearing in the above equation. Once the weights are estimated via OLS, one can always rescale them to obtain a slope coefficient (assuming the weights

[1] To eliminate irregular behavior of the polynomial for some values of θ at the ends of [0,1] interval we use instead $x_i = eps + i/(N+1)(1 - eps)$, where eps is a machine 0 for MATLAB.

do not sum up to zero).

5. Polynomial specification with step functions (not normalized)

$$
\begin{aligned}
b_h c(i;\theta=[\theta_1,\ldots,\theta_P]) &= \theta_1 I_{i\in[a_0,a_1]} + \sum_{p=2}^{P} \theta_p I_{i\in(a_{p-1},a_p]} \\
a_0 = 1 &< a_1 < \ldots < a_P = N \\
I_{i\in[a_{p-1},a_p]} &= \begin{cases} 1, & a_{p-1} \le i \le a_p \\ 0, & otherwise \end{cases}
\end{aligned}
\tag{2.11}
$$

where $a_0 = 1 < a_1 < \ldots < a_P = N$.

2.4 ADL-MIDAS regressions

Andreou, Ghysels, and Kourtellos (2013b) introduce the class of ADL-MIDAS regressions, extending the structure of ARDL models to a mixed frequency setting. Assuming an autoregressive augmentation of order one, the model can be written as:

$$
y^L_{t_L+h} = a_h + \lambda_h y^L_{t_L} + b_h C(L^{1/m};\theta_h) x^H_{t_L} + \varepsilon^L_{t_L+h} \tag{2.12}
$$

Hence, an ADL-MIDAS regression is a direct forecasting tool projecting a low frequency series, at some horizon h, namely $y^L_{t_L+h}$ onto $y^L_{t_L}$ (or more lags if we consider higher order AR augmentations) and high frequency data $x^H_{t_L}$. Nowcasting, or MIDAS with leads, can again be obtained via shifting forward the high frequency data with $1/m$ increments. The parameters are again horizon specific and the forecast is one that is direct (instead of iterated).

2.5 Model selection

A few words about model selection are in order. First, how do we decide on K, the maximal lag in the MIDAS polynomial? It might be tempted to use say an information criterion as is typically done in ARMA or ARDL models. However, the number of lags in the high frequency polynomial is not affecting the number of parameters. Hence, the usual penalty functions such as those in the Akaike (AIC), Schwarz (SIC) or Hannan-Quinn (HQ) criteria will not apply. The only penalty of picking K too large is that we require more (high frequency) data at the beginning of the sample as the weights typically vanish to zero with K too large. Picking K too small is more problematic. This issue has been discussed extensively in the standard literature on distributed lag models, see e.g. Judge, Hill, Griffiths, Lütkepohl, and Lee (1988, Chapters 8 & 9). Nevertheless, using information criteria will be useful once we introduce lagged depend variables, see the next subsection, as the selection of AR augmentations falls within the realm of IC-based model selection. For this reason Andreou, Ghysels, and Kourtellos (2013b) recommend using AIC or SIC for example. Finally, Kvedaras and Zemlys (2012) present model specification tests for the polynomial choices in MIDAS regressions.

2.6 Factors and other regressors in ADL-MIDAS models

Recently, a large body of recent work has developed factor model techniques that are tailored to exploit a large cross-sectional dimension; see for instance, Bai and Ng (2002), Bai (2003), Forni, Hallin, Lippi, and Reichlin (2000), Forni, Hallin, Lippi, and Reichlin (2005), Stock and Watson (1989), Stock and Watson (2003), among many others. These factors are usually estimated at quarterly frequency using a large cross-section of time-series. Following this literature Andreou, Ghysels, and Kourtellos (2013a) investigate whether one can improve factor model forecasts by augmenting such models with high frequency information, especially daily financial data.

We therefore augment the aforementioned MIDAS models with factors, F_t, obtained by following dynamic factor model

$$\begin{aligned} X_t &= \Lambda_t F_t + u_t \qquad (2.13) \\ F_t &= \Phi F_{t-1} + \eta_t \\ u_{it} &= a_{it}(L) u_{it-1} + \varepsilon_{it}, \quad i = 1, 2, ..., n \end{aligned}$$

where the number of factors is computed using criteria proposed by Bai and Ng (2002). The data used to implement the factor representation will be described in the next section. Suffice it here to say that we use series similar to those used by Stock and Watson (2008a).

Augmenting the MIDAS regression models from the previous subsection with the factors, we obtain a richer family of models that includes monthly frequency lagged dependent variable, quarterly factors, and a daily financial indicator. For instance, equation (2.12) generalizes to the FADL-MIDAS model:

$$y_{t_L+h}^L = a_h + \sum_{i=0}^{p_F} \beta_{i,h}^F F_{t-i}^Q + \sum_{i=0}^{p_y} \lambda_{i,h} y_{t_L}^L + b_h C(L^{1/m};\theta_h) x_{t_L}^H + \varepsilon_{t_L+h}^L \tag{2.14}$$

or factor augmented ADL-MIDAS regression.

Equation (2.14) simplifies to the traditional factor model with additional regressors when the MIDAS features are turned off - i.e. say a flat aggregation scheme is used. When the lagged dependent variable is excluded then we have a projection on daily data, combined with aggregate factors.

It should finally be noted that we can add any low frequency regressor, not just factors. The software is written such that one can add any type of low frequency regressor.

To conclude it should be noted that two modes of forecasting can be used in the Matlab MIDAS Toolbox. The first is fixed in-sample estimation and fixed out-of-sample prediction and the second is a rolling window approach. For details, see Section 5.

2.7 Forecast combinations

There is a large literature on forecast combinations, see Timmermann (2006) for an excellent survey. Although there is a consensus that forecast combinations improve forecast accuracy there is no consensus concerning how to form the forecast weights.

Given the findings in Stock and Watson (2004), Stock and Watson (2008b) and Andreou, Ghysels, and Kourtellos (2013a) we focus primarily on the Squared Discounted MSFE forecast combinations method, which delivers the highest forecast gains relative to other methods in many applications. The software also includes a BIC-based criterion as an option.

Let $\hat{y}_{j;t+h|t}^L$ denote the j^{th} individual out-of-sample forecast of y_{t+h}^L computed at date t. The forecast combination made at time t is a (time-varying) weighted average of n individual h-step ahead out-of-sample forecasts, $(\hat{y}_{1;t+h|t}^L, \ldots, \hat{y}_{n;t+h|t}^L)$, given as:

$$f_{c_M,t+h|t} = \sum_{j=1}^{n} w_{j,t}^h \hat{y}_{j;t+h|t}^L \tag{2.15}$$

where $(w^h_{1,t}, \ldots, w^h_{n,t})$ is the vector of combination weights formed at time t and c_M emphasizes the fact that the combined forecast depends on the class of models producing individual forecasts. A class of models is a collection of models involving either: (a) different high frequency series (the most common application) with each individual forecast $\hat{y}_{i,t+h|t}$ produced by a ADL-MIDAS regression involving the same type of polynomial and lag lengths for both the low and high frequency data, (b) different high frequency series with each individual forecast $\hat{y}^L_{j;t+h|t}$ produced by a ADL-MIDAS regression involving the different polynomial and lag lengths - for example selecting the best specification obtained with each individual series. In the latter case $\hat{y}^L_{k;t+h|t}$ and $\hat{y}^L_{j;t+h|t}$, for any k and j, differ not only because of different high frequency series but also with regards to polynomial and/or lag lengths. In principle one could also consider forecast combinations involving the same high frequency series, but different polynomial and/or lag lengths. Finally, one could consider *mega-combination* simply combining all the series, all the polynomial specifications and with different lag lengths. Obviously the user has to define the class of models that are considered for the forecast combination exercise.

We consider four different weighting schemes:

- Equally weighted weights

$$w_{i,t} = \frac{1}{n} \tag{2.16}$$

- BIC-weighted forecast

$$w_{i,t} = \frac{\exp(-BIC_i)}{\sum_{i=1}^{n} \exp(-BIC_i)} \tag{2.17}$$

- MSFE-related model averaging:

$$\begin{aligned} w_{i,t} &= \frac{m_{i,t}^{-1}}{\sum_{i=1}^{n} m_{i,t}^{-1}} \\ m_{i,t} &= \sum_{i=T_0}^{t} \delta^{t-i} (y^h_{s+h} - \hat{y}^h_{i,s+h|s})^2 \end{aligned} \tag{2.18}$$

  where T_0 is the first out-of-sample observation, $\hat{y}^h_{i,s+h|s}$ – out-of-sample forecast, δ – exponential averaging parameter.

  1. MSFE averaging: $\delta = 1$
  2. DMSFE averaging: $\delta = .9$

The BIC- and MSFE-based forecast combinations involve an estimation sample for all the models - involving either rolling windows or recursive window samples. In case of rolling windows, the user will have to specify the length of the window as well as the starting date. The BIC-weighted forecasts use the BIC from the latest available estimation sample. Hence, the forecast combination at time t for horizon h uses the BIC from the latest estimation sample - either rolling or recursive - with data up to time t. For MSFE-related model averaging we need - in addition to the estimation sample - to define a forecast evaluation sample which is expressed in formula (2.18) as T_0 to $.t$ This means that the estimation sample ends in T_0. All the parameter estimates for the class of models are taken as given - they are produced by either the rolling or recursive sample with data until T_0 - and forecasts $\hat{y}^h_{i,s+h|s}$ are produced over the sample starting $\hat{y}^h_{i,T_0+h|T_0}$ until $\hat{y}^h_{i,T_0+t+h|T_0+t}$. These h-step ahead forecasts yield a MFSE $m_{i,t}$ for each member i of the class of models c_M.

In a typical application, see e.g. Andreou, Ghysels, and Kourtellos (2013a), involving quarterly data (low frequency) and either daily or monthly high frequency series, the estimation sample is usually 10 years (rolling sample) whereas the forecast evaluation sample is two years - or 8 quarters. This means that the first forecast combinations can be produced after 12 years (10 years for estimating the first models and 2 to appraise their out-of-sample performance). Then, for every additional quarter in the sample, one can update the estimates, produce new out-of-sample forecasts and finally generate additional forecast combinations.

2.8 Nuts and bolts issues

It is important to warn the user upfront that when creating data input files the dates need to be saved as text in Excel (American format). Any other format (even if it shows dates as mm/dd/yy) will not work. Other data formats will create errors which, on first sight, may appear unrelated to dates.

Different data providers have different data storing conventions. The approach we took is that the user is responsible for arranging the data in the appropriate format. All that matters is that for each low frequency period there are m high frequency data points and both high and low frequency data start and end at the same time.

We opted for the user to arrange the data properly rather than provide a general approach.

Nevertheless, we briefly describe a typical situation encountered in MIDAS regression applications. Suppose quarterly data start in 1980Q1 and end in 2009Q2. Then the monthly data should start and 1980M01 and end at 2009M06. If there is insufficient data, i.e. some months at the beginning and/or the end are missing, NA values should be used. The 2009Q2 data should be aligned with 3 monthly observations 2009 M04, 2009 M05 and 2009 M06. Typically, the quarterly value of 2009Q2 becomes available after 2009M06. But this is a choice of the user. Ultimately, it is part of MIDAS regression models to specify which data is available at which time.

The historical data should be stored in a format compatible to the MIDAS toolbox. For instance the data input file of a quarterly sampled variable should look like the following:

```
DATE VALUE
1947-03-01 237.2
1947-06-01 240.4
1947-09-01 244.5
1947-12-01 254.3
. . . . . .

2010-09-01 14605.5
2010-12-01 14755.0
2011-03-01 14867.8
2011-06-01 14996.8
```

In this file, the field VALUE is the value of the input variable in the quarter starting with the month appearing in the DATE field. For example in the figure above, 14867.8 refers to the quarter 2011Q1. If you are using a different format of dating, you will need to align low frequency date to make sure the match with the high frequency data is correct. Similarly in a data input file of a monthly sampled variable, such as, date value

```
DATE VALUE
1947-01-01 235.8
1947-02-01 250.3
1947-03-01 247.5
. . . . . .
2010-02-01 456.0
2011-03-01 442.3
```

```
2011-04-01 473.6
```

the field VALUE is the value of the input variable in the month corresponding DATE field. Therefore in this table, 473.6 refers to 2011 M04.

In principle, you don't need to have any other Matlab Toolbox to work with MIDAS Toolbox. There is only one simple $m.file$ you may want to put into MIDAS Toolbox directory to be able to print plots. It is called *suptitle.m*, a function that puts a title above all subplots. If you receive a message stating that this $m.file$ is missing, then please add it into your MIDAS Toolbox folder. It is available online.

Practical implementation of MIDAS involves issues that are typical for regression analysis, yet there are some not commonly encountered in standard regression problems and they pertain to the mixed sampling nature of the data.

Take for example a quarterly/daily combination and consider the situation of holidays occurring throughout a calendar year. This will create an unequal number of days on a quarter by quarter basis. While one can take different approaches towards this, we treat the holidays as missing values in the MIDAS polynomial. They will be linearly interpolated using various schemes.

The algorithms can be grouped into (1) specifications with the same number of MIDAS lags each period and (2) specifications that cover the same time span each period.

Define a sequence of MIDAS polynomial weights $c_{\tau_1}, c_{\tau_2}, \ldots$. Then we have the following:

1. Equally-spaced specification.

   (a) It is characterized by the fact that each observation point $\{y_t, Xfactor_t, Xmidas_t\}$ has the same number of MIDAS lags $Xmidas_t$. As a result, different periods may have different time span coverage *but the same number of lags*. The sequence of weights $c_{\tau i}, c_{\tau_{i+1}}, \ldots$ is defined in this case as $c_i, c_{i+1}, \ldots$.

2. Real-time specifications. They are characterized by unequal number of MIDAS lags over time that cover *the same time span*.

   (a) Real time specification. The distance between $c_{\tau i}$ and $c_{\tau_{i+1}}$ is proportionate to $\tau_i - \tau_{i+1}$. No artificial observations are inserted in the MIDAS polynomial.

(b) Real time specification with zeros at the end. Depending on the number of calendar days within a given time interval all missing days are added as zeros to the end of Xmidas lag structure. MIDAS weights are constructed as in the equally-spaced case.[2]

2.9 Timing of lags

It is worth briefly elaborating on the timing of high frequency data. Recall that with i/m additional observations the horizon h shrinks to $h - i/m$, and as noted earlier equation (2.1) becomes:

$$\hat{y}^L_{T+h|T+i/m} = \hat{a}_{h-i/m,T} + b_{h-i/m,T} C(L^{1/m}; \theta_{h-i/m,T}) x^H_{t_L+i/m}$$

There are both issues of convention/notation and issues of substance when we discuss the timing of lags. What matters in MIDAS regressions - and for that matter pretty much any time series model - is to properly take into account the alignment of information sets. Now, real-time forecasters will tell you that many macro data are released with delays. Some are delayed by one month, some by even more delays. So, when one runs a regression with say quarterly GDP growth and monthly employment, with info prior to Q1 (say end of sample T) one has to decide whether the December employment data is in the info set at time T. In real-time one may only have the November data available at the end of December. Does one call this lag $T - 1/3$ or rather T since it is released by end of December. Does one ignore the publication lag - as many applied econometricians do - then one could use the December figure and thus x^H_T. The same applies to nowcasting, namely does $T + 1/3$ refer to the January figure, or if publication delays are taken into account only the December number released in January? There is no general answer here. Notation-wise we keep it in line with information sets, where the user decides and consequently aligns the data properly.[3]

[2] Please note that normalization of the polynomial in this case is different from the equally-spaced specification.

[3] For a discussion of publication delays and their impact on estimation see for instance Ghysels, Horan, and Moench (2014).

3 GARCH-MIDAS and DCC-MIDAS

The GARCH-MIDAS model decomposes the conditional variance into the short-run and long-run components. The former is a mean-reverting GARCH(1,1)-like process, while the latter is determined by the history of the realized volatility or macroeconomic variables weighted by the MIDAS polynomials.

The DCC-MIDAS model is a multivariate extension to the GARCH-MIDAS model with dynamic correlations. The DCC-MIDAS model decomposes the conditional covariance matrix into the variances and the correlation matrix, with a two-step model specification and estimation strategy. In the first step, conditional variances are estimated by the univariate GARCH-MIDAS models. In the second step, observations are deflated by the estimated means and conditional variances, and the standardized residuals are thus constructed. The standardized residuals have a correlation matrix with GARCH-MIDAS-like dynamics. The long-run component is determined by the history of sample autocorrelations under the MIDAS weights.

Following Engle, Ghysels, and Sohn (2013), we specify a GARCH-MIDAS model by equation (3.19) to (3.25).

$$r_{it} = \mu + \sqrt{\tau_t g_{it}}\varepsilon_{it}, \tag{3.19}$$

$$g_{it} = (1-\alpha-\beta) + \alpha\frac{(r_{i-1,t}-\mu)^2}{\tau_t} + \beta g_{i-1,t}, \tag{3.20}$$

$$\tau_t = m + \theta\sum_{k=1}^{K}\psi_k(w)V_{t-k}, \tag{3.21}$$

$$V_t = \sum_{i=1}^{N} r_{it}^2, \tag{3.22}$$

$$or, V_t = \frac{1}{N}\sum_{i=1}^{N} x_{it}, \tag{3.23}$$

$$\psi_k(w) \propto (1-\frac{k}{K})^{w-1}, \tag{3.24}$$

$$or, \psi_k(w) \propto (1-\frac{k}{K})^{c_1-1}(\frac{k}{K})^{c_2-1}. \tag{3.25}$$

Take daily/monthly aggregation as an example. In equation (3.19), r_{it} denotes an observation (say, an asset return) of day i in month t. The conditional variance is decomposed into the short-run component g_{it} and the long-run component τ_t. The former has a GARCH(1,1)-like recursion specified by equation (3.20), while the latter is determined by the realized volatility or macroeconomic series. V_t in equation (3.22) is the realized volatility of the month, and V_t in equation (3.23) represents the monthly average of an exogenous variable. If the macroeconomic variable x_{it} is sampled at the monthly frequency, then its value is fixed for $i = 1, ..., N$. A history of $V_{t-1}, V_{t-2}, ..., V_{t-k}$ weighted by Beta polynomials (i.e., equation (3.24) or (3.25)) captures the long-run volatility. Of course, other weight specifications in Section 2.3 are also good.

Colacito, Engle, and Ghysels (2011) extend the model to the multivariate case. In the DCC-MIDAS model, the observations are m dimensional time series data. The conditional covariance matrix is decomposed into m conditional variances and a $m \times m$ conditional correlation matrix, hence a two-step specification strategy. Each of the conditional variances is assumed to follow a GARCH-MIDAS model. The correlation matrix evolves over time. Consider a quasi-correlation matrix Q_t whose (i, j) element q_{ijt} has the dynamics

$$q_{ijt} = \rho_{ijt}(1 - a - b) + a\varepsilon_{i,t-1}\varepsilon_{j,t-1} + bq_{ij,t-1}, \tag{3.26}$$

where $\varepsilon_{i,t-1}$ is the standardized residuals of the i^{th} series in period $t - 1$, so q_{ijt} has a GARCH(1,1)-like dynamics. The long-run component ρ_{ijt} is the (i, j) element of ρ_t, namely the MIDAS weighted-sum of the sample correlation matrices

$$\rho_t = \sum_{k=1}^{K} \psi_k(w)c_{t-k}, \tag{3.27}$$

where c_t is computed by the sample correlation matrix from the observations.

The correlation matrix is a rescale of the quasi-correlation matrix so that the diagonals are unity:

$$R_t = diag(Q_t)^{-1/2} Q_t diag(Q_t)^{-1/2}. \tag{3.28}$$

4 MIDAS quantile regressions

We use the MIDAS quantile model in Ghysels, Plazzi, and Valkanov (2016). The α conditional quantile of the $n-$period return $r_{t,n}$ is an affine function of predetermined variables. The regressors are daily returns weighed by the MIDAS polynomial.

The model can be written as

$$q_\alpha(r_{t,n}) = \beta_0 + \beta_1 Z_{t-1}(\kappa), \tag{4.29}$$

$$Z_{t-1}(\kappa) = \sum_{d=0}^{D} \psi_d(\kappa) x_{t-1-d}, \tag{4.30}$$

where $q_\alpha(r_{t,n})$ is the α conditional quantile of the $n-$period return $r_{t,n}$, and x_{t-1-d} is the high-frequency conditioning variable with the MIDAS weight $\psi_d(\kappa)$. The conditioning variable can be chosen as the absolute returns, which capture the temporal variation in the conditional distribution of returns (see Ghysels, Plazzi, and Valkanov (2016)).

Suppose that we have the daily return series $r_t, t = 1, ..., T$, the software implements MIDAS quantile regression in this way: First, it generates $n-$period returns by aggregating daily returns: $r_{t,n} = \sum_{i=0}^{n-1} r_{t+i}$. Second, it generates the conditioning variable by taking the absolute values of returns: $x_{t-1-d} = |r_{t-1-d}|$. Third, it chooses the MIDAS Beta polynomial $\psi_d(\kappa) \propto (1 - \frac{d}{D})^{\kappa-1}$ and compute the weighted average of the conditioning variable. Fourth, it estimates the unknown parameters β_0, β_1, κ by minimizing the asymmetric loss function $\sum_{t=1}^{T} \rho_\alpha(e_t)$, where ρ_α is the check function, namely $\rho_\alpha(x) = x(\alpha - I(x < 0))$, and $e_t = r_{t,n} - \beta_0 - \beta_1 Z_{t-1}(\kappa)$.

5 Software Usage

5.1 MIDAS regression

To use the MIDAS package, first prepare the mixed frequency data: DataY, DataYdate, DataX, DataXdate. As the name suggests, DataY is the low frequency dependent variable data specified as a column vector. DataYdate indicates the dates corresponding to the low frequency observations. A variety of date formats are supported. For instance, '1985-01-01'

, '01/01/1985', 'January 1, 1985' are all legitimate dates. DataYdate is a cell array in which each element is a date string. Similarly, DataX and DataXdate are the high frequency data and dates.[4]

The function MIDAS_ADL.m in the software package is the gateway to the MIDAS regression. The required input arguments are DataY, DataYdate, DataX, DataXdate. In addition, optional input arguments are specified as name-value pairs, which detail the mixed frequency model specifications. The options include:

- 'Xlag': the number of lagged the high frequency explanatory variables. It can be a scalar or descriptive string such as '3m','1q'. The default value is 9, which means that the explanatory variables include 9 lagged high frequency variables. 'Xlag' = 0 will yield what is essentially the OLS output of a low frequency data AR regression model. The out-of-sample forecast results with MSE are produced as well.

- 'Ylag': the number of lags of the autoregressive low frequency variables. It can be a scalar or descriptive string such as '3m','1q'. The default value is 1, which means that the predictors also include a lagged low frequency variable. When 'Ylag' = 4, for example, then the regression will include Y^Q_{t-1}, Y^Q_{t-2}, Y^Q_{t-3} and Y^Q_{t-4}. If the user only wants Y^Q_{t-4} one needs to put may put 'Ylag' = 4. Similarly, we one can put something like Ylag = 3,6,9.

- 'Horizon': MIDAS lead/lag specification. It can be a scalar or descriptive string such as '3m','1q'. The default value is 1, which implies that dependent variables in period t is accompanied by high frequency regressors in period $t-1$, $t-2$, etc. If 'Horizon' is reset to 2, dependent variables in period t will be regressed on high frequency regressors in period $t-2$, $t-3$, etc. A negative integer value of 'Horizon' is also supported. In that case, it is a MIDAS with leads of high frequency regressors. Proper setting of 'Horizon' can offset the impact of different date styles of the low frequency data. For example, if the quarterly dates are coded as '01/01/1985', 'Horizon' = 1 implies that lagged high frequency monthly regressors start from '12/01/1984'. However, if the same quarterly data is recorded as '03/01/1985' instead, 'Horizon' can be set to 3 so that the lagged high frequency data still start from '12/01/1984'. In case of any confusion on the regression dates, refer to the time frame displayed on the screen.

[4] As noted earlier, it is that when creating data input files the dates need to be saved as text in Excel (American format). Any other format (even if it shows dates as mm/dd/yy) will not work. Other data formats will create errors which, on first sight, may appear unrelated to dates.

- 'EstStart': start date of the estimation window, specified as a date string. By default, estimation starts from the beginning of the sample, adjusted by lagged values. It is illegal to set the 'EstStart' out of the sample range. In that case, the program will explain the earliest date that can be supported.

- 'EstEnd': terminal date of the estimation window, specified as a date string. By default, estimation terminates at the end of the sample, adjusted by the 'Horizon' value. If 'EstEnd' is earlier than the (adjusted) last observation date, out-of-sample forecast will be performed and the forecast values will be compared with the unused observations. Best practice is to leave some observations for the out-of-sample forecast, which provides some assessment of the model performance.

- 'ExoReg': Exogenous low-frequency regressors specified as a T-by-k matrix, where T is the length of the data, k is the number of exogenous regressors. The frequency of exogenous regressors must be the same as the low frequency dependent variable DataY. The sample size must be at least as large as DataY. Do not include a constant, for it is automatically added to the regression. For instance, if the MIDAS is augmented by known factors, 'ExoReg' accommodates the factors data.

- 'ExoRegDate': Dates associated with exogenous regressors data specified as a T-by-1 cell array in which each element is a date string. All exogenous regressors share the same dates.

- 'Method': an option for estimation methods. Its value can be

  – 'FixedWindow' (default): Estimation window is defined as [estStart, estEnd]. Then the multi-step forecast values are compared with the unused observations.

  – 'RollingWindow': Multiple windows are defined as [estStart+i, estEnd+i]. Then the one-step forecast value is compared with the observations in estEnd+i+1.

  – 'Recursive': Multiple windows are defined as [estStart, estEnd+i]. Then the one-step forecast value is compared with the observations in estEnd+i+1.

- 'Polynomial': functional form of the MIDAS weights. Its value can be

  – 'Beta' (default): Normalized beta density with a zero last lag

  – 'BetaNN': Normalized beta density with a non-zero last lag

  - 'ExpAlmon': Normalized exponential Almon lag polynomial
  - 'UMIDAS': Unrestricted coefficients
  - 'Step': Polynomial with step functions
  - 'Almon': Almon lag polynomial of order p

- 'PolyStepFun': thresholds of the step function. This option is relevant only if 'Polynomial' is set to 'Step'.
- 'AlmonDegree': number of lags of the Almon lag. This option is relevant only if 'Polynomial' is set to 'Almon'.
- 'Discount': discount factor to compute the discounted mean squared error of forecast. The default value is 0.9.
- 'DiscountIncrease': logical value indicating increasing or decreasing sequence of discounted mean squared error of forecast. An increasing sequence indicates larger weights on dates that have lower forecast uncertainty. Decreasing sequence puts larger weights on more recent dates. The default value is true.
- 'Display': the screen display style. Its value can be
  - 'full' (default): full display of the regression time frame, and the estimator summary
  - 'time': display of the regression time frame
  - 'estimate': display of the estimator summary
  - 'off': no display on the screen
- 'PlotWeights': logical value indicating whether to plot the MIDAS weights after parameter estimation. The default is true.

When the function MIDAS_ADL.m is called, it will first parse the mixed frequency data and model specifications. Intermediate results are stored in a struct array called 'MixedFreqData'. After that stage, a MIDAS regression is well defined and nonlinear least squares is employed to obtain the estimated model parameters. The estimation results are stored in a struct array called 'OutputEstimate'. Lastly, if the 'EstEnd' is earlier than the last observation, out-of-sample forecast is performed. The forecast values are compared with the realized values so as to evaluate the forecasting power of the model. The forecast results are stored in a struct array called 'OutputForecast'. To evaluate the mean squared errors of predictions, we may set the name-value pair 'Discount' and 'DiscountIncrease' if the default values are not satisfactory. For example, if it is desirable to put larger weights on more recent dates, we may try MIDAS_ADL(DataY, DataYdate, DataX, DataXdate, 'DiscountIncrease', 0).

'OutputForecast' includes the following fields:

- Yf: point forecast of the low frequency data after 'EstEnd'
- RMSE: root mean squared error of forecast
- MSFE: mean squared error of forecast
- DMSFE: discounted mean squared error of forecast
- aic: Akaike information criteria of the regression (a copy from OutputEstimate)
- bic: Bayesian information criteria of the regression (a copy from OutputEstimate)

'OutputEstimate' includes the following fields:

- model: description of the MIDAS weight polynomial
- paramName: description of the model parameters
- estParams: estimated parameters
- EstParamsCov: covariance matrix of the estimated parameters
- se: standard errors of the estimated parameters
- tstat: t statistics of the estimated parameters
- sigma2: disturbance variance of the mixed frequency regression
- yfit: fitted low frequency data
- resid: residual of the mixed frequency regression

- estWeights: estimated coefficients of high frequency regressors (weights)
- logL: log likelihood of the low frequency data
- r2: R2 statistics of the regression
- aic: Akaike information criteria of the regression
- bic: Bayesian information criteria of the regression

'MixedFreqData' includes the following fields:

- EstY: low frequency data in the estimation periods, a T1-by-1 vector
- EstYdate: dates of low frequency data in the estimation periods, a T1-by-1 vector of MATLAB serial date numbers
- EstX: high frequency data in the estimation periods, a T1-by-Xlag matrix
- EstXdate: dates of high frequency data in the estimation periods, a T1-by-Xlag matrix of MATLAB serial date numbers
- EstLagY: low frequency lagged regressors in the estimation periods, a T1-by-Ylag matrix
- EstLagYdate: dates of low frequency lagged regressors in the estimation periods, a T1-by-Ylag matrix of MATLAB serial date numbers
- OutY: low frequency data in the forecasting periods, a T2-by-1 vector
- OutYdate: dates of low frequency data in the forecasting periods, a T2-by-1 vector of MATLAB serial date numbers
- OutX: high frequency data in the forecasting periods, a T2-by-Xlag matrix
- OutXdate: dates of high frequency data in the forecasting periods, a T2-by-Xlag matrix of MATLAB serial date numbers
- OutLagY: low frequency lagged regressors in the forecasting periods, a T2-by-Ylag matrix

- OutLagYdate: dates of low frequency lagged regressors in the forecasting periods, a T2-by-Ylag matrix of MATLAB serial date numbers
- Xlag: number of lagged the high frequency explanatory variables, in numerical format
- Ylag: number of lagged the low frequency explanatory variables, in numerical format

We revisit some of the examples in Armesto, Engemann, and Owyang (2010). In particular we run ADL-MIDAS regressions to forecast GDP growth with monthly employment growth. Seasonally adjusted real GDP quarterly data are taken from St. Louis FRED website and the real GDP growth is computed as log-quarterly first difference. Monthly total employment non-farm payrolls data are also taken from FRED and log-monthly first differences are computed.

The data are stored in the spreadsheet 'mydata.xlsx'. First, we load the data:

```
[DataY,DataYdate] = xlsread('mydata.xlsx','sheet1');
DataYdate = DataYdate(2:end,1);
[DataX,DataXdate] = xlsread('mydata.xlsx','sheet2');
DataXdate = DataXdate(2:end,1);

DataXgrowth = log(DataX(2:end)./DataX(1:end-1))*100;
DataYgrowth = log(DataY(2:end)./DataY(1:end-1))*100;
DataX = DataXgrowth;
DataY = DataYgrowth;
DataYdate = DataYdate(2:end);
DataXdate = DataXdate(2:end);
```

Then we estimate the model with a variety of weight polynomials by calling the function MIDAS_ADL.m. Note that all optional input arguments have default values. We use verbose syntax for illustration of those name-value pairs.

```
Xlag = 9;
Ylag = 1;
Horizon = 3;
EstStart = '1985-01-01';
EstEnd = '2009-01-01';
Method = 'fixedWindow';
```

```
[OutputForecast1,OutputEstimate1,MixedFreqData]...
     = MIDAS_ADL(DataY,DataYdate,DataX,DataXdate,...
    'Xlag',Xlag,'Ylag',Ylag,'Horizon',Horizon,'EstStart',EstStart,'EstEnd',...
    EstEnd,'Polynomial','beta','Method',Method,'Display','full');
[OutputForecast2,OutputEstimate2] = MIDAS_ADL(DataY,DataYdate,DataX,DataXdate,...
    'Xlag',Xlag,'Ylag',Ylag,'Horizon',Horizon,'EstStart',EstStart,'EstEnd',...
    EstEnd,'Polynomial','betaNN','Method',Method,'Display','estimate');
[OutputForecast3,OutputEstimate3] = MIDAS_ADL(DataY,DataYdate,DataX,DataXdate,...
    'Xlag',Xlag,'Ylag',Ylag,'Horizon',Horizon,'EstStart',EstStart,'EstEnd',...
    EstEnd,'Polynomial','expAlmon','Method',Method,'Display','estimate');
[OutputForecast4,OutputEstimate4] = MIDAS_ADL(DataY,DataYdate,DataX,DataXdate,...
    'Xlag',Xlag,'Ylag',Ylag,'Horizon',Horizon,'EstStart',EstStart,'EstEnd',...
    EstEnd,'Polynomial','umidas','Method',Method,'Display','estimate');
[OutputForecast5,OutputEstimate5] = MIDAS_ADL(DataY,DataYdate,DataX,DataXdate,...
    'Xlag',Xlag,'Ylag',Ylag,'Horizon',Horizon,'EstStart',EstStart,'EstEnd',...
    EstEnd,'Polynomial','step','Method',Method,'Display','estimate');
[OutputForecast6,OutputEstimate6] = MIDAS_ADL(DataY,DataYdate,DataX,DataXdate,...
    'Xlag',Xlag,'Ylag',Ylag,'Horizon',Horizon,'EstStart',EstStart,'EstEnd',...
    EstEnd,'Polynomial','Almon','Method',Method,'Display','estimate');
```

In the full display mode, the time frame of the regression is shown on the screen, which helps to verify the mixed frequency date specification. The estimation results will also be reported on the screen. Occasionally, numerical optimization routine does not yield convergent results and it is possible that the returned estimator covariance matrix is not positive definite. In that case, model specification should be carefully reviewed. Diagnostics and new proposals might be in need.

```
Frequency of Data Y: 3 month(s)
Frequency of Data X: 1 month(s)
Start    Date: 01-Jan-1985
Terminal Date: 01-Jan-2009

Mixed frequency regression time frame:
Reg Y(01/01/85) on Y(10/01/84),X(10/01/84),X(09/01/84),...,X(02/01/84)
Reg Y(04/01/85) on Y(01/01/85),X(01/01/85),X(12/01/84),...,X(05/01/84)
...
Reg Y(01/01/09) on Y(10/01/08),X(10/01/08),X(09/01/08),...,X(02/01/08)

MIDAS: Normalized beta density with a zero last lag
```

```
    ''                  ' Estimator'    'SE'         't-stat'
    'Const'             [     0.6656]   [0.1353]     [ 4.9184]
    'HighFreqSlope'     [     1.9121]   [0.5592]     [ 3.4190]
    'Beta1'             [     0.9904]   [0.0672]     [14.7435]
    'Beta2'             [     6.6157]   [9.6620]     [ 0.6847]
    'Ylag1'             [     0.2847]   [0.1156]     [ 2.4619]
```

Since the estimation sample runs from 1985-01-01 to 2009-01-01 and the data for GDP growth in the example runs until the second quarter of 2011, there are nine quarters left for the out-of-sample evaluation. By extracting the RMSE of each model, we can compare their forecasting power:

```
fprintf('RMSE Beta:          %5.4f\n',OutputForecast1.RMSE);
fprintf('RMSE Beta Non-Zero: %5.4f\n',OutputForecast2.RMSE);
fprintf('RMSE Exp Almon:     %5.4f\n',OutputForecast3.RMSE);
fprintf('RMSE U-MIDAS:       %5.4f\n',OutputForecast4.RMSE);
fprintf('RMSE Stepfun:       %5.4f\n',OutputForecast5.RMSE);
fprintf('RMSE Almon:         %5.4f\n',OutputForecast6.RMSE);
```

In this example, the weight function of the normalized beta density with a non-zero last lag outperforms other models, though other weight specifications are not obviously inferior.

```
RMSE Beta:          0.5650
RMSE Beta Non-Zero: 0.5210
RMSE Exp Almon:     0.5641
RMSE U-MIDAS:       0.5424
RMSE Stepfun:       0.5252
RMSE Almon:         0.5329
```

Though the function MIDAS_ADL.m can plot the weights by setting the name-value pair 'PlotWeights', it is more desirable to have multiple curves in one figure for comparison. So we extract the weights from the estimation output and plot them manually.

```
Xlag = MixedFreqData.Xlag;
for m = 1:6
    weights = eval(['OutputEstimate',num2str(m),'.estWeights']);
```

```
    subplot(2,3,m);plot(1:Xlag,weights);title(['Model ',num2str(m)])
end
```

Users are encouraged to modify the model specification and see how the estimation/forecast results change accordingly. For example, consider resetting the name-value pair 'Horizon':

```
% Reg Y(01/01/85) on Y(10/01/84),X(10/01/84),X(09/01/84),...,X(02/01/84)
MIDAS_ADL(DataY,DataYdate,DataX,DataXdate,'EstStart',EstStart,'Horizon',3);

% Reg Y(01/01/85) on Y(10/01/84),X(11/01/84),X(10/01/84),...,X(03/01/84)
MIDAS_ADL(DataY,DataYdate,DataX,DataXdate,'EstStart',EstStart,'Horizon',2);

% Reg Y(01/01/85) on Y(10/01/84),X(12/01/84),X(11/01/84),...,X(04/01/84)
MIDAS_ADL(DataY,DataYdate,DataX,DataXdate,'EstStart',EstStart,'Horizon',1);
```

We can slightly tweak the program to make it suitable for nowcasting. We estimate an ADL-MIDAS with two months of leads. If we reset 'Horizon' to 1, we will be forecasting with one month horizon rather than one quarter (we changed $1q$ to $1m$).

```
Mixed frequency regression time frame:
Reg Y(01/01/85) on Y(10/01/84),X(12/01/84),X(11/01/84),...,X(04/01/84)
Reg Y(04/01/85) on Y(01/01/85),X(03/01/85),X(02/01/85),...,X(07/01/84)
...
Reg Y(01/01/09) on Y(10/01/08),X(12/01/08),X(11/01/08),...,X(04/01/08)
```

```
RMSE Beta:          0.5214
RMSE Beta Non-Zero: 0.5176
RMSE Exp Almon:     0.5238
RMSE U-MIDAS:       0.5150
RMSE Stepfun:       0.5244
RMSE Almon:         0.5041
```

Note that we have made improvements in the RMSE across all polynomial specifications with the two extra months of information. The output structure allows one to appraise the new forecasts, parameter estimates, etc.

We turn our attention to the recursive estimation by setting the name-value pair 'Method', 'rollingwindow'. When either rolling or recursive estimation is chosen, the program re-estimates the model recursively. At each iteration, the program produces a rolling or recursive estimation/forecast of one step ahead. Substantial improvement are made in the recursive updates of the parameter estimates.

```
RMSE Beta:           0.3146
RMSE Beta Non-Zero: 0.3311
RMSE Exp Almon:      0.3280
RMSE U-MIDAS:        0.3272
RMSE Stepfun:        0.3245
RMSE Almon:          0.3376
```

Finally, we consider the model averaging by adding the industrial production as a second high frequency series. In the first model, we use the monthly total employment non-farm payrolls to predict GDP growth, while the second model uses the industrial production as the high frequency predictors. With two sets of forecast outputs, we use the function ForecastCombine.m to combine the forecast according to the MSFE, MSFE, aic/bic and flat weights respectively.

```
YfMSFE = ForecastCombine(OutputForecast1,OutputForecast2);
YfDMSFE = ForecastCombine(OutputForecast1,OutputForecast2,'DMSFE');
YfAIC = ForecastCombine({OutputForecast1,OutputForecast2},'aic');
YfBIC = ForecastCombine({OutputForecast1,OutputForecast2},'bic');
YfFlat = ForecastCombine(OutputForecast1,OutputForecast2,'flat');
```

```
Forecast by Model 1
    0.8820    1.3111    1.4065    1.4384    1.1836    1.2412    1.1257

Forecast by Model 2
    0.7436    1.1792    1.1371    1.2726    0.9524    1.2767    1.0058

Combined forecast by MSFE
    0.8121    1.2444    1.2704    1.3546    1.0668    1.2592    1.0651

Combined forecast by DMSFE
```

```
    0.8184    1.2505    1.2827    1.3622    1.0774    1.2576    1.0706

Combined forecast by AIC
    0.7439    1.1794    1.1376    1.2729    0.9528    1.2767    1.0060

Combined forecast by BIC
    0.7439    1.1794    1.1376    1.2729    0.9528    1.2767    1.0060

Combined forecast by equal weight
    0.8128    1.2451    1.2718    1.3555    1.0680    1.2590    1.0657
```

5.2 GARCH-MIDAS and DCC-MIDAS

GarchMidas is a MATLAB function for estimating a GARCH-MIDAS model. The syntax is

```
[...] = GarchMidas(y,name,value,...)
```

The required input argument is y, a $T \times 1$ vector of observations. The optional name-value pairs include:

- 'X': $T \times 1$ macroeconomic data that determines the long-run conditional variance. If X is not specified, realized volatility will be used. X should be of the same length as y; repeat X values to match the date of y if necessary. Only one regressor is supported. The default is empty (realized volatility).
- 'Period': A scalar integer that specifies the aggregation periodicity (N). How many days in a week/month/quarter/year? How long is the secular component (τ_t) fixed? The default is 22 (as in a daily/monthly aggregation).
- 'NumLags': A scalar integer that specifies the number of lags (K) in filtering the secular component by MIDAS weights. The default is 10 (say a history of 10 weeks/months/quarters/years).
- 'EstSample': A scalar integer that specifies a subsample y(1:EstSample) for parameter estimation. The remaining sample is used for conditional variance forecast and validation. The default is length(y), no forecast.

- 'RollWindow': A logical value that indicates rolling window estimation on the long-run component. If true, the long-run component varies every period. If false, the long-run component will be fixed for a week/month/quarter/year. The default is false.

- 'LogTau': A logical value that indicates logarithmic long-run volatility component. The default is false.

- 'Beta2Para': A logical value that indicates two-parameter Beta MIDAS polynomial (equation (3.25)). The default is false (one-parameter Beta polynomial,equation (3.24))

- 'Options': The FMINCON options for numerical optimization. For example,
  Display iterations: optimoptions('fmincon','Display','Iter');
  Change solver: optimoptions('fmincon','Algorithm','active-set');
  The default is the FMINCON default choice.

- 'Mu0': MLE starting value for the location-parameter (μ). The default is the sample average of observations.

- 'Alpha0': MLE starting value for α in the short-run GARCH(1,1) component. The default is 0.05.

- 'Beta0': MLE starting value for β in the short-run GARCH(1,1) component. The default is 0.9.

- 'Theta0': MLE starting value for the MIDAS coefficient $\sqrt{\theta}$ in the long-run component. If the name-value pair 'ThetaM' is true, it is θ. The default is 0.1.

- 'W0': MLE starting value for the MIDAS parameter in the long-run component. The default is 5.

- 'M0': MLE starting value for the location-parameter $\sqrt{m}$ in the long-run component. If the name-value pair 'ThetaM' is true, it is m. The default is 0.01.

- 'Gradient': A logical value that indicates analytic gradients in MLE. The default is false.

- 'AdjustLag': A logical value that indicates MIDAS lag adjustments for initial observations due to missing presample values. The default is false.

- 'ThetaM': A logical value that indicates not taking squares for the parameter θ and m in the long-run volatility component. The default is false (they are squared).
- 'Params': Parameter values for $\mu, \alpha, \beta, \theta, w, m$. In that case, the program will skip MLE, and just infer the conditional variances based on the specified parameter values. The default is empty (need parameter estimation).
- 'ZeroLogL': A vector of indices between 1 and T, which selects a subset of dates and forcefully resets the likelihood values of those dates to zero. For example, use ZeroLogL to ignore initial likelihood values. The default is empty (no reset).

The output arguments include:

- EstParams: Estimated parameters for $\mu, \alpha, \beta, \theta, w, m$.
- EstParamCov: Estimated parameter covariance matrix.
- Variance: $T \times 1$ conditional variances.
- LongRunVar: $T \times 1$ long-run component of the conditional variances.
- ShortRunVar: $T \times 1$ short-run component of the conditional variances.
- logL: $T \times 1$ log likelihood. Initial observations may be assigned a flag of zero.

DccMidas is a MATLAB function for estimating a DCC-MIDAS model. The syntax is

```
[...] = DccMidas(Data,name,value,...)
```

The required input argument is Data, a $T \times n$ matrix of observations. The optional name-value pairs include:

- 'Period': A scalar integer that specifies the aggregation periodicity (N). How many days in a week/month/quarter/year? How long is the secular component (τ_t) fixed? The default is 22 (as in a daily/monthly aggregation).

- 'NumLagsVar': A scalar integer that specifies the number of lags (K) in filtering the secular component by MIDAS weights. This is for the first step GARCH-MIDAS model. The default is 10 (say a history of 10 weeks/months/quarters/years).

- 'NumLagsCorr': A scalar integer that specifies the number of lags (K) in filtering the secular component by MIDAS weights. This is for the second step estimation of correlation matrix. The default is 10 (say a history of 10 weeks/months/quarters/years).

- 'EstSample': A scalar integer that specifies a subsample y(1:EstSample) for parameter estimation. The remaining sample is used for conditional variance forecast and validation. The default is length(y), no forecast.

- 'RollWindow': A logical value that indicates rolling window estimation on the long-run component. If true, the long-run component varies every period. If false, the long-run component will be fixed for a week/month/quarter/year. The default is false.

- 'LogTau': A logical value that indicates logarithmic long-run volatility component. The default is false.

- 'Beta2Para': A logical value that indicates two-parameter Beta MIDAS polynomial (equation (3.25)). The default is false (one-parameter Beta polynomial,equation (3.24))

- 'Options': The FMINCON options for numerical optimization. For example,
  Display iterations: optimoptions('fmincon','Display','Iter');
  Change solver: optimoptions('fmincon','Algorithm','active-set');
  The default is the FMINCON default choice.

- 'Mu0': MLE starting value for the location-parameter (μ). The default is the sample average of observations.

- 'Alpha0': MLE starting value for α in the short-run GARCH(1,1) component. The default is 0.05.

- 'Beta0': MLE starting value for β in the short-run GARCH(1,1) component. The default is 0.9.

- 'Theta0': MLE starting value for the MIDAS coefficient $\sqrt{\theta}$ in the long-run component. If the name-value pair 'ThetaM' is true, it is θ. The default is 0.1.

- 'W0': MLE starting value for the MIDAS parameter in the long-run component. The default is 5.
- 'M0': MLE starting value for the location-parameter $\sqrt{m}$ in the long-run component. If the name-value pair 'ThetaM' is true, it is m. The default is 0.01.
- 'CorrA0': MLE starting value for a in the GARCH(1,1) component. It is either a scalar (if all variables share it) or a column vector (if each variable has its own parameter). This is for the second step correlation matrix estimation. The default is 0.05 (or a vector expansion).
- 'CorrB0': MLE starting value for b in the GARCH(1,1) component. It is either a scalar (if all variables share it) or a column vector (if each variable has its own parameter). This is for the second step correlation matrix estimation. The default is 0.05 (or a vector expansion).
- 'CorrW0': MLE starting value for the MIDAS parameter w in the long-run component. It is a scalar. Vector is not supported. The default is 0.05.
- 'MorePara': A logical value that indicates multivariate series have different a, b. However, the program only supports a single w. This is for the second step correlation matrix estimation. The default is false (parameters a, b, w are shared by all variables).
- 'Gradient': A logical value that indicates analytic gradients in MLE. The default is false.
- 'AdjustLag': A logical value that indicates MIDAS lag adjustments for initial observations due to missing presample values. The default is false.
- 'ThetaM': A logical value that indicates not taking squares for the parameter θ and m in the long-run volatility component. The default is false (they are squared).
- 'Params': Parameter values for $\mu, \alpha, \beta, \theta, w, m$. In that case, the program will skip MLE, and just infer the conditional variances based on the specified parameter values. The default is empty (need parameter estimation).
- 'ZeroLogL': A vector of indices between 1 and T, which select a subset of dates and forcefully reset the likelihood values of those dates to zero. For example, use ZeroLogL to ignore initial likelihood values. The default is empty (no reset).

The output arguments include:

- EstParamsStep1: $6 \times n$ estimated parameters for $\mu, \alpha, \beta, \theta, w, m$.
- EstParamCovStep1: $6 \times 6 \times n$ estimated parameter covariance matrix.
- EstParamsStep2: 3×1 or $(2n+1) \times 1$ estimated parameters, obtained from the second-step autocorrelation matrix estimation.
- EstParamCovStep2: 3×3 or $(2n+1) \times (2n+1)$ estimated parameter covariance matrix.
- Variance: $T \times n$ conditional variances.
- LongRunVar: $T \times n$ long-run component of the conditional variances.
- CorrMatrix: $n \times n \times T$ conditional correlation matrices.
- LongRunCorrMatrix: $n \times n \times T$ long-run component of the correlation matrices.
- logL: $T \times 1$ log likelihood. Initial observations may be assigned a flag of zero.

We first consider a GARCH-MIDAS example. We downloaded the NASDAQ Composite Index daily return data (1971 - 2015) from the FRED Economic Data (NASDAQCOM). Though our data are not the same as those used in Engle, Ghysels, and Sohn (2013), we try if we could obtain a similar volatility estimator after 1970s.

To run the program, we could simply type *GarchMidas(y)* and accept all the default settings. However, there are some name-value pairs we may want to fine tune. 'Period' specifies aggregation periodicity. If we put 22, it is roughly a daily/monthly aggregation. 'NumLags' specifies the number of MIDAS lags. Here we put 24, meaning a history of 24 months' realized volatility will be averaged by the MIDAS weights to determine the long-run conditional variance. As we can see on the screen display, the adjusted sample size is 11120, while the dataset contains 11648 observations. The 24 lag months cost 528 observations for initialization. If you cannot afford a pre-sample of that size, you may consider setting the name-value pair 'AdjustLag'.

```
% NASDAQ Composite Index, daily percentage change 1971 - 2015
% Data Source: FRED Economic Data
```

```
% https://research.stlouisfed.org/fred2/series/NASDAQCOM
y = xlsread('NASDAQCOM.xls','B22:B11669') ./ 100;

% Estimate the GARCH-MIDAS model, and extract the volatilities
period = 22;
numLags = 24;
[estParams,EstParamCov,Variance,LongRunVar] = ...
GarchMidas(y,'Period',period,'NumLags',numLags);
```

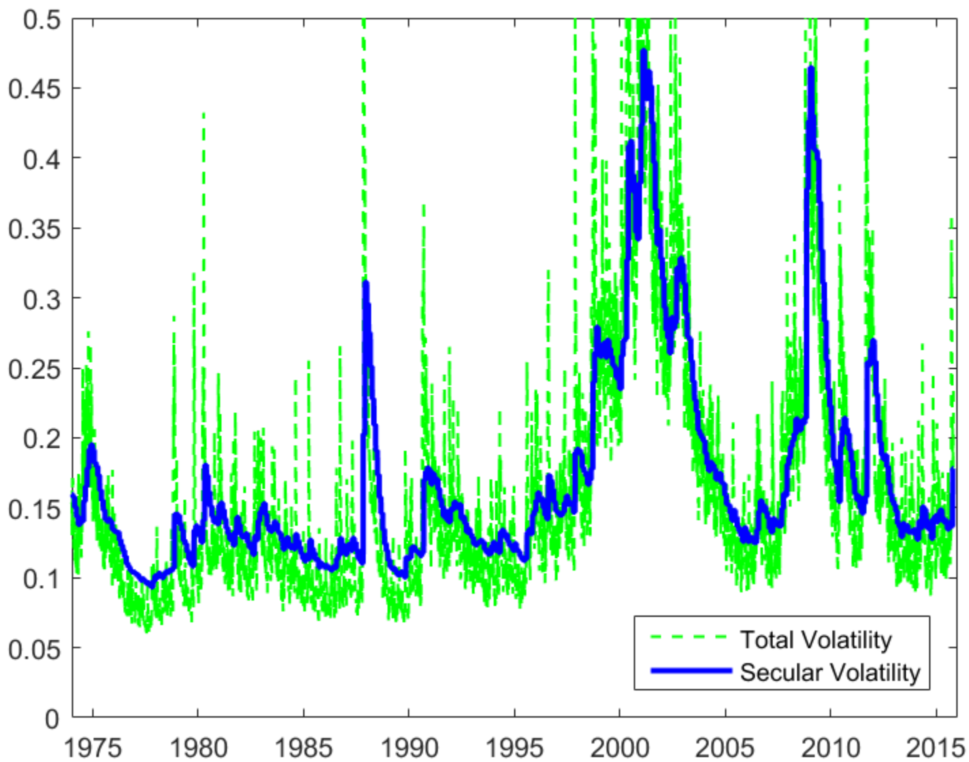
```
Method: Maximum likelihood
Sample size: 11648
Adjusted sample size: 11120
Logarithmic  likelihood:      36393.4
Akaike   info criterion:     -72774.9
Bayesian info criterion:     -72730.7
               Coeff         StdErr      tStat     Prob
             ----------    ----------    ------    ----

    mu       0.00080314    7.3864e-05    10.873    0
    alpha       0.12607     0.0043966    28.674    0
    beta        0.81026     0.0083922    96.549    0
    theta        0.1849     0.0050338    36.733    0
    w            5.8269       0.68289    8.5328    0
    m         0.0050642    0.00025503    19.858    0
```

Our estimated conditional volatility and its secular component in 1975 - 2010 have similar patterns as those reported in Figure 2 of Engle, Ghysels, and Sohn (2013). The long-run component exhibits spikes in years around 1975, 1989, 2002, 2008, etc. The total volatility jumps upwards during the recession periods. It confirms the empirical regularity of the counter-cyclical stock market volatility.

The rolling-window specification has a different weight scheme for the realized volatility. To check whether it will produce similar results or not, we may run the program with the name-value pair 'RollWindow'. The codes run a little slower due to more MIDAS weighed terms, but the results appear close to those under the fixed-window specification.

```
% Estimate the rolling window version of the GARCH-MIDAS model
```

Figure 1: Conditional volatility and its secular component



The figure illustrates a GARCH-MIDAS example using the NASDAQ Composite Index daily return data (1971 - 2015). The model is fitted by maximum likelihood with MIADS Beta weights of 24 months' of lags. The dashed line plots the conditional variance series and the solid line shows the long-run component series.

```
[estParams,EstParamCov,Variance,LongRunVar]...
    = GarchMidas(y,'Period',period,'NumLags',numLags,'RollWindow',1);
```

```
Method: Maximum likelihood
Sample size: 11648
Adjusted sample size: 11120
Logarithmic  likelihood:      36399.9
Akaike   info criterion:     -72787.7
Bayesian info criterion:     -72743.5
               Coeff          StdErr       tStat      Prob
             ----------     ----------     ------     ----

    mu       0.00080289    7.4187e-05    10.823     0
    alpha       0.13281     0.0048218    27.543     0
    beta         0.7822      0.010471    74.701     0
    theta       0.19053     0.0043393    43.908     0
    w            8.7578       0.96896    9.0383     0
    m         0.0045943    0.00022618    20.312     0
```

The realized volatility could be a noisy proxy for the macro-volatility. We may replace the realized volatility by some direct measure of economic activities. We downloaded the Industrial Production Index growth rate data (1971-2015) from the FRED database (INDPRO). The program requires the exogenous variable formatted as a vector with the same length as the observation series y. So we just repeat the monthly values throughout the days. Then we can run the program with the name-value pair 'X'.

```
% Industrial Production Index growth rate, 1971-2015
% Data Source: FRED database
% https://research.stlouisfed.org/fred2/series/INDPRO
xMonth = xlsread('INDPRO.xls','B42:B576') ./ 100;

% Repeat the monthly value throughout the days in that month
[¬,yDate] = xlsread('NASDAQCOM.xls','A22:A11669');
[¬,yDateMonth] = datevec(yDate);
xDay = NaN(nobs,1);
count = 1;
for t = 1:nobs
```

```
    if t > 1 && yDateMonth(t) ≠ yDateMonth(t-1)
        count = count + 1;
        if count > length(xMonth)
            break
        end
    end
    xDay(t) = xMonth(count);
end

% Estimate the GARCH-MIDAS model
[estParams,EstParamCov,Variance,LongRunVar] = ...
GarchMidas(y,'Period',period,'NumLags',32,'X',xDay,'ThetaM',1);
```

```
Method: Maximum likelihood
Sample size: 11648
Adjusted sample size: 10944
Logarithmic  likelihood:        35774
Akaike   info criterion:     -71536.1
Bayesian info criterion:     -71491.9
               Coeff          StdErr        tStat         Prob
             ----------     ----------     -------     ----------

    mu       0.00077109     7.2524e-05     10.632               0
    alpha        0.1022      0.0030644     33.352               0
    beta        0.88682      0.0033269     266.56               0
    theta     -0.014074      0.0029087    -4.8384      1.3089e-06
    w            2.4424        0.41713     5.8552               0
    m        0.00016871     2.2356e-05     7.5465               0
```

Lastly, we do some forecast exercise. We may run a subsample estimation and leave some observations for the one-step forecast validation by setting the name-value pairs 'EstSample'. For example, we use 8000 observations for parameter estimation and the remaining observations for forecast validation. The software reports on the screen the root mean squared errors (RMSE) of the one-step forecast on the conditional variance.

```
% In-sample forecast validation
GarchMidas(y,'Period',period,'NumLags',numLags,'estSample',8000);
```

```
Method: Maximum likelihood
Sample size: 8000
Adjusted sample size: 7472
Logarithmic  likelihood:      25254.1
Akaike   info criterion:     -50496.2
Bayesian info criterion:     -50454.2
               Coeff         StdErr      tStat     Prob
             ----------    ----------    ------    ----

    mu       0.00088309    8.2387e-05    10.719    0
    alpha       0.17105     0.0068514    24.966    0
    beta        0.71813      0.013063    54.975    0
    theta       0.19193     0.0050618    37.918    0
    w            8.4068       0.99459    8.4526    0
    m         0.0043922    0.00023089    19.023    0

RMSE of one-step variance forecast (period 1 to 8000): 4.558e-04.
RMSE of one-step variance forecast (period 8001 to 11648): 5.160e-04.
```

We may want to perform an out-of-sample volatility forecast. Equation (3.20) specifies the conditional variance recursion: $g_{it} = (1 - \alpha - \beta) + \alpha \frac{(r_{i-1,t}-\mu)^2}{\tau_t} + \beta g_{i-1,t}$. Note that g_{it} is a deterministic function of $r_{i-1,t}$ and historical observations. For out-of-sample forecast, $(r_{i-1,t} - \mu)^2$ is not available. We may replace such unavailable observations by the forecasted variance, similar to the way we iteratively forecast an autoregressive process. We may call *GarchMidas* recursively to forecast future variances.

```
% Out-of-sample forecast
estParams = GarchMidas(y,'Period',period,'NumLags',numLags);
nForecast = 5;
yBig = [y;0];
for t = 1:nForecast
    [¬,¬,Variance,LongRunVar] = ...
 GarchMidas(yBig,'Period',period,'NumLags',numLags,'Params',estParams);
    yPseudo = estParams(1) + sqrt(Variance(end));
    yBig = [yBig(1:end-1);yPseudo;0];
end
VarianceForecast = Variance(nobs+1:nobs+nForecast);
LongRunVarForecast = LongRunVar(nobs+1:nobs+nForecast);
```

Now we consider a DCC-MIDAS example. We try to use the *DccMidas* program to reproduce the results in Colacito, Engle, and Ghysels (2011). The tri-variate DCC-MIDAS model consists of Energy and Hi-Tech portfolios and a 10 year bond. Users are responsible for obtaining their original data. Alternatively, the program will load a different dataset containing the NASDAQ daily returns, JPY/USD exchange rates percentage change and 10-Year treasury rates percentage change, downloaded from FRED Economic Data (NASDAQCOM, DEXJPUS, DGS10, respectively).

To use the software, users may simply type *DccMidas(Data)*. Similar to *GarchMidas*, setting some of the name-value pairs may be helpful. 'Period' specifies aggregation periodicity. If we put 22, it is roughly a daily/monthly aggregation. 'NumLagsVar' specifies the number of MIDAS lags for the univariate GARCH-MIDAS for the first-step variance estimation. Here we put 36, meaning a history of 36 months realized volatility will be averaged by the MIDAS weights to determine the long-run conditional variance. 'NumLagsCorr' specifies the number of MIDAS lags for the second-step correlation matrix estimation. In this application, we put the lagged values of 144 months, but users may reduce the number of lags if the sample size is smaller.

To reproduce the results of the paper, we will overload some of the default name-value pairs of *DccMidas*, because their results were estimated by different codes. 'Options' is the FMINCON options for numerical optimization. We use the legacy 'active-set', though the default choice is 'interior-point'. Also, by setting 'ZeroLogL' to 1:3600, we forcefully suppress the contribution of the initial 3600 observations to the likelihood function, though the default initialization scheme does not have a burn-in of that amount. Also, we reset the MLE starting values 'mu0' to 0.001. Numerical optimization will not work well unless starting values are carefully chosen.

```
% Estimate the DCC-MIDAS model
options = optimoptions('fmincon','Algorithm','active-set');
[estParamsStep1,¬,estParamsStep2,¬,Variance,LongRunVar,CorrMatrix,LongRunCorrMatrix]...
    = DccMidas(Data,'Period',20,'NumLagsVar',36,'NumLagsCorr',144,...
'options',options,'ZeroLogL',1:3600,'mu0',0.001);
CorrMatrix = reshape(CorrMatrix,9,nobs)';
LongRunCorrMatrix = reshape(LongRunCorrMatrix,9,nobs)';
```

The program first estimates three univariate GARCH-MIDAS models for the conditional variances, and then constructs the standardized residuals and estimates the correlation matrix.

```
Method: Maximum likelihood
Sample size: 8827
Adjusted sample size: 8107
Logarithmic  likelihood:     -12291.1
Akaike   info criterion:      24594.3
Bayesian info criterion:      24636.8
              Coeff       StdErr      tStat     Prob
             --------    ---------    ------    ----

    mu        0.06978     0.011112    6.2798    0
    alpha    0.088307    0.0031895    27.687    0
    beta      0.79673     0.016579    48.055    0
    theta     0.19945    0.0049751    40.089    0
    w          13.356       1.7771    7.5158    0
    m         0.54037       0.0328    16.475    0
```

```
Method: Maximum likelihood
Sample size: 8827
Adjusted sample size: 8107
Logarithmic  likelihood:       -13535
Akaike   info criterion:      27082.1
Bayesian info criterion:      27124.6
              Coeff       StdErr      tStat        Prob
             --------    ---------    ------    ----------

    mu       0.062998     0.012912    4.8791    1.0659e-06
    alpha    0.086857    0.0030831    28.172             0
    beta      0.83684     0.014522    57.624             0
    theta     0.18674    0.0064655    28.882             0
    w          10.169       1.8541    5.4846             0
    m         0.72539     0.045025    16.111             0
```

```
Method: Maximum likelihood
Sample size: 8827
```

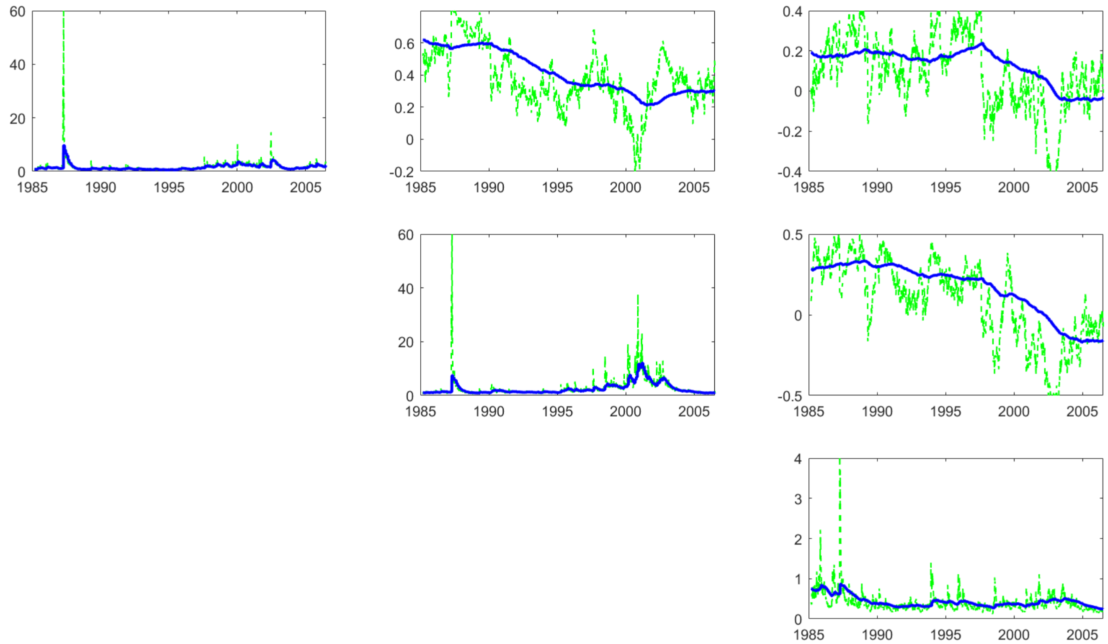
```
Adjusted sample size: 8107
Logarithmic  likelihood:     -7797.56
Akaike   info criterion:      15607.1
Bayesian info criterion:      15649.6
             Coeff       StdErr      tStat        Prob
            --------    ---------    ------    ----------

    mu       0.022186    0.0059381    3.7362    0.00018686
    alpha    0.059198    0.0033665    17.584             0
    beta      0.91586    0.0066424    137.88             0
    theta     0.20377    0.0065901     30.92             0
    w          2.9872      0.73878    4.0434    5.2681e-05
    m         0.29003     0.028859     10.05             0
```

```
Method: Two-Step Maximum likelihood
Sample size: 8827
Adjusted sample size: 5227
Logarithmic  likelihood:     -21598.1
Akaike   info criterion:      43202.1
Bayesian info criterion:      43223.4
         Coeff       StdErr      tStat       Prob
        --------    ---------    ------    ---------

    a    0.017591    0.0013486    13.044           0
    b     0.97741     0.001994    490.18           0
    w      1.8246      0.70672    2.5818   0.0098302
```

The program nearly reproduces the results; both the estimated parameters and the volatility estimation are close to Table 1 and Figure 1 of Colacito, Engle, and Ghysels (2011).

```
                mu         alpha        beta        theta        w          m
             --------    --------    -------     -------     ------     -------

    Energy      0.06978    0.088307    0.79673    0.19945    13.356    0.54037
    Hi-Tech    0.062998    0.086857    0.83684    0.18674    10.169    0.72539
    Bond       0.022186    0.059198    0.91586    0.20377    2.9872    0.29003
```

Figure 2: Long and short run volatilities and correlations



The figure illustrates a DCC-MIDAS example using the energy and hi-tech portfolios and the 10 year bond. The model is fitted by maximum likelihood with MIADS Beta weights of 36 (first step) and 144 (second step) months' of lags. In each diagonal panel the solid line refers to the long-run volatility and the dashed line represents the short-run volatility. In each off-diagonal panel the solid line is the long-run correlation and the dashed line is the total correlation.

```
                  a           b          w
               --------    -------    ------

    DCC-MIDAS  0.017591    0.97741    1.8246
```

If the users cannot obtain the original data used by the paper, the program will load an alternative dataset consisting of stock returns, exchange rate returns and bond yields percentage changes. Using the same codes, the estimation results are the following:

```
Method: Maximum likelihood
```

```
Sample size: 9236
Adjusted sample size: 8516
Logarithmic  likelihood:       -10909
Akaike   info criterion:      21829.9
Bayesian info criterion:      21872.7
              Coeff       StdErr      tStat     Prob
             --------    ---------    ------    ----

    mu       0.086574    0.0079921    10.832    0
    alpha     0.15278    0.0059163    25.823    0
    beta       0.7429      0.01208      61.5    0
    theta      0.2036    0.0047777    42.614    0
    w          7.8053      0.82288    9.4853    0
    m         0.39976     0.023648    16.905    0
```

```
Method: Maximum likelihood
Sample size: 9236
Adjusted sample size: 8516
Logarithmic  likelihood:     -7296.97
Akaike   info criterion:      14605.9
Bayesian info criterion:      14648.7
               Coeff       StdErr       tStat       Prob
             ---------    ---------    --------    -------

    mu       -0.001992    0.0056466    -0.35278    0.72426
    alpha      0.14489    0.0032909      44.028          0
    beta       0.68896     0.010587      65.076          0
    theta      0.23074    0.0025254      91.369          0
    w           12.267      0.62979      19.479          0
    m          0.16631    0.0094764       17.55          0
```

```
Method: Maximum likelihood
Sample size: 9236
Adjusted sample size: 8516
Logarithmic  likelihood:     -10295.3
Akaike   info criterion:      20602.7
Bayesian info criterion:      20645.4
               Coeff       StdErr       tStat      Prob
```

```
         ---------     ---------     -------     -------

    mu       0.0021314    0.0072295    0.29482    0.76813
    alpha     0.055403    0.0029636     18.695          0
    beta       0.91936    0.0056068     163.97          0
    theta      0.22148    0.0054202     40.862          0
    w           2.7536      0.53081     5.1875          0
    m          0.24627     0.036903     6.6735          0
```

```
Method: Two-Step Maximum likelihood
Sample size: 9236
Adjusted sample size: 5636
Logarithmic  likelihood:     -24124.3
Akaike   info criterion:      48254.7
Bayesian info criterion:      48276.1
           Coeff       StdErr     tStat        Prob
         ---------    --------    ------    ----------

    a    0.0077311    0.002346    3.2954    0.00098303
    b      0.97311      0.0231    42.126             0
    w       14.176      2.6517     5.346             0
```

5.3 MIDAS quantile regressions

MidasQuantile is a MATLAB function for the MIDAS quantile regression. The syntax is

```
[...] = MidasQuantile(y, name,value,...)
```

The required input argument is y, a $T \times 1$ vector of observations. The optional name-value pairs include:

- 'Quantile': A scalar between zero and one that specifies the level (i.e., α) of quantile. The default is 0.05.
- 'X': $T \times 1$ high frequency conditioning variable (predictor). This variable must have the same length as y, and only one predictor is supported. By default, it is $|y|$, as

"absolute returns successfully capture time variation in the conditional distribution of returns".

- 'Period':A scalar integer that specifies the aggregation periodicity. y will be aggregated so as to formulate n-period returns. How many days in a week/month/quarter/year? The default is 22 (as in a day-month aggregation)

- 'NumLags': A scalar integer that specifies the number of lags for the high frequency predictor, to which MIDAS weights is assigned. The default is 250.

- 'Dates': $T \times 1$ vector or cell array for the dates of y. This is for book-keeping purpose and does not affect estimation. The default is $1 : length(y)$.

- 'Smoother': A non-negative scalar that specifies how to smooth the non-differentiable objective function. If it is zero, there is no smoothing. The default is average absolute residuals. This is the starting smoother. The software will run a series of optimizations; each time the smoother will be reduced by an half.

- 'Search': A logical value that indicates numerical minimization via pattern search in the Global Optimization Toolbox. If not available, it resorts to fminsearch in base MATLAB. The default is false (and will use gradient-based methods under smoothed objective functions).

- 'Options': The options for numerical optimization. The default is the FMINCON default choice.

- 'Gradient': A logical value that indicates analytic gradients in MLE. The default is false.

- 'Bootstrap': A character vector that specifies bootstrap standard error method: 'Residual' (Default) or 'XY'.

- 'Params': Parameter values for [intercept;slope;k].In that case, the program skips estimation, and just infers conditional quantiles based on the specified parameters. The default is empty (need parameter estimation).

- 'Params0': Starting parameter values of [intercept;slope;k] for numeric optimization. This will overload the software default choice, which starts from the OLS estimator.

The output arguments include:

- estParams: Estimated parameters for [intercept;slope;k], where intercept and slope are the coefficients of the quantile regression, and k is the parameter in the MIDAS Beta polynomial.
- condQuantile: $R \times 1$ conditional quantile. This is the fitted value of the right-hand-side of the quantile regression. $R = T - Period - NumLags + 1$.
- yLowFreq: $R \times 1$ n-period returns obtained from overlapping aggregation of y. This is the left-hand-side of the quantile regression.
- xHighFreq: $R \times NumLags$ high-frequency predictors used by the quantile regression.
- yDates: $R \times 1$ serial dates for the output variables.

We consider a quantile regression example.

First, we load some daily returns data, which contains 4435 observations.

```
% load the price data and form returns
P = xlsread('DataQuantile.xlsx','Sheet1','B2:B4437');
[¬,dates] = xlsread('DataQuantile.xlsx','Sheet1','A2:A4437');
dates = datenum(dates);
y = log(P(2:end)./P(1:end-1));
dates = dates(2:end);
```

Second, we estimate the 0.25 conditional quantile for the 3-month returns. We set the name-value pairs 'Period' and 'Quantile'.

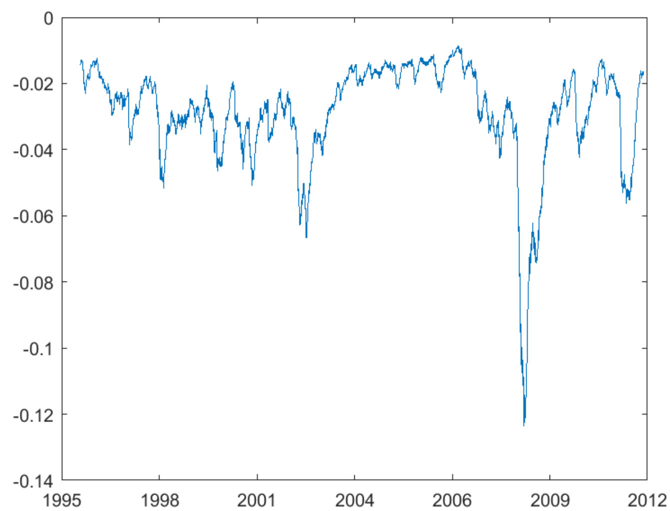
```
MidasQuantile(y,'Dates',dates,'Period',66,'Quantile',0.25);
```

Finally, we plot the conditional quantiles, which are returned as the second output argument.

6 Other MIDAS Applications

Ghysels (2016) introduces a relatively simple mixed sampling frequency VAR model. By

Figure 3: Midas Quantile regression



The figure illustrates a MIDAS quantile regression example using daily-quarterly aggregation. The solid line plots the 0.25 conditional quantile of the returns.

simple we mean, (1) a specification that does not involve latent shocks, (2) a specification that allows us to measure the impact of high frequency data onto low frequency ones and vice versa, (3) as far as VAR models go parsimonious, (4) a specification that can be estimated and analyzed with standard VAR analysis tools - such as impulse response analysis, and can be estimated with standard VAR estimation procedure (5) one that can track the proper timing of low and high frequency data - that may include releases of quarterly data in the middle of the next quarter along with the releases of monthly data or daily data.

The mixed frequency VAR provides an alternative to commonly used state space models involving mixed frequency data.[5] State space models involve latent processes, and therefore rely on filtering to extract hidden states that are used in order to predict future outcomes. State space models are, using the terminology of Cox (1981), parameter-driven models. The mixed frequency VAR models are, using again the same terminology, observation-driven

[5]See for example, Harvey and Pierse (1984), Bernanke, Gertler, and Watson (1997), Zadrozny (1990), Mariano and Murasawa (2003), Mittnik and Zadrozny (2004), and more recently Aruoba, Diebold, and Scotti (2009), Ghysels and Wright (2009), Kuzin, Marcellino, and Schumacher (2011a), Marcellino and Schumacher (2010), among others.

models as they are formulated exclusively in terms of observable data. The fact we rely only on observable shocks has implications with respect to impulse response functions. Namely, we formulate impulse response functions in terms of observable data - high and low frequency - instead of shocks to some latent processes. Finally, mixed frequency VAR models, like MIDAS regressions, may be relatively frugal in terms of parameterization.

Technically speaking Ghysels (2016) adapts techniques typically used to study seasonal time series with hidden periodic structures, to multiple time series that have different sampling frequencies. The techniques we adapt relate to work by Gladyshev (1961), Pagano (1978), Tiao and Grupe (1980), Hansen and Sargent (1990, Chap. 17), Hansen and Sargent (1993), Ghysels (1994), Franses (1996), among others. In addition, the mixed frequency VAR model is a multivariate extension of MIDAS regressions proposed in recent work by Ghysels, Santa-Clara, and Valkanov (2006), Ghysels and Wright (2009), Andreou, Ghysels, and Kourtellos (2010) and Chen and Ghysels (2011), among others.

Ghysels (2016) also characterizes the mapping between the mixed frequency VAR model and (1) a traditional VAR model where all the data are sampled at a common low frequency as well as (2) a hidden state high frequency VAR commonly used in a state space model setting. This mapping allows us to study the mis-specification of impulse response functions of traditional VAR models. The VAR models we propose can also handle time-varying mixed frequencies. Not all months have the same number of trading days, not all quarters have the same number of weeks, etc. Assuming a deterministic calendar effect, which makes all variation in changing mixed frequencies perfectly predictable, we are able to write a VAR with time-varying high frequency data structures.

Ghysels (2016) studies two classes of estimation procedures, classical and Bayesian, for mixed frequency VAR models. For the former Ghysels (2016) characterizes how the mis-specification of traditional VAR models translates into pseudo-true VAR parameter and impulse response estimates. Parameter proliferation is an issue in both mixed frequency and traditional VAR models. A Bayesian approach which easily accommodates the potentially large set of parameters to be estimated is therefore also considered. While this software is not yet featured in the Tooblox, the programs are available upon request.

Econometric analysis of mixed frequency models is not limited to linear regressions. The purpose of this concluding section is to briefly discuss other applications of MIDAS beyond linear regression analysis.

Chen and Ghysels (2011) introduce semi-parametric estimation of MIDAS regression models. They consider a regression model with a MIDAS polynomial (the parametric part) involving functional transformations of high frequency data, where the function is estimated via kernel methods.

The initial work on MIDAS and volatility involved a likelihood based approach on the risk return tradeoff. In particular the monthly variance is specified in Ghysels, Santa-Clara, and Valkanov (2005) as a weighted average of lagged daily squared returns and estimated via a QMLE similar a GARCH-in-mean approach. Hence, they estimate the coefficients of the conditional variance process jointly with the expected return equation. Hence, this approach is very different from the MIDAS regressions discussed in the previous sections. The similarity, however, is that in both MIDAS regressions and in the likelihood based MIDAS specification use the same type of parsimoniously specified lag polynomials.

The volatility specification in Ghysels, Santa-Clara, and Valkanov (2005) involves a single polynomial applied to daily data. Similar to the specification of the MIDAS regression one could think of introducing lagged volatilities. This approach would be similar to the specification of a GARCH model. This insight has recently been pursued by Chen, Ghysels, and Wang (2014). A key ingredient of conditional volatility models is that more weight is attached to the most recent returns (i.e. information). The model, however, involves returns sampled at different frequencies. For example, daily volatility can be predicted using intra-daily data. However, unlike realized volatility measures, intra-daily returns get different weights. Indeed, if volatility is a persistent process, it would be natural to weight intra-daily data differently. This is one example of the class of models Chen, Ghysels, and Wang (2014) called HYBRID GARCH models. They provide a unifying framework, based on a generic GARCH-type model, that addresses the issue of volatility forecasting involving forecast horizons of a different frequency than the information set. Hence, they propose a class of GARCH models that can handle volatility forecasts over the next five business days and use past daily data, or tomorrows expected volatility while using intra- daily returns. The models are called HYBRID GARCH, which stands for High FrequencY Data Based PRojectIon Driven GARCH models.

References

Alper, C.E., S. Fendoglu, and B. Saltoglu, 2008, Forecasting Stock Market Volatilities Using MIDAS Regressions: An Application to the Emerging Markets, Discussion paper, MPRA Paper No. 7460.

Andreou, Elena, Eric Ghysels, and Andros Kourtellos, 2010, Regression models with mixed sampling frequencies, *Journal of Econometrics* 158, 246–261.

———, 2011, Forecasting with mixed-frequency data, in Clements. Michael P., and David F. Hendry, ed.: *Oxford Handbook of Economic Forecasting* pp. 225–245. Oxford UP.

Andreou, E., E. Ghysels, and A. Kourtellos, 2013a, Should macroeconomic forecasters look at daily financial data?, *Journal of Business and Economic Statistics* 31, 240–251.

Andreou, Elena, Eric Ghysels, and Andros Kourtellos, 2013b, Should macroeconomic forecasters use daily financial data and how?, *Journal of Business and Economic Statistics* 31, 240–251.

Armesto, Michelle T., Kristie M. Engemann, and Michael T. Owyang, 2010, Forecasting with mixed frequencies, *Federal Reserve Bank of Saint Louis Review* 92, 521–536.

Armesto, Michelle T., Rubén Hernandez-Murillo, Michael Owyang, and Jeremy Piger, 2009, Measuring the information content of the beige book: A mixed data sampling approach, *Journal of Money, Credit and Banking* 41, 35–55.

Aruoba, S. Boragan, Frank X. Diebold, and Chiara Scotti, 2009, Real-time measurement of business conditions, *Journal of Business and Economic Statistics* 27, 417–427.

Baele, Lieven, Geert Bekaert, and Koen Inghelbrecht, 2010, The determinants of stock and bond return comovements, *Review of Financial Studies* 23, 2374–2428.

Bai, Jushan, 2003, Inferential theory for factor models of large dimensions, *Econometrica* pp. 135–171.

Bai, J., E. Ghysels, and J. Wright, 2013, State space models and midas regressions, *Econometric Reviews* 32, 779–813.

Bai, Jushan, and Serena Ng, 2002, Determining the number of factors in approximate factor models, *Econometrica* 70, 191–221.

Bernanke, Ben, Mark Gertler, and Mark Watson, 1997, Systematic monetary policy and the effects of oil price shocks, *Brookings Papers on Economic Activity* 1, 91–157.

Bhansali, Rajendra J., 1999, Parameter estimation and model selection for multistep prediction of a time series: a review, in S. Ghosh, ed.: *Asymptotics, Nonparametrics and Time Series* pp. 201–225. Marcel Dekker.

Brown, David P., and Miguel A. Ferreira Ferreira, 2003, The Information in the Idiosyncratic Volatility of Small Firms, Working paper, Univesrity of Wisconsin and ISCTE.

Chen, X., and E. Ghysels, 2011, News-Good or Bad-and its Impact on Volatility Forecasts over Multiple Horizons, *Review of Financial Studies* 24, 46–81.

Chen, Xilong, Eric Ghysels, and Fangfang Wang, 2011, On the role of Intra-Daily Seasonality in HYBRID GARCH Models, *Journal of Time Series Econometrics* 3.

——— , 2014, The HYBRID GARCH class of models, *Statistica Sinica*, forthcoming.

Chevillon, Guillaume, and David F. Hendry, 2005, Non-parametric direct multi-step estimation for forecasting economic processes, *International Journal of Forecasting* 21, 201–218.

Clements, Michael P., and Ana B. Galvão, 2008, Macroeconomic Forecasting with Mixed Frequency Data: Forecasting US output growth, *Journal of Business and Economic Statistics* 26, 546–554.

——— , 2009, Forecasting US output growth using Leading Indicators: An appraisal using MIDAS models, *Journal of Applied Econometrics* 24, 1187–1206.

——— , and Jae H. Kim, 2008, Quantile forecasts of daily exchange rate returns from forecasts of realized volatility, *Journal of Empirical Finance* 15, 729–750.

Clements, Michael P, and David F Hendry, 1996, Multi-step estimation for forecasting, *Oxford Bulletin of Economics and Statistics* 58, 657–684.

Colacito, Ricardo, Robert F. Engle, and Eric Ghysels, 2011, A component model for dynamic correlations, *Journal of Econometrics* 164, 45–59.

Corsi, F., 2009, A simple approximate long-memory model of realized volatility, *Journal of Financial Econometrics* 7, 174–196.

Cox, D.R., 1981, Statistical analysis of time series: Some recent developments [with discussion and reply], *Scandinavian Journal of Statistics* 8, 93–115.

Engle, Robert F, Eric Ghysels, and Bumjean Sohn, 2013, Stock market volatility and macroeconomic fundamentals, *Review of Economics and Statistics* 95, 776–797.

Engle, Robert F., and J. Gonzalo Rangel, 2008, The Spline-GARCH Model for Low-Frequency Volatility and Its Global Macroeconomic Causes, *Review of Financial Studies* 21, 1187–1222.

Findley, David F., 1983, On the use of multiple models for multi-period forecasting, in *Proceedings of Business and Economic Statistics, American Statistical Association* pp. 528–531.

———, 1985, Model selection for multi-step-ahead forecasting, in H. A. (Baker, and P. C. Young, ed.: *Proceedings of the 7th Symposium on Identification and System Parameter* pp. 1039–1044.

Forni, Mario, Marc Hallin, Marco Lippi, and Lucrezia Reichlin, 2000, The generalized dynamic-factor model: Identification and estimation, *Review of Economics and statistics* 82, 540–554.

———, 2005, The generalized dynamic factor model, *Journal of the American Statistical Association* 100, 830–840.

Foroni, Claudia, Massimiliano Marcellino, and Christian Schumacher, 2015, Unrestricted mixed data sampling (midas): Midas regressions with unrestricted lag polynomials, *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 178, 57–82.

Forsberg, Lars, and Eric Ghysels, 2006, Why do absolute returns predict volatility so well?, *Journal of Financial Econometrics* 6, 31–67.

Frale, Cecilia, and Libero Monteforte, 2011, FaMIDAS: A Mixed Frequency Factor Model with MIDAS structure, Bank of Italy Temi di Discussione -Working Paper No. 788.

Franses, Philip H., 1996, *Periodicity and stochastic trends in economic time series* (Oxford University Press, USA).

Ghysels, Eric, 1994, On the periodic structure of the business cycle, *Journal of Business and Economic Statistics* pp. 289–298.

——— , 2016, Macroeconomics and the reality of mixed frequency data, *Journal of Econometrics*, forthcoming.

——— , Casidhe Horan, and Emanuel Moench, 2014, Forecasting through the rear-view mirror: Data revisions and bond return predictability, FRB of New York Staff Report 581.

Ghysels, Eric, Alberto Plazzi, and Rossen Valkanov, 2016, Why invest in emerging markets? the role of conditional return asymmetry, *Journal of Finance*, (forthcoming).

Ghysels, E., and H. Qiang, 2016, Estimating midas regressions via ols with polynomial parameter profiling, Discussion paper, MathWorks and UNC.

Ghysels, E., A. Rubia, and R. Valkanov, 2008, Multi-Period Forecasts of Variance: Direct, Iterated, and Mixed-Data Approaches, Working paper, Alicante, UCSD and UNC.

Ghysels, Eric, Pedro Santa-Clara, and Rossen Valkanov, 2002, The MIDAS touch: Mixed data sampling regression models, Working paper, UNC and UCLA.

Ghysels, Eric, Pedro Santa-Clara, and Ross Valkanov, 2004, The MIDAS touch: Mixed Data Sampling Regressions, Discussion paper UNC and UCLA.

——— , 2005, There is a risk-return tradeoff after all, *Journal of Financial Economics* 76, 509–548.

Ghysels, E., P. Santa-Clara, and R. Valkanov, 2006, Predicting volatility: getting the most out of return data sampled at different frequencies, *Journal of Econometrics* 131, 59–95.

Ghysels, E., and A. Sinko, 2006, Comment on realized variance and market microstructure noise by p. r. hansen and asger lunde, *Journal of Business and Economic Statistics* 24, 192–194.

Ghysels, Eric, and Arthur Sinko, 2011, Volatility prediction and microstructure noise, *Journal of Econometrics* 160, 257–271.

——— , and Ross Valkanov, 2006, MIDAS regressions: Further results and new directions, *Econometric Reviews* 26, 53–90.

Ghysels, E., and R. Valkanov, 2012, Forecasting volatility with MIDAS, in Luc Bauwens, Christian Hafner, and Sébastien Laurent, ed.: *Handbook of Volatility Models and Their Applications* pp. 383–401. Wiley.

Ghysels, Eric, and Jonathan Wright, 2009, Forecasting professional forecasters, *Journal of Business and Economic Statistics* 27, 504–516.

Gladyshev, E. G., 1961, Periodically correlated random sequences, *Soviet Mathematics* 2, 385–388.

Hamilton, James D., 2008, Daily monetary policy shocks and new home sales, *Journal of Monetary Economics* 55, 1171–1190.

Hansen, L.P., and T.J. Sargent, 1990, *Recursive linear models of dynamic economies* (NYU and University of Chicago).

———, 1993, Seasonality and approximation errors in rational expectations models* 1, *Journal of Econometrics* 55, 21–55.

Harvey, Andrew C., and Richard G. Pierse, 1984, Estimating missing observations in economic time series, *Journal of the American Statistical Association* 79, 125–131.

Judge, George G., R. Carter Hill, William Griffiths, Helmut Lütkepohl, and Tsoung-Chao Lee, 1988, *Introduction to the Theory and Practice of Econometrics* (John Wiley & Sons, New York).

Kuzin, Vladimir, Massimiliano Marcellino, and Christian Schumacher, 2011a, MIDAS versus mixed-frequency VAR: nowcasting GDP in the euro area, *International Journal of Forecasting* 27, 529–542.

Kuzin, Vladimir, Massimiliano G. Marcellino, and Christian Schumacher, 2011b, MIDAS versus mixed-frequency VAR: Nowcasting GDP in the Euro Area, *International Journal of Forecasting* 27, 529–542.

Kvedaras, Virmantas, and Alfredas Račkauskas, 2010, Regression models with variables of different frequencies: The case of a fixed frequency ratio*, *Oxford Bulletin of Economics and Statistics* 72, 600–620.

Kvedaras, Virmantas, and Vaidotas Zemlys, 2012, Testing the functional constraints on parameters in regressions with variables of different frequency, *Economics Letters* 116, 250–254.

León, Ángel., Juan M. Nave, and Gonzalo Rubio, 2007, The relationship between risk and expected return in Europe, *Journal of Banking and Finance* 31, 495–512.

Lin, Jin-Lung, and Clive W.J. Granger, 1994, Forecasting from non-linear models in practice, *Journal of Forecasting* 13, 1–9.

Lütkepohl, Helmut, 2012, *Forecasting aggregated vector ARMA processes* (Springer Science & Business Media).

Marcellino, Massimiliano G., and Christian Schumacher, 2010, Factor MIDAS for Nowcasting and Forecasting with Ragged-Edge Data: A Model Comparison for German GDP, *Oxford Bulletin of Economics and Statistics* 72, 518–550.

Marcellino, Massimiliano G., James H. Stock, and Mark W. Watson, 2006, A comparison of direct and iterated multistep ar methods for forecasting macroeconomic time series, *Journal of Econometrics* 135, 499–526.

Mariano, Roberto S., and Yasutomo Murasawa, 2003, A new coincident index of business cycles based on monthly and quarterly series, *Journal of Applied Econometrics* 18, 427–443.

Mittnik, Stefan, and Peter A. Zadrozny, 2004, *Forecasting quarterly German GDP at monthly intervals using monthly Ifo business conditions data* (Springer).

Monteforte, Libero, and Gianluca Moretti, 2013, Real-time forecasts of inflation: The role of financial variables, *Journal of Forecasting* 32, 51–61.

Pagano, M., 1978, On periodic and multiple autoregressions, *The Annals of Statistics* pp. 1310–1317.

Rodriguez, Abel, and Gavino Puggioni, 2010, Mixed frequency models: Bayesian approaches to estimation and prediction, *International Journal of Forecasting* 26, 293–311.

Schumacher, Christian, and Jörg Breitung, 2008, Real-time Forecasting of German GDP Based on a Large Factor Model with Monthly and Quarterly Data, *International Journal of Forecasting* 24, 386–398.

Stock, James H., and Mark W. Watson, 1989, New indexes of coincident and leading economic indicators, *NBER Macroeconomics Annual* pp. 351–394.

——— , 2002, Macroeconomic forecasting using diffusion indexes, *Journal of Business and Economic Statistics* 20, 147–162.

——— , 2003, Forecasting output and inflation: the role of asset prices, *Journal of Economic Literature* pp. 788–829.

——— , 2004, Combination forecasts of output growth in a seven-country data set, *Journal of Forecasting* 23, 405–430.

——— , 2008a, Forecasting in Dynamic Factor Models Subject to Structural Instability, in Jennifer Castle, and Neil Shephard, ed.: *The Methodology and Practice of Econometrics, A Festschrift in Honour of Professor David F. Hendry.* Oxford University Press.

——— , 2008b, Phillips curve inflation forecasts, NBER Working paper.

Tiao, George C., and M. R. Grupe, 1980, Hidden periodic autoregressiye-moving average models in time series data, *Biometrika* 67, 365.

Timmermann, Allan, 2006, Forecast combinations, in G. Elliott, C. Granger, and A. Timmermann, ed.: *Handbook of Economic Forecasting, Volume 1* . pp. 136–196 (Elsevier).

Wohlrabe, Klaus, 2009, Forecasting with mixed-frequency time series models, Ph.D. thesis Ludwig-Maximilians-Universität, München.

Zadrozny, Peter A., 1990, Forecasting US GNP at monthly intervals with an estimated bivariate time series model, *Federal Reserve Bank of Atlanta Economic Review* 75, 2–15.