Aerosol and Air Quality Research, 15: 1270–1280, 2015
Copyright © Taiwan Association for Aerosol Research
ISSN: 1680-8584 print / 2071-1409 online
doi: 10.4209/aaqr.2015.02.0103

Adaptive Decomposition of Highly Resolved Time Series into Local and Non-Local Components

Ram Vedantham[1], Gayle S.W. Hagler[2*], Kathleen Holm[1], Sue Kimbrough[2], Richard Snow[2]

[1] United States Environmental Protection Agency, Office of Research and Development, National Exposure Research Laboratory, Research Triangle Park, North Carolina, 27711, USA
[2] United States Environmental Protection Agency, Office of Research and Development, National Risk Management Research Laboratory, Research Triangle Park, North Carolina, 27711, USA

ABSTRACT

Highly time-resolved air monitoring data are widely being collected over long time horizons in order to characterize ambient and near-source air quality trends. In many applications, it is desirable to split the time-resolved data into two or more components (e.g., local and regional) for apportionment and mitigation purposes. While there may be increased information content in highly time-resolved data, the temporal resolution may also increase entropic effects on the data, thereby dramatically clouding the very information sought in time-resolved data. Specialized methods such as filtering may be required to extract the underlying information content. Constrained and Adaptive Decomposition of Time Series (CADETS) is a new method that can help carve out components of time series based on the content of the frequencies present in the time series. CADETS is also a flexible approach that allows the user to choose the bifurcation point with minimal negative impacts. Using this algorithm, we demonstrate that a time series signal may be decomposed into two useful and interpretable signals that can help identify aspects that may otherwise be hidden or distorted. Using the output from the CADETS algorithm, we show that ultrafine particles (30–100 nm) collected near a major highway may be split into a 64:36 ratio of highly varying (local) and slowly varying (regional) components, meanwhile identical measurements at a background location were estimated to split into a 56:44 local versus regional ratio.

Keywords: Air pollution; Time series; Ultrafine particles; Signal decomposition.

INTRODUCTION

Air monitoring technology advancement has supported increased temporal resolution of data – seconds to minutes – for many pollutants of interest, allowing a clear understanding of time-varying and spatial trends. Long-term application of higher time resolution monitoring is particularly of interest in near-source areas, where changes in local wind speed and direction or emissions from a nearby source (such as traffic) can significantly affect downwind dispersion of emissions (Zhu *et al.*, 2004). For many ambient and near-source studies, monitoring and data analysis strategies are conducted to identify causes of variation, such as apportionment to specific sources via chemical fingerprint methods (Schauer and Cass, 1996), estimation of local versus regional influences using upwind-downwind monitoring and analysis (Hagler *et al.*, 2006), and source location estimation via local scale back-trajectory analysis (Henry *et al.*, 2011). As air monitoring data sets improve in temporal resolution, signal processing becomes an additional promising strategy for trend analysis. In particular, this strategy may help maximize the information obtained in air monitoring studies that are limited in the number of pollutants or locations measured. This approach may allow for a degree of source separation without requiring extensive chemical speciation (e.g., particulate trace elements or organics) and may also be useful in environments with similar pollutant mixtures of local and distant sources.

The idea of decomposition of time series using signal processing is neither novel nor unique. Temporal signals have been decomposed using a variety of techniques in almost all subject areas. Fourier and wavelet analyses have been employed to achieve the desired decomposition. For the most part, the efforts have been to remove the noise aspect of the signal (Ganguli, 2002; Geerken *et al.*, 2005; Lu *et al.*, 2007). Applications of these methods can be found in many fields including signal processing, and epidemiology (Bauch and Earn, 2003). Also, a straightforward decomposition of a signal (measured air pollutant concentration, in this case) using a filter typically results in having a large number of negative values in at least one component. This is due to

[*] Corresponding author.
Tel.: 919-541-2827
E-mail address: hagler.gayle@epa.gov

lack of constraints associated with filters designed to capture the desired aspect (e.g., slowly varying) of the signal. Non-negativity in the decomposed signals would be an example of a constraint. In mathematical terms, the following two requirements summarize the stipulations of idealized signal decomposition.

- The input time series is split into two independent time series. That is $C(t) = C_1(t) + C_2(t)$, where $C(t)$ is the original time series, and $C_1(t)$ and $C_2(t)$ are the decomposed time series signals.
- The ranges of frequencies in the components are complimentary subsets of the frequencies associated with the original time series. That is $\Omega = \Omega_1 \cup \Omega_2$ and $\Omega_1 \cap \Omega_2 \cong \emptyset$, where Ω is the set of frequencies of the original time series, $\Omega_i(\mathrm{t})$, i = 1, 2 are the *contiguous* sets of frequencies associated with the decomposed time series $C_1(t)$ and $C_2(t)$ and Ø denotes the empty set. (Note: Even under ideal conditions, decomposition using filters produces sets of frequencies that are only almost-disjoint ($\Omega_1 \cap \Omega_2 \cong \emptyset$). Completely disjoint sets ($\Omega_1 \cap \Omega_2 = \emptyset$) are possible only in theory. That is, the so-called spectral leakage is impossible to avoid in practical applications.)

After decomposition into components, interpretation is based on expert judgment of the pollutant and measurement environment. For example, component descriptors for traffic-related emissions may include: Slow and Fast Varying, Primary vs. Secondary, Local and Regional. Using decomposition, the components may be analyzed more effectively. For instance, it might be helpful to estimate the local contribution of a certain pollutant to the local air shed for possible mitigation steps. Combining decomposed components with ancillary data (meteorology, source activity) might support source contribution and origin identification.

This study presents a new algorithm that decomposes a given time series signal into two temporal components that satisfies the above mentioned specifications. Called Constrained and Adaptive Decomposition of Time Series (CADETS), the algorithm allows the user the flexibility to fine tune the results, particularly:

- Selection of the separation point to divide the set of frequencies into two. In practical use, this selected bifurcation point relies upon knowledge of the data measurement rate, duration, and analysis goals. For example, ambient air measurements collected at high-time resolution (e.g., 5 minute, 15 minute) over a long time horizon may opt for a sub-24 hour cycling period to separate slow- and fast-varying components of the total signal.
- Selection of the negative values tolerance level that is computed as a percent of total number of observations in the decomposed time series signal. For instance, the user may choose to terminate the algorithm when the number of negative values in the component time series is less than 2% of the total number of observations.

The CADETS algorithm is demonstrated on multiple extended time series of data collection involving particle number concentration measured in varying environments, including roadside and background locations. This data set is unique in allocating particle counts to multiple size-based bins, which allows hypotheses to be explored, such as an increased slow varying component (i.e., regional or secondary) attribution for larger versus small particles, and background versus roadside environments.

METHODS

Particle Count Field Data

Size-resolved particle counts were measured over a several year time horizon at three sequential locations – a near-road environment in Las Vegas, Nevada, a near-road location in Detroit, MI, and a suburban background location in Durham, NC (Table 1, Fig. S1 in Supplemental Information). Ultrafine particle (UFP) and accumulation-mode particle count measurements were conducted using an Ultrafine Particle Monitor (Model 3031, TSI, Inc., Shoreview, Minnesota, USA), which sizes particles using a differential mobility analyzer and detects via an electrometer. The UFP 3031 performs a single scan during the measurement time period (~11 minutes) and output counts associated with six size ranges (20–30, 30–50, 50–70, 70–100, 100–200, > 200 nm) every 15 minutes.

Notable advantages of the UFP 3031 instrument utilized in this study are the size-segregated counts and the lack of consumable operating fluid required to maintain the monitoring instrument. However, a weakness of the instrument is having less sensitivity than other methodologies – such as condensational particle counters – at low number count concentrations. Particle number count instruments can be described as having lower particle size detection limits as well as having lower number count detection limits. The UFP 3031 has a lower particle size detection of 20 nm and a lower count detection that varies per size bin, due to the amount of charge required for detection by electrometer. Based upon manufacturer correspondence, estimated lower detection limits of UFP3031 particle number concentrations were 408 cm^{-3}, 258 cm^{-3}, 169 cm^{-3}, 120 cm^{-3}, 71 cm^{-3}, and 50 cm^{-3} for bin ranges 20–30, 30–50, 50–70, 70–100, 100–200, and >200 nm, respectively. Each data set was evaluated for detection limit; where the instrument reported values below the detection limit, a value of ½ the detection limit was substituted for uniformity and to prevent positive bias introduced by alternatively treating these time periods as missing values. Based upon detection limit evaluation, the CADETS analysis was constrained to the size range of 30–200 nm (central four size bins), given the higher fraction of data (e.g., >20%) below the detection limit for the other size bins. Based on previous research, the smallest particle size bin (30–50 nm) is hypothesized to be most affected by nearby traffic emissions at the roadside sites, whereas the larger particles measured (50–200 nm) are expected to have progressively more contribution from secondary processes (Zhang *et al.*, 2004). The CADETS algorithm is demonstrated on the particle data on an individual size bin basis, as well as summed over the four bins. The lower size limit of 30 nm should be kept in mind when observing the total particle number results – were particles < 30 nm included in the summation, studies such

Table 1. Field measurement collection locations and times.

Site name	Location	Latitude, Longitude	Sampling Duration
LV-ROAD	10 meters from I-15 in Las Vegas, NV USA	36° 4'35.25"N, 115°10'48.75"W	June, 2009–May, 2010
LV-300m	300 meters from I-15 in Las Vegas, NV USA	36° 4'32.72"N, 115°10'37.82"W	June, 2009–May, 2010
DT-ROAD	10 meters from I-96 in Detroit, MI USA	42°23'12.60"N, 83°16'13.74"W	August, 2010–June, 2011
RTP-BGD	Background location in Research Triangle Park, NC USA	35°52'50.11"N, 78°52'9.91"W	March, 2012–January, 2013

as Zhang *et al.* (2004) would predict greater degree of local source influence on the roadside total particle count.

Constrained and Adaptive Decomposition of Time Series (CADETS) Algorithm

The overarching goal of the CADETS algorithm is to create a lower envelope of the original time series data. This envelope will reflect the slowly varying contribution to the air shed at the receptor site. Instead of creating an arbitrary envelope, the method uses a derived low frequency component (LFC) to adaptively create an envelope of the original data, one piece-wise sample of the LFC at a time. The CADETS algorithm has multiple steps that are listed below, which are demonstrated using the UFP data sets in the following section. All the results presented here were generated using MATLAB version 2009b (Mathworks Inc., Natick, MA) and apply some built-in functions in the MATLAB software. Steps 1–2 can be applied by the user using basic functions in MATLAB; the code associated with step 3 through step 7 is provided in supplemental information.

1. Use the Fourier functionality on the input data to plot the Power Spectral Density (PSD) using a periodogram. The PSD estimates the energy in the component frequencies. Instead of the usual frequency vs. energy plot, it may be preferable to plot period (inverse of frequency) vs. energy. This will likely provide the significant periodicities present in the data (e.g., diurnal, daily, weekend/weekday etc.).
2. Choose an appropriate frequency value to be used as the cut-off point. The best choice for the cut-off is typically the frequency associated with the first significant peak (fundamental frequency) in the PSD. The peak is a local maximum (may even be a global maximum) implying a significant presence of the associated frequency in the data signal. We refer to this frequency as the cut-off frequency.
3. Apply an appropriate low pass filter on the input data to generate the component that embodies the contribution from all frequencies lower than the cut-off frequency. This component will be referred as the low frequency component (LFC). The Butterworth filter – a built-in low pass digital filter function in MATLAB that provides the user a cut-off frequency option – was used for the analysis to follow.
4. In the resulting LFC, find all of the location of local minima. The ordinates of these minima will be referred to as the valley points.
5. For each two neighboring valley points at a time, the LFC segment between the chosen valley points is adjusted up or down until the percent of LFC segment values that are greater than the input data values within the valley points does not exceed the user-chosen tolerance value (2% for the data sets considered here).
6. The next two valley points are considered and the same exercise from step 6 is repeated until all valley points are processed pair-wise.
7. This exercise will likely create a set of disconnected lower envelope pieces. A curve-fitting tool is then used to patch the disconnected set of lower envelope pieces. The resulting curve will now constitute the lower envelope of the input data.

RESULTS AND DISCUSSION

Determination of the Cut-off Frequency

The CADETS algorithm was applied to the multiple long-term particle count data sets (Table 1) representing a variety of environments. The data are analyzed both as a total particle count (30–200 nm) and isolated into the four middle bins recorded by the instrument (30–50, 50–70, 70–100, and 100–200 nm). The first step of the algorithm is to determine an appropriate frequency to separate the time series signal, with the objective of decomposing the time series using into two components; one component is driven primarily by highly local and likely anthropomorphic events characterized by higher frequency components and the rest of the time series signal influenced primarily by regional-scale factors such as meteorology and long-range transport. The choice of the cut-off frequency is dependent on the data set and the intended use of the decomposed data sets. In this case, multiple data sets were used to justify the choice of the cut-off frequency. It is not necessary that every application of this methodology follow the same procedure. However, an example of a rigorous approach in selecting the cut-off frequency is demonstrated.

The periodograms associated with total particle count at all monitoring locations (Fig. 1) show the energy content in the frequencies associated with the data. Instead of the usual Energy versus Frequency, the plots show Energy versus Period (the inverse of frequency) for ease of interpretation. All sites uniformly exhibit a 1-day time-scale variation, which is likely strongly tied to the diurnal change in atmospheric mixing height. The presence of the same one-day cycle, even at the background site (RTP) further confirms the diurnal mixing effects to be the dominant driver of this periodicity. However, there may still be a nonzero contribution to this frequency from local emissions during favorable meteorological conditions. But, PSD analysis is not enough to investigate the composite effects of anthropogenic and environmental factors. More than anthropogenic effects, in many locations environmental factors can have a much larger impact on frequencies due to their diurnal nature.

At both Las Vegas sites (LV-road and LV-300m), the fundamental frequency of a roughly 12-hour time scale (Fig. 1) appears to be a good candidate for decomposition. The wind flow patterns and the traffic counts confirm that the 12-hour frequency is unlikely due to the daily rush hour traffic (Fig. S2), although it should be noted that traffic patterns in Las Vegas do not have a typical bimodal nature (Kimbrough *et al.*, 2013). Hourly wind direction with shaded area shows winds from the direction of the I-15 indicates that the evening traffic hour rush may be missed almost entirely as favorable winds reset only around 6 PM (Fig. S2). Nevertheless, caution must be exercised when decisions are made based on rush hour traffic patterns.

In many other studies of metropolitan areas, the 12-hour timescale has been noted as a result of the dilution effects of the morning rush hour traffic due to the expansion of the Planetary Boundary Layer (PBL) over the rest of the day (Kimbrough *et al.*, 2013). Prior near-road studies confirm that traffic-related ultrafine particle concentrations are generally asymmetrical comparing morning to afternoon periods, as the greater atmospheric mixing in the afternoon tends to decrease roadside pollution concentrations (Hagler *et al.*, 2009, Janhall *et al.*, 2004). Nevertheless, a 12-hour cycle still adequately captures the effects of the local activity (morning rush hour traffic) around the LV sites, and hence may be appropriate to decompose the time series signal into two components, with 12 hours as the time separation point of the two components. Such decompositions may be useful in other types of analyses where the goal is to separate local-scale and regional-scale contributions.

Decomposition of Time Series and Estimating Components

Following the selection of the cut-off frequency, the next steps of the CADETS algorithm decompose input data into two components, one that is more slowly varying when compared to the other. Step 3 of the CADETS algorithm builds the slow varying signal utilizing all frequencies below the selected cut-off 12 hr frequency, while steps 4–7 provide an empirical tuning of the estimated slow-varying time series (Fig. 2). Applying the CADETS method on LV-300m Aitken mode UFP data (30–100 nm), we obtain the result whose snapshot is shown in Fig. 3(a). As shown, the empirical tuning step minimizes the extent to which the slowly varying component can overpredict actual observations. The dashed line shows the initial reconstructed signal ("FFT-slowly varying"), whereas the solid black line shows the final reconstructed signal after empirical tuning. The solid black line marks the CADETS output that captures the slowly varying component of the original data (marked in gray). Subtracting this from the raw data gives the fast varying component (black line) shown in Fig. 3(b). For this data set, representing a monitoring location 300 m away from a major highway, the mean of ratio of the CADETS based highly varying data and the input data is approximately 0.64. That is, according to this operationally defined signal separation, approximately 36% of the signal may be due to slowly-varying and regional-scale effects and 64% would be attributed to local-scale factors.

After applying the CADETS algorithm with identical inputs to different monitoring locations and also separating by size bin, interesting results are revealed (Fig. 4). As expected, the background site location (RTP) had overall a higher contribution from the derived slow-varying factor in comparison to the Las Vegas locations for each of the four size bins ranging from 30–200 nm. In addition, the nonlocal contribution appeared to increase with larger particle size, where the accumulation mode particles tend to represent more aged air masses whereas the smaller particles are more associated with fresh emissions. Of the two Las Vegas locations shown, the roadside location had the lowest nonlocal contribution, as anticipated being so close to fresh exhaust emissions. The strength of the CADETS algorithm is the quantifiable estimate of nonlocal versus local contributions and ability to compare trends between locations.

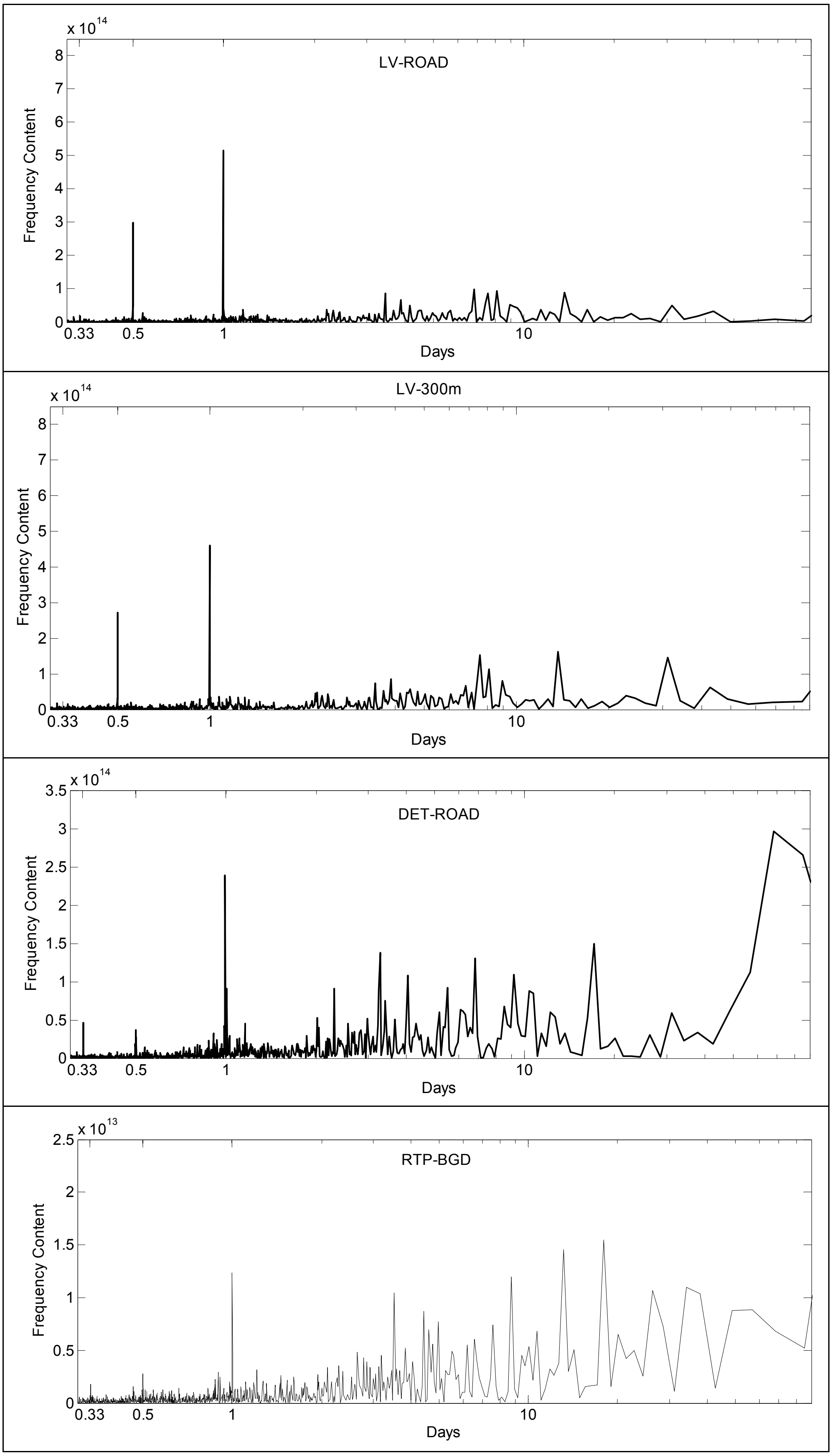

Fig. 1. Power Spectral Density of sites in Las Vegas (LV-ROAD, LV-300m), Detroit (DET-ROAD) and Research Triangle Park (RTP-BGD). The x-axis is log-scaled and y-axis is scaled linearly.

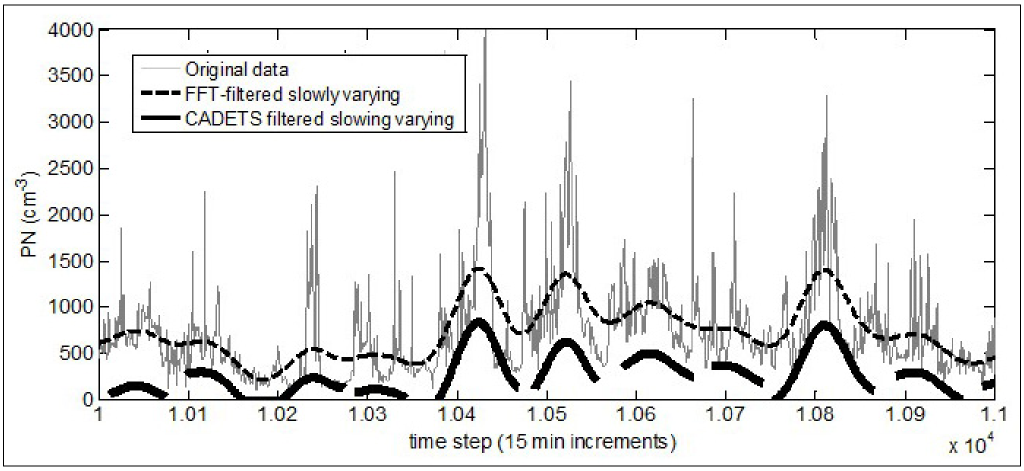

Fig. 2. This figure displays several processing steps of the CADETS algorithm. The raw input data (gray line) in the background is the LV-300m particle counts data (30 nm–200 nm). The dashed line is the raw filtered slowly-varying data. The heavy black lines are the initial CADETS representation of the dashed lines, which are then finalized with a spline interpolation.

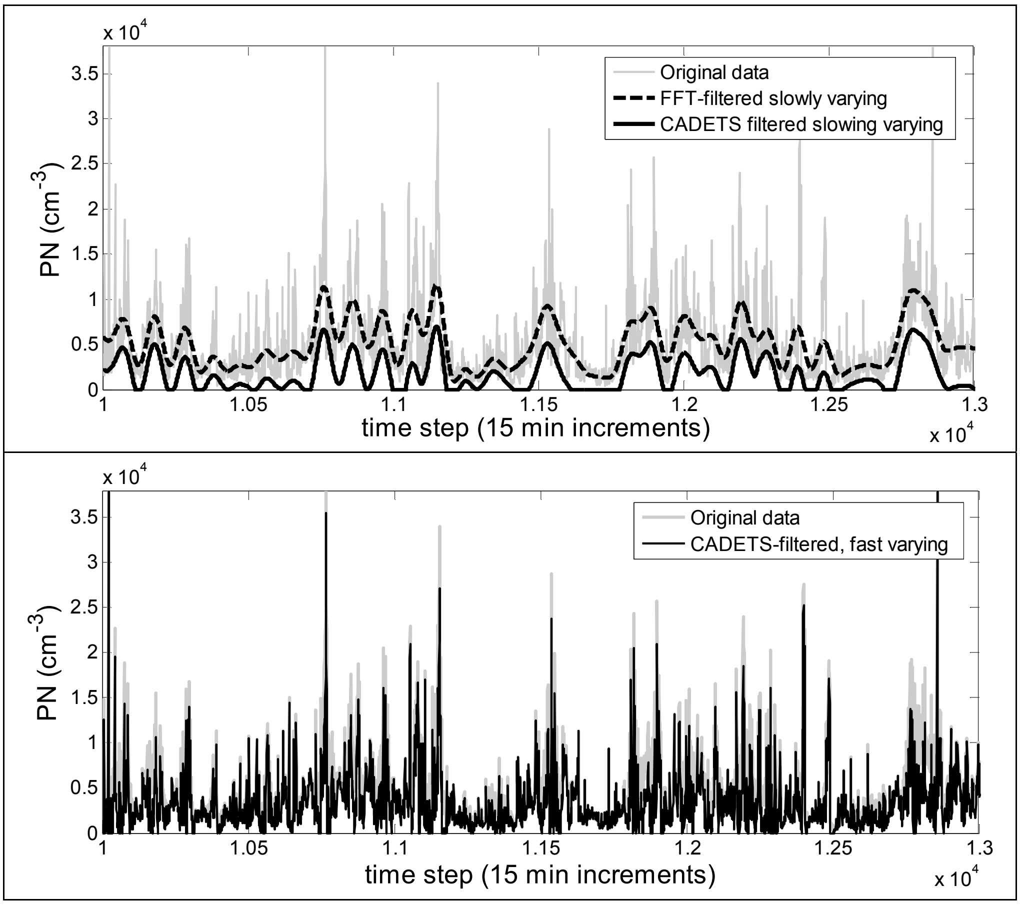

Fig. 3. The top figure shows the slowly time varying component of the LV-300m data, shown for the FFT-filtered step (dashed line) and then final CADETS estimate (solid black line). The bottom figure shows the final CADETS estimate for the fast-varying component of the data.

Sensitivity to Low Pass Filter Characteristics

The selection of the low pass filter in Step 3 may affect results. The sensitivity of the filter is evaluated. For this analysis, the Butterworth filter was used to obtain the initial decomposition. This filter belongs to a broader class of filters called the Infinite Impulse Response (IIR) filters (Oppenheim and Schafer, 2009). The alternate sets of filters are the Finite Impulse Response (FIR) filters. FIR

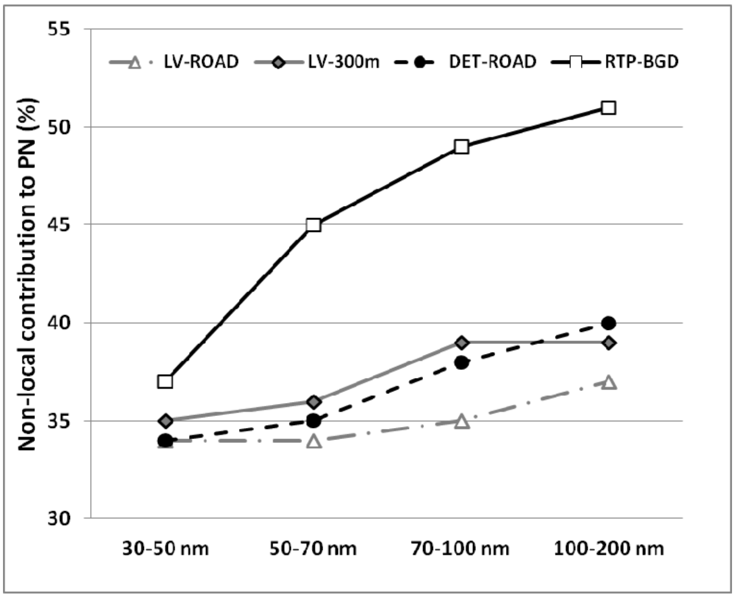

Fig. 4. Bin-based breakdown of the percent of slowly-varying ("non-local") contribution to the UFP time series, with a line connecting the markers for easy of viewing.

filters offer advantages over IIR filters that are more important in design of instruments. Speed and stability of filter responses and phase characteristics are critical in design of instruments and are not important in this application. The frequency response function for the Butterworth filter, as a function of the frequency ω, for the Nth order Butterworth filter is defined by

$$|B(j\omega)|^2 = \frac{1}{1+\left(\dfrac{j\omega}{j\omega_c}\right)^{2N}} \tag{1}$$

where ω_c is the cut-off frequency and $j = \sqrt{-1}$, the imaginary number common in complex number analysis.

The Butterworth filter was chosen due to maximally flat magnitude and phase response as compared to other filters, and it has a wide transition band from the pass band to the stop band domains leading to slightly higher spectral leakage. Other well-known IIR filters include the Chebyshev and Elliptic filters. Nevertheless, IIR filters best meet the needs of this analysis in providing an ideal "brick-wall" shaped lowpass magnitude response with almost no ripples in either passband or the stopband domains.

The initial FFT output was obtained using a 5th order (N = 5), relatively lower order (less computationally intensive), Butterworth filter. The Butterworth filter is a *lowpass* filter because frequencies lower (slower) than the specified cut-off frequency, ω_c, remain in the filtered output, whereas frequencies higher (faster) than the cut-off frequency are attenuated in order to smooth the signal. The initial cut-off frequency was based on the fundamental frequency present in the data, which can be observed by the first prominent peak in the Power Spectral Density (PSD) plot. Using the PSD in Fig. 1, an initial normalized cut-off frequency, ω_c, of 0.0209 was selected. This corresponds to the 12-hour cycle present in the data. The FFT output is the result of applying the 5th order lowpass Butterworth filter with normalized cut-off frequency of 0.0209 (Fig. 3(a), dashed line). However, possibly due to spectral leakage, the filtering at this initial cut-off frequency resulted in a decomposition that picked up some content from undesired frequencies (associated with 1-day period in Fig. 5(a)). By adjusting the normalized cut-off frequency and reviewing the resulting PSD of the component frequencies, the value of 0.023 was selected as being close enough to the theoretical 12-hour value without excessive leakage (Fig. 5(b)).

The same holds true for the choice of the order of the filter: The higher the filter order, the sharper the drop-off in the transition band. The Butterworth 10th order (Fig. 6(a)) and 50th order (Fig. 6(b)) filters shows a sharper drop-off for the higher order filter. However, the results did not vary significantly when the choices were varied. Also, even with other IIR filters, the average CADETS highly-varying decomposition to input data ratio varied by less than 1%. Thus, the choice of filter and its associated parameters were negligent in comparison to the effect of the choosing the cut-off frequency as discussed next.

The choice of cut-off the frequency is critical since this selection can have a significant effect on the resulting decomposition. For instance, the choice of 0.0167 as the normalized cut-off frequency (corresponds to the period of 15 hours), the slowly-varying (e.g., "non-local") component of the LV-300m UFP data was reduced to 32%. Similarly, the percent of slowly-varying contribution was reduced to 25% when the normalized frequency associated with 18 hours was chosen as the cut-off frequency. Apart from drops in estimation of non-local contributions, other aspects such as filter stability are also affected by the choice of the cut-off frequency as IIR filters are sensitive to errors introduced by rounding of coefficients used to construct the filters.

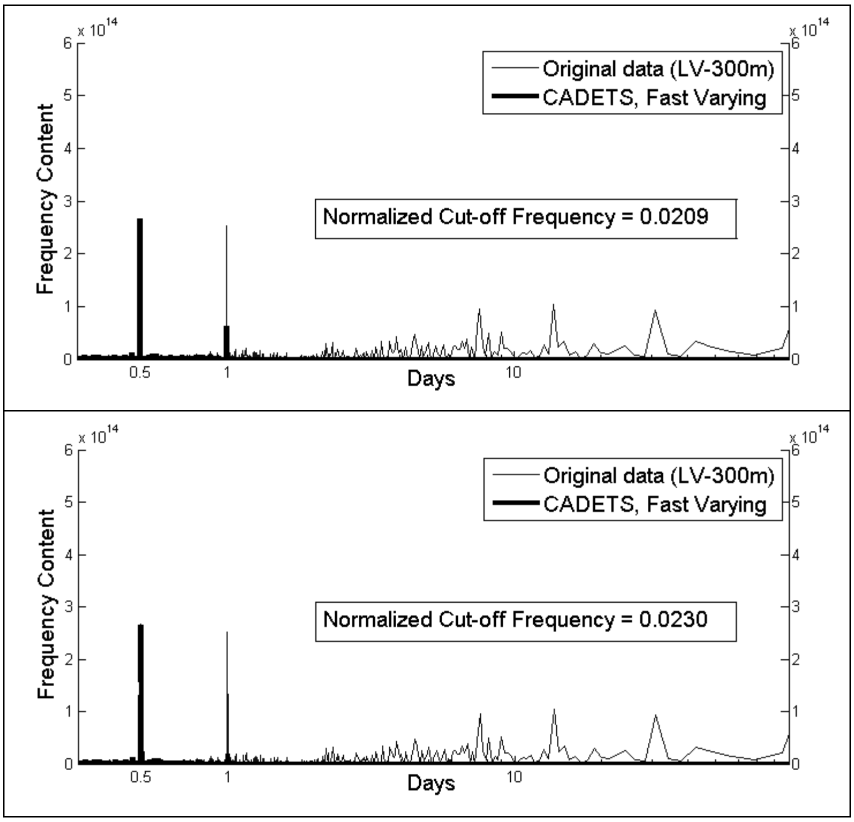

Fig. 5. Periodograms (frequency content) comparing cut-off frequencies success in decomposing the raw signal. Fig. 5(a) shows the possible consequence (leak) of an incorrect cut-off frequency, with significant content from a 1-day cycle.

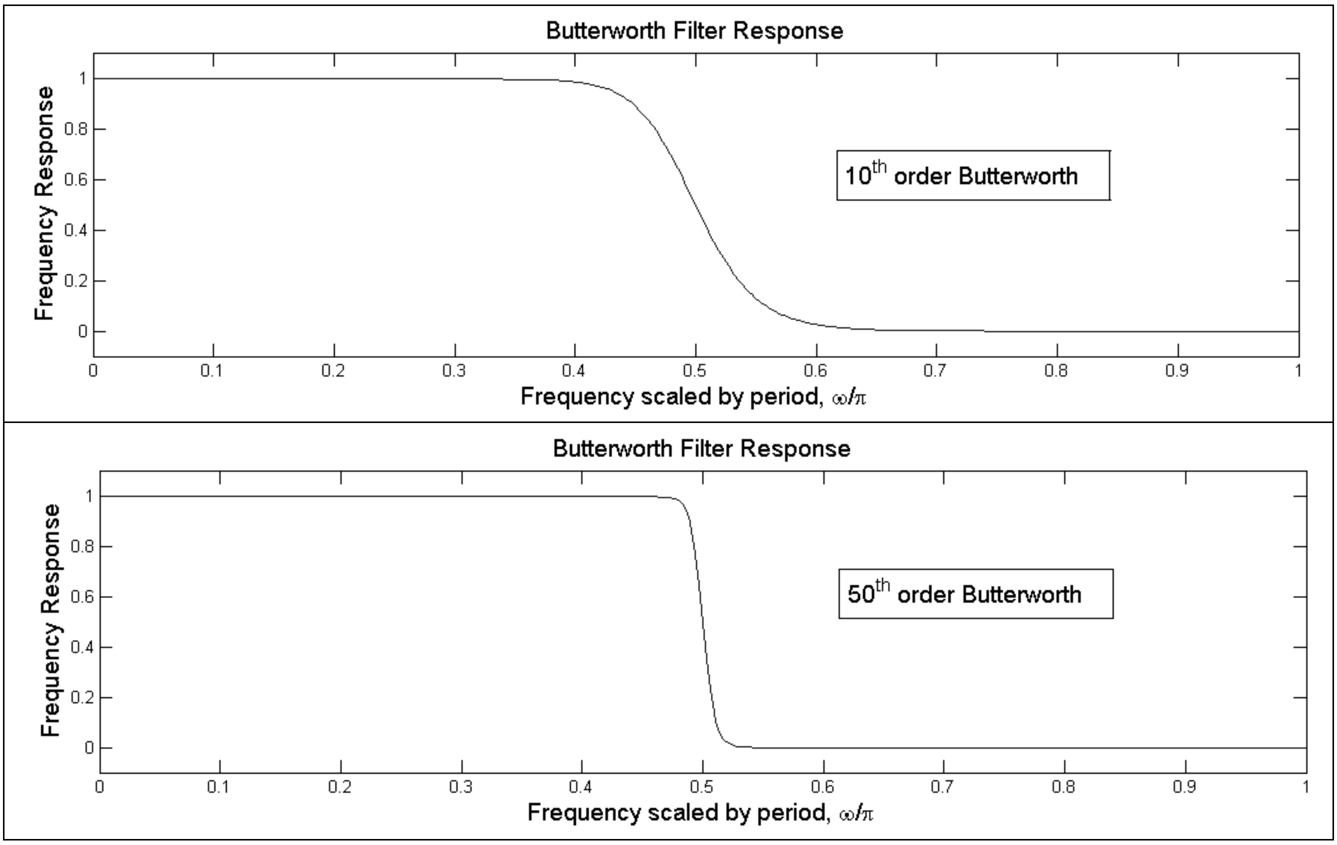

Fig. 6. Choice of orders for the Butterworth filter. The y-axis is the normalized magnitude response.

Energy Distribution Theorem and Its Implications

Finally, the PSD of the decomposed signals were compared with the original signal to confirm that the decomposed signals satisfied the stated goal of the frequency domain based on decomposition of signals (Fig. 6). The fast varying component (Fig. 7(a)) shows that CADETS was efficient in capturing the signal associated with a 12-hour cycle. The difference in the energy content in the 1-day period (Fig. 7(b)) can be interpreted in at least two different ways. The first was the loss of temporal signal in the decomposed signal and the other was the presence of harmonics of the fundamental frequency. We explored both possibilities.

A fundamental result in frequency analysis of time series data, Parceval's theorem, states that the energy content of a temporal signal remains invariant under a transformation using a complete set of orthogonal functions. In mathematical notation, if $f(t)$ and $\hat{f}(\omega)$ are the time series and its Fourier representations respectively, then the theorem states that

$$\int_{T_1}^{T_2} |f(t)|^2\, dt = \int_0^{\infty} \left|\hat{f}(\omega)\right|^2 d\omega. \qquad (2)$$

But, Fig. 7(b) suggests the CADETS output of the slowly varying component is lower than the slowly varying component of the input data. Using this theorem, we would expect that the temporal signal would be preserved. However, the discrepancy shown in Fig. 7(b) suggests that the CADETS output displays a loss of temporal content near the 1-day peak. The CADETS algorithmic step to stop reconstructing the LFC between any two valley points once the tolerance level has been reached (Step 6 of the algorithm – Fig. 2(b)) could be the cause of this loss. This tolerance level may introduce a small amount of error depending on how strict the level is set (here is was 2%). However, the energy is not actually lost; upon closer observation, it becomes clear the energy is preserved. Within the CADETS decomposition, the slowly varying and rapidly varying components can be added to achieve the original data. The algorithm has essentially transferred the temporal energy content from the slowly varying component to the rapidly varying component (12-hour cycle in Fig. 7(a)).

Another explanation for this loss may be that some fraction of energy in the 1-day period may be attributable to the harmonics associated with the fundamental frequency. A close scrutiny of the PSD (Fig. 1) shows small amount energy in the 8-hour and 12-hour cycles even in the RTP background site with their harmonics contributing to the energy in the 1-day period. However, Fourier analysis alone is not capable of detecting and quantitatively determining the contribution of the fundamental frequency to its harmonics. If the cause of the energy transfer was due to

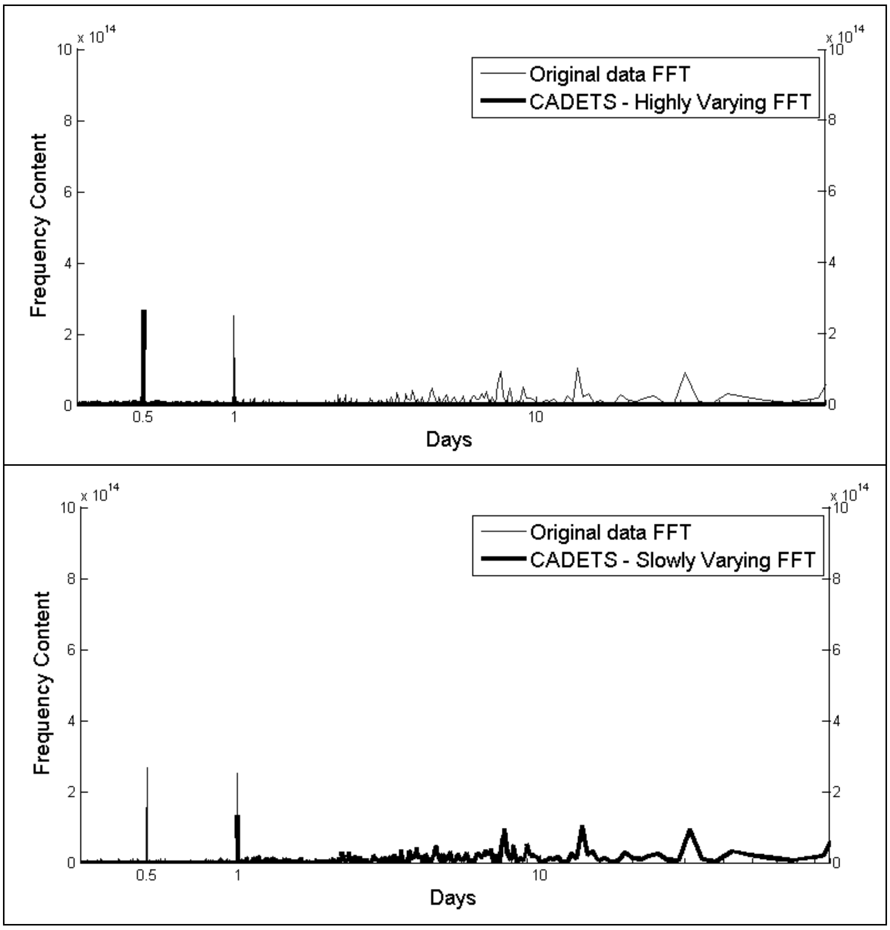

Fig. 7. FFT analysis confirming the efficacy of the CADETS decomposition into highly- and slowly-varying components.

the CADETS algorithm, it can be minimized by raising the tolerance limit (2% for this application – Step 6 of the algorithm). It is highly likely that a combination of the two reasons stated above is the cause of the energy transfer.

Comparing Fourier and Wavelet Transform Approaches to Decompose Signals

A similar method has been explored by Klems *et al.* (2010) using one-second particle number concentrations (using a condensation particle counter) collected near a busy intersection to study its origins and to apportion the collected data to appropriate sources. In that study, the authors used wavelet transforms to parse out the high frequency portion of the data. They report that the high frequency contribution (presumably local in nature) accounted for 6–35% of the daily ambient concentration, but some of the hourly contributions topped 50%. Using CADETS, we conclude that even in our background site particle count data, the local source contribution exceeded 50%. The difference in the attribution could be due to many reasons. Primary among them are the signal processing methodology, sampling interval (1 second vs. 15 minute), and measurement environment.

While CADETS allows the user to choose the cut-off frequency and apply a lowpass filter on the data using the chosen frequency, the method proposed by Klems *et al.* (2010) suggests that data be repeatedly filtered until the resulting output reaches the point of minimal improvement. The latter approach is practical in separating the signal into two parts, but does not provide any information about the separated components. Even though the bifurcation point (in CADETS) is a choice made by the user, we are suggesting an approach that encourages a more rigorous argument to justify the choice of the bifurcation point and allows for the same processing to be applied to multiple data sets for comparability.

While both approaches (wavelet and Fourier) can decompose data, there are differences in how the data are filtered. Wavelets are ideal in capturing isolated peaks (as in seismic events) or sharp changes in features such as images, but the filters have leakage issues due to significant ripples in the pass band and stop band that reduces its effectiveness on time series data. Also, wavelet filters with very similar magnitude responses have different phase responses which can greatly affect filter response in some important frequency intervals. As an example, when the winds favor the receptor site, traffic related events may occur as a rapid increase followed by gradual decline and not necessarily as isolated sharp peaks. Thus, marker-based traffic related data may not be suitable data for wavelet analysis. Nevertheless, wavelet analysis may be more suitable than Fourier analysis for small data sets or those collected for short durations where adequate information of component frequencies (necessary for Fourier analysis) may not be available.

CONCLUSIONS

Air monitoring datasets are evolving towards higher time resolution, with newer technology allowing long-term field studies to report pollution data at subhourly (even second by second) time intervals. In addition, the emergence of air sensor technology – such as particulate sensors currently commercially available at low price points (e.g., under 2000 USD) – will likely produce a significant increase in real-time air pollution data that may advance our understanding of temporal and spatial trends. Advanced time series approaches – such as CADETS – will support the retrieval of meaningful information from long-term and high-time resolution time series. The CADETS algorithm, demonstrated on multiple highly variable ultrafine particle datasets, separates a single time series into slowly-varying and rapidly-varying components. This approach is most effective for measurement scenarios where the input data is of high-time resolution and contains high amplitude and short-duration peaks, characteristic of nearby source emissions. For data sets that are of lower time resolution or where source signals may have low frequency fluctuation – such as emissions of interest located at a distance from the measurement location or for pollutants formed secondarily through post-emission processes – the separation of slowly-varying and rapidly-varying components via CADETS may be limited or more challenging to interpret. Overall, this separation strategy is anticipated to provide new insight in causes of air pollution variation in complex environments, particularly for directly emitted pollutants.

ACKNOWLEDGMENTS

Authors wish to express appreciation to the EPA near-road research team who facilitated the collection of UFP data, including Richard Shores, Richard Baldauf, Bill Squier, and Bill Mitchell. Support from TSI inc. (the instrument manufacturer) staff is also appreciated, specifically Jeff Baker and Maynard Havlicek. In addition, insight and guidance on the algorithm and paper from staff scientists in the EPA National Exposure Research Laboratory, especially Gary Norris, is appreciated.

DISCLAIMER

The United States Environmental Protection Agency through its Office of Research and Development funded and managed the research described here. It has been subjected to Agency's administrative review and approved for publication.

SUPPLEMENTARY MATERIALS

Supplementary data associated with this article can be found in the online version at http://www.aaqr.org.

REFERENCES

Bauch, C.T. and Earn, D.J.D. (2003). Transients and Attractors in Epidemics. *Proc. R. Soc. London, Ser. B* 270: 1573–1578.

Ganguli, R. (2002). Noise And Outlier Removal From Jet Engine Health Signals Using Weighted Fir Median Hybrid Filters. *Mech. Syst. Sig. Process.* 16: 967–978.

Geerkan, R., Zaitchik, B. and Evans, J.P. (2005) Classifying Rangeland Vegetation Type and Coverage from NDVI Time Series Using Fourier Filtered Cycle Similarity. *Int. J. Remote Sens.* 26: 5535–5554.

Hagler, G.S.W., Baldauf, R.W., Thoma, E.D., Long, T.R., Snow, R.F., Kinsey, J.S., Oudejans, L. and Gullett, B.K. (2009) Ultrafine Particles near a Major Roadway in Raleigh, North Carolina: Downwind attenuation and Correlation with Traffic-related Pollutants. *Atmos. Environ.* 43: 1229–1234.

Hagler, G.S.W., Bergin, M.H., Salmon, L.G., Yu, J.Z., Wan, E.C.H., Zheng, M., Zeng, L.M., Kiang, C.S., Zhang, Y.H., Lau, A.K.H and Schauer, J.J. (2006). Source Areas and Chemical Composition of Fine Particulate Matter in the Pearl River Delta Region of China. *Atmos. Environ.* 40: 3802–3815.

Henry, R.C., Vette, A., Norris, G., Vedantham, R., Kimbrough, S. and Shores, R. (2011). Separating the Air Quality Impact of a Major Highway and Nearby Sources by Nonparametric Trajectory Analysis. *Environ. Sci. Technol.* 45: 10471–10476.

Janhäll, S., Jonsson, A.M., Molnár, P., Svensson, E.A. and Hallquist, M. (2004). Size Resolved Traffic Emission Factors of Submicrometer Particles. *Atmos. Environ.* 38: 4331–4340.

Kimbrough, S., Baldauf, R.W., Hagler, G.S.W., Shores, R.C., Mitchell, W., Whitaker, D.A., Croghan, C.W. and Vallero, D.A. (2013) Long-term Continuous Measurement of Near-road Air Pollution in Las Vegas: Seasonal Variability in Traffic Emissions Impact on Local Air Quality. *Air Qual. Atmos. Health* 6: 295–305.

Klems, J.P., Pennington, M.R., Zordan, C.A. and Johnston, M.V. (2010). Ultrafine Particles near a Roadway Intersection: Origin and Apportionment of Fast Changes in Concentration. *Environ. Sci. Technol*. 44: 7903–7907.

Lu, X., Liu, R., Liu, J. and Liang, S. (2007). Removal of Noise by Wavelet Method to Generate High Quality Temporal Data of Terrestrial MODIS Products. *Photogramm. Eng. Remote Sens.* 73: 1129–1139.

Oppenheim and Schafer (2009). *Digital Signal Processing*. 3 ed, ed. P.H.S. Processing.

Schauer, J.J. and Cass, G.R. (1996). Source Apportionment of Airborne Particulate Matter Using Organic Compounds as Tracers. *Atmos. Environ.* 30: 3837–3855.

Zhang, K.M., Wexler, A.S., Zhu, Y.F., Hinds, W.C., and Sioutas, C. (2004). Evolution of Particle Number Distribution near Roadways. Part II: The 'Road to Ambient' Process. *Atmos. Environ.* 38: 6655–6665.

Zhu, Y.F., Hinds, W.C., Shen, S. and Sioutas, C. (2004). Seasonal Trends of Concentration and Size Distribution of Ultrafine Particles near Major Highways in Los Angeles. *Aerosol Sci. Technol.* 38: 5–13.

Received for review, February 19, 2015
Revised, May 11, 2015
Accepted, June 22, 2015

Supplemental Information

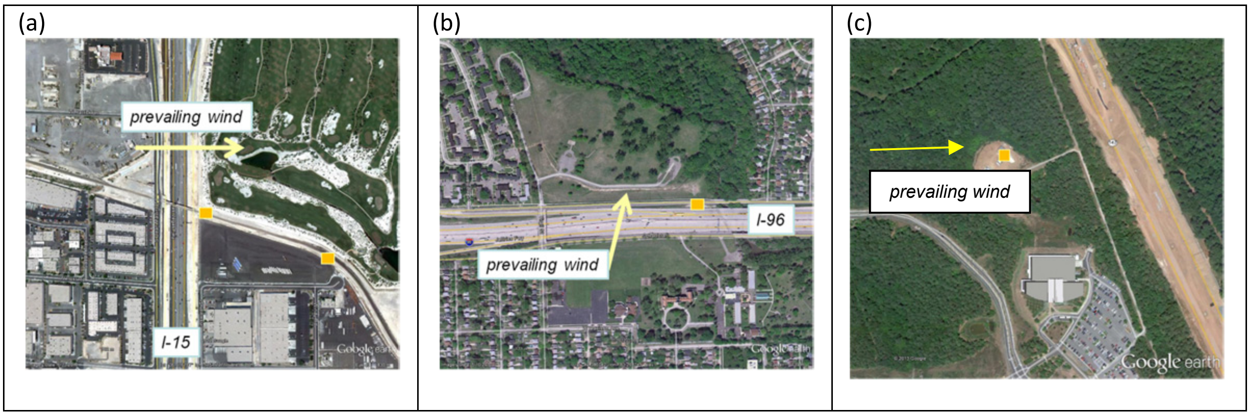

Figure S1: Location of field monitoring sites, LV-road and LV-300m (a), DT-road (b), and RTP-BGD (c). The orange squares note the locations of the UFP monitoring stations. Note that a roadway was under construction approximately 150 m from the RTP-BGD monitoring station at the time of sampling (c).

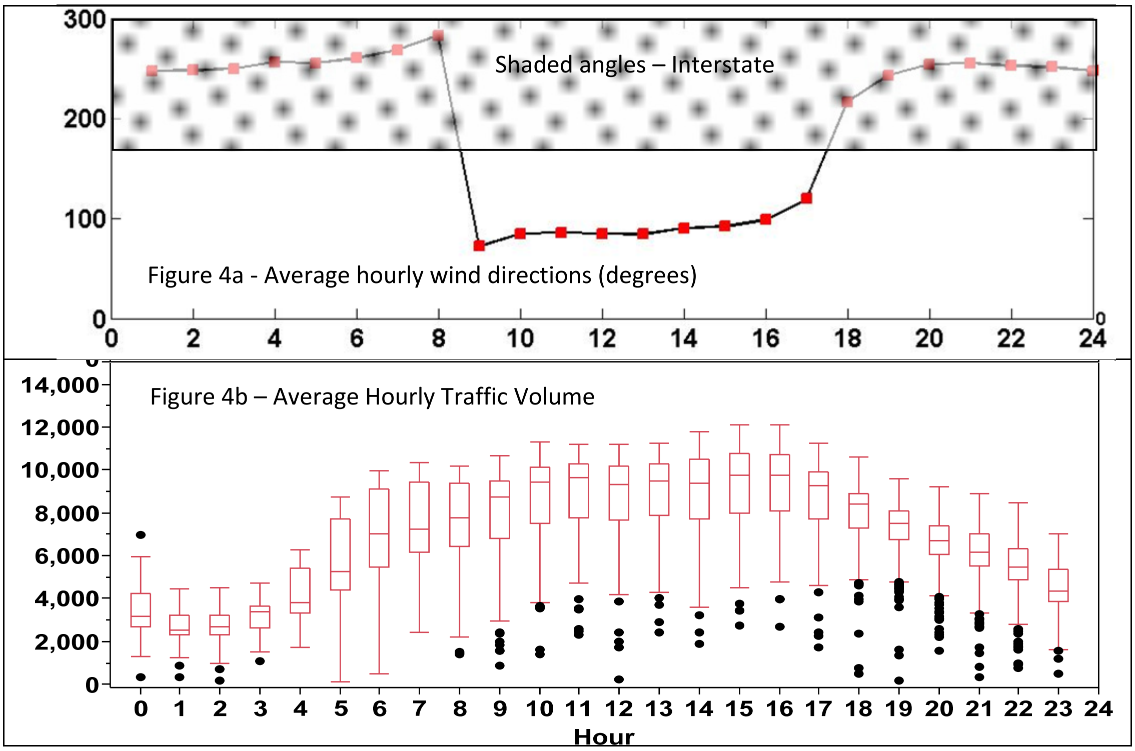

Figure S2: Wind Directions (Degrees) and Traffic Volumes (count) measured at the Las Vegas near road site. The boxes in Figure 4b represents the inter-quartile range while the whiskers represent the 5th (bottom) and 95th (top) percentiles.

CADETS MATLAB Code

```
%  CADETS code
% This code expects the user to provide several pieces of information:
% 1. data: the measurement data (e.g., a single column of values)
% 2. timeint: the time base of the data (e.g., 15 min)
% 3. butterval: the user provided cut-off frequency.
% 4. butterorder: this value sets the order of the lowpass digital Butterworh filter
% 5. convert2hrs: value to convert timeint to hours (e.g., 15 min / 1440
% min/hr)
% 6. peakwindow: value to set length of the window for finding peaks

%user provided information
%load data
%data is a single column of measurement values provided by the user
timeint = 15; %example provided setting the data time interval at 15 min
butterval = 0.023; % Vedantham et al. (2014) utilized value of 0.023.
butterorder = 10; % Vedantham et al. (2014) utilized value of 10.
convert2hrs = 1440; %converts minutes to days
peakwindow = 50;  % 12 hour fundamental frequency for a 15-minute data translates to
~50 (12
% times 4) as the minpeakdistance. Otherwise, way too many peaks and valley points are
found.

%code begins here
NFFT = 2^nextpow2(length(data))/2;  %sets the number of frequencies to be tested in
fft
freqs = [1:NFFT]'/NFFT;  %normalized to give array of frequencies
periods = 1./freqs';
days = periods*timeint/convert2hrs; % division to convert timebase to hours
data(isnan(data)) = []; %remove any rows with missing values
%%
[b,a] = butter(butterorder,butterval); %10th order butterworth filter with cut-off
frequency of 0.023 (ref. Vedantham et al.)
output = filtfilt(b,a,data); %application of butterworth filter to data set
clear a b
%%
datafft = fft(data,NFFT); %Fast Fourier transfor of the raw data
outputFFT = fft(output,NFFT); %Fast Fourier transform of the data passed through the
butterworth filter
theoreticalLocal = data-output; %subtraction of the low frequency signal from the raw
data
theoLocalFFT = fft(theoreticalLocal,NFFT); %Fast Fourier transform of the estimated
"local" timeseries

clear freqs periods
[valleys,valleyLocs] = findpeaks(-output,'minpeakdistance',peakwindow);  %find local
minima in set window length, using findpeaks command on inversion of output variable
[peaks,peakLocs] = findpeaks(output,'minpeakdistance',peakwindow);  %find local maxima
in set window length

%% Now, using the peak and the valley points, the slowly varying part is reconstructed
extValleyLocs = union(union(1,valleyLocs),length(data)); %appends indices 1 and end of
data to valleyLocs
notLocal = NaN(size(output)); %create notLocal variable, representing slowly varying
timeseries component
exceeds = 0;
needPatchup = 0;
prevRange = 1;
prevMaxLoc = peakLocs(1);
prevPatch = [];
for i = 2:length(extValleyLocs)
```

```
    currRange = extValleyLocs(i-1):extValleyLocs(i);  %sets window to span two local
minima
    maxLoc = currRange(find(ismember(currRange,peakLocs)));  %finds local maxima
indices within the set range
    if isempty(maxLoc)  %if no local maxima is within window, sets the max value as
the highest value within the range
        [maxVal,loc] = max(output(currRange));
        maxLoc = currRange(loc(1));
    else
        maxLoc = maxLoc(1); %otherwise, selects first occurrence of a local maxima in
the window
    end
    spanVal = 5;
    while exceeds < 2
        thisRange = max(maxLoc-
spanVal,currRange(1)):min(maxLoc+spanVal,currRange(end));  %creates a window spanning
spanVal indices +/- the location of the maxLoc
        if length(find(ismember(thisRange,currRange))) < length(currRange)
            thisPatch = NaN(size(output));
            thisPatch(thisRange) = output(thisRange);  %apply output values for the
local maxima window
            thisPatch = thisPatch - nanmin(thisPatch);
            exceeds = length(find(thisPatch(currRange)>...
                data(currRange)))*100/length(currRange);  %compares local output
maxima against original raw data
            spanVal = spanVal + 10;  %expands span window and repeats
        else
            break;
        end
    end
    if length(find(ismember(thisRange,currRange))) == length(currRange)
        maxValDiff = output(maxLoc) - max(thisPatch); %compares output local maximum
versus maximum estimated for window
        factorVal = 0.1;
        thisPatch = NaN(size(output));
        thisPatch(thisRange)  = output(thisRange);
        exceeds = length(find(thisPatch(currRange)>...
            data(currRange)))*100/length(currRange);
        if exceeds <= 2
            break
        else
            while exceeds > 2 %2% is the user-chosen tolerance (step 5 of the paper)
                thisPatch = thisPatch - factorVal*maxValDiff;   %decreases estimated
value
                exceeds = length(find(thisPatch(currRange)>...
                    data(currRange)))*100/length(currRange);
                factorVal = factorVal + 0.05;                   %increases factorVal
slightly and repeats
            end
            thisPatch(thisPatch < 0) = NaN;
        end
    end
    if prevRange(end) == thisRange(1) % Arrive at the patch up if the previous patch
was the entire intrval
        cutoffLoc = find(prevPatch(prevMaxLoc:prevRange(end))<thisPatch(1),1,'first');
        if isempty(cutoffLoc)
            shortStretch = min(maxLoc - currRange(1),currRange(end)-maxLoc);
            cutoff = floor(0.3*shortStretch);
            thisPatch([currRange(1):currRange(1)+cutoff ...
                currRange(end)-cutoff:currRange(end)]) = NaN;

        end
        nonLocal(cutoffLoc:prevRange(end)) = NaN;
    end
    nonLocal(currRange) = thisPatch(currRange);
    lastNonzeroVal = find(~isnan(nonLocal(prevRange)),1,'last');
    firstNonzeroVal = find(~isnan(nonLocal(thisRange)),1,'first');
    if nonLocal(prevRange(lastNonzeroVal)) == 0 & ...
            nonLocal(thisRange(firstNonzeroVal)) == 0
        nonLocal(prevRange(lastNonzeroVal):thisRange(firstNonzeroVal)) = 0;
    end

    prevRange = currRange;
    prevPatch = thisPatch;
    prevMaxLoc = maxLoc;
    exceeds = 0;
end

missingVals = find(isnan(nonLocal));
nonMissingVals = setdiff(1:length(nonLocal),missingVals);
%the next line interpolates the missing spots using cubic smoothing splines
connections = csaps(nonMissingVals,nonLocal(nonMissingVals),1,missingVals);
completeNonLocal = nonLocal';
completeNonLocal(missingVals) = connections;
completeNonLocal(completeNonLocal < 0) = 0; %set negative values to zero
completeNonLocalFFT = fft(completeNonLocal,NFFT);
completeLocal = data-completeNonLocal(1:length(data));
completeLocal(completeLocal<0) = 0; %set negative values to zero
completeLocalFFT = fft(completeLocal,NFFT);
data(data == 0) = NaN;
nanmean(completeNonLocal(1:length(data))./data)
%remove temporary variables
clear NFFT exceeds connections prevPatch prevMaxLoc prevRange lastNonzeroVal
firstNonzeroVal i needPatchup maxLoc maxVal
clear maxValDif peaks peakLocs thisPatch thisRange timeint valleys currRange
convert2hrs loc spanVal shortStretch cutoff
clear butterorder ans utterval allTogether cuttoff cutoffLoc maxValDiff factorVal
extValleyLocs peakwindow valleyLocs butterval
return

%%
figure
dataPlot = plot(data,'DisplayName','data','YDataSource','data');figure(gcf)
axis tight
ylim([0 38000])
xlim([12000 19000])
hold on
% thisPlot = plot(output,'c','linewidth',2);
nonLocal = plot(completeNonLocal,'r');

%%
figure;
fftPlot = plot(days,abs(datafft).^2);
axis tight
currAxes = gca;
set(currAxes,'xscale','log');
xlims = [0.3 90];
xlim(xlims)
ylim([0 2.5e15]);
ax2 = axes('Position',get(currAxes,'Position'));
localFFT = plot(ax2,days,abs(theoLocalFFT).^2,'r','linewidth',3);
set(ax2,'xscale','log','yaxislocation','right','box','off','xlim',xlims,'color','none')
set(currAxes,'box','off');
set(ax2,'box','off');
set(ax2,'color','none','xlim',xlims);
ylim(currAxes,[0 1e15])
ylim(ax2,[0 1e15]);
```

```
legend([fftPlot localFFT],{'Original data FFT','CADETS - Highly Varying FFT'});
%%
figure;
fftPlot = plot(days,abs(datafft).^2);
axis tight
currAxes = gca;
set(currAxes,'xscale','log');
xlims = [0.3 90];
xlim(xlims)
ylim([0 2.5e15]);
ax2 = axes('Position',get(currAxes,'Position'));
decomposedFFT = plot(ax2,days,abs(outputFFT).^2,'r','linewidth',3);
set(ax2,'xscale','log','yaxislocation','right','box','off','xlim',xlims,'color','none')
set(currAxes,'box','off');
set(ax2,'box','off');
set(ax2,'color','none','xlim',xlims);
ylim(currAxes,[0 1e15])
ylim(ax2,[0 1e15]);
legend([fftPlot decomposedFFT],{'Original data FFT','CADETS - Slowly Varying FFT'});
```