

## State-Space Models – Introduction

- through two seemingly simple equations, *state-space models* define a rich class of processes that have served well as models for time series
  - special cases: ARMA, ARIMA and SARIMA models
- so-called *Kalman recursions* for state-space models offer an elegant solution not only for forecasting time series, but also for filtering and smoothing (see forthcoming definitions)
- state-space models and Kalman recursions can be readily adapted to handle time series with missing values (‘gappy’ series)
- just as ARMA processes are built upon white noise, state-space models are built upon a certain type of uncorrelated noise

## Multivariate Uncorrelated Noise: I

- let  $\mathbf{W}_t$  be a random vector (i.e., a vector of random variables (RVs)), where  $t$  is a time index
- as a running example, consider the bivariate case:

$$\mathbf{W}_t = \begin{bmatrix} W_{1,t} \\ W_{2,t} \end{bmatrix},$$

where  $W_{1,t}$  and  $W_{2,t}$  are two real-valued RVs

- as usual, expected value of random vector is denoted as  $E\{\mathbf{W}_t\}$ , which is a vector of expectations of components of  $\mathbf{W}_t$ :

$$E\{\mathbf{W}_t\} = \begin{bmatrix} E\{W_{1,t}\} \\ E\{W_{2,t}\} \end{bmatrix}$$

- $E\{\mathbf{W}_t\} = \mathbf{0}$  says all RVs in  $\mathbf{W}_t$  have mean zero ( $\mathbf{0}$  denotes a vector of zeros)

## Multivariate Uncorrelated Noise: II

- assuming  $E\{\mathbf{W}_t\} = \mathbf{0}$ , covariance matrix for  $\mathbf{W}_t$  is given by

$$\begin{aligned} R_t &\stackrel{\text{def}}{=} E\{\mathbf{W}_t\mathbf{W}_t'\} = E\left\{ \begin{bmatrix} W_{1,t} \\ W_{2,t} \end{bmatrix} \begin{bmatrix} W_{1,t} & W_{2,t} \end{bmatrix} \right\} \\ &= \begin{bmatrix} \text{var}\{W_{1,t}\} & \text{cov}\{W_{1,t}, W_{2,t}\} \\ \text{cov}\{W_{2,t}, W_{1,t}\} & \text{var}\{W_{2,t}\} \end{bmatrix} \end{aligned}$$

- $\mathbf{W}_1, \mathbf{W}_2, \dots$  said to be *uncorrelated noise* if  $E\{\mathbf{W}_s\mathbf{W}_t'\} = \mathbf{0}$  for all  $1 \leq s < t$  (note: here 0 is a matrix of zeros)
- note carefully:
  - $R_t$  need *not* be a diagonal matrix; i.e., RVs within a given  $\mathbf{W}_t$  can be correlated (in contrast to RVs from distinct  $\mathbf{W}_t$ 's)
  - RVs in  $\mathbf{W}_t$  need *not* have the same variance
  - $R_s$  and  $R_t$  need *not* be the same

## Multivariate Uncorrelated Noise: III

- $\mathbf{W}_t \sim \text{UN}(\boldsymbol{\mu}_t, R_t)$  denotes uncorrelated noise with mean vectors  $\boldsymbol{\mu}_t$  and covariance matrices  $R_t$ ,  $t = 1, 2, \dots$
- $\mathbf{W}_t \sim \text{WN}(\boldsymbol{\mu}, R)$  denotes white noise, i.e., uncorrelated noise with time-independent mean vectors and covariance matrices
- note: Brockwell & Davis use  $\mathbf{W}_t \sim \text{WN}(\mathbf{0}, R_t)$  to denote uncorrelated noise with zero mean, but with (possibly) time-varying covariance matrices (precisely defined, but, when  $\mathbf{W}_t$  is one dimensional, does *not* reduce to what we've been calling white noise)

## State-Space Models: I

- let  $\mathbf{Y}_1, \mathbf{Y}_2, \dots$  be a time series of dimension  $w$
- state-space model for this series consists of two equations

1. *observation equation* takes form

$$\mathbf{Y}_t = G_t \mathbf{X}_t + \mathbf{W}_t, \quad t = 1, 2, \dots,$$

where

- $\mathbf{X}_t$  is  $v$ -dimensional *state vector* (stochastic in general)
  - $G_t$  is  $w \times v$  observation matrix (deterministic)
  - $\mathbf{W}_t \sim \text{UN}(\mathbf{0}, R_t)$  is observation noise, with  $\mathbf{W}_t$  &  $R_t$  having dimensions  $w$  &  $w \times w$  (can be degenerate, i.e.,  $R_t = 0$ )
- observation equation essentially says that we can observe linear combinations of variables in state vector, but typically only in presence of noise (noise present if  $R_t \neq 0$ )

## State-Space Models: II

2. *state-transition equation* takes form

$$\mathbf{X}_{t+1} = F_t \mathbf{X}_t + \mathbf{V}_t, \quad t = 1, 2, \dots,$$

where

- $F_t$  is  $v \times v$  state transition matrix (deterministic)
- $\mathbf{V}_t \sim \text{UN}(\mathbf{0}, Q_t)$  is state-transition noise, with  $\mathbf{V}_t$  &  $Q_t$  having dimensions  $v$  &  $v \times v$
- two additional assumptions
  - a.  $E\{\mathbf{W}_s \mathbf{V}_t'\} = 0$  for all  $s$  and  $t$  (here 0 is a  $w \times v$  matrix of zeros; in words, every observation noise RV is uncorrelated with every state-transition noise RV)
  - b. assuming  $E\{\mathbf{X}_1\} = \mathbf{0}$  for convenience,  $E\{\mathbf{X}_1 \mathbf{W}_t'\} = 0$  &  $E\{\mathbf{X}_1 \mathbf{V}_t'\} = 0$  for all  $t$  (in words, initial state vector RVs are uncorrelated with observation & state-transition noise)

## AR(1) Process as a State-Space Model: I

- state-transition equation

$$\mathbf{X}_{t+1} = F_t \mathbf{X}_t + \mathbf{V}_t$$

is reminiscent of a causal AR(1) model:

$$\mathcal{X}_{t+1} = \phi \mathcal{X}_t + \mathcal{Z}_{t+1} \text{ with } \{\mathcal{Z}_t\} \sim \text{WN}(0, \sigma^2) \text{ and } |\phi| < 1$$

- can express AR(1) in state-space formulation by setting

- $\mathbf{X}_{t+1} = \mathcal{X}_{t+1}$

- $F_t = \phi$

- $\mathbf{V}_t = \mathcal{Z}_{t+1}$  along with  $Q_t \stackrel{\text{def}}{=} E\{\mathbf{V}_t \mathbf{V}_t'\} = E\{\mathcal{Z}_{t+1}^2\} = \sigma^2$

and by using a degenerate form of the observation equation

$$\mathbf{Y}_t = G_t \mathbf{X}_t + \mathbf{W}_t \text{ in which}$$

$$G_t = 1 \text{ and } \mathbf{W}_t = 0 \text{ (implying } R_t = 0) \text{ so that } \mathbf{Y}_t = \mathcal{X}_t$$

## AR(1) Process as a State-Space Model: II

- to complete model, need to define initial state  $\mathbf{X}_1$
- natural choice is

$$\mathcal{X}_1 = \sum_{j=0}^{\infty} \phi^j \mathcal{Z}_{1-j}, \text{ for which } \text{var} \{ \mathcal{X}_1 \} = \frac{\sigma^2}{1 - \phi^2}$$

- with this choice, required conditions, namely,  $E\{\mathbf{X}_1 \mathbf{W}_t'\} = 0$   
&  $E\{\mathbf{X}_1 \mathbf{V}_t'\} = 0$ , hold:  $E\{\mathbf{X}_1 \mathbf{W}_t'\} = E\{\mathcal{X}_1 \times 0\} = 0$  and

$$E\{\mathbf{X}_1 \mathbf{V}_t'\} = E\{\mathcal{X}_1 \mathcal{Z}_{t+1}\} = \sum_{j=0}^{\infty} \phi^j E\{\mathcal{Z}_{1-j} \mathcal{Z}_{t+1}\} = 0 \text{ for } t = 1, 2, \dots$$

- could also set  $\mathbf{X}_1 = \mathcal{Z}_1 \frac{\sigma}{\sqrt{1-\phi^2}}$  to get AR(1) process, but using, e.g.,  $\mathbf{X}_1 = \mathcal{Z}_1$  would lead to a valid state-space model that is *not* a true AR(1) model (is such in an asymptotic sense)

## AR(1) Process as a State-Space Model: III

- AR(1) process with  $0 < \phi < 1$  is known as ‘red noise’
- red noise is related to a 1st-order stochastic differential equation, rendering it a model for various geophysical processes
- can usually only observe red noise process of interest in presence of observational noise (often taken to be white noise)
  - Problem 20: AR(1) + white noise yields ARMA(1,1)
- by modifying state-space formulation for an AR(1) process, can model this setup by changing observational noise from  $\mathbf{W}_t = 0$  to  $\mathbf{W}_t = \mathcal{W}_t \sim \text{WN}(0, \sigma_{\mathcal{W}}^2)$  (with  $R_t$  changing from 0 to  $\sigma_{\mathcal{W}}^2$ ), where  $\mathcal{W}_t$  is uncorrelated with  $\mathcal{Z}_t$ 's (and hence  $\mathcal{X}_1$  also)
- observation and state-transition equations become

$$\mathcal{Y}_t = \mathcal{X}_t + \mathcal{W}_t \quad \text{and} \quad \mathcal{X}_{t+1} = \phi \mathcal{X}_t + \mathcal{Z}_{t+1}$$

## ARMA(1,1) Process as a State-Space Model: I

- formulating ARMA(1,1) process  $\mathcal{Y}_t - \phi\mathcal{Y}_t = \mathcal{Z}_t + \theta\mathcal{Z}_{t-1}$  as a state-space model is trickier
- expressing model as  $\phi(B)\mathcal{Y}_t = \theta(B)\mathcal{Z}_t$ , note that we can create  $\mathcal{Y}_t$  by taking causal AR(1) process  $\phi(B)\mathcal{X}_t = \mathcal{Z}_t$ , i.e.,  $\mathcal{X}_t = \phi^{-1}(B)\mathcal{Z}_t$ , and subjecting it to a  $\theta(B)$  filter to obtain output  $\mathcal{Y}_t = \theta(B)\mathcal{X}_t = \theta(B)\phi^{-1}(B)\mathcal{Z}_t$
- can express filtering of AR(1) process by

$$\mathcal{Y}_t = \begin{bmatrix} 1 & \theta \end{bmatrix} \begin{bmatrix} \mathcal{X}_t \\ \mathcal{X}_{t-1} \end{bmatrix},$$

which matches up with observation equation

$$\mathbf{Y}_t = G_t\mathbf{X}_t + \mathbf{W}_t$$

if  $\mathbf{Y}_t = \mathcal{Y}_t$ ,  $G_t = \begin{bmatrix} 1 & \theta \end{bmatrix}$ ,  $\mathbf{X}_t = \begin{bmatrix} \mathcal{X}_t \\ \mathcal{X}_{t-1} \end{bmatrix}$  &  $\mathbf{W}_t = 0$  (thus  $R_t = 0$ )

## ARMA(1,1) Process as a State-Space Model: II

- given  $\mathbf{X}_t = \begin{bmatrix} \mathcal{X}_t \\ \mathcal{X}_{t-1} \end{bmatrix}$ , can express  $\mathcal{X}_{t+1} = \phi\mathcal{X}_t + \mathcal{Z}_{t+1}$  in 1st row of matrix equation

$$\begin{bmatrix} \mathcal{X}_{t+1} \\ \mathcal{X}_t \end{bmatrix} = \begin{bmatrix} \phi & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \mathcal{X}_t \\ \mathcal{X}_{t-1} \end{bmatrix} + \begin{bmatrix} \mathcal{Z}_{t+1} \\ 0 \end{bmatrix},$$

which matches up with state-transition equation

$$\mathbf{X}_{t+1} = F_t \mathbf{X}_t + \mathbf{V}_t$$

if  $F_t = \begin{bmatrix} \phi & 0 \\ 1 & 0 \end{bmatrix}$  &  $\mathbf{V}_t = \begin{bmatrix} \mathcal{Z}_{t+1} \\ 0 \end{bmatrix}$  with  $Q_t \stackrel{\text{def}}{=} E\{\mathbf{V}_t \mathbf{V}_t'\} = \begin{bmatrix} \sigma^2 & 0 \\ 0 & 0 \end{bmatrix}$

- to complete model, let

$$\mathbf{X}_1 = \begin{bmatrix} \mathcal{X}_1 \\ \mathcal{X}_0 \end{bmatrix} = \begin{bmatrix} \sum_{j=0}^{\infty} \phi^j \mathcal{Z}_{1-j} \\ \sum_{j=0}^{\infty} \phi^j \mathcal{Z}_{-j} \end{bmatrix},$$

noting that  $\mathbf{X}_1$  &  $\mathbf{V}_t$  for  $t \geq 1$  are uncorrelated, as required

## ARMA(1,1) Process as a State-Space Model: III

- since

$$E\{\mathbf{X}_1\mathbf{X}'_1\} = \begin{bmatrix} \gamma(0) & \gamma(1) \\ \gamma(1) & \gamma(0) \end{bmatrix} = \frac{\sigma^2}{1-\phi^2} \begin{bmatrix} 1 & \phi \\ \phi & 1 \end{bmatrix},$$

can alternatively stipulate

$$\mathbf{X}_1 = \begin{bmatrix} 1 & \frac{\phi}{\sqrt{(1-\phi^2)}} \\ 0 & \frac{1}{\sqrt{(1-\phi^2)}} \end{bmatrix} \begin{bmatrix} \mathcal{Z}_1 \\ \mathcal{Z}_0 \end{bmatrix},$$

yielding

$$E\{\mathbf{X}_1\mathbf{X}'_1\} = \begin{bmatrix} 1 & \frac{\phi}{\sqrt{(1-\phi^2)}} \\ 0 & \frac{1}{\sqrt{(1-\phi^2)}} \end{bmatrix} \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{\phi}{\sqrt{(1-\phi^2)}} & \frac{1}{\sqrt{(1-\phi^2)}} \end{bmatrix} = \frac{\sigma^2}{1-\phi^2} \begin{bmatrix} 1 & \phi \\ \phi & 1 \end{bmatrix},$$

as required (recall that, if  $\mathbf{U}$  has covariance  $\Sigma$ , then  $A\mathbf{U}$  has covariance  $A\Sigma A'$ )

## ARMA(1,1) Process as a State-Space Model: IV

- stipulation

$$\mathbf{X}_1 = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} \mathcal{Z}_1 \\ \mathcal{Z}_0 \end{bmatrix}$$

with arbitrary  $a$ ,  $b$ ,  $c$  and  $d$  still leads to a valid state-space model, but one that need not correspond to elements of  $\mathbf{X}_t$  (and hence  $\mathbf{Y}_t$ ) being part of a stationary process (but asymptotically they will be so)

- as in case of AR(1) process, can formulate state-space model for ARMA(1,1) process + observational noise by replacing  $\mathbf{W}_t = 0$  and  $R_t = 0$  with  $W_t \sim \text{WN}(0, \sigma_W^2)$  and  $R_t = \sigma_W^2$ , where  $W_t$ 's are uncorrelated with  $Z_t$ 's

## Classical Decomposition Model Revisited (Again!)

- recall classical decomposition model for time series  $Y_t$ , namely,

$$Y_t = m_t + s_t + W_t,$$

where  $m_t$  is trend;  $s_t$  is periodic; and  $W_t$  is a stationary process

- $m_t$  &  $s_t$  can be treated as deterministic or stochastic
- to introduce basic ideas behind state-space methodology, will consider simple version of above known as a *local level* model in which  $m_t$  is stochastic and  $s_t = 0$ :

$$\begin{aligned} Y_t &= X_t + W_t, & \{W_t\} &\sim \text{WN}(0, \sigma_W^2) \\ X_{t+1} &= X_t + V_t, & \{V_t\} &\sim \text{WN}(0, \sigma_V^2), \end{aligned}$$

where  $E\{W_s V_t\} = 0$  for all  $s$  &  $t$

- déjà vu all over again: same model as used to motivate seasonal component in SARIMA models (overheads XV-40 & 41)

## Local Level Model: I

- comparing observation equation

$$\mathbf{Y}_t = G_t \mathbf{X}_t + \mathbf{W}_t, \quad \mathbf{W}_t \sim \text{UN}(\mathbf{0}, R_t)$$

with

$$Y_t = X_t + W_t, \quad \{W_t\} \sim \text{WN}(0, \sigma_W^2)$$

yields correspondences

$$\mathbf{Y}_t = Y_t, \quad G_t = 1, \quad \mathbf{X}_t = X_t, \quad \mathbf{W}_t = W_t \quad \text{and} \quad R_t = \sigma_W^2$$

- comparing state-transition equation

$$\mathbf{X}_{t+1} = F_t \mathbf{X}_t + \mathbf{V}_t, \quad \mathbf{V}_t \sim \text{UN}(\mathbf{0}, Q_t)$$

with

$$X_{t+1} = X_t + V_t, \quad \{V_t\} \sim \text{WN}(0, \sigma_V^2)$$

yields correspondences

$$F_t = 1, \quad \mathbf{V}_t = V_t \quad \text{and} \quad Q_t = \sigma_V^2$$

## Local Level Model: II

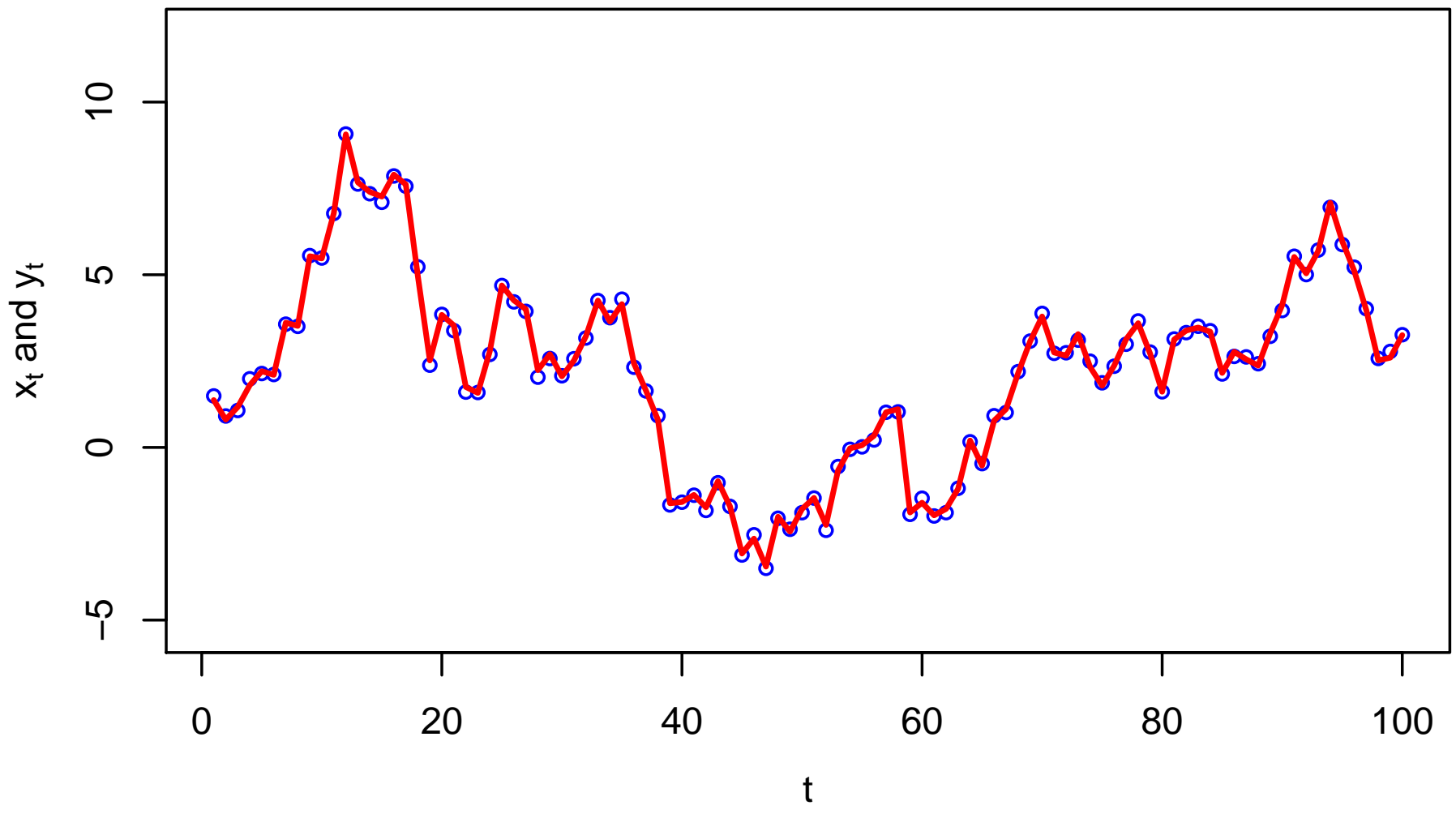
- to fully specify state-space model, need to define initial state  $\mathbf{X}_1 = X_1$  to be an RV that is uncorrected with  $W_t$ 's and  $V_t$ 's
- in addition, will assume  $E\{X_1\} = m_1$  and  $\text{var}\{X_1\} = P_1$
- model thus has 4 parameters:  $\sigma_W^2 = R_t$ ,  $\sigma_V^2 = Q_t$ ,  $m_1$  &  $P_1$
- since  $X_{t+1} = X_t + V_t$ , state variable  $X_t$  is a random walk starting from  $m_1$  (intended to model a slowly varying trend)
- since  $V_t$  and  $X_t$  are uncorrelated (why?),

$$E\{X_{t+1} | X_t\} = E\{X_t + V_t | X_t\} = X_t + E\{V_t\} = X_t;$$

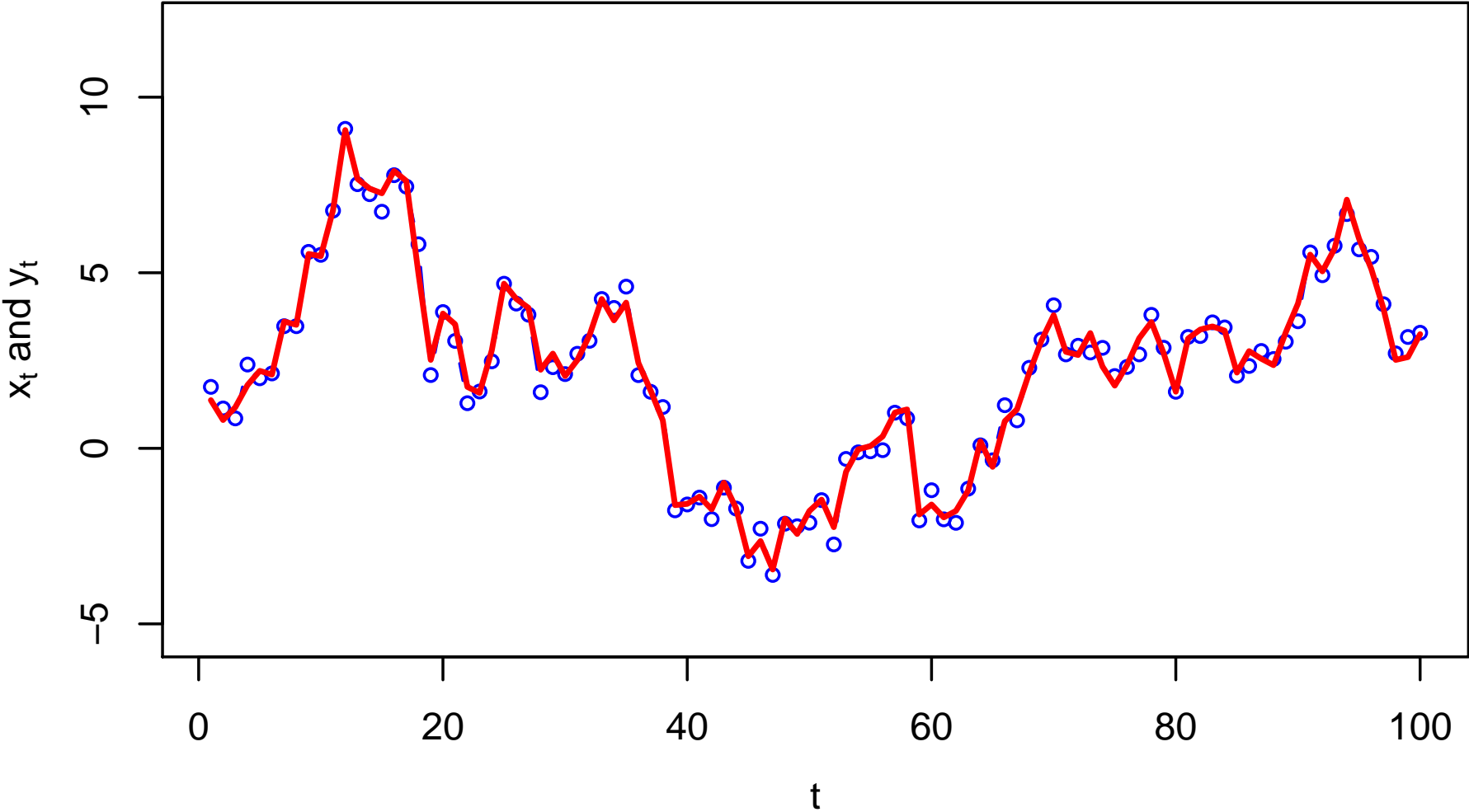
i.e., if state variable is at a certain 'level' at time  $t$ , we can expect no change in its level at time  $t + 1$

- when  $\sigma_W^2 > 0$ , trend is corrupted by noise, so ability to pick out trend depends upon 'signal to noise' (SNR) ratio  $\sigma_V^2 / \sigma_W^2$

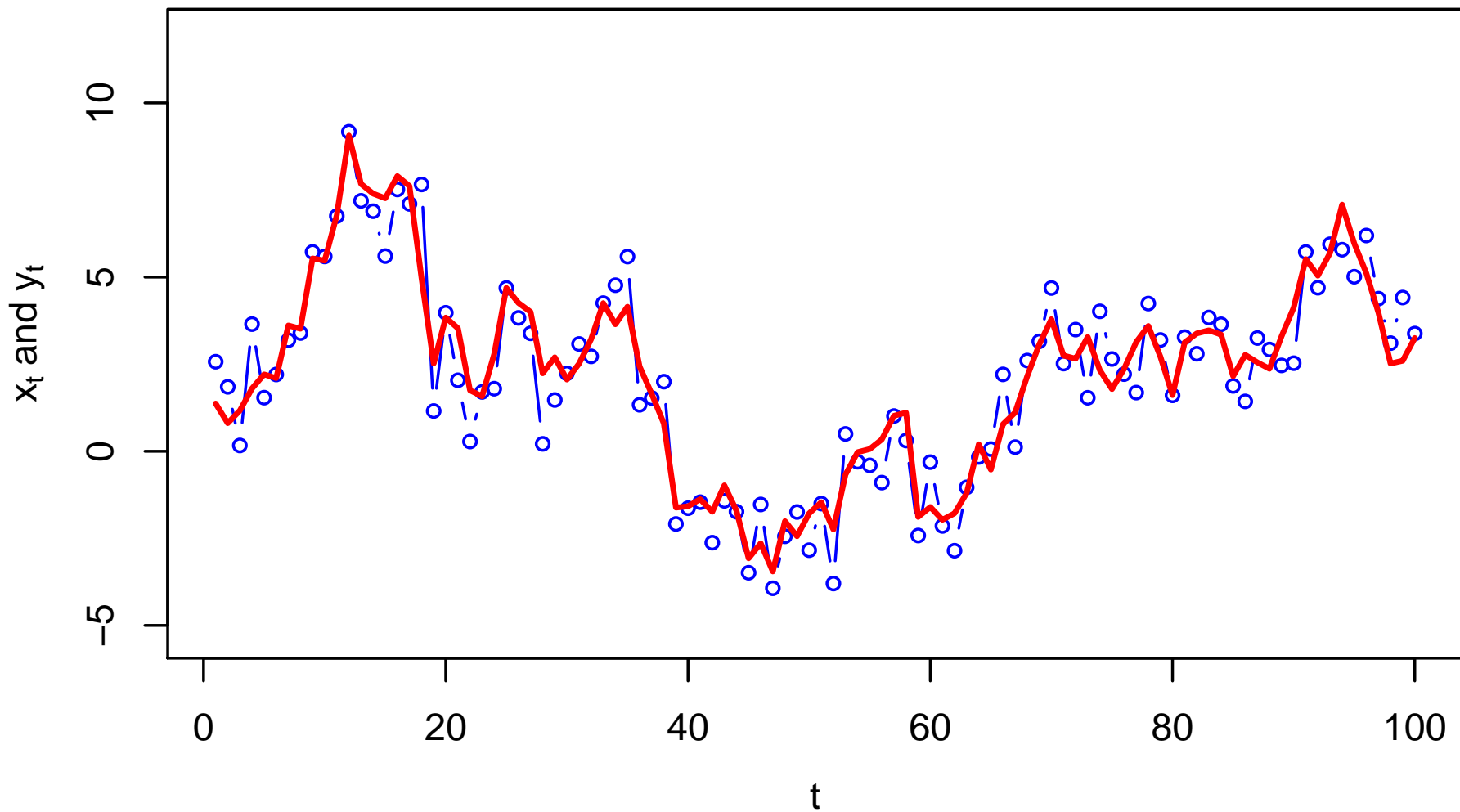
# Time Series $Y_t$ and Underlying State $X_t$ , SNR = 100



# Time Series $Y_t$ and Underlying State $X_t$ , SNR = 10



# Time Series $Y_t$ and Underlying State $X_t$ , SNR = 1



## Four Classical Problems in State-Space Models

- given observations  $Y_1, \dots, Y_t$  of a local level process,
  1. what is best predictor of state  $X_t$ ? (filtering)
  2. what is best predictor of state  $X_{t+1}$ ? (forecasting)
  3. what is best predictor of state  $X_s$  for  $s < t$ ? (smoothing)
  4. what are best estimates of model parameters? (estimation)
- note: if  $\sigma_W^2 = 0$  (no observation noise), filtering and smoothing are not of interest (unless some observations are missing)
- will concentrate first on filtering & forecasting problems with ‘best’ taken to be minimum mean square error (MSE)
- to facilitate discussion, will assume that  $X_1$ ,  $V_t$ ’s and  $W_t$  are multivariate normal (Gaussian)
  - implies that  $Y_t$  and remaining  $X_t$ ’s are also such

## Regression Lemma: I

- suppose random vectors  $\mathbf{X}$  &  $\mathbf{Y}$  are jointly normal with mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\Sigma$ , to be denoted by

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$$

- can partition both  $\boldsymbol{\mu}$  and  $\Sigma$ :

$$\begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu}_X \\ \boldsymbol{\mu}_Y \end{bmatrix}, \begin{bmatrix} \Sigma_{XX} & \Sigma_{XY} \\ \Sigma'_{XY} & \Sigma_{YY} \end{bmatrix}\right);$$

here  $\boldsymbol{\mu}_X$  &  $\Sigma_{XX}$  are mean vector & covariance matrix for  $\mathbf{X}$  and  $\boldsymbol{\mu}_Y$  &  $\Sigma_{YY}$  are those for  $\mathbf{Y}$ ; i.e.,

$$\mathbf{X} \sim \mathcal{N}(\boldsymbol{\mu}_X, \Sigma_{XX}) \quad \text{and} \quad \mathbf{Y} \sim \mathcal{N}(\boldsymbol{\mu}_Y, \Sigma_{YY});$$

$\Sigma_{XY}$  is cross-covariance matrix between  $\mathbf{X}$  &  $\mathbf{Y}$ ; i.e.,  $(i, j)$ th element is covariance between elements  $i$  &  $j$  of  $\mathbf{X}$  &  $\mathbf{Y}$

## Regression Lemma: II

- lemma: conditional distribution of  $\mathbf{X}$  given  $\mathbf{Y} = \mathbf{y}$  is multivariate normal with mean vector

$$\boldsymbol{\mu}_{\mathbf{X}|\mathbf{y}} = \boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}^{-1}(\mathbf{y} - \boldsymbol{\mu}_{\mathbf{Y}}) \quad (*)$$

and covariance matrix

$$\boldsymbol{\Sigma}_{\mathbf{X}|\mathbf{y}} = \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{X}} - \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}^{-1}\boldsymbol{\Sigma}'_{\mathbf{X}\mathbf{Y}}$$

- $\boldsymbol{\mu}_{\mathbf{X}|\mathbf{y}}$  is mean of  $\mathbf{X}$  given  $\mathbf{Y} = \mathbf{y}$
- replacing realization  $\mathbf{y}$  in right-hand side of (\*) with RV  $\mathbf{Y}$  yields  $E\{\mathbf{X} | \mathbf{Y}\}$ , the conditional mean of  $\mathbf{X}$  given  $\mathbf{Y}$ :

$$E\{\mathbf{X} | \mathbf{Y}\} = \boldsymbol{\mu}_{\mathbf{X}} + \boldsymbol{\Sigma}_{\mathbf{X}\mathbf{Y}}\boldsymbol{\Sigma}_{\mathbf{Y}\mathbf{Y}}^{-1}(\mathbf{Y} - \boldsymbol{\mu}_{\mathbf{Y}}) \quad (\dagger)$$

- best predictor (here linear) of  $\mathbf{X}$  given  $\mathbf{Y}$  is  $E\{\mathbf{X} | \mathbf{Y}\}$ , which (\dagger) makes clear is an RV constructed from  $\mathbf{Y}$
- can see from (\dagger) that  $E\{E\{\mathbf{X} | \mathbf{Y}\}\} = \boldsymbol{\mu}_{\mathbf{X}}$

## Regression Lemma: III

- recall that, if random vector  $\mathbf{u}$  has covariance matrix  $\Sigma_{\mathbf{u}}$ , then covariance matrix for  $A\mathbf{u}$  is  $A\Sigma_{\mathbf{u}}A'$ 
  - implies covariance matrix for  $\mathbf{c} + A(\mathbf{u} - \mu_{\mathbf{u}})$  is also  $A\Sigma_{\mathbf{u}}A'$  ( $\mathbf{c}$  and  $\mu_{\mathbf{u}}$  are vectors with constants)
- covariance matrix for

$$E\{\mathbf{X} \mid \mathbf{Y}\} = \mu_{\mathbf{X}} + \Sigma_{\mathbf{XY}}\Sigma_{\mathbf{YY}}^{-1}(\mathbf{Y} - \mu_{\mathbf{Y}})$$

is thus

$$\Sigma_{\mathbf{XY}}\Sigma_{\mathbf{YY}}^{-1}\Sigma_{\mathbf{YY}}\Sigma_{\mathbf{YY}}^{-1}\Sigma'_{\mathbf{XY}} = \Sigma_{\mathbf{XY}}\Sigma_{\mathbf{YY}}^{-1}\Sigma'_{\mathbf{XY}}$$

(note: *not* the same as  $\Sigma_{\mathbf{X}|\mathbf{y}} = \Sigma_{\mathbf{XX}} - \Sigma_{\mathbf{XY}}\Sigma_{\mathbf{YY}}^{-1}\Sigma'_{\mathbf{XY}}$ , the covariance matrix for  $\mathbf{X}$  given  $\mathbf{Y} = \mathbf{y}$ )

## Regression Lemma: IV

- consider prediction error  $\mathbf{U}$  associated with best linear predictor of  $\mathbf{X}$ :

$$\mathbf{U} = \mathbf{X} - E\{\mathbf{X} \mid \mathbf{Y}\}$$

- since  $E\{E\{\mathbf{X} \mid \mathbf{Y}\}\} = \boldsymbol{\mu}_X$ , follows that  $E\{\mathbf{U}\} = \mathbf{0}$
- covariance matrix for  $\mathbf{U}$  is given by

$$\begin{aligned} E\{\mathbf{U}\mathbf{U}'\} &= E\{(\mathbf{X} - E\{\mathbf{X} \mid \mathbf{Y}\})(\mathbf{X} - E\{\mathbf{X} \mid \mathbf{Y}\})'\} \\ &= E\{\mathbf{X}\mathbf{X}'\} + E\{E\{\mathbf{X} \mid \mathbf{Y}\}E\{\mathbf{X} \mid \mathbf{Y}\}'\} \\ &\quad - E\{\mathbf{X}E\{\mathbf{X} \mid \mathbf{Y}\}'\} - E\{E\{\mathbf{X} \mid \mathbf{Y}\}\mathbf{X}'\} \\ &= \Sigma_{\mathbf{X}\mathbf{X}} - \Sigma_{\mathbf{X}\mathbf{Y}}\Sigma_{\mathbf{Y}\mathbf{Y}}^{-1}\Sigma'_{\mathbf{X}\mathbf{Y}}, \end{aligned}$$

which is equal to  $\Sigma_{\mathbf{X}|\mathbf{y}}$ , the conditional covariance matrix (see next overhead for details)

## Some Details

- without loss of generality, can assume  $\boldsymbol{\mu}_X = \mathbf{0}$  &  $\boldsymbol{\mu}_Y = \mathbf{0}$
- by definition,  $E\{\mathbf{X}\mathbf{X}'\} = \Sigma_{\mathbf{X}\mathbf{X}}$
- since  $E\{\mathbf{X} \mid \mathbf{Y}\} = \Sigma_{\mathbf{X}\mathbf{Y}}\Sigma_{\mathbf{Y}\mathbf{Y}}^{-1}\mathbf{Y}$ ,

$$\begin{aligned} E\{E\{\mathbf{X} \mid \mathbf{Y}\}E\{\mathbf{X} \mid \mathbf{Y}\}'\} &= E\{\Sigma_{\mathbf{X}\mathbf{Y}}\Sigma_{\mathbf{Y}\mathbf{Y}}^{-1}\mathbf{Y}\mathbf{Y}'\Sigma_{\mathbf{Y}\mathbf{Y}}^{-1}\Sigma'_{\mathbf{X}\mathbf{Y}}\} \\ &= \Sigma_{\mathbf{X}\mathbf{Y}}\Sigma_{\mathbf{Y}\mathbf{Y}}^{-1}\Sigma'_{\mathbf{X}\mathbf{Y}} \end{aligned}$$

$$\begin{aligned} E\{\mathbf{X}E\{\mathbf{X} \mid \mathbf{Y}\}'\} &= E\{\mathbf{X}\mathbf{Y}'\Sigma_{\mathbf{Y}\mathbf{Y}}^{-1}\Sigma'_{\mathbf{X}\mathbf{Y}}\} \\ &= \Sigma_{\mathbf{X}\mathbf{Y}}\Sigma_{\mathbf{Y}\mathbf{Y}}^{-1}\Sigma'_{\mathbf{X}\mathbf{Y}} \end{aligned}$$

$$\begin{aligned} E\{E\{\mathbf{X} \mid \mathbf{Y}\}\mathbf{X}'\} &= E\{\Sigma_{\mathbf{X}\mathbf{Y}}\Sigma_{\mathbf{Y}\mathbf{Y}}^{-1}\mathbf{Y}\mathbf{X}'\} \\ &= \Sigma_{\mathbf{X}\mathbf{Y}}\Sigma_{\mathbf{Y}\mathbf{Y}}^{-1}\Sigma'_{\mathbf{X}\mathbf{Y}}, \end{aligned}$$

the last line being because  $E\{\mathbf{Y}\mathbf{X}'\} = \Sigma_{\mathbf{Y}\mathbf{X}} = \Sigma'_{\mathbf{X}\mathbf{Y}}$

- putting all the pieces together yields desired result

## Regression Corollary

- specialize now to case where  $\mathbf{X}$  has just one element, say,  $X$
- corollary: conditional distribution of  $X$  given  $\mathbf{Y} = \mathbf{y}$  is normal with mean

$$\mu_X + \boldsymbol{\Sigma}'_{XY} \boldsymbol{\Sigma}_{YY}^{-1} (\mathbf{y} - \boldsymbol{\mu}_Y)$$

and variance

$$\Sigma_{X|\mathbf{y}} = \sigma_X^2 - \boldsymbol{\Sigma}'_{XY} \boldsymbol{\Sigma}_{YY}^{-1} \boldsymbol{\Sigma}_{XY},$$

where  $\sigma_X^2 = \text{var}\{X\}$  and  $\boldsymbol{\Sigma}_{XY}$  is a column vector containing covariance between  $X$  and RVs in  $\mathbf{Y}$

- since conditional variance is same as MSE for  $X$ , will refer to  $\Sigma_{X|\mathbf{y}}$  as MSE

## Aside – Revisiting an Old Friend: I

- suppose  $\{X_t\}$  is zero mean stationary process with ACVF  $\gamma(h)$
- set  $X$  to  $X_{n+1}$  and put  $X_1, \dots, X_n$  into  $\mathbf{Y}$
- corollary says best linear predictor  $\hat{X}_{n+1}$  of  $X_{n+1}$  given  $X_n, \dots, X_1$  is

$$\hat{X}_{n+1} = \Sigma'_{XY} \Sigma_{YY}^{-1} \mathbf{Y} = \boldsymbol{\gamma}'_n \Gamma_n^{-1} \mathbf{Y} \stackrel{\text{def}}{=} \boldsymbol{\phi}'_n \mathbf{Y},$$

where

- $\boldsymbol{\gamma}_n = [\gamma(1), \gamma(2), \dots, \gamma(n)]' = \Sigma_{XY}$
- $(i, j)$ th entry of matrix  $\Gamma_n = \Sigma_{YY}$  is  $\gamma(i - j)$
- $\boldsymbol{\phi}'_n \stackrel{\text{def}}{=} \boldsymbol{\gamma}'_n \Gamma_n^{-1}$  and hence  $\boldsymbol{\phi}_n = \Gamma_n^{-1} \boldsymbol{\gamma}_n$
- $\boldsymbol{\phi}_n$  solves  $\Gamma_n \boldsymbol{\phi}_n = \boldsymbol{\gamma}_n$ , a result previously seen (overhead XI-2)

## Aside – Revisiting an Old Friend: II

- overhead XI-3 states that MSE for  $\hat{X}_{n+1}$  is

$$\begin{aligned}v_n &= \text{var} \{X_{n+1}\} - \phi'_n \gamma_n \\&= \sigma_X^2 - \gamma'_n \Gamma_n^{-1} \gamma_n \quad (\text{making use of } \phi'_n \stackrel{\text{def}}{=} \gamma'_n \Gamma_n^{-1}) \\&= \sigma_X^2 - \Sigma'_{XY} \Sigma_{YY}^{-1} \Sigma_{XY} \\&= \Sigma_{X|y}\end{aligned}$$

- thus ‘old friend’ is a special case of regression corollary

## Filtering for Local Level Model: I

- return now to local level model:

$$\begin{aligned} Y_t &= X_t + W_t, \quad \{W_t\} \sim \text{WN}(0, \sigma_W^2) \\ X_{t+1} &= X_t + V_t, \quad \{V_t\} \sim \text{WN}(0, \sigma_V^2), \end{aligned}$$

and  $\mathbf{X}_1 = X_1$  is an RV that

- is uncorrected with  $W_t$ 's and  $V_t$ 's
- has  $E\{X_1\} = m_1$  and  $\text{var}\{X_1\} = P_1$
- filtering problem is to predict unknown state  $X_t$  based on data up to time  $t$ , i.e.,  $Y_1, \dots, Y_t$
- in what follows, let  $\mathbf{Y}_t \stackrel{\text{def}}{=} [Y_1, \dots, Y_t]'$

## Filtering for Local Level Model: II

- best linear predictor of  $X_t$  given  $\mathbf{Y}_t$  is

$$\hat{X}_{t|t} \stackrel{\text{def}}{=} E\{X_t \mid \mathbf{Y}_t\} = m_t + \boldsymbol{\Sigma}'_{t,t} \boldsymbol{\Sigma}_t^{-1} (\mathbf{Y}_t - \mathbf{m}_t), \text{ where}$$

- $m_t = E\{X_t\}$
- vector  $\boldsymbol{\Sigma}_{t,t}$  contains covariances between  $X_t$  &  $Y_1, \dots, Y_t$
- $(j, k)$ th element of matrix  $\boldsymbol{\Sigma}_t$  is covariance between  $Y_j$  &  $Y_k$
- $\mathbf{m}_t$  is a vector containing, for  $j = 1, \dots, t$ ,

$$m_j \stackrel{\text{def}}{=} E\{X_j\} = E\{X_j + W_j\} = E\{Y_j\}$$

- note:  $E\{\hat{X}_{t|t}\} = E\{E\{X_t \mid \mathbf{Y}_t\}\} = E\{X_t\} = m_t$

- with  $P_t \stackrel{\text{def}}{=} \text{var}\{X_t\}$ , MSE for predictor is

$$E\{(X_t - \hat{X}_{t|t})^2\} = P_t - \boldsymbol{\Sigma}'_{t,t} \boldsymbol{\Sigma}_t^{-1} \boldsymbol{\Sigma}_{t,t} \stackrel{\text{def}}{=} P_{t|t} \text{ (see overhead XVIII-26)}$$

## Forecasting for Local Level Model: I

- best linear predictor of  $X_{t+1}$  given  $\mathbf{Y}_t$  is

$$\hat{X}_{t+1|t} \stackrel{\text{def}}{=} E\{X_{t+1} \mid \mathbf{Y}_t\} = m_{t+1} + \boldsymbol{\Sigma}'_{t+1,t} \boldsymbol{\Sigma}_t^{-1} (\mathbf{Y}_t - \mathbf{m}_t),$$

where vector  $\boldsymbol{\Sigma}_{t+1,t}$  has covariances between  $X_{t+1}$  &  $\mathbf{Y}_t$

- note:  $E\{\hat{X}_{t+1|t}\} = E\{E\{X_{t+1} \mid \mathbf{Y}_t\}\} = E\{X_{t+1}\} = m_{t+1}$
- MSE for predictor is

$$E\{(X_{t+1} - \hat{X}_{t+1|t})^2\} = P_{t+1} - \boldsymbol{\Sigma}'_{t+1,t} \boldsymbol{\Sigma}_t^{-1} \boldsymbol{\Sigma}_{t+1,t} \stackrel{\text{def}}{=} P_{t+1|t}$$

## Forecasting for Local Level Model: II

- let's also consider best linear predictor of  $Y_{t+1}$  given  $\mathbf{Y}_t$ :

$$\hat{Y}_{t+1|t} \stackrel{\text{def}}{=} E\{Y_{t+1} \mid \mathbf{Y}_t\} = m_{t+1} + \tilde{\Sigma}'_{t+1,t} \Sigma_t^{-1} (\mathbf{Y}_t - \mathbf{m}_t),$$

where vector  $\tilde{\Sigma}_{t+1,t}$  has covariances between  $Y_{t+1}$  &  $\mathbf{Y}_t$

- however, note that, for  $j = 1, \dots, t$

$$\text{cov}\{Y_{t+1}, Y_j\} = \text{cov}\{X_{t+1} + W_{t+1}, Y_j\} = \text{cov}\{X_{t+1}, Y_j\}$$

- thus  $\tilde{\Sigma}_{t+1,t} = \Sigma_{t+1,t}$  (since latter has covariances between  $X_{t+1}$  &  $\mathbf{Y}_t$ ), yielding

$$\hat{Y}_{t+1|t} = m_{t+1} + \Sigma'_{t+1,t} \Sigma_t^{-1} (\mathbf{Y}_t - \mathbf{m}_t) = \hat{X}_{t+1|t},$$

an 'obvious' result: difference between  $Y_{t+1}$  and  $X_{t+1}$  is  $W_{t+1}$ , for which best predictor at time  $t$  is zero!

## Forecasting for Local Level Model: III

- although  $\hat{Y}_{t+1|t}$  &  $\hat{X}_{t+1|t}$  are identical, their MSEs differ:

$$\begin{aligned} E\{(Y_{t+1} - \hat{Y}_{t+1|t})^2\} &= E\{(X_{t+1} + W_{t+1} - \hat{X}_{t+1|t})^2\} \\ &= E\{(X_{t+1} - \hat{X}_{t+1|t})^2\} + \sigma_W^2 \\ &= P_{t+1|t} + \sigma_W^2 \end{aligned}$$

## Filtering for Local Level Model: III

- to implement filtering (i.e., compute  $\hat{X}_{t|t}$ ), need to determine
  1.  $m_j = E\{X_j\}$ ,  $j = 1, \dots, t$
  2. elements of  $\Sigma_{t,t}$ , i.e., covariances between  $X_t$  &  $Y_1, \dots, Y_t$
  3. elements of  $\Sigma_t$ , i.e., covariances between  $Y_j$  &  $Y_k$ ,  $1 \leq j \leq k \leq t$
- to compute  $P_{t|t}$ , i.e., MSE for  $\hat{X}_{t|t}$ , need  $P_t = \text{var}\{X_t\}$  in addition to 2 and 3
- since  $X_t = X_{t-1} + V_{t-1}$  and  $Y_t = X_t + W_t$ , telescoping yields
$$X_j = X_1 + \sum_{l=1}^{j-1} V_l \quad \text{and} \quad Y_j = X_1 + \sum_{l=1}^{j-1} V_l + W_j, \quad j = 1, \dots, t$$
(above holds for  $j = 1$  if we interpret  $\sum_{l=1}^0 V_l$  as zero)

## Filtering for Local Level Model: IV

- using

$$X_j = X_1 + \sum_{l=1}^{j-1} V_l \text{ and } Y_j = X_1 + \sum_{l=1}^{j-1} V_l + W_j, \quad j = 1, \dots, t,$$

get  $m_j = E\{X_j\} = E\{X_1\} = m_1$  and (assuming  $j \leq k \leq t$ )

$$\begin{aligned} \text{cov}\{X_t, Y_j\} &= \text{cov}\left\{X_1 + \sum_{l=1}^{t-1} V_l, X_1 + \sum_{l=1}^{j-1} V_l + W_j\right\} \\ &= P_1 + (j-1)\sigma_V^2 \end{aligned}$$

$$\begin{aligned} \text{cov}\{Y_j, Y_k\} &= \text{cov}\left\{X_1 + \sum_{l=1}^{j-1} V_l + W_j, X_1 + \sum_{l=1}^{k-1} V_l + W_k\right\} \\ &= P_1 + (j-1)\sigma_V^2 + \delta_{j,k}\sigma_W^2, \end{aligned}$$

where  $\delta_{j,k} = 1$  if  $j = k$  and  $\delta_{j,k} = 0$  if  $j \neq k$

## Filtering for Local Level Model: V

- using

$$X_t = X_1 + \sum_{l=1}^{t-1} V_l,$$

get

$$P_t = \text{var} \{X_t\} = P_1 + (t - 1)\sigma_V^2$$

- now have all the pieces needed to form  $\hat{X}_{t|t}$  and its MSE  $P_{t|t}$
- note: similar argument leads to pieces needed to form forecast  $\hat{X}_{t+1|t}$  and its MSE  $P_{t+1|t}$

## Kalman Recursions for Filtering/Forecasting: I

- critique: while straightforward conceptually, forming

$$\hat{X}_{t|t} = m_t + \Sigma'_{t,t} \Sigma_t^{-1} (\mathbf{Y}_t - \mathbf{m}_t)$$

and

$$\hat{X}_{t+1|t} = m_{t+1} + \Sigma'_{t+1,t} \Sigma_t^{-1} (\mathbf{Y}_t - \mathbf{m}_t)$$

via these equations requires inversion of matrix  $\Sigma_t$  whose dimension  $t \times t$  can become problematic as  $t$  gets large

- celebrated *Kalman recursions* give a recipe that avoids explicit matrix inversion
- mindset is that, at time  $t - 1$ , we have 4 quantities of interest:
  - filtered value  $\hat{X}_{t-1|t-1}$  and forecast  $\hat{X}_{t|t-1}$
  - associated MSEs  $P_{t-1|t-1}$  and  $P_{t|t-1}$
- note:  $\hat{X}_{t-1|t-1} = \hat{X}_{t|t-1}$  for local level model (but not others)

## Kalman Recursions for Filtering/Forecasting: II

- at time  $t$ , new observation  $Y_t$  becomes available
- Kalman recursion takes  $\hat{X}_{t|t-1}$ ,  $P_{t|t-1}$  &  $Y_t$  and yields
  - filtered value  $\hat{X}_{t|t}$  and forecast  $\hat{X}_{t+1|t}$
  - associated MSEs  $P_{t|t}$  and  $P_{t+1|t}$
- there are six steps in the Kalman recursion:
  - steps 1 & 2 are preparatory
  - steps 3 & 4 yield  $\hat{X}_{t|t}$  and  $P_{t|t}$  (filtering)
  - steps 5 & 6 yield  $\hat{X}_{t+1|t}$  and  $P_{t+1|t}$  (forecasting)

## Kalman Recursions for Filtering/Forecasting: III

1. compute innovation:

$$U_t = Y_t - \hat{Y}_{t|t-1} = Y_t - \hat{X}_{t|t-1}$$

2. compute MSE for  $\hat{Y}_{t|t-1}$  (i.e.,  $\text{var}\{U_t\}$  since  $E\{U_t\} = 0$ ):

$$P_{t|t-1} + \sigma_W^2 \stackrel{\text{def}}{=} F_t$$

3. compute new filtered value:

$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + K_t U_t,$$

where  $K_t \stackrel{\text{def}}{=} P_{t|t-1}/F_t$  is the so-called *Kalman gain*

4. compute MSE for new filtered value:

$$P_{t|t} = P_{t|t-1}(1 - K_t)$$

## Kalman Recursions for Filtering/Forecasting: IV

5. compute new forecast:

$$\hat{X}_{t+1|t} = \hat{X}_{t|t-1} + K_t U_t = \hat{X}_{t|t}$$

6. compute MSE for new forecast:

$$P_{t+1|t} = P_{t|t-1}(1 - K_t) + \sigma_V^2 = P_{t|t} + \sigma_V^2$$

(makes sense in view of  $X_{t+1} = X_t + V_t$ : since predictor  $\hat{X}_{t|t}$  for  $X_t$  has MSE  $P_{t|t}$ , MSE for predictor  $\hat{X}_{t+1|t}$  for  $X_{t+1}$  must be  $P_{t|t} + \sigma_V^2$ )

- recursions are carried out for  $t = 1, \dots, n$ , with inputs at  $t = 1$  being  $\hat{X}_{1|0} \stackrel{\text{def}}{=} m_1 = E\{X_1\}$ ,  $P_{1|0} \stackrel{\text{def}}{=} P_1 = \text{var}\{X_1\}$  and  $Y_1$

## Another Regression Corollary

- can prove that Kalman recursion works as advertised using regression lemma specialized to case where both  $\mathbf{X}$  and  $\mathbf{Y}$  have a single element
- corollary: conditional distribution of  $X$  given  $Y$  is normal with mean

$$E\{X | Y\} = \mu_X + \frac{\sigma_{XY}}{\sigma_Y^2} (Y - \mu_Y)$$

and conditional variance

$$\sigma_{X|Y}^2 = \sigma_X^2 - \frac{\sigma_{XY}^2}{\sigma_Y^2},$$

where  $\mu_X = E\{X\}$ ,  $\mu_Y = E\{Y\}$ ,  $\sigma_X^2 = \text{var}\{X\}$ ,  $\sigma_Y^2 = \text{var}\{Y\}$  and  $\sigma_{XY} = \text{cov}\{X, Y\}$

## Kalman Recursions for Filtering/Forecasting: V

- to prove validity of steps 3 and 4, need to show that

- $\hat{X}_{t|t-1} + K_t U_t$  is equal to  $\hat{X}_{t|t}$

- $P_{t|t-1}(1 - K_t)$  is equal to  $P_{t|t}$

- in corollary, let  $X$  be  $X_t$  with  $\mathbf{Y}_{t-1}$  held fixed so that

$$\mu_X = E\{X_t \mid \mathbf{Y}_{t-1}\} = \hat{X}_{t|t-1} \quad \text{and} \quad \sigma_X^2 = \text{var}\{X_t \mid \mathbf{Y}_{t-1}\} = P_{t|t-1}$$

- let  $Y$  be  $U_t$  so  $\mu_Y = E\{U_t\} = 0$ ,  $\sigma_Y^2 = \text{var}\{U_t\} = F_t$  and

$$\begin{aligned} \sigma_{XY} &= \text{cov}\{X_t, U_t \mid \mathbf{Y}_{t-1}\} \\ &= \text{cov}\{X_t, Y_t - \hat{Y}_{t|t-1} \mid \mathbf{Y}_{t-1}\} \\ &= \text{cov}\{X_t, X_t + W_t \mid \mathbf{Y}_{t-1}\} \\ &= \text{var}\{X_t \mid \mathbf{Y}_{t-1}\} = P_{t|t-1} \end{aligned}$$

## Kalman Recursions for Filtering/Forecasting: VI

- key fact:  $X_t$  conditioned on both  $U_t = Y_t - \hat{Y}_{t|t-1}$  and  $\mathbf{Y}_{t-1}$  is *same* as  $X_t$  conditioned on  $\mathbf{Y}_t$  ( $\hat{Y}_{t|t-1}$  is set if  $\mathbf{Y}_{t-1}$  is known)
- plugging all the pieces into

$$E\{X | Y\} = \mu_X + \frac{\sigma_{XY}}{\sigma_Y^2} (Y - \mu_Y) \quad \text{and} \quad \sigma_{X|Y}^2 = \sigma_X^2 - \frac{\sigma_{XY}^2}{\sigma_Y^2}$$

yields

$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + \frac{P_{t|t-1}}{F_t} U_t \quad \text{and} \quad P_{t|t} = P_{t|t-1} - \frac{P_{t|t-1}^2}{F_t};$$

i.e., since  $K_t \stackrel{\text{def}}{=} P_{t|t-1}/F_t$ , we get required

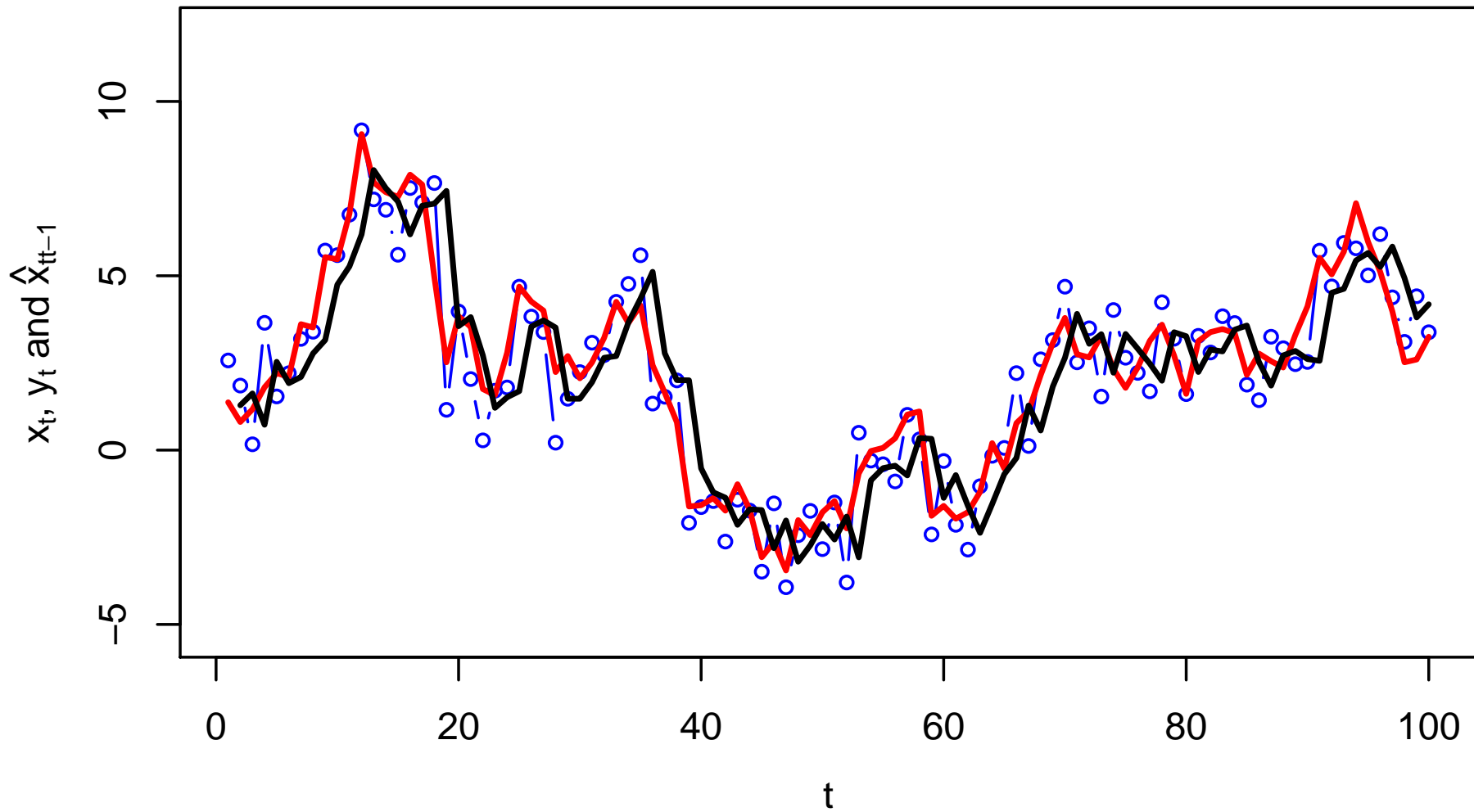
$$\hat{X}_{t|t} = \hat{X}_{t|t-1} + K_t U_t \quad \text{and} \quad P_{t|t} = P_{t|t-1}(1 - K_t)$$

- note: proof of steps 5 and 6 of Kalman recursion is analogous

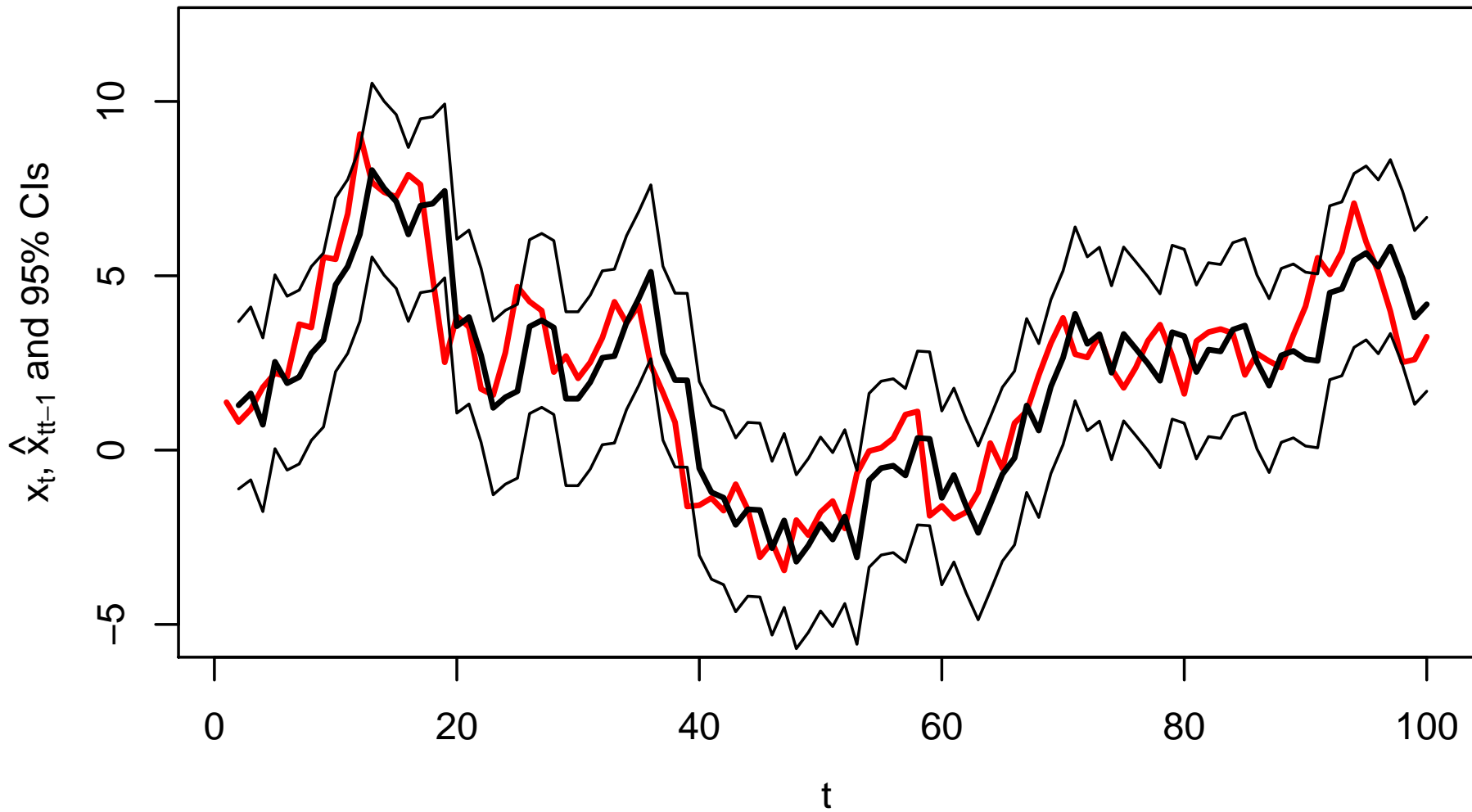
## Kalman Recursions for Filtering/Forecasting: VII

- proof of Kalman recursion for local level model adapted from Durbin & Koopman (2012)
- proofs for general state-space models are in Brockwell & Davis and Shumway & Stoffer textbooks and also in
  - Diderrich (1985)
  - Eubank (2006)
  - Meinhold & Singpurwalla (1983)
  - Petris, Petrone & Campagnoli (2009)
- as an example, let's apply Kalman filter to time series from simulated local level model with  $\text{SNR} = 1$

# Time Series $Y_t$ , States $X_t$ and Forecasts $\hat{X}_{t|t-1}$



States  $X_t$ , Forecasts  $\hat{X}_{t|t-1}$  & 95% CIs Based on  $P_{t|t-1}$



## Kalman Recursions for Filtering/Forecasting: VIII

- for a time series of length  $n$ , last use of Kalman recursions gives forecast  $\hat{X}_{n+1|n}$  for future state  $X_{n+1}$ , along with error variance  $P_{n+1|n}$  useful for constructing 95% confidence interval:

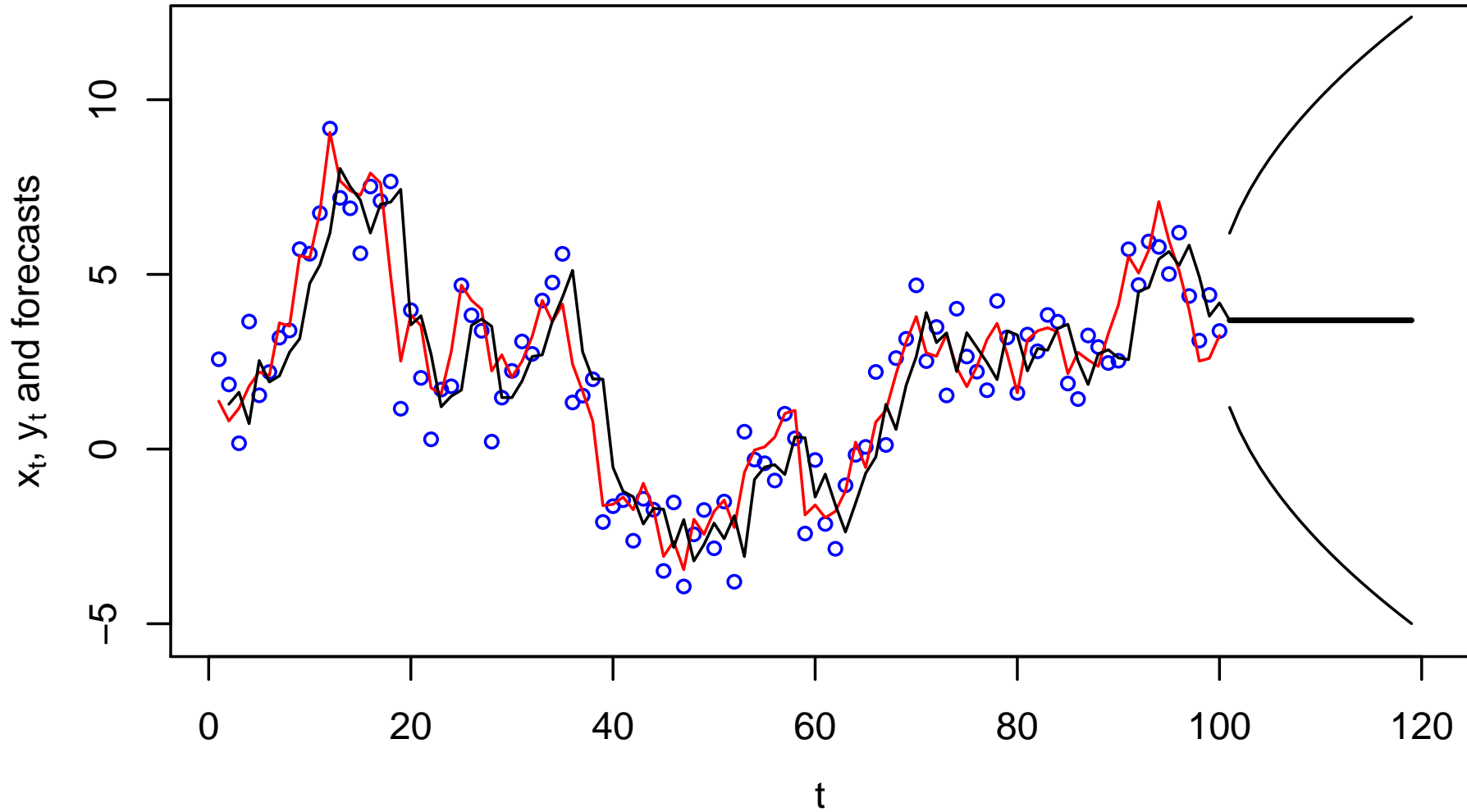
$$\left[ \hat{X}_{n+1|n} - 1.96\sqrt{P_{n+1|n}}, \hat{X}_{n+1|n} + 1.96\sqrt{P_{n+1|n}} \right]$$

- to forecast  $X_{n+2}$ , enter  $\hat{X}_{n+1|n}$  and  $P_{n+1|n}$  into *modified recursion* that differs from usual six-step recursion as follows:
  1. skip computation of innovation ( $Y_{n+1}$  is unavailable)
  2. do as usual
  3. do as usual, but with  $K_t U_t$  set to 0
  4. do as usual, but with Kalman gain  $K_t$  set to 0
  5. do as usual, but with  $K_t U_t$  set to 0
  6. do as usual, but with  $K_t$  set to 0

## Kalman Recursions for Filtering/Forecasting: IX

- in short: proceed as usual, but set Kalman gain  $K_t$  to 0
- recursion for  $t = n + 1$  yields  $\hat{X}_{n+2|n}$  and associated error variance  $P_{n+2|n}$
- to forecast  $X_{n+3}$ , enter  $\hat{X}_{n+2|n}$  and  $P_{n+2|n}$  into modified recursion to obtain  $\hat{X}_{n+3|n}$  and  $P_{n+3|n}$
- repeat as needed to get forecasts as far into future as desired
- as an example, let's forecast  $X_{101}, \dots, X_{120}$  for simulated local level model with  $\text{SNR} = 1$

# Time Series $Y_t$ , States $X_t$ , Forecasts and 95% CIs



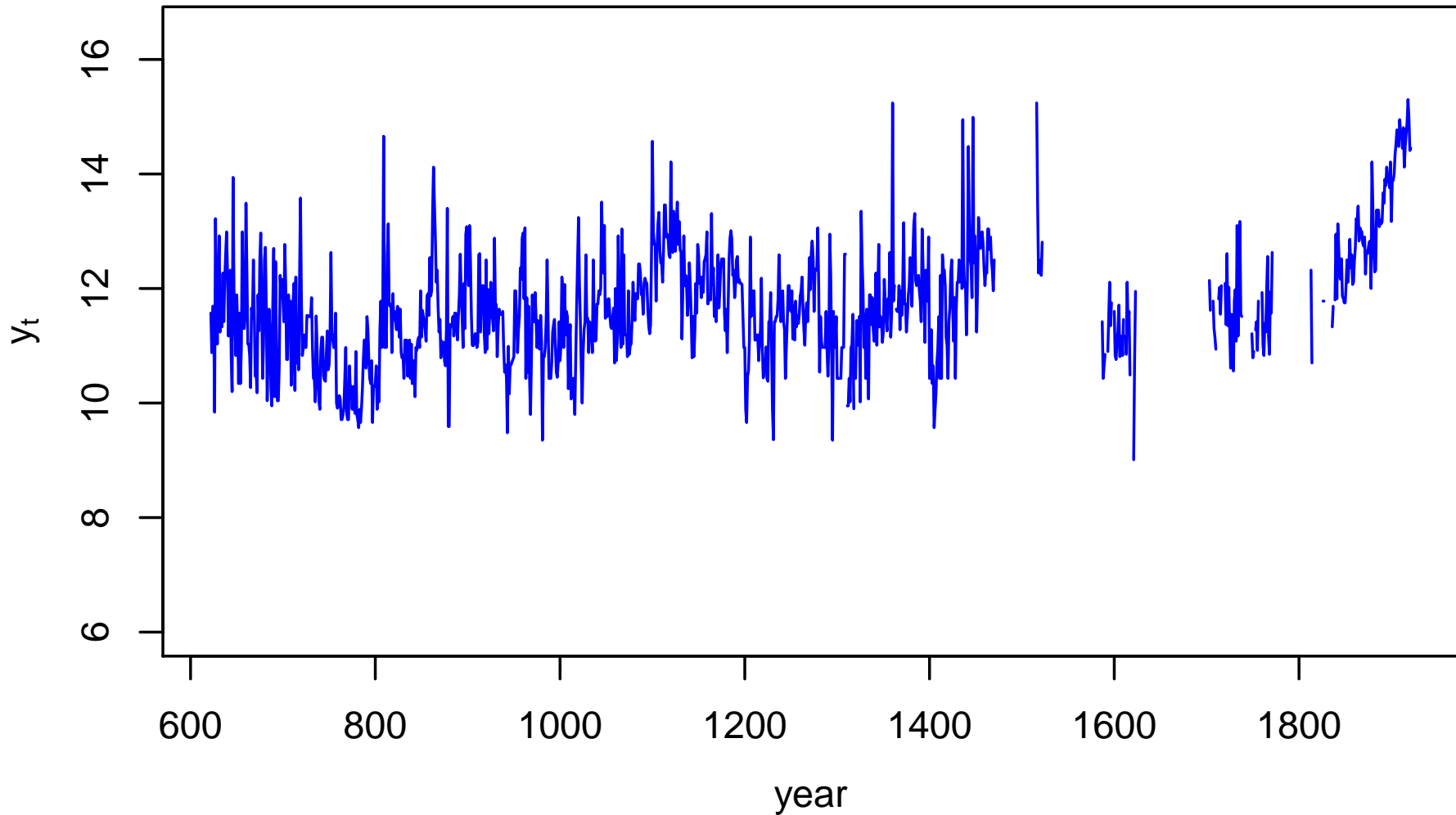
## Kalman Recursions for Gappy Time Series: I

- one of the strengths of state-space formulation is ease with which it can handle time series with missing values (gaps)
- suppose  $Y_1, \dots, Y_t$  &  $Y_{t+3}$  are observed, but not  $Y_{t+1}$  &  $Y_{t+2}$
- use modified recursion
  - with  $\hat{X}_{t+1|t}$  &  $P_{t+1|t}$  to get  $\hat{X}_{t+2|t}$  &  $P_{t+2|t}$
  - with  $\hat{X}_{t+2|t}$  &  $P_{t+2|t}$  to get  $\hat{X}_{t+3|t}$  &  $P_{t+3|t}$
- enter  $\hat{X}_{t+3|t}$ ,  $P_{t+3|t}$  and  $Y_{t+3}$  into usual recursion to obtain  $\hat{X}_{t+3|t+3}$  &  $\hat{X}_{t+4|t+3}$  along with  $P_{t+3|t+3}$  &  $P_{t+4|t+3}$
- need to interpret ‘given  $t + 3$ ’ as conditioning on everything available at time  $t + 3$ , i.e.,  $Y_1, \dots, Y_t$  and  $Y_{t+3}$

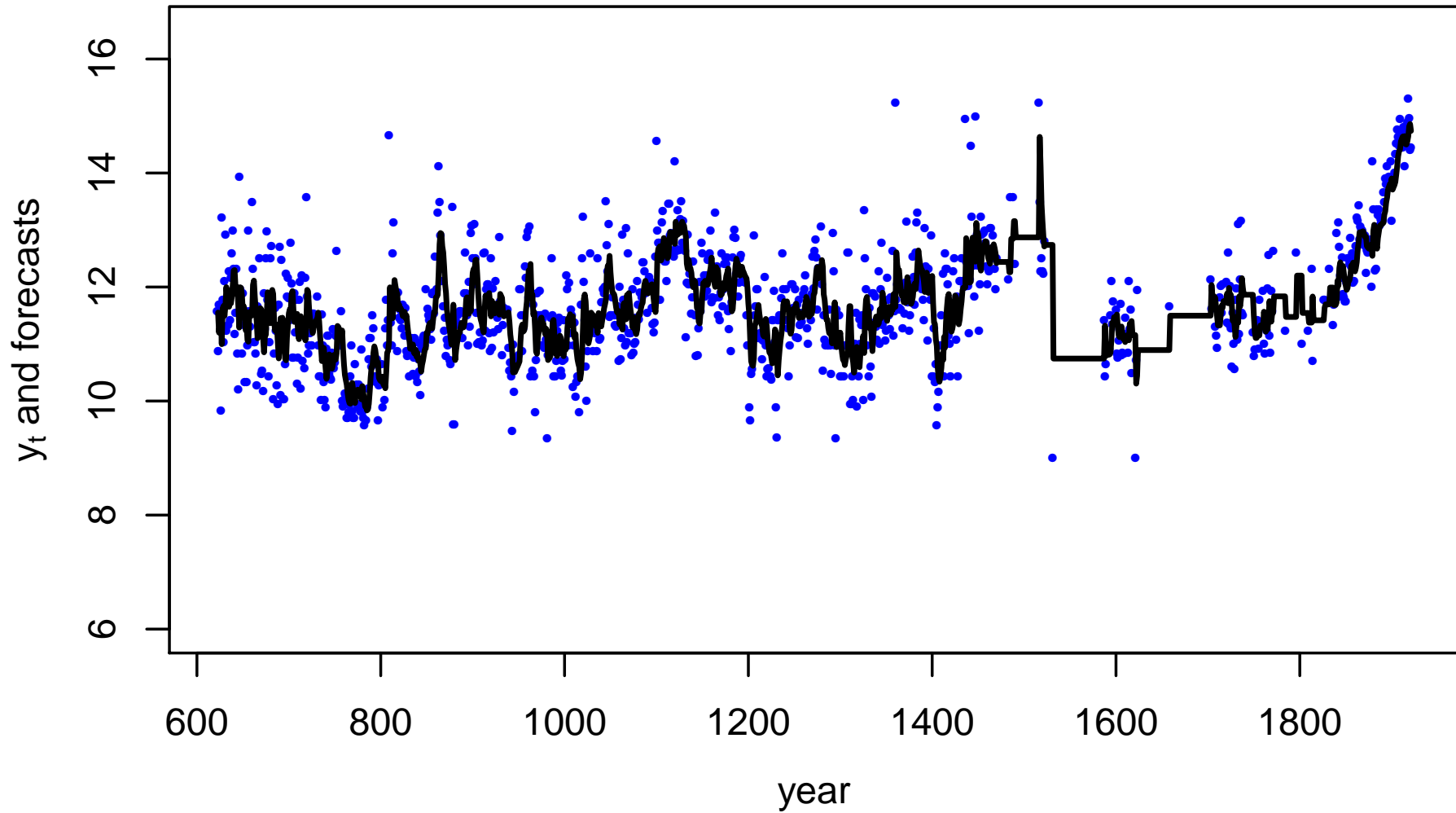
## Kalman Recursions for Gappy Time Series: II

- repeat as needed to handle other gaps in time series
- as an example, let's reconsider Nile River minima

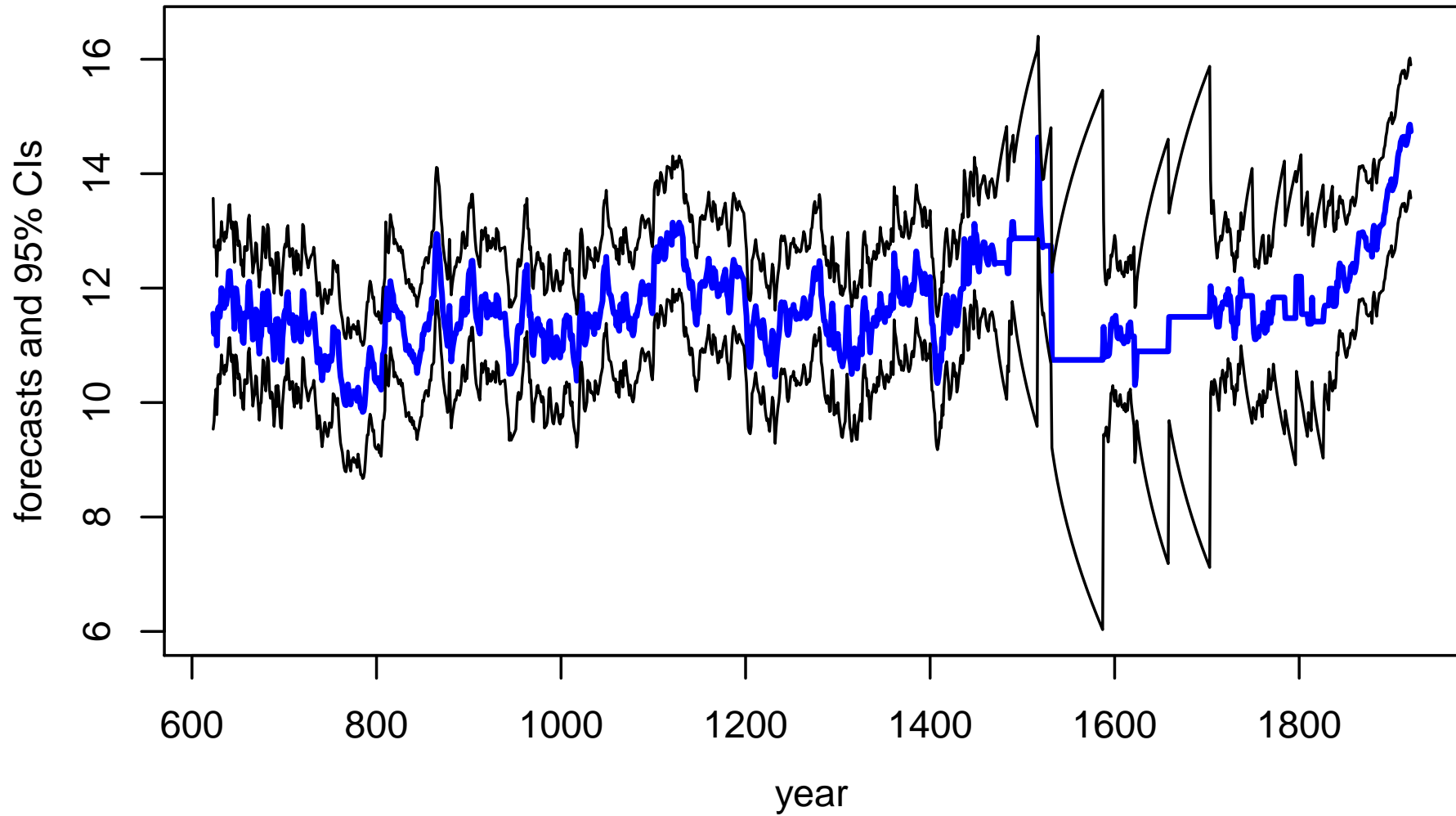
# Nile River Minima Series, 622 to 1921



# Nile River Series and Forecasts



## Forecasts for Nile River Series and 95% CIs



## Kalman Recursions for Smoothing: I

- given time series  $Y_1, \dots, Y_n$ , Kalman filter recursions give us  $\hat{X}_{t|t}$  for  $t = 1, \dots, n$
- predictor  $\hat{X}_{t|t}$  of unknown state variable  $X_t$  only makes use of  $Y_1, \dots, Y_t$  (appropriate mindset for ‘real-time’ applications)
- makes sense that best linear predictor of  $X_t$  given  $Y_1, \dots, Y_n$  might significantly outperform  $\hat{X}_{t|t}$  (particularly for small  $t$ )
- regression lemma says solution to *smoothing problem* is

$$\hat{X}_{t|n} \stackrel{\text{def}}{=} E\{X_t \mid \mathbf{Y}_n\} = m_t + \boldsymbol{\Sigma}'_{t,n} \boldsymbol{\Sigma}_n^{-1} (\mathbf{Y}_n - \mathbf{m}_n),$$

where vector  $\boldsymbol{\Sigma}_{t,n}$  has covariances between  $X_t$  &  $\mathbf{Y}_n$

- MSE for predictor, i.e.,  $E\{(X_t - \hat{X}_{t|n})^2\}$ , is

$$P_t - \boldsymbol{\Sigma}'_{t,n} \boldsymbol{\Sigma}_n^{-1} \boldsymbol{\Sigma}_{t,n} \stackrel{\text{def}}{=} P_{t|n}, \quad \text{where } P_t \stackrel{\text{def}}{=} \text{var}\{X_t\}$$

## Kalman Recursions for Smoothing: II

- using innovations  $U_t$ , innovation variances  $F_t$ , Kalman gains  $K_t$ , forecasts  $\hat{X}_{t|t-1}$  and associated MSEs  $P_{t|t-1}$ ,  $t = 1, \dots, n$  (all computed by Kalman filter recursions), *Kalman smoother recursions* allow efficient computation of  $\hat{X}_{t|n}$ ,  $t = 1, \dots, n$
  - there are four steps in the Kalman recursion
  - first two steps yield desired predictor  $\hat{X}_{t|n}$
1. manipulate innovations: starting with  $r_n = 0$ , recursively form

$$r_{t-1} = \frac{U_t}{F_t} + (1 - K_t)r_t, \quad t = n, \dots, 1$$

2. combine manipulated innovations and forecasts:

$$\hat{X}_{t|n} = \hat{X}_{t|t-1} + P_{t|t-1}r_{t-1}, \quad t = 1, \dots, n$$

## Kalman Recursions for Smoothing: III

- next two steps yield MSE for predictor  $\hat{X}_{t|n}$
- 3. manipulate innovation variances: starting with  $N_n = 0$ , recursively form

$$N_{t-1} = \frac{1}{F_t} + (1 - K_t)^2 N_t, \quad t = n, \dots, 1$$

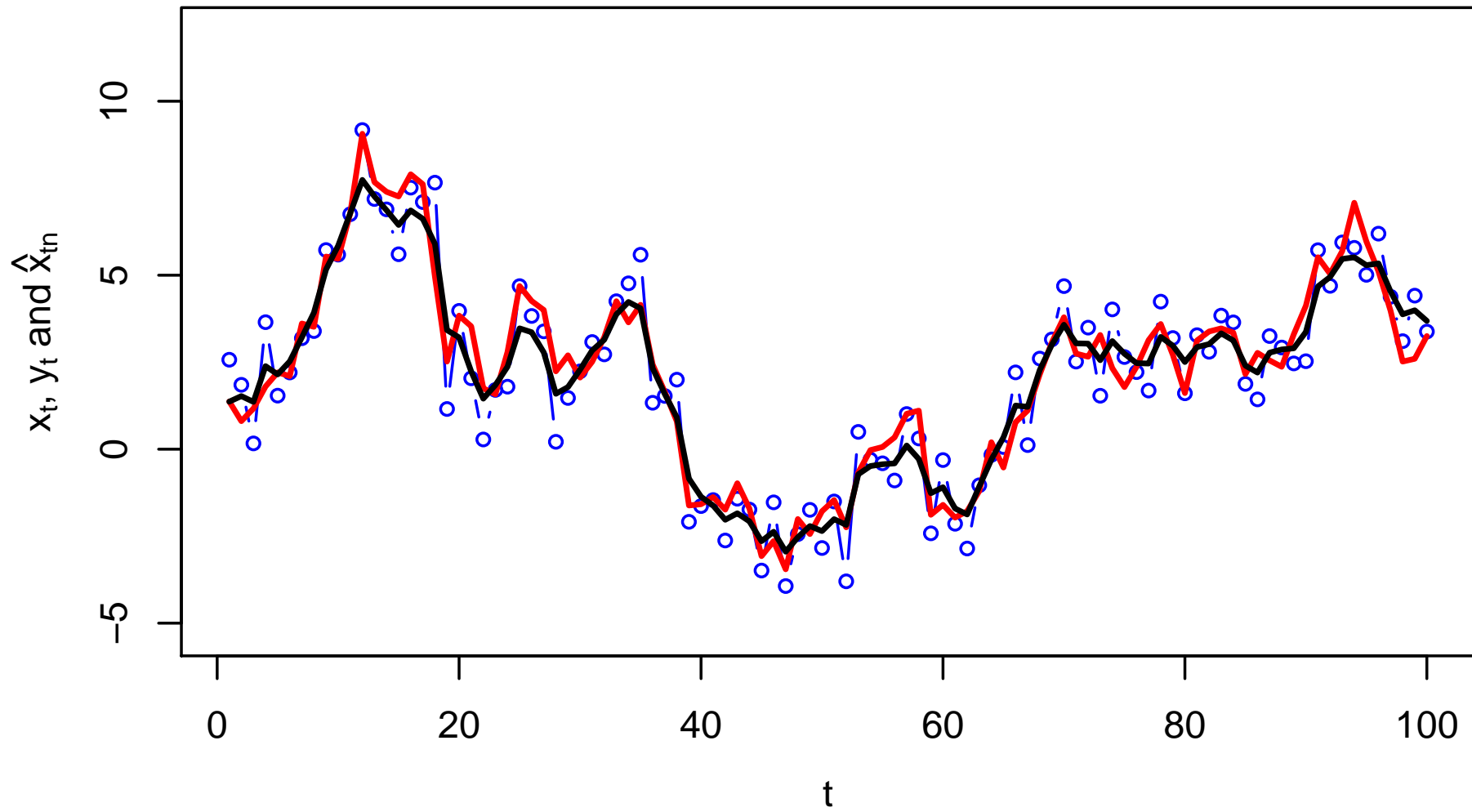
- 4. combine manipulated innovation variances and forecast MSEs:

$$V_t = P_{t|t-1} - P_{t|t-1}^2 N_{t-1}, \quad t = 1, \dots, n,$$

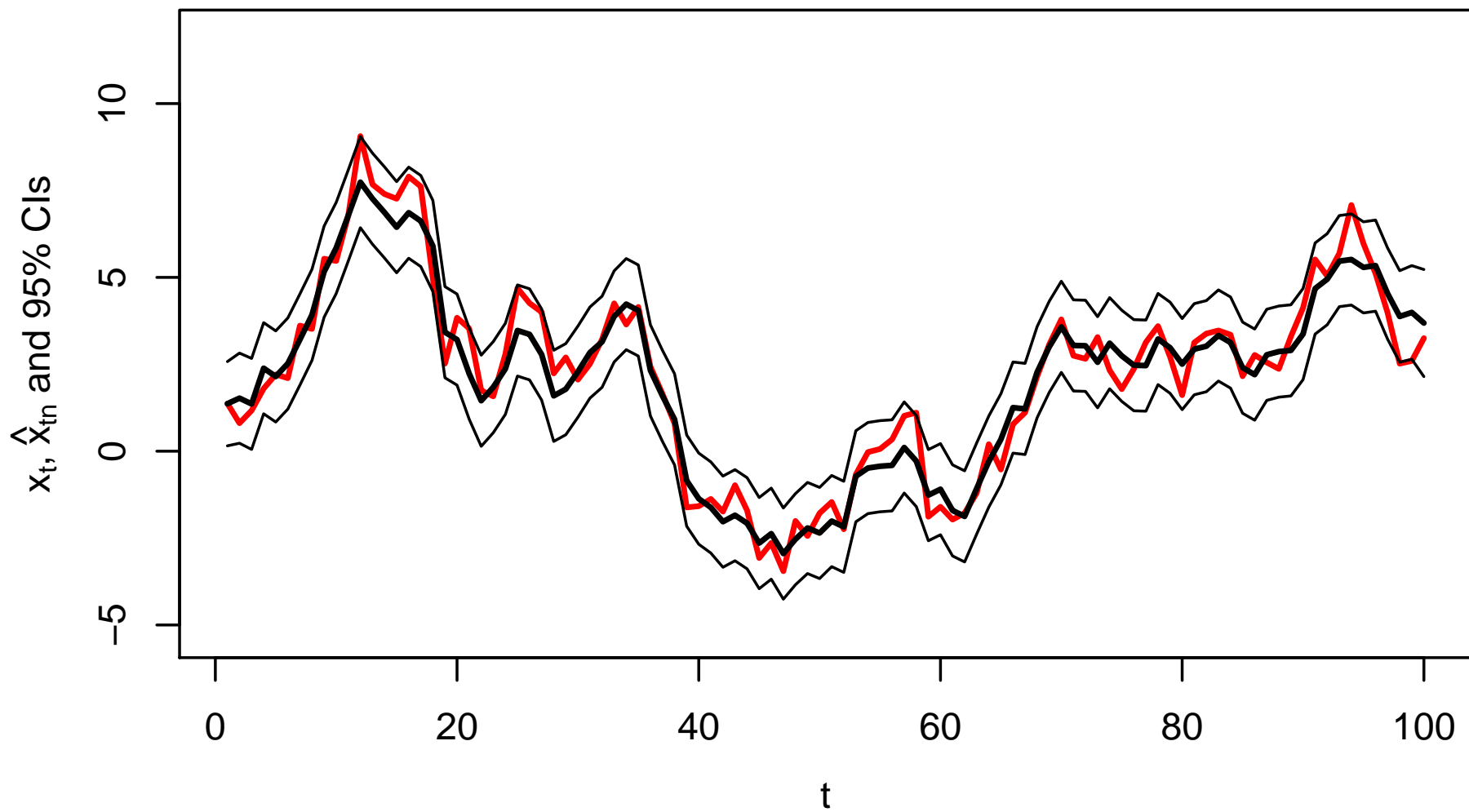
where  $V_t$  is desired MSE

- as an example, let's apply Kalman smoother to time series from simulated local level model with  $\text{SNR} = 1$

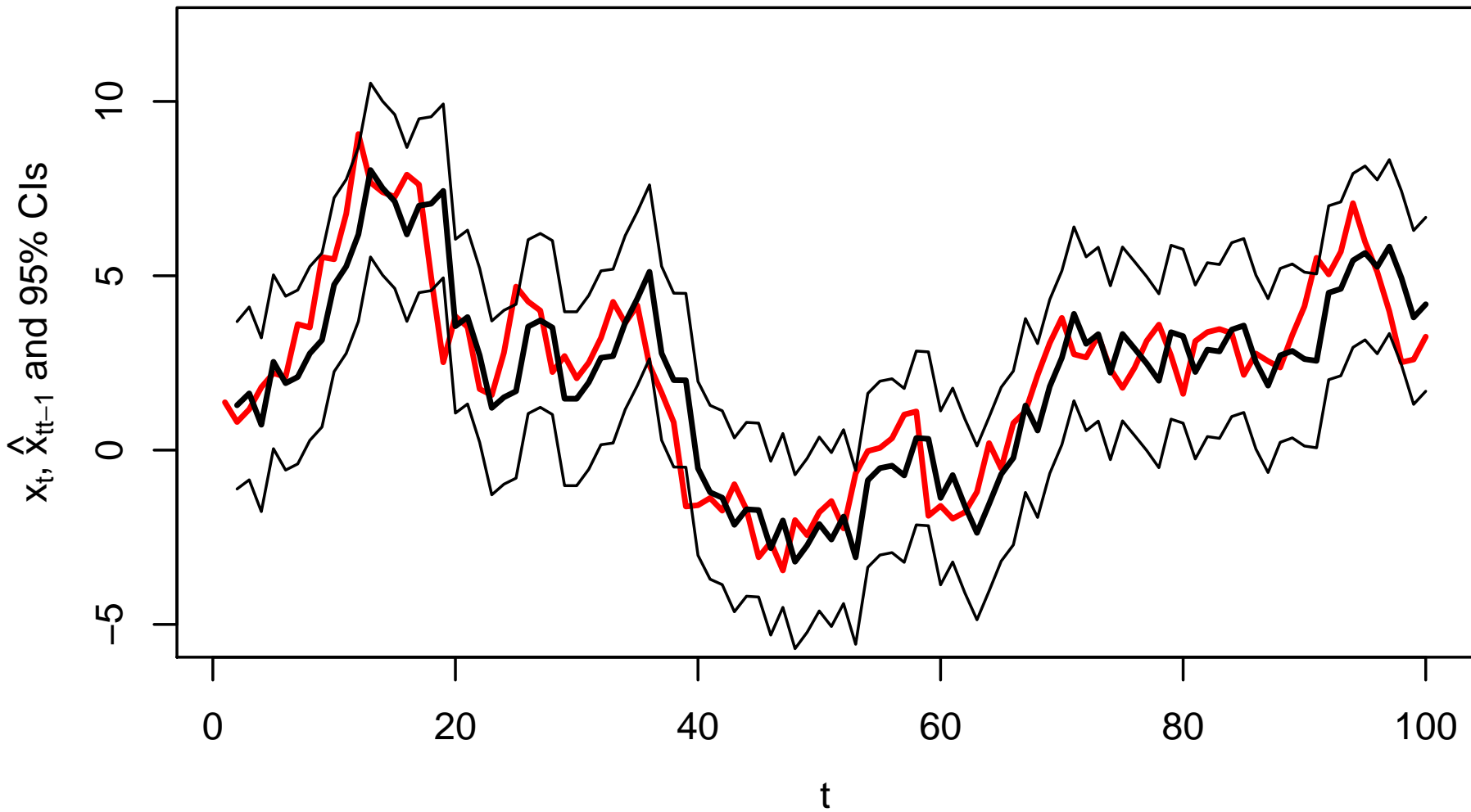
# Time Series $Y_t$ , States $X_t$ and Smooths $\hat{X}_{t|n}$



States  $X_t$ , Smoother  $\hat{X}_{t|n}$  and 95% CIs



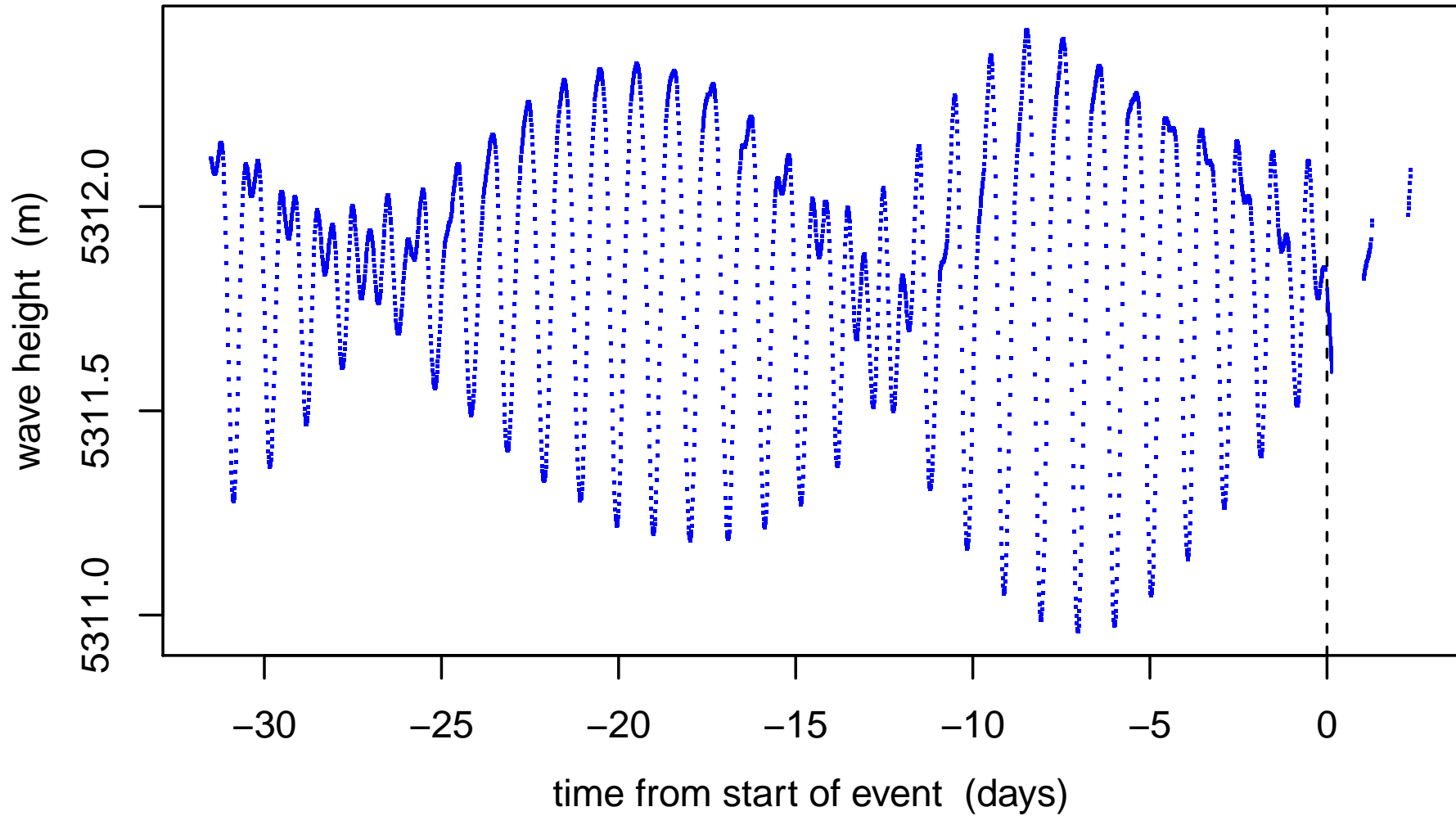
States  $X_t$ , Forecasts  $\hat{X}_{t|t-1}$  and 95% CIs



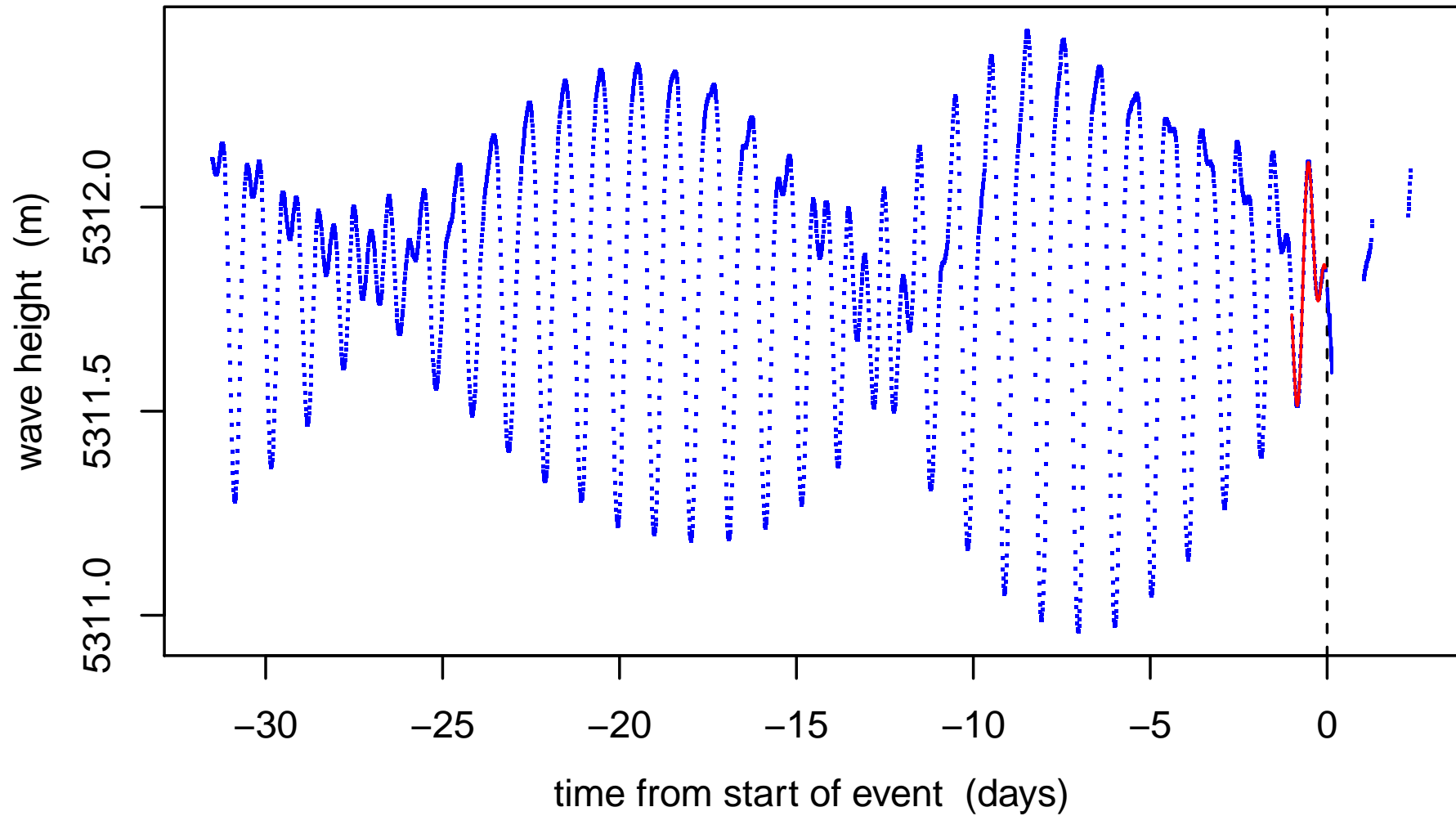
## Detiding DART™ Buoy Data

- though simple, local level model useful in practical applications
- as example, consider use of model to remove tidal component from data recorded by DART™ buoy 21414 prior to and after passage of tsunami wave in November, 2006 (generated by an earthquake near Kuril Islands)

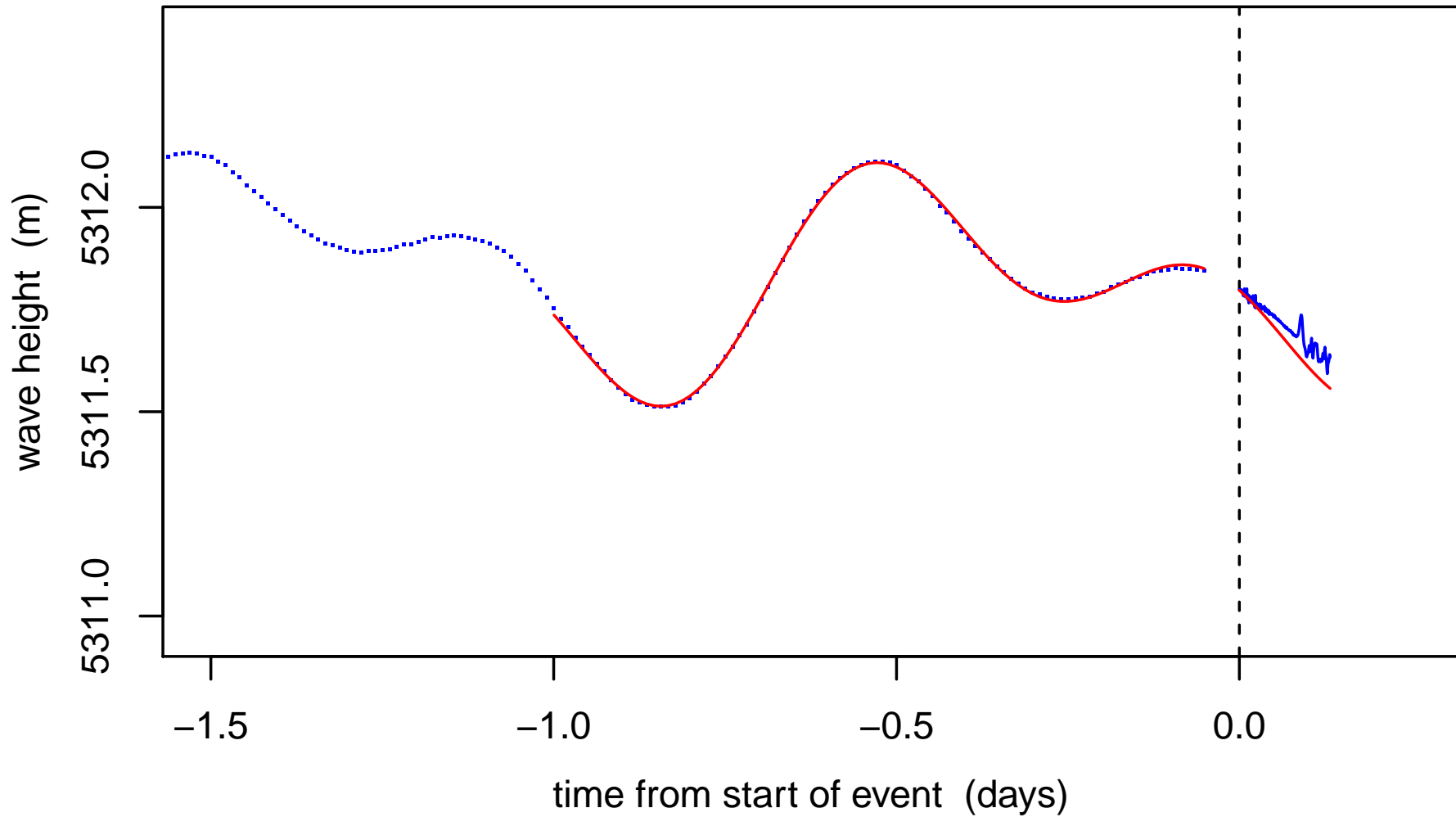
# Buoy 21414 Data for Kuril Islands Event



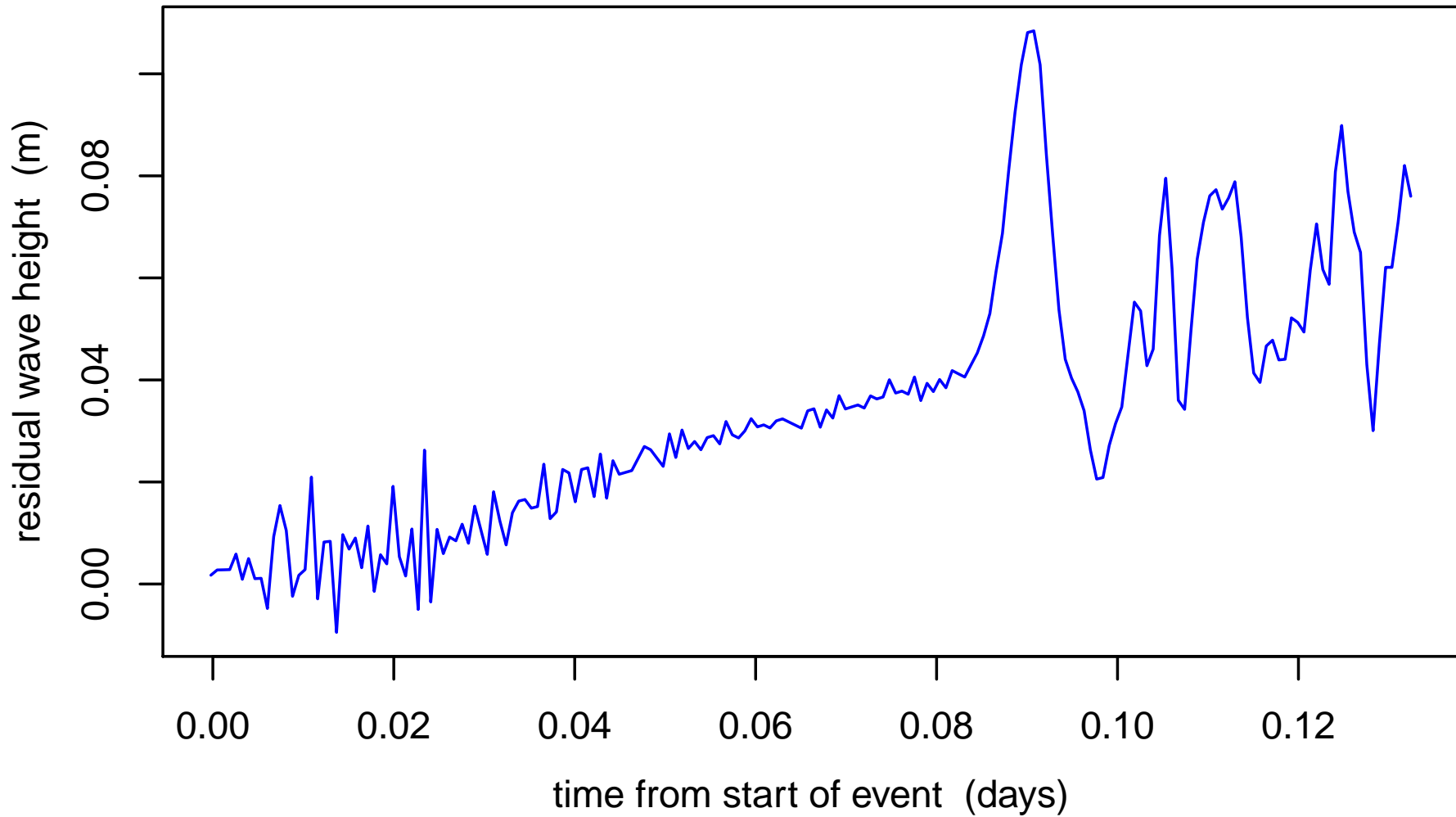
# Model Based on One Day Harmonic Regression



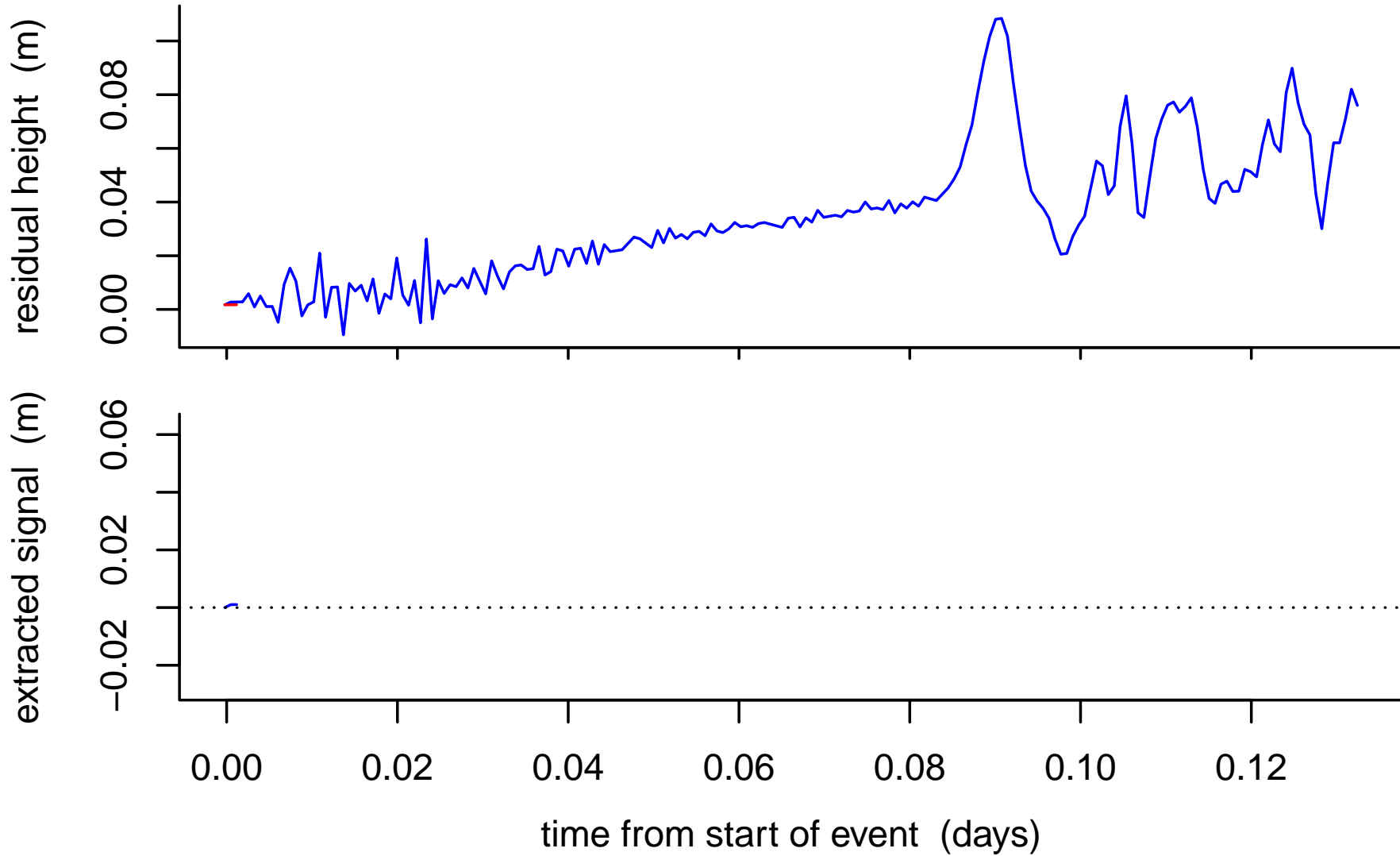
# Forecasts from Harmonic Regression Model



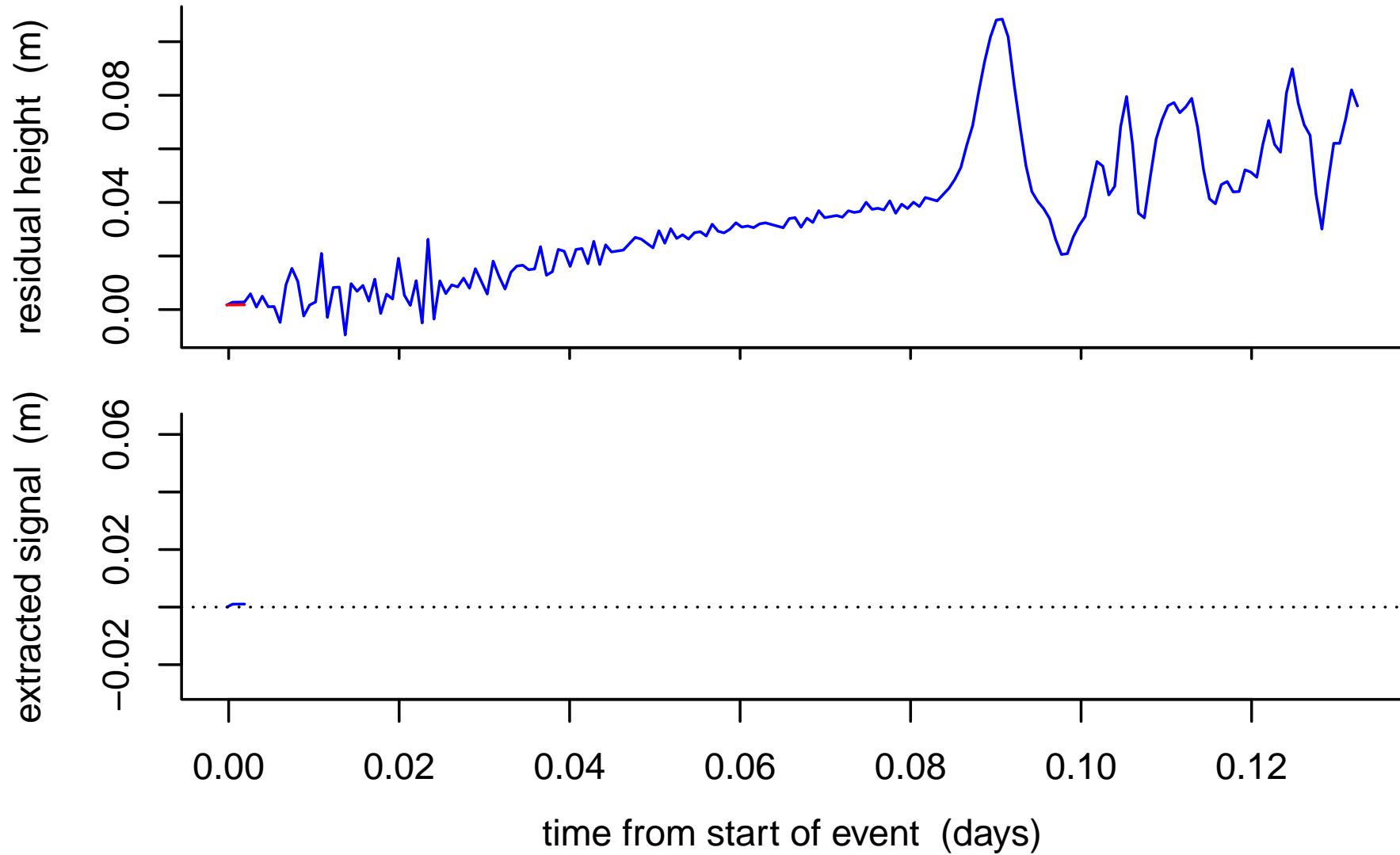
# Prediction Errors from Harmonic Regression Model



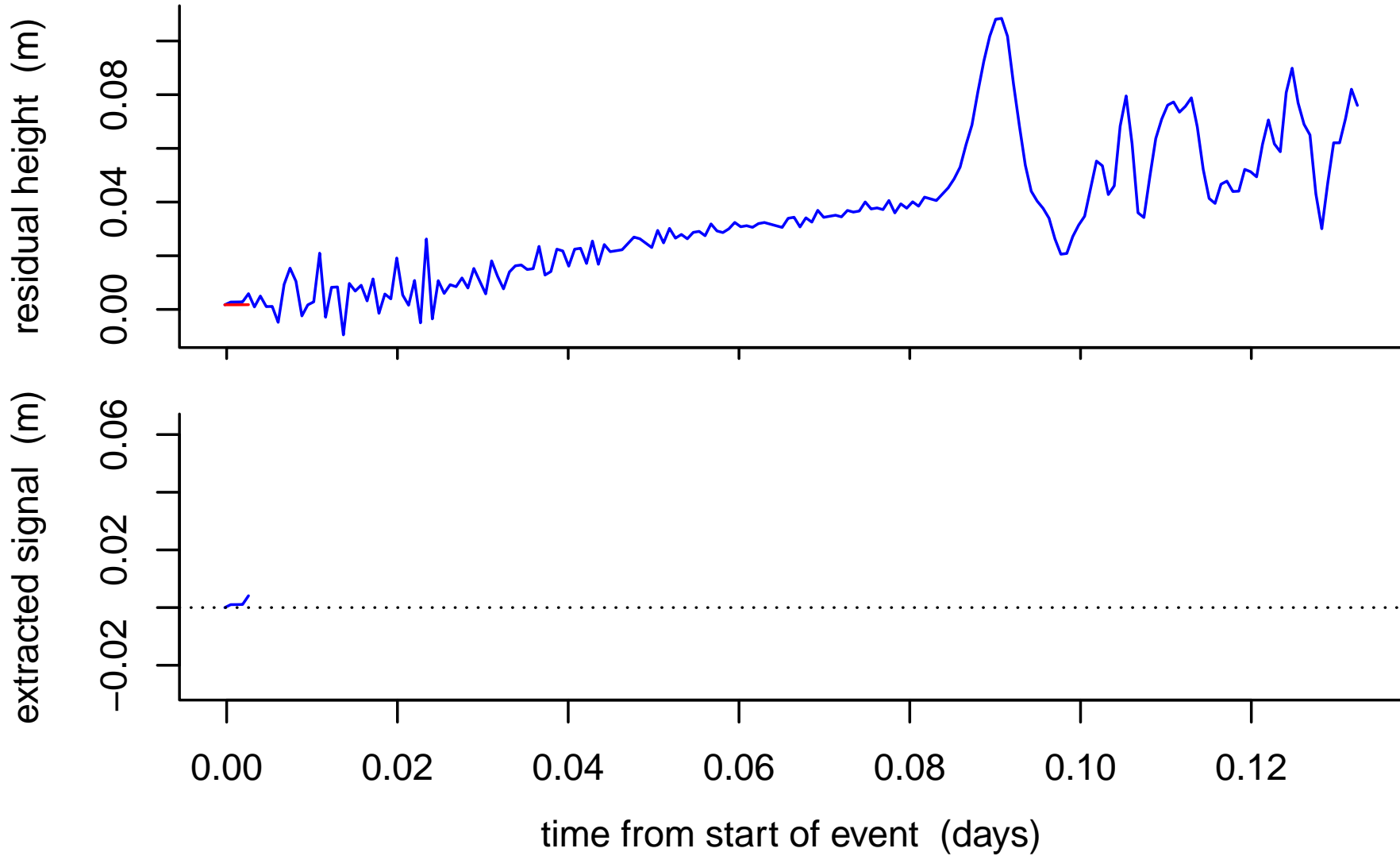
# Kalman Filtering/Smoothing (Local Level Model)



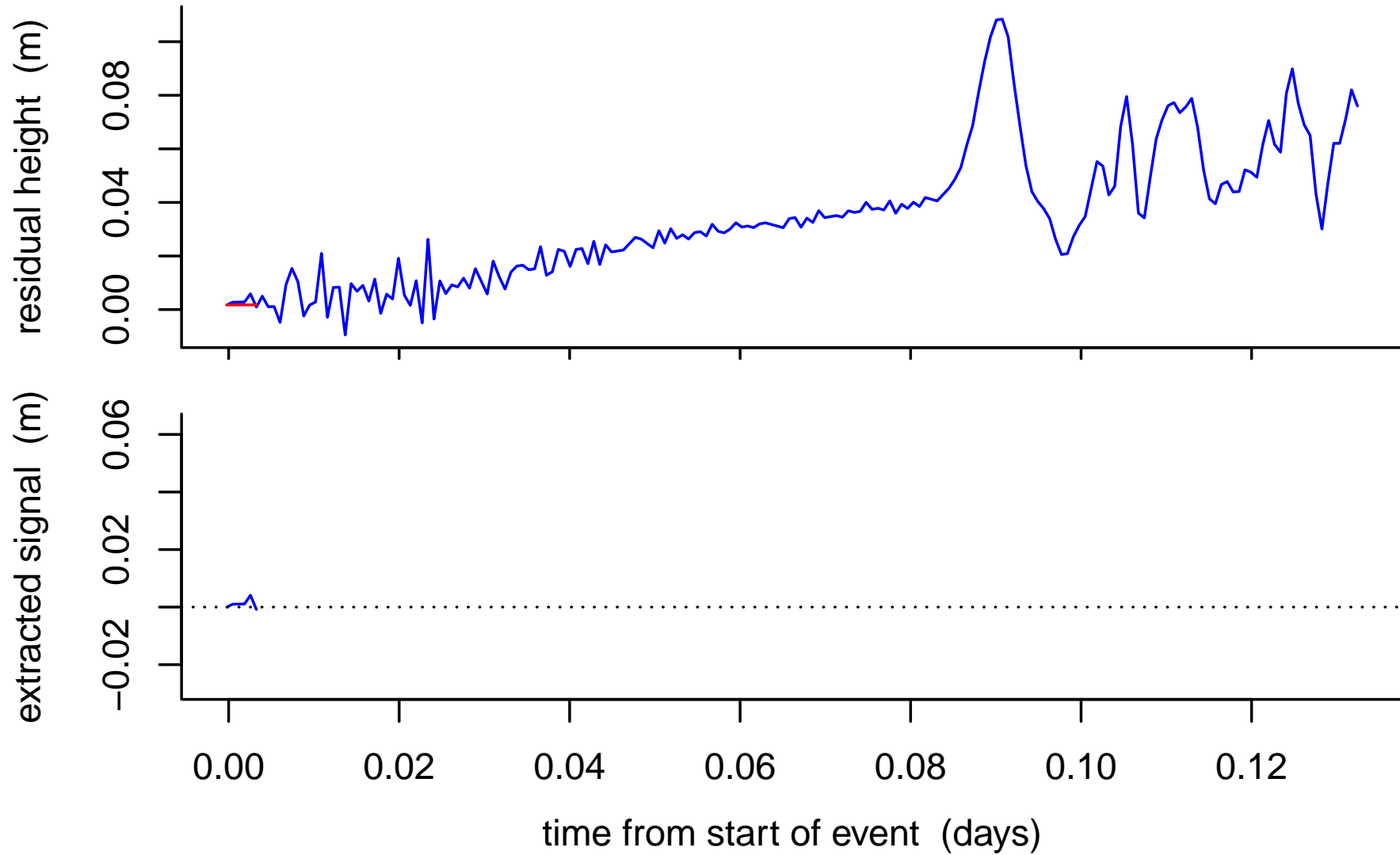
# Kalman Filtering/Smoothing (Local Level Model)



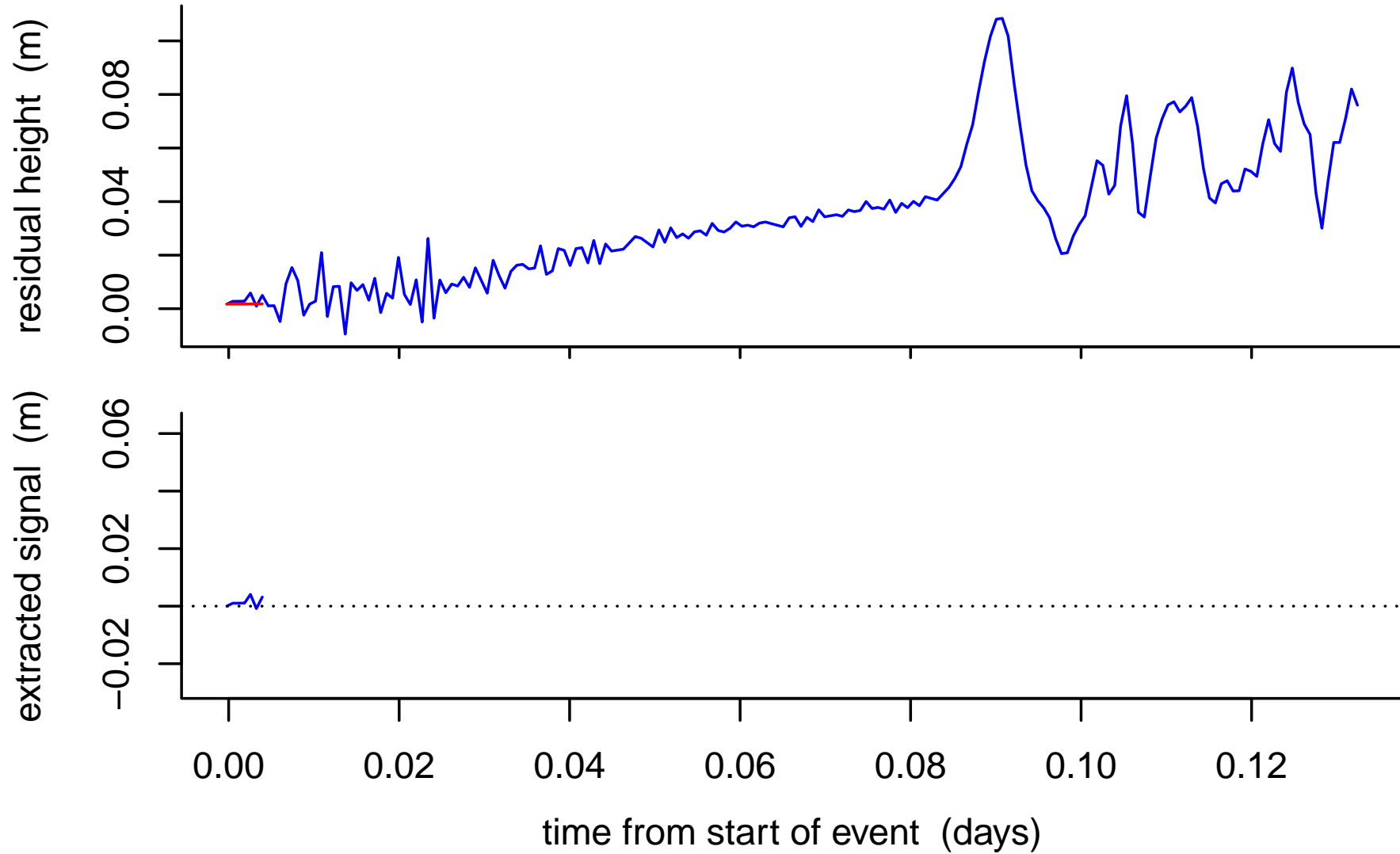
# Kalman Filtering/Smoothing (Local Level Model)



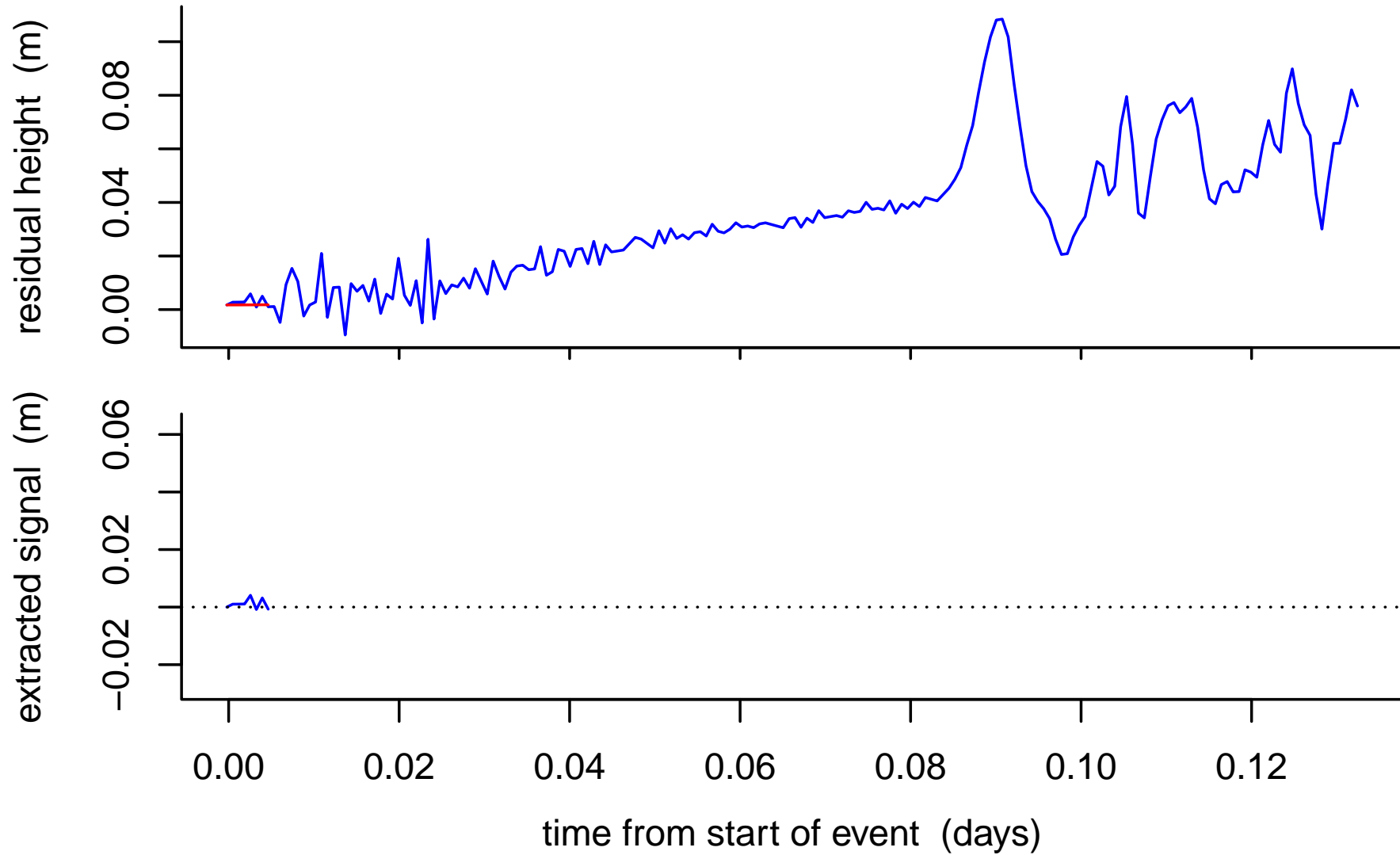
# Kalman Filtering/Smoothing (Local Level Model)



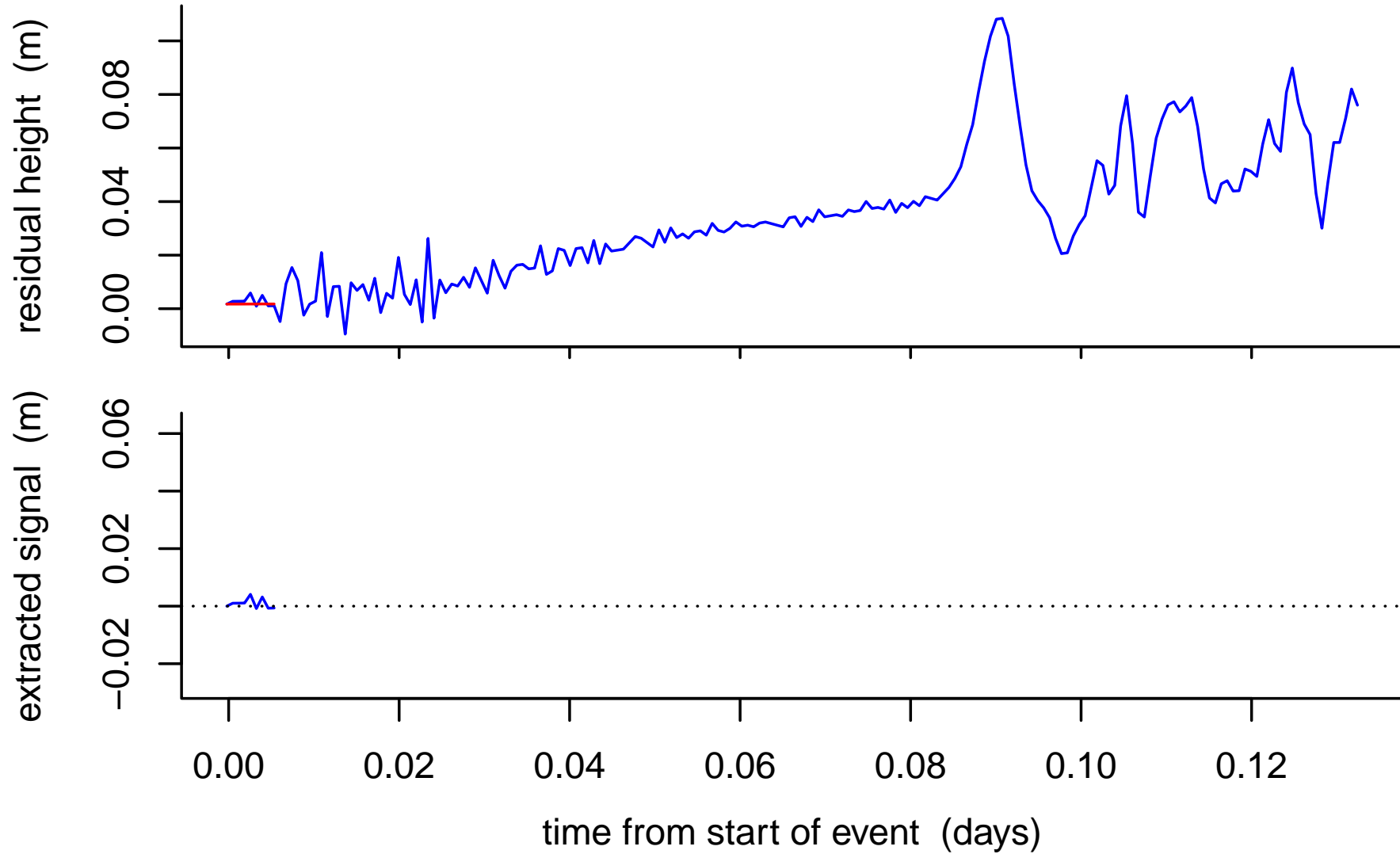
# Kalman Filtering/Smoothing (Local Level Model)



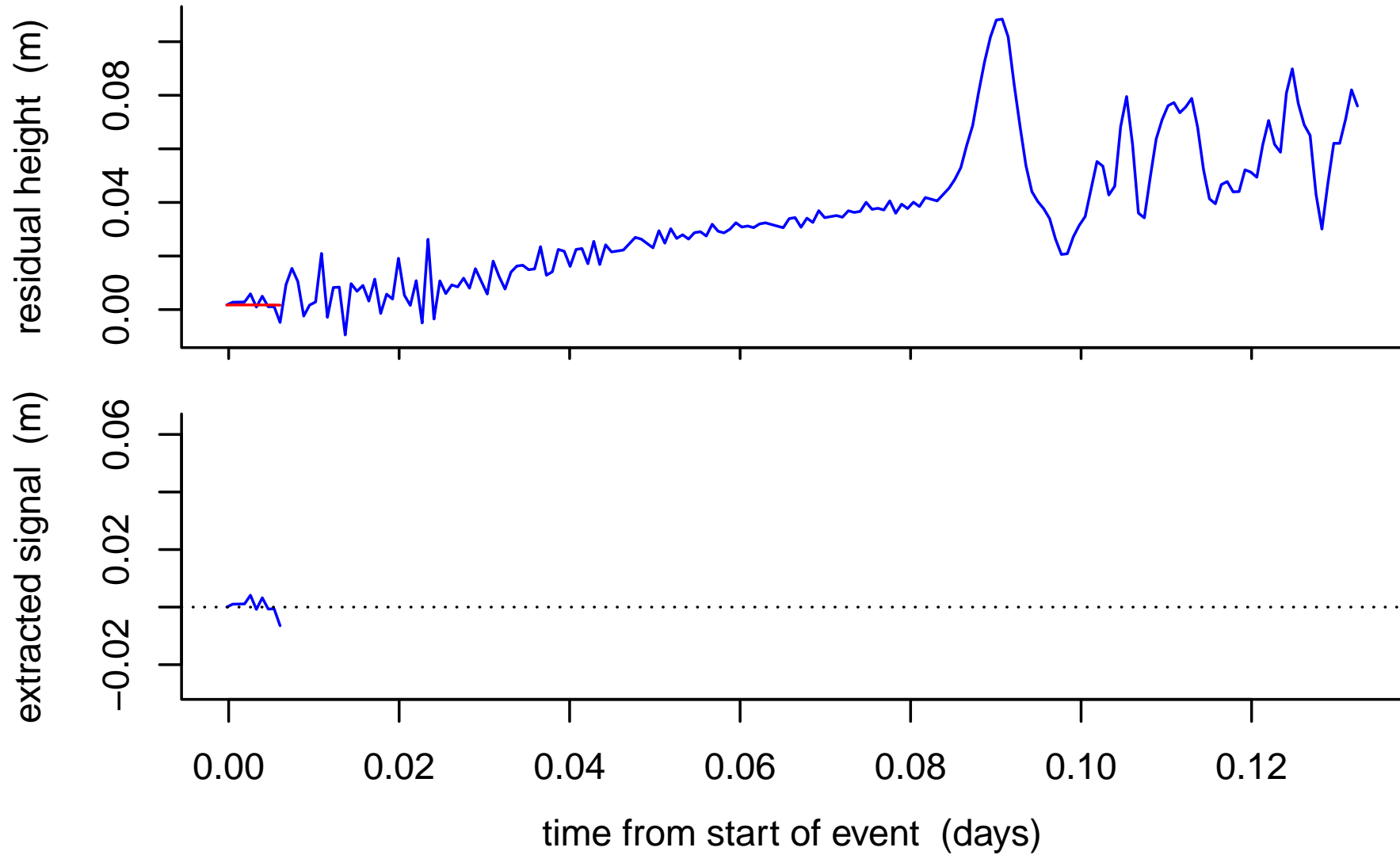
# Kalman Filtering/Smoothing (Local Level Model)



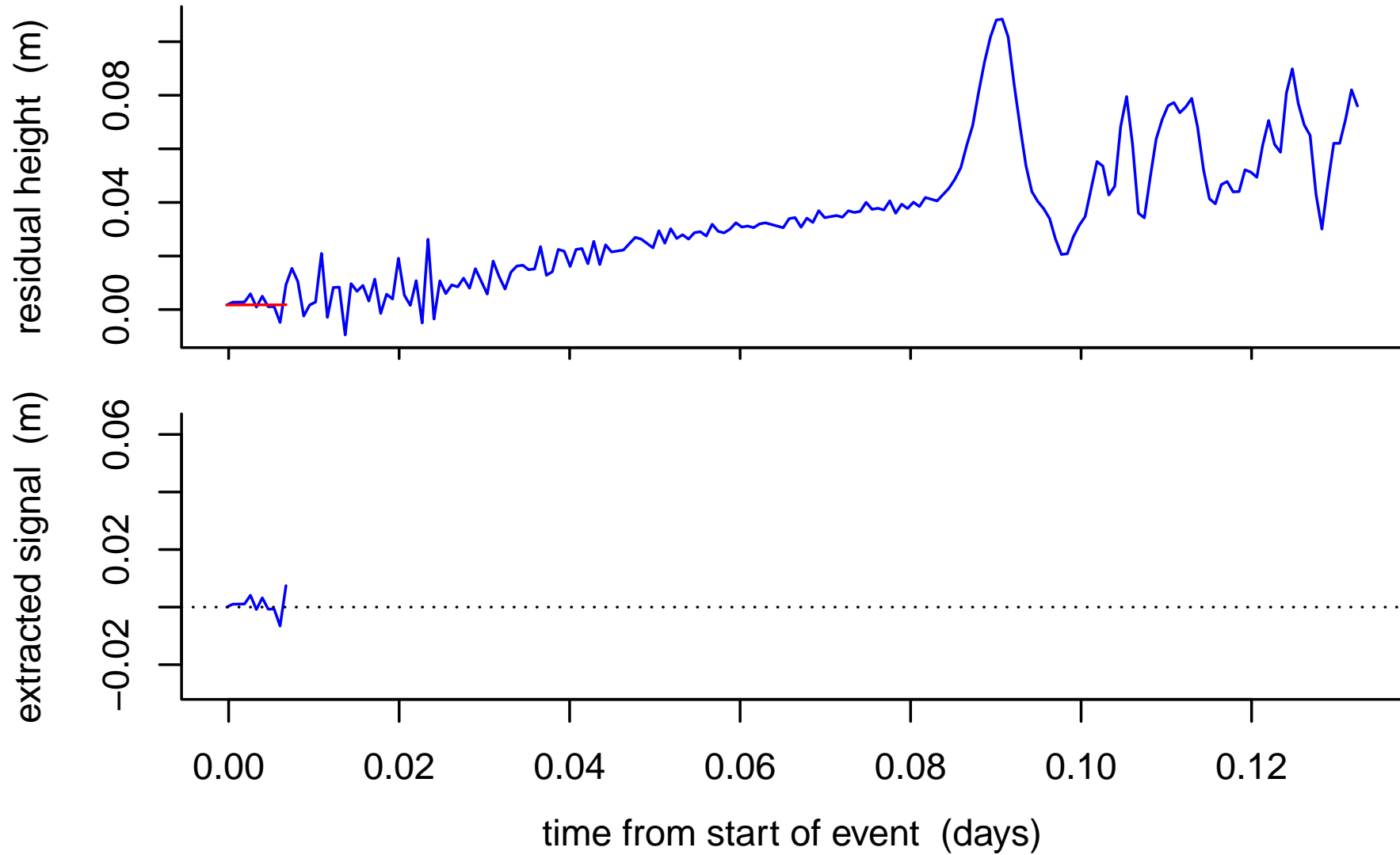
# Kalman Filtering/Smoothing (Local Level Model)



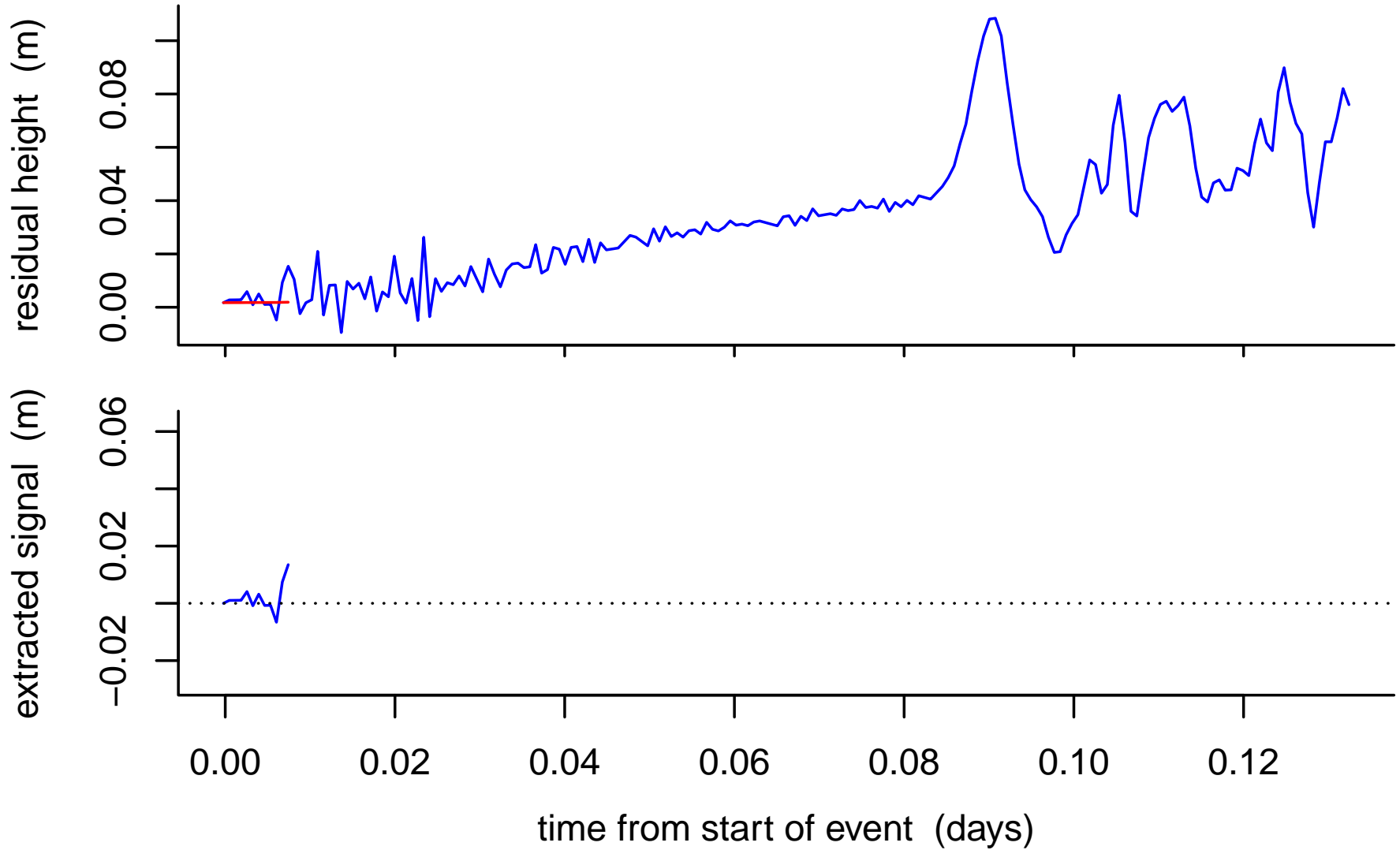
# Kalman Filtering/Smoothing (Local Level Model)



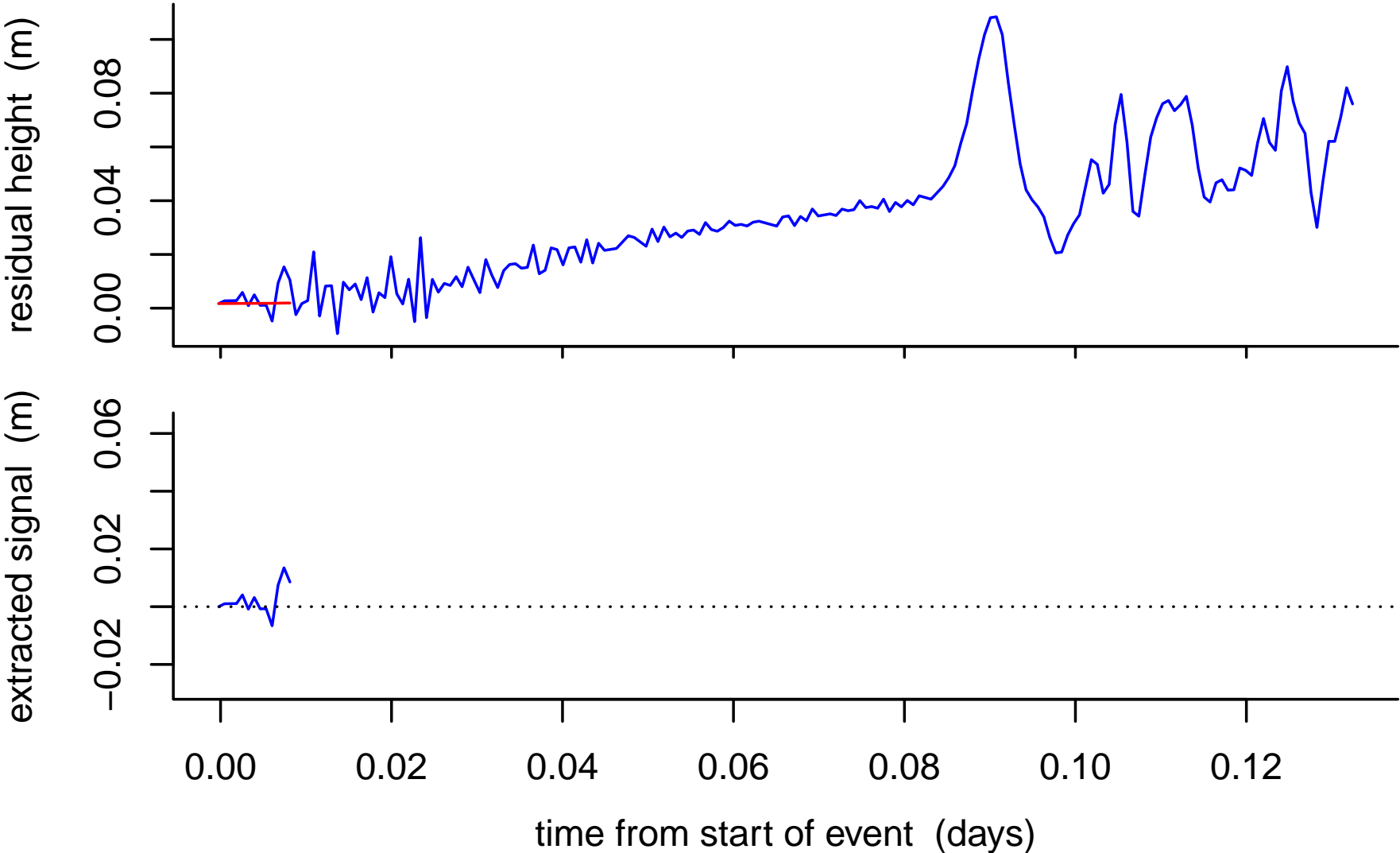
# Kalman Filtering/Smoothing (Local Level Model)



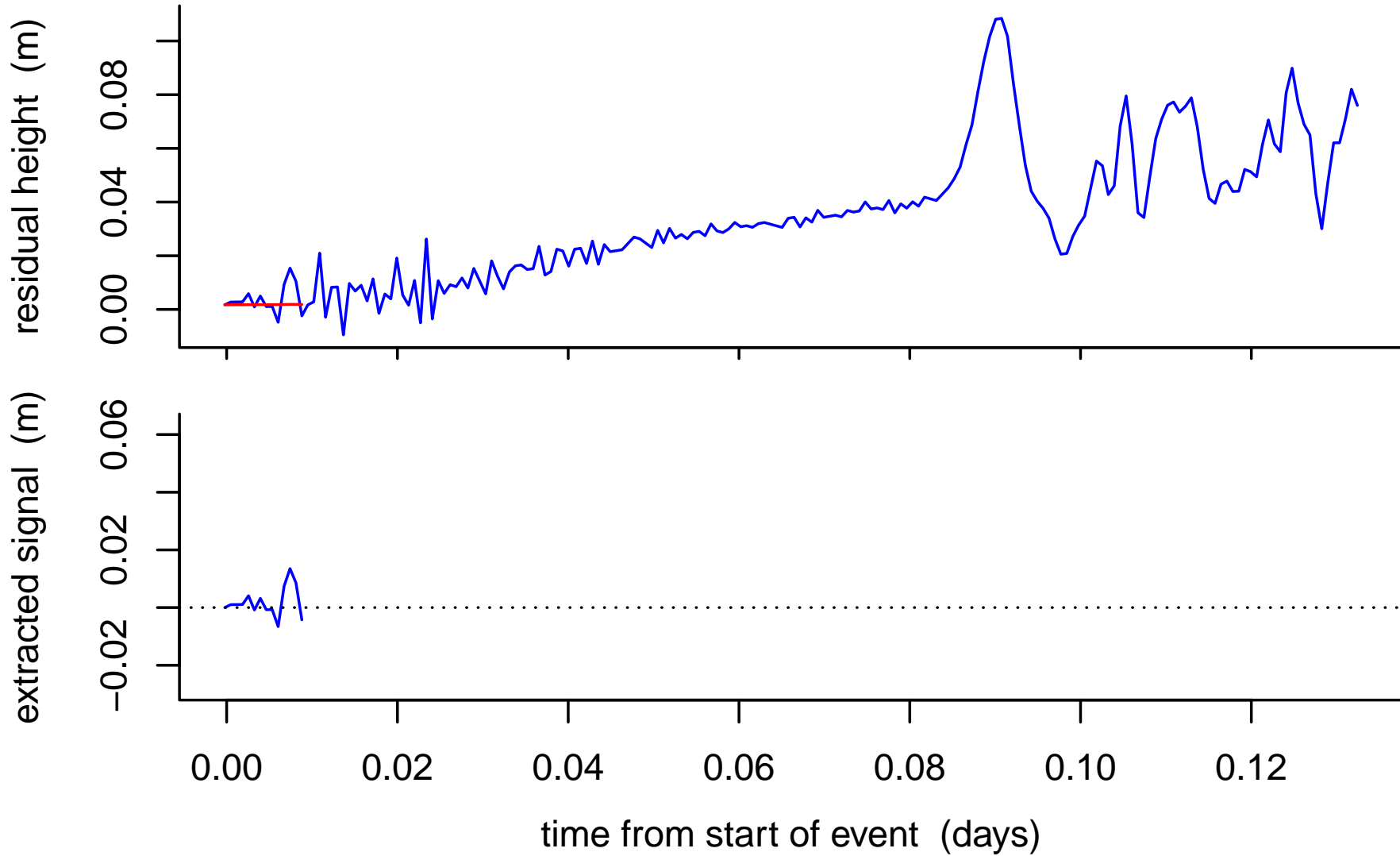
# Kalman Filtering/Smoothing (Local Level Model)



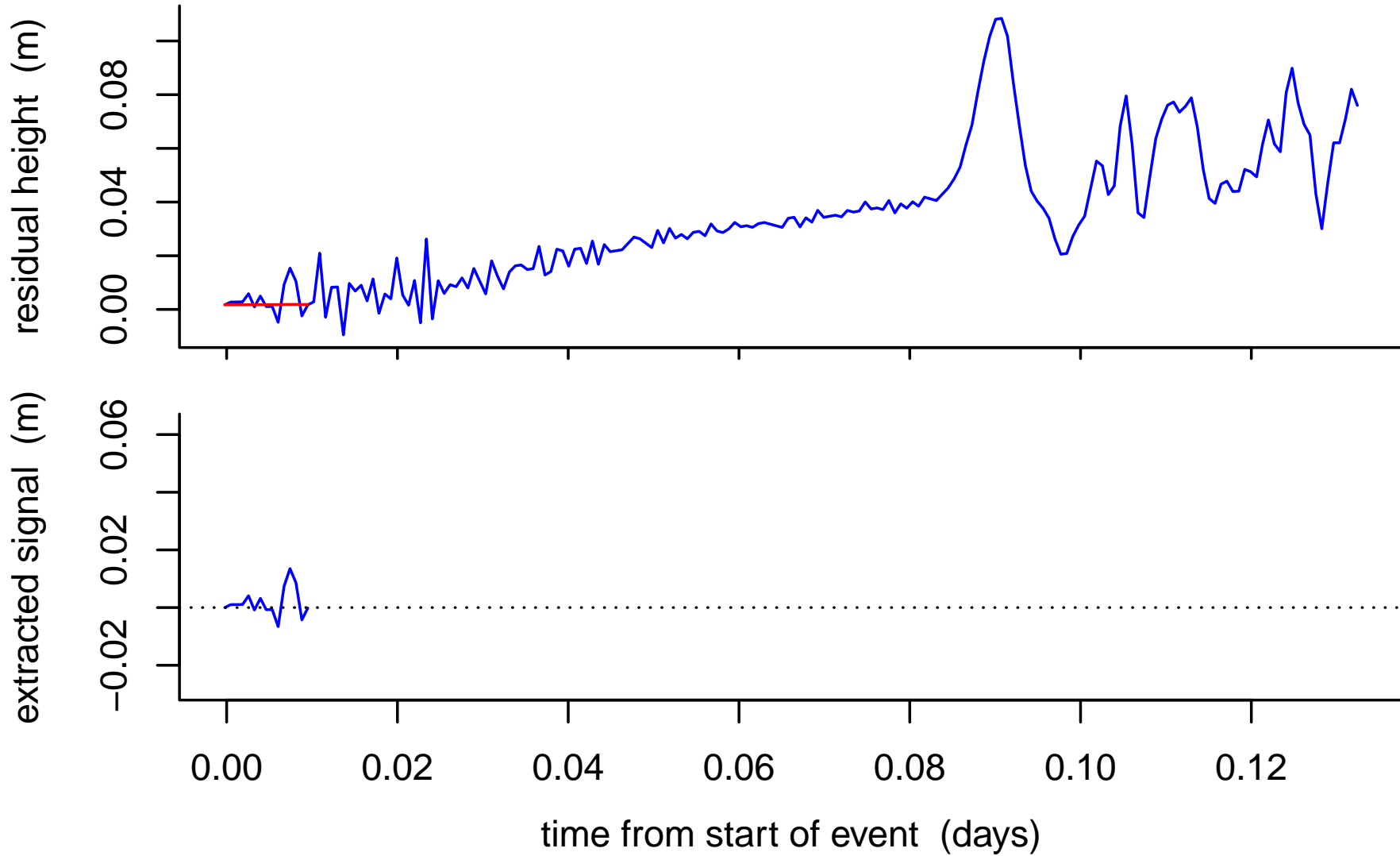
# Kalman Filtering/Smoothing (Local Level Model)



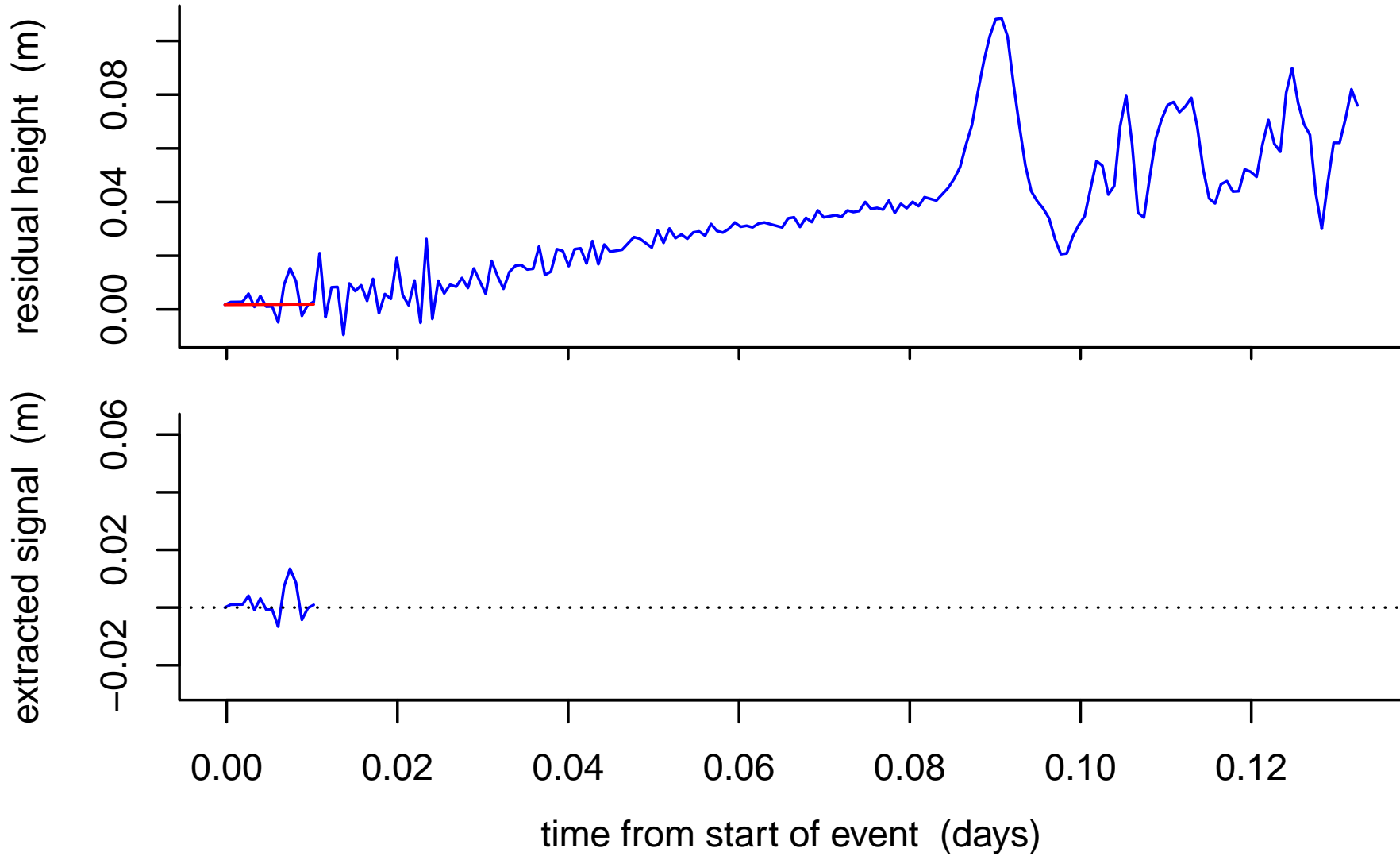
# Kalman Filtering/Smoothing (Local Level Model)



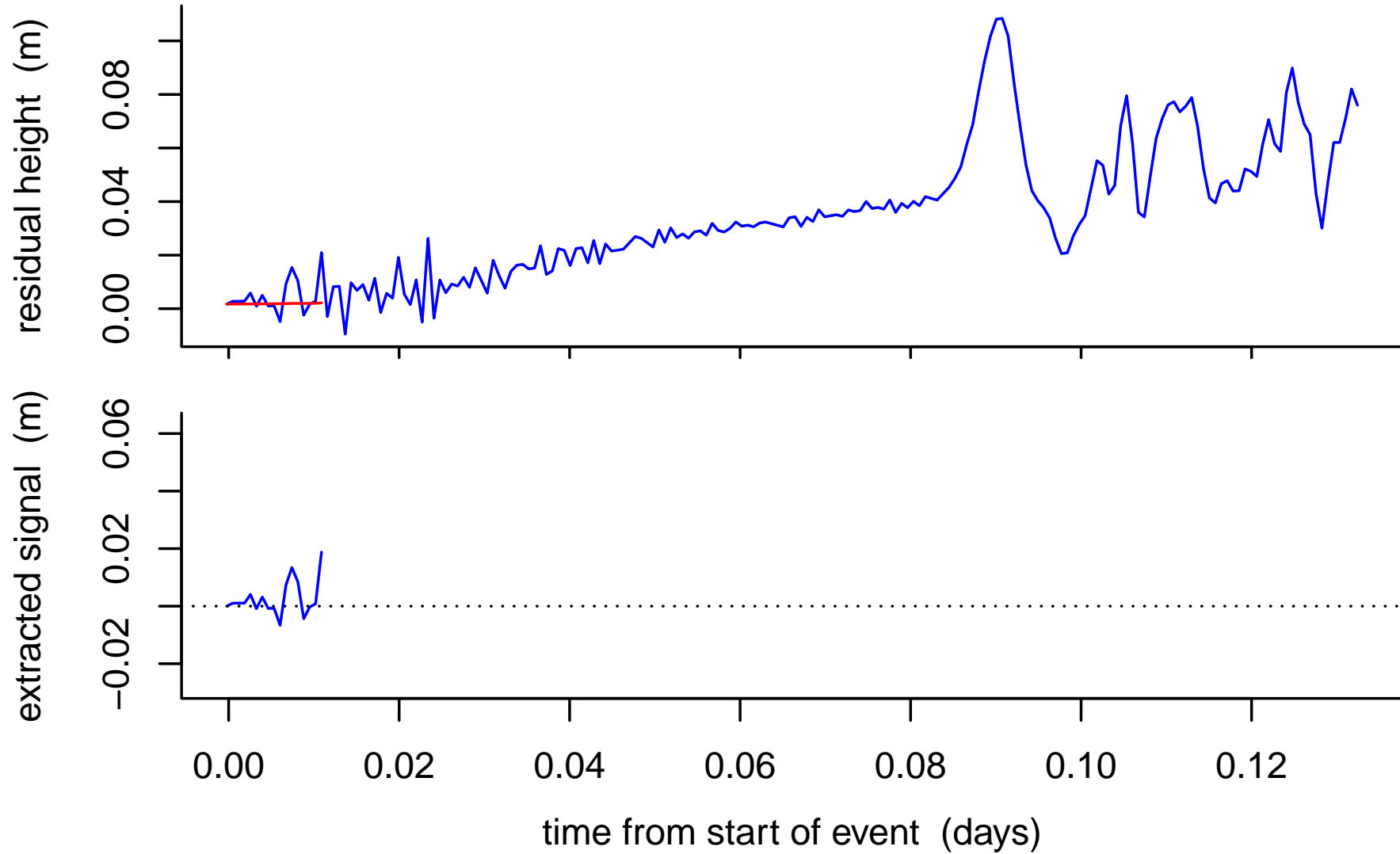
# Kalman Filtering/Smoothing (Local Level Model)



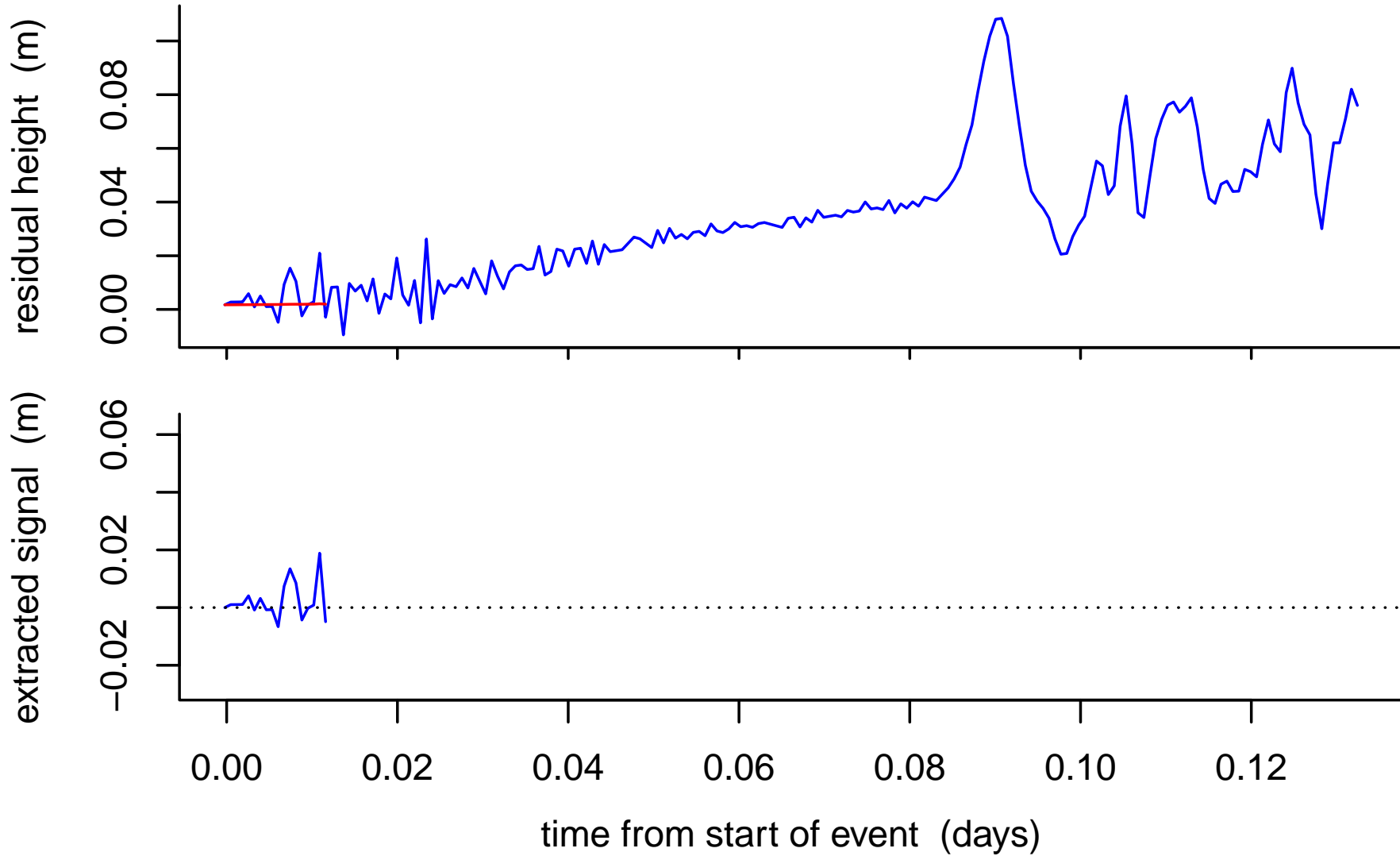
# Kalman Filtering/Smoothing (Local Level Model)



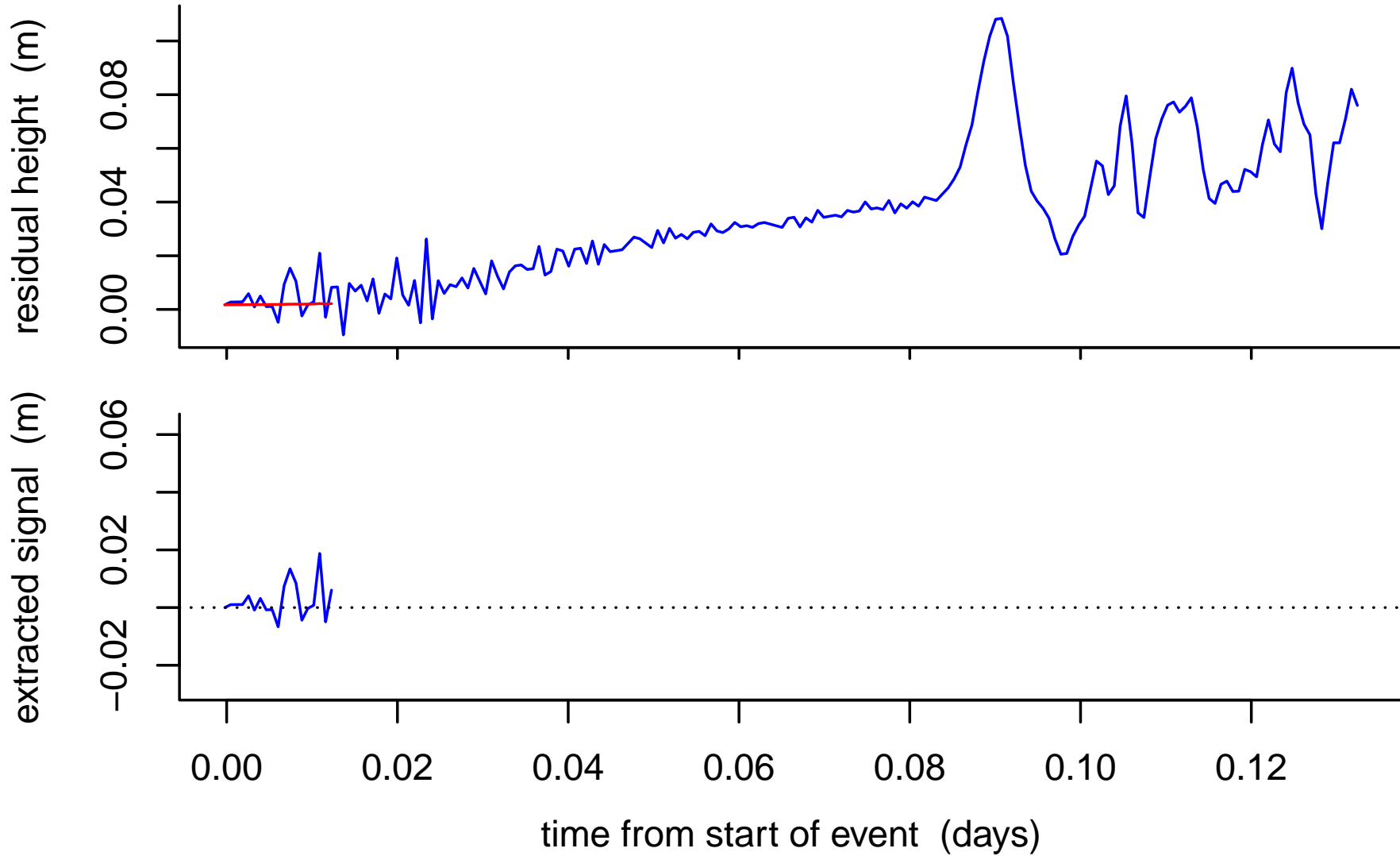
# Kalman Filtering/Smoothing (Local Level Model)



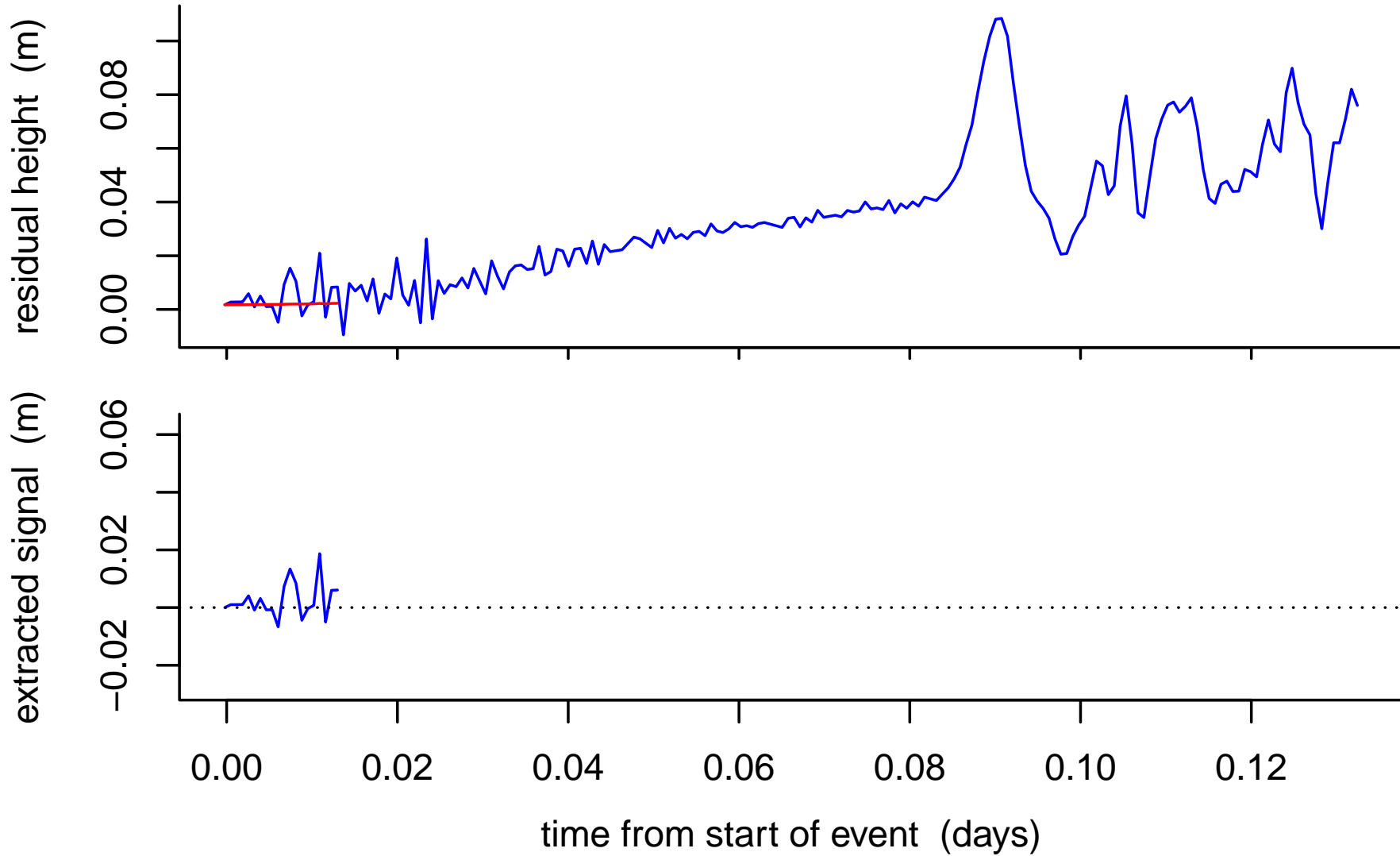
# Kalman Filtering/Smoothing (Local Level Model)



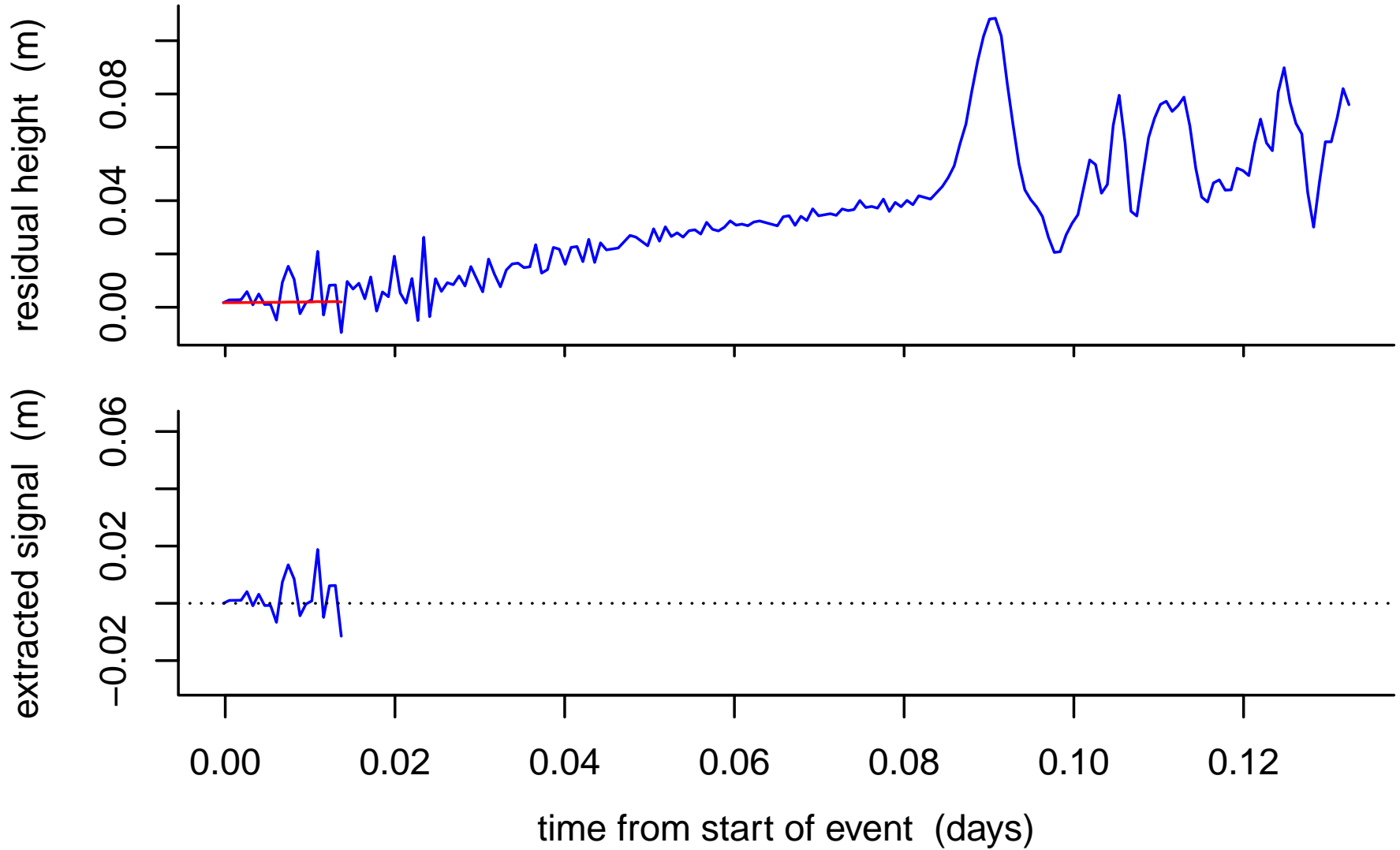
# Kalman Filtering/Smoothing (Local Level Model)



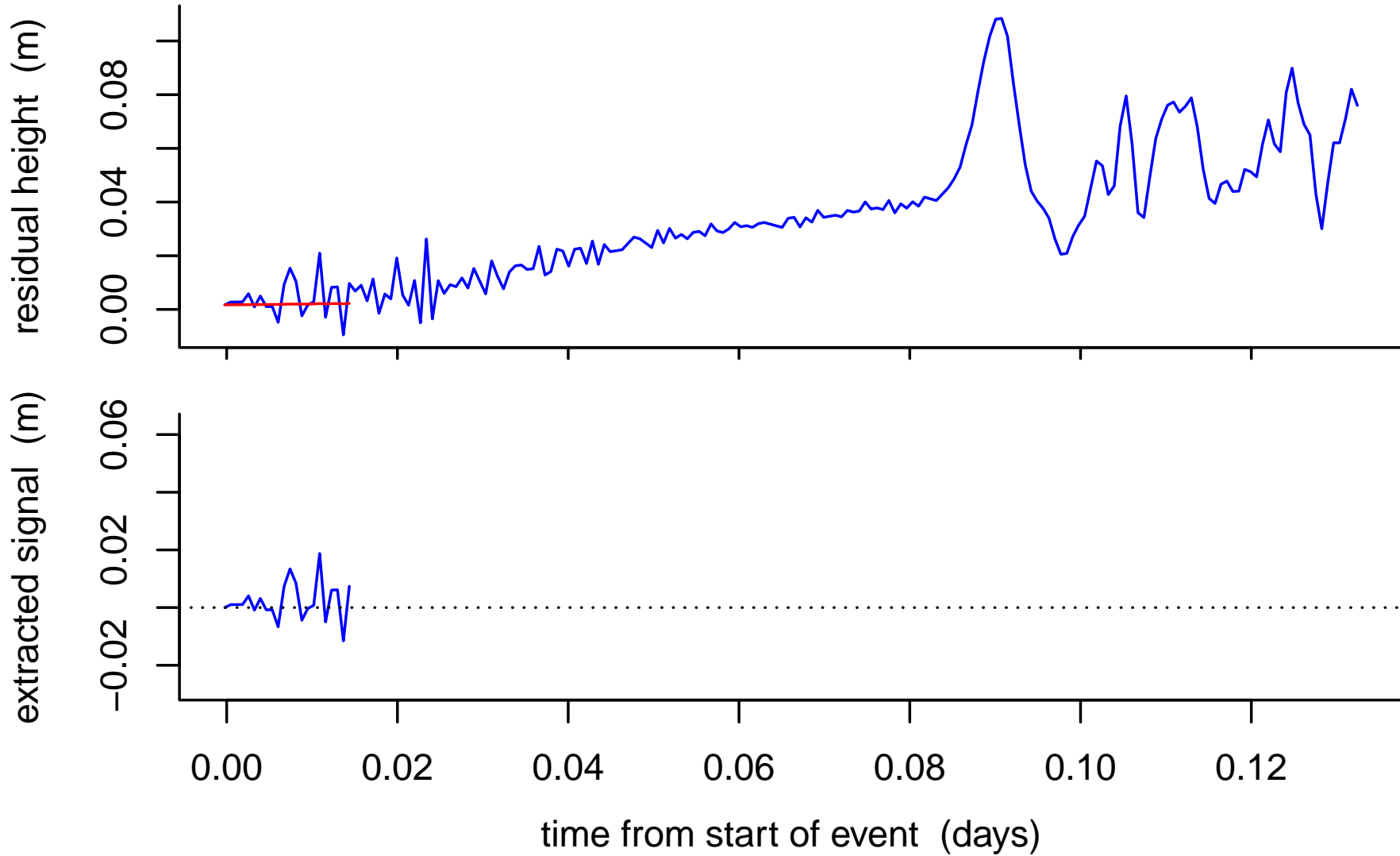
# Kalman Filtering/Smoothing (Local Level Model)



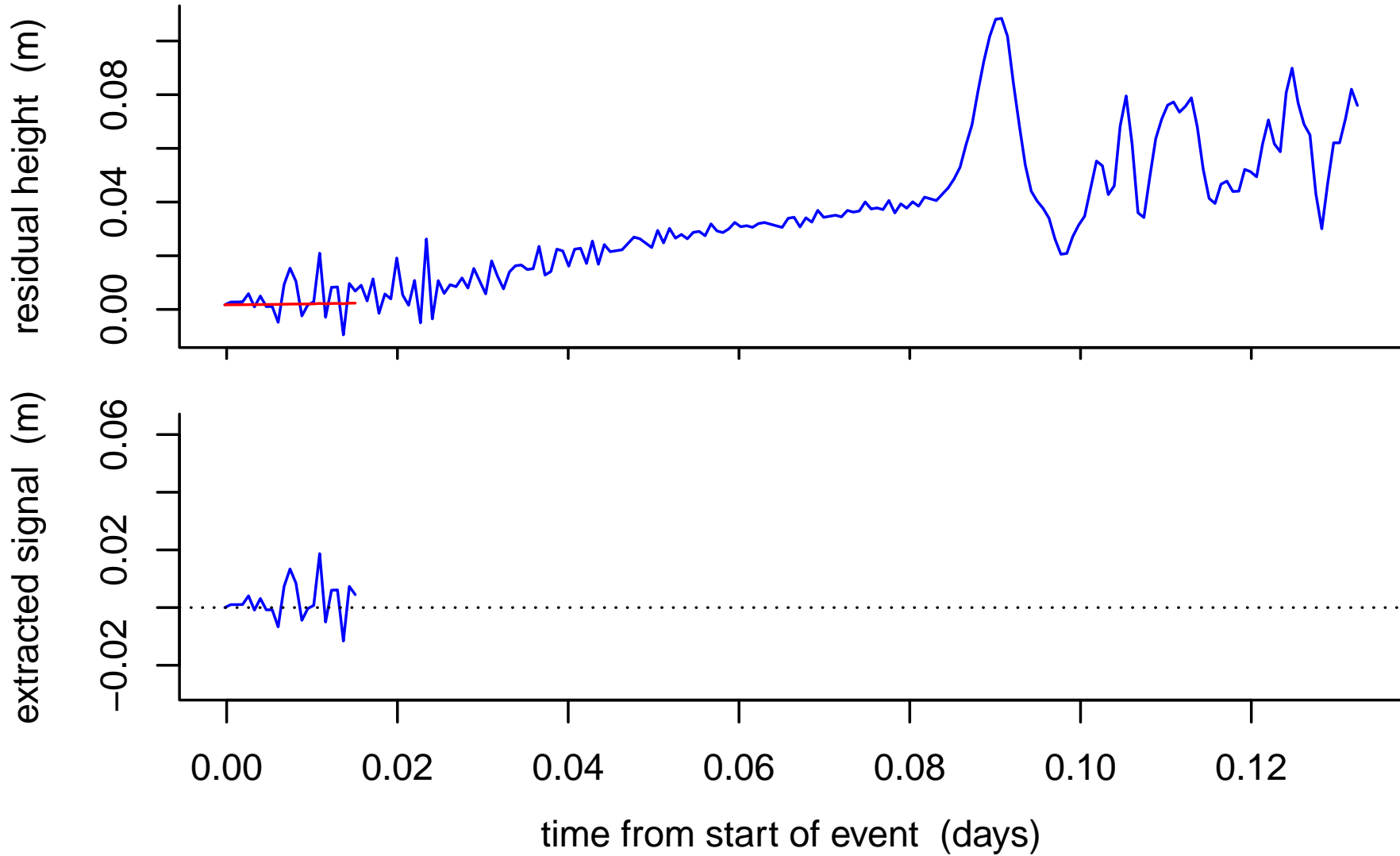
# Kalman Filtering/Smoothing (Local Level Model)



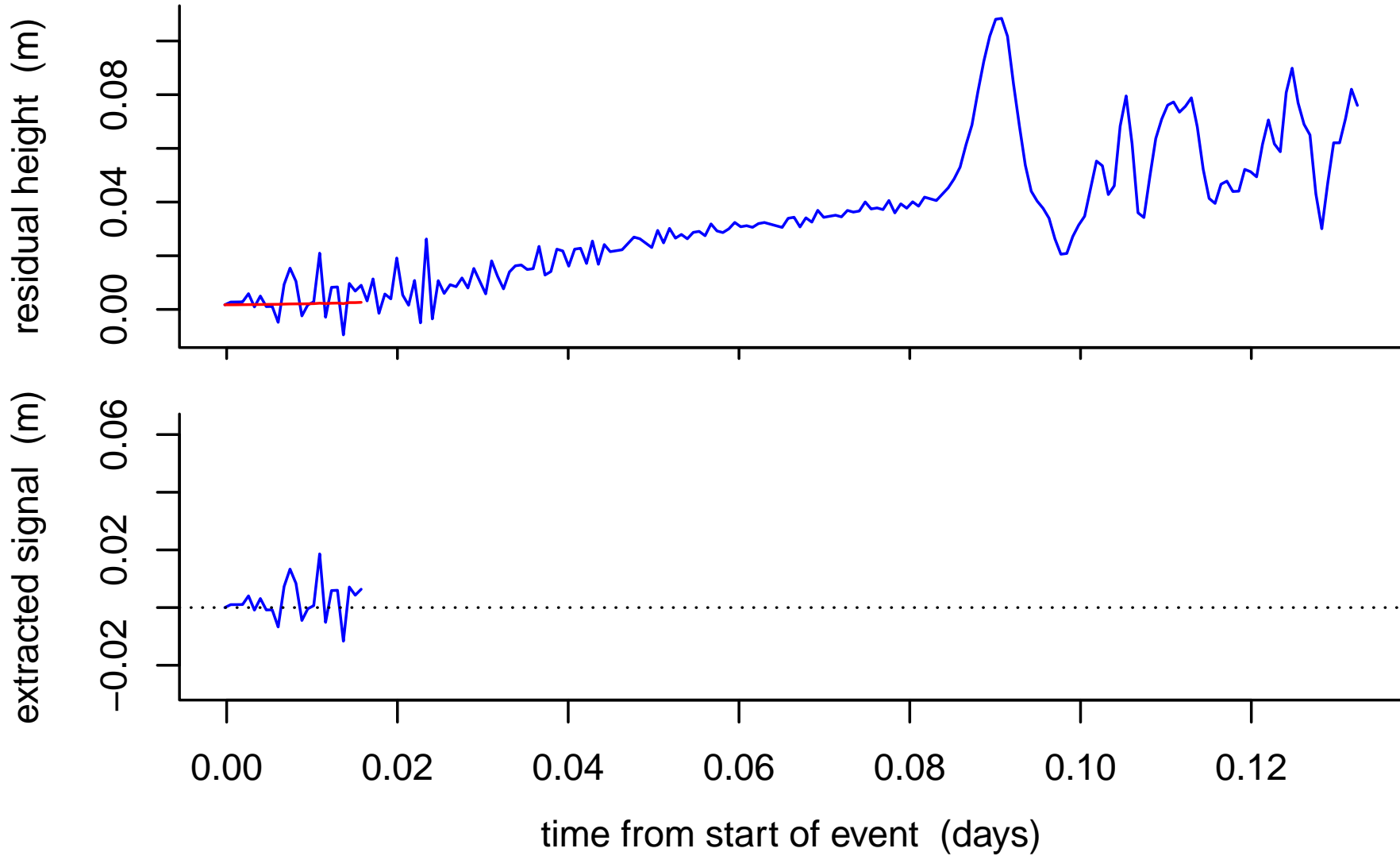
# Kalman Filtering/Smoothing (Local Level Model)



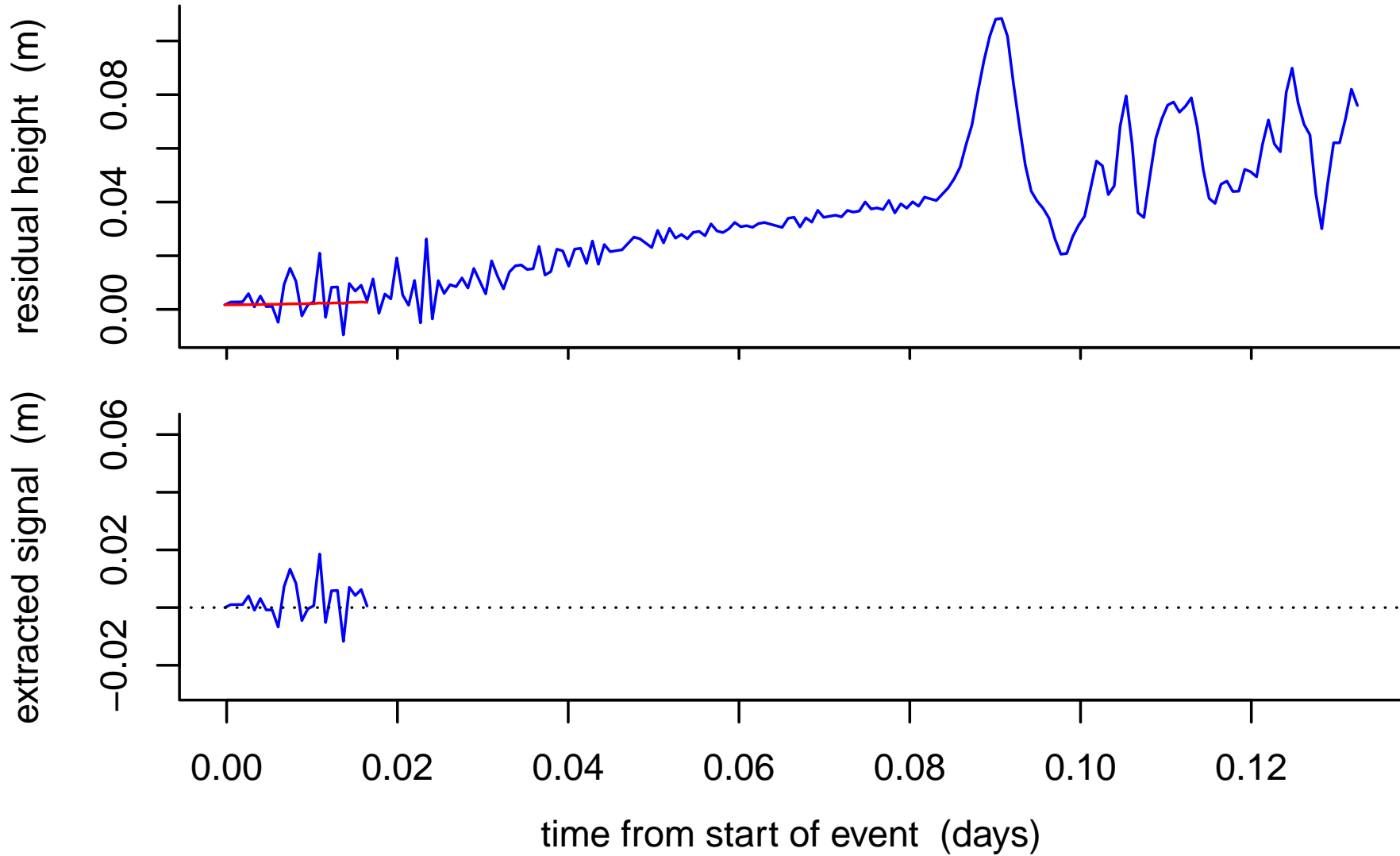
# Kalman Filtering/Smoothing (Local Level Model)



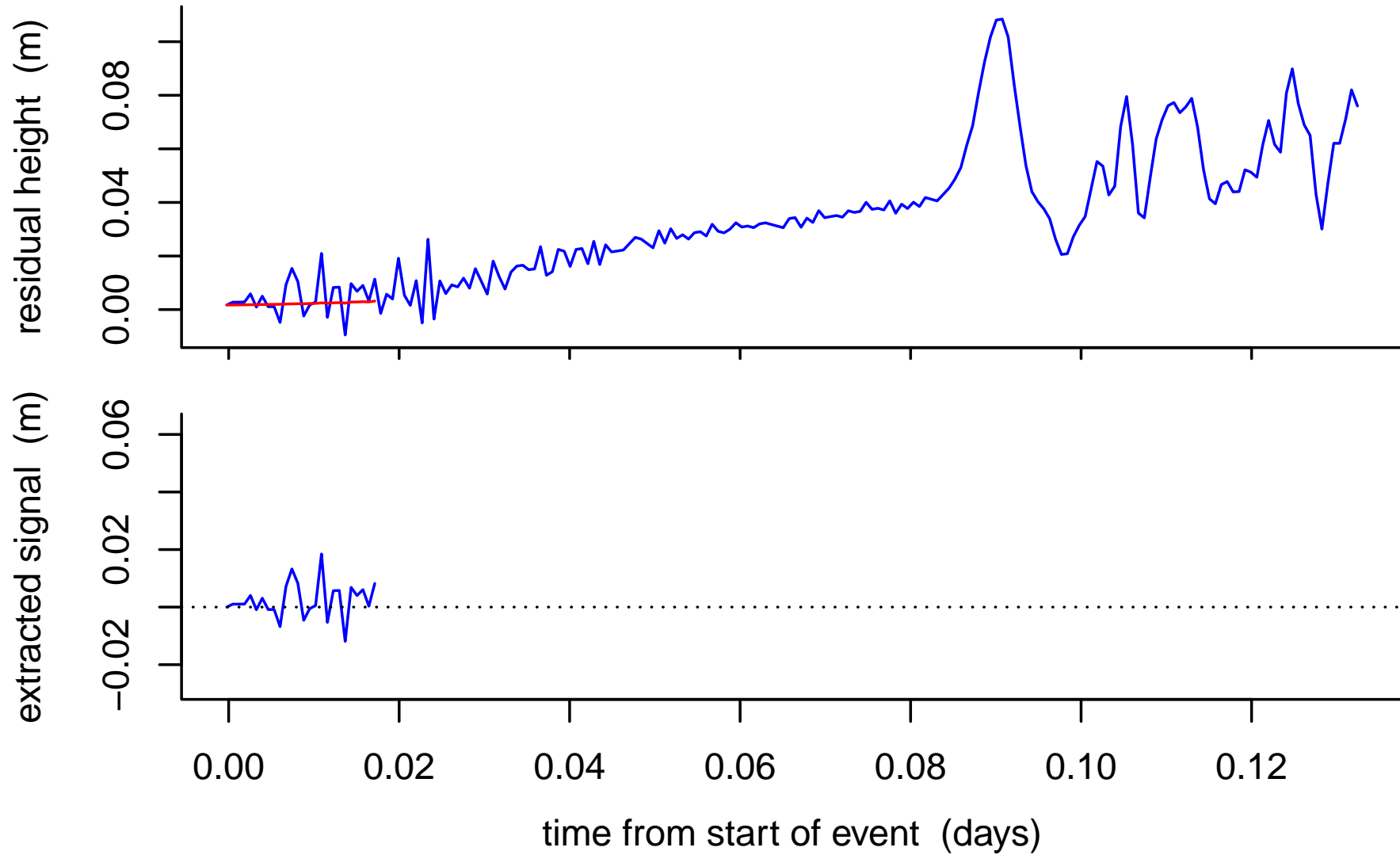
# Kalman Filtering/Smoothing (Local Level Model)



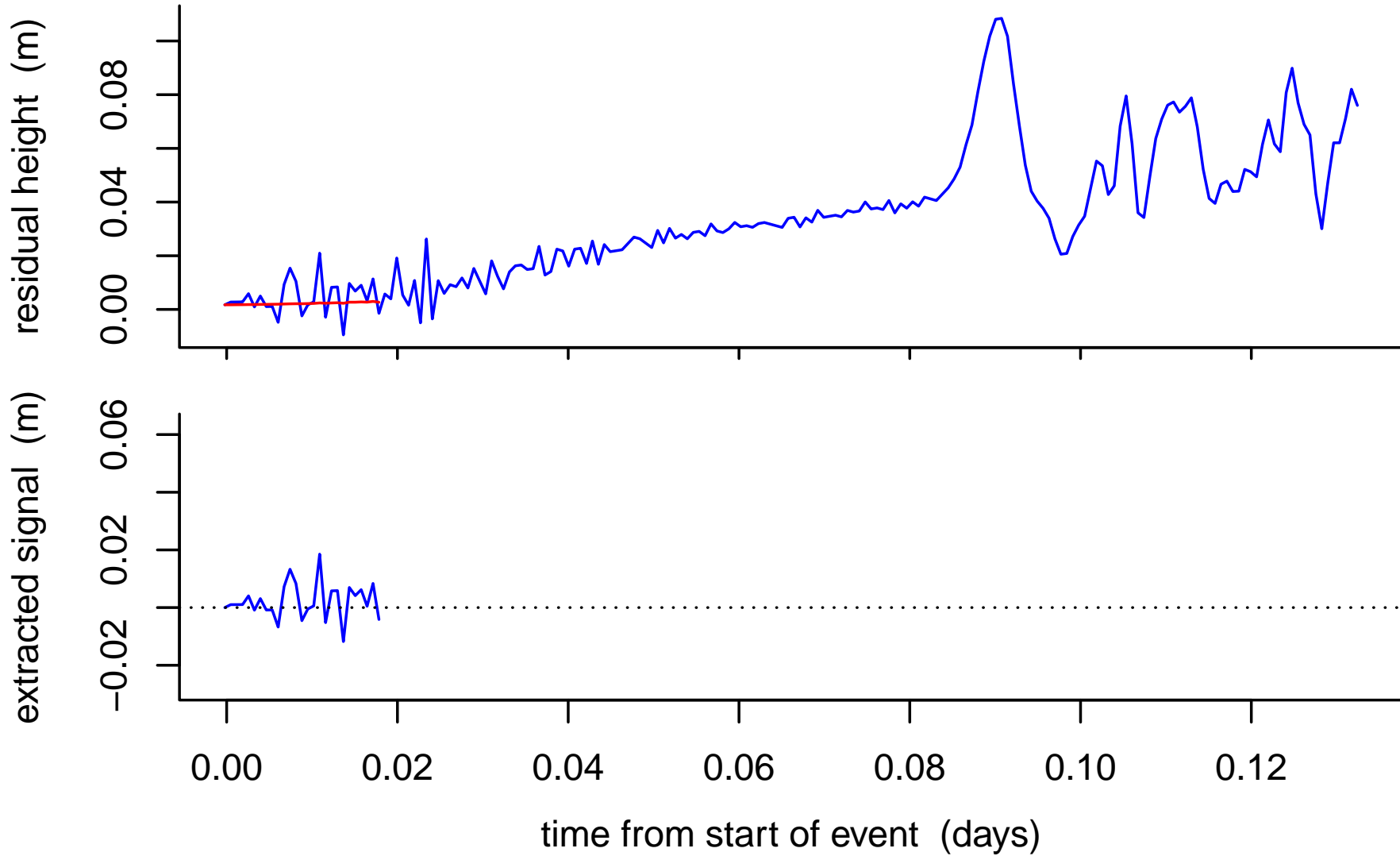
# Kalman Filtering/Smoothing (Local Level Model)



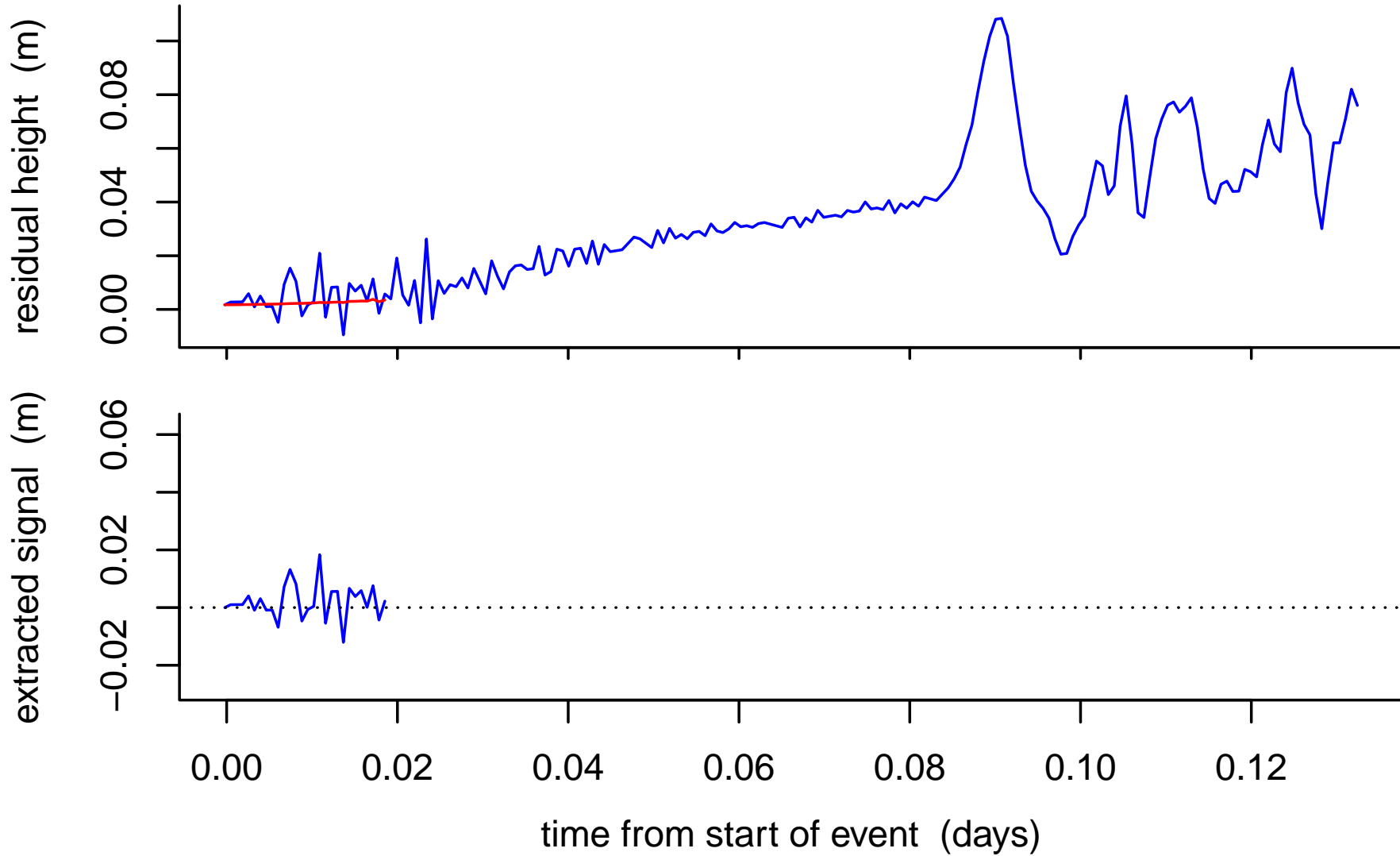
# Kalman Filtering/Smoothing (Local Level Model)



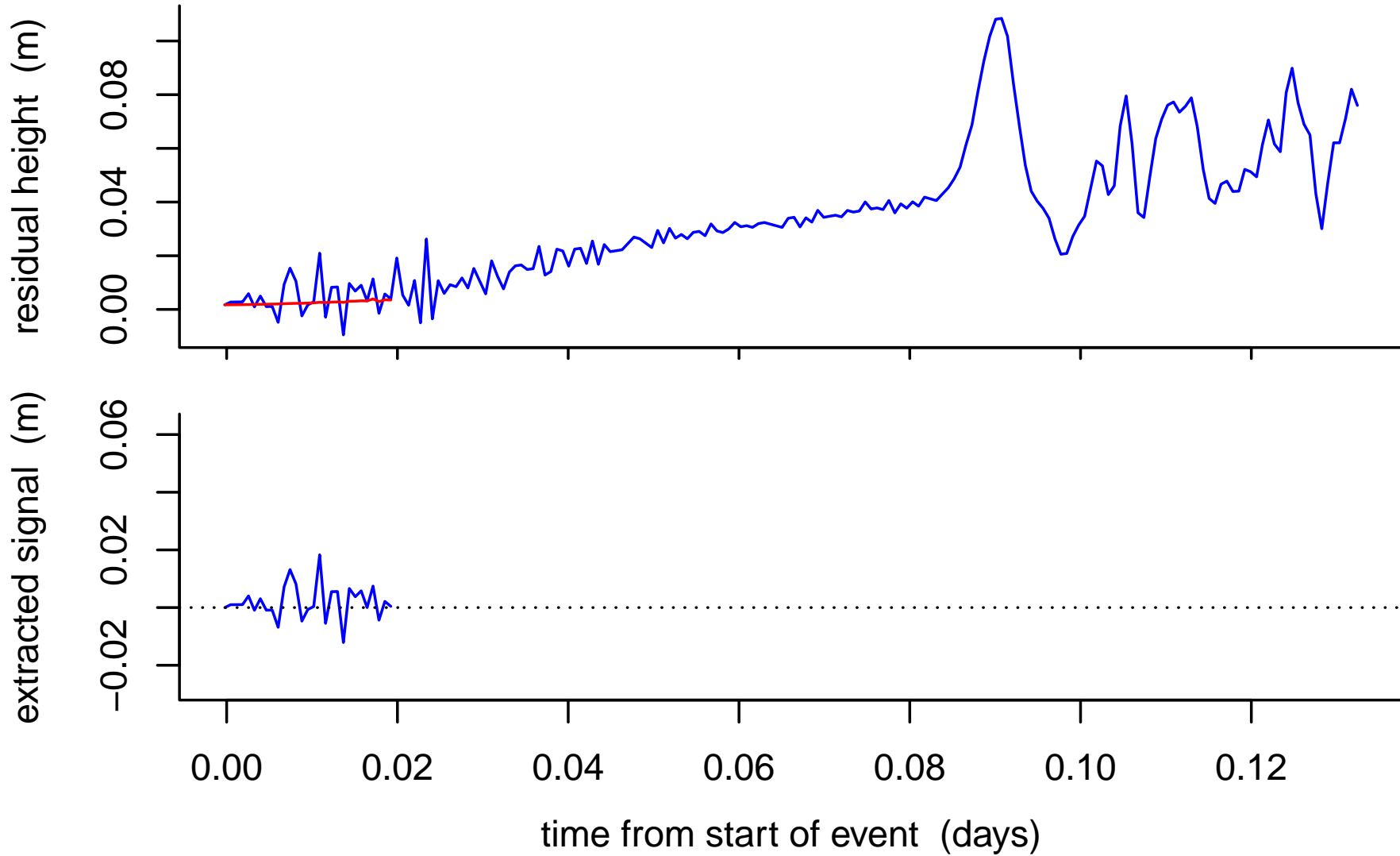
# Kalman Filtering/Smoothing (Local Level Model)



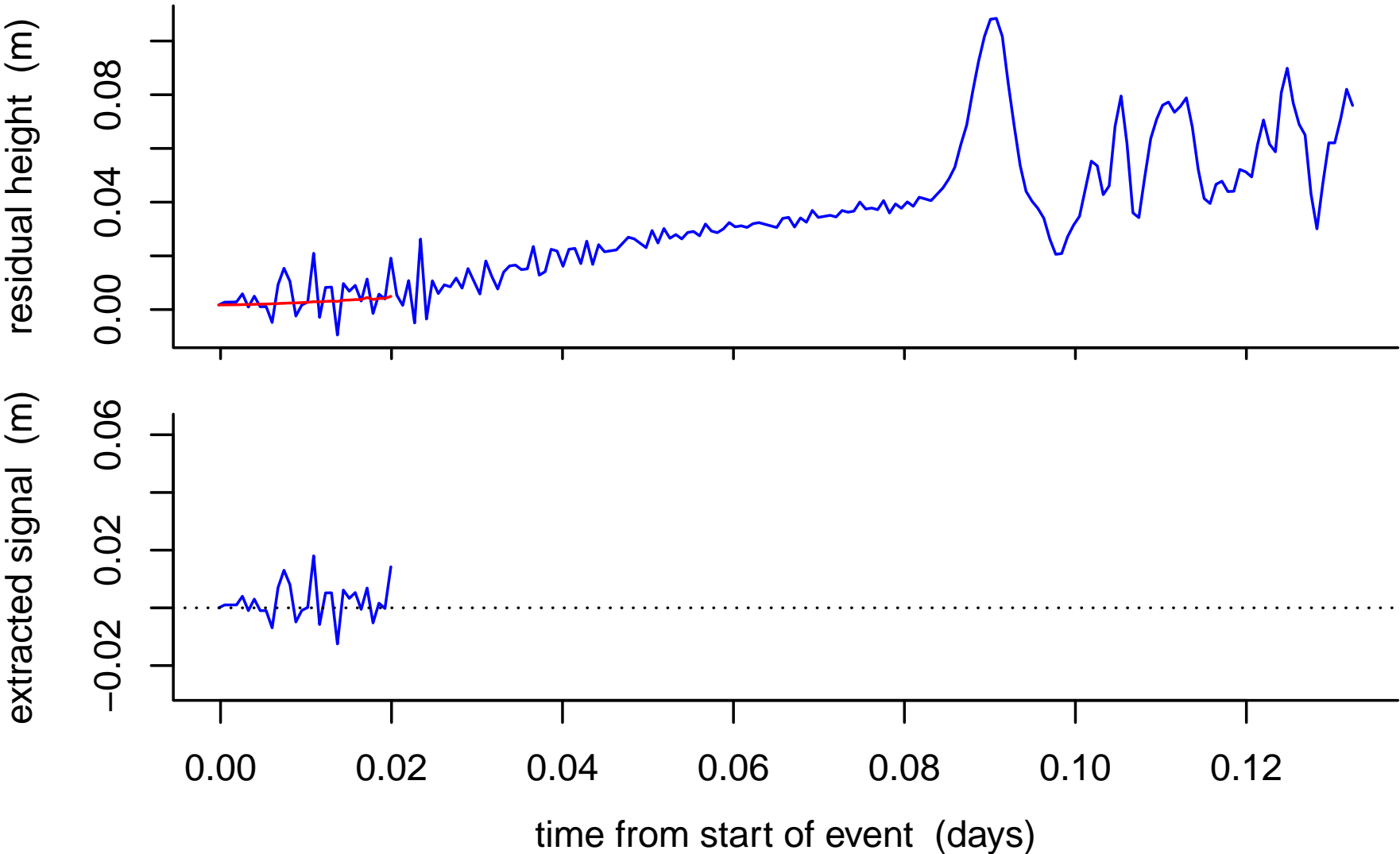
# Kalman Filtering/Smoothing (Local Level Model)



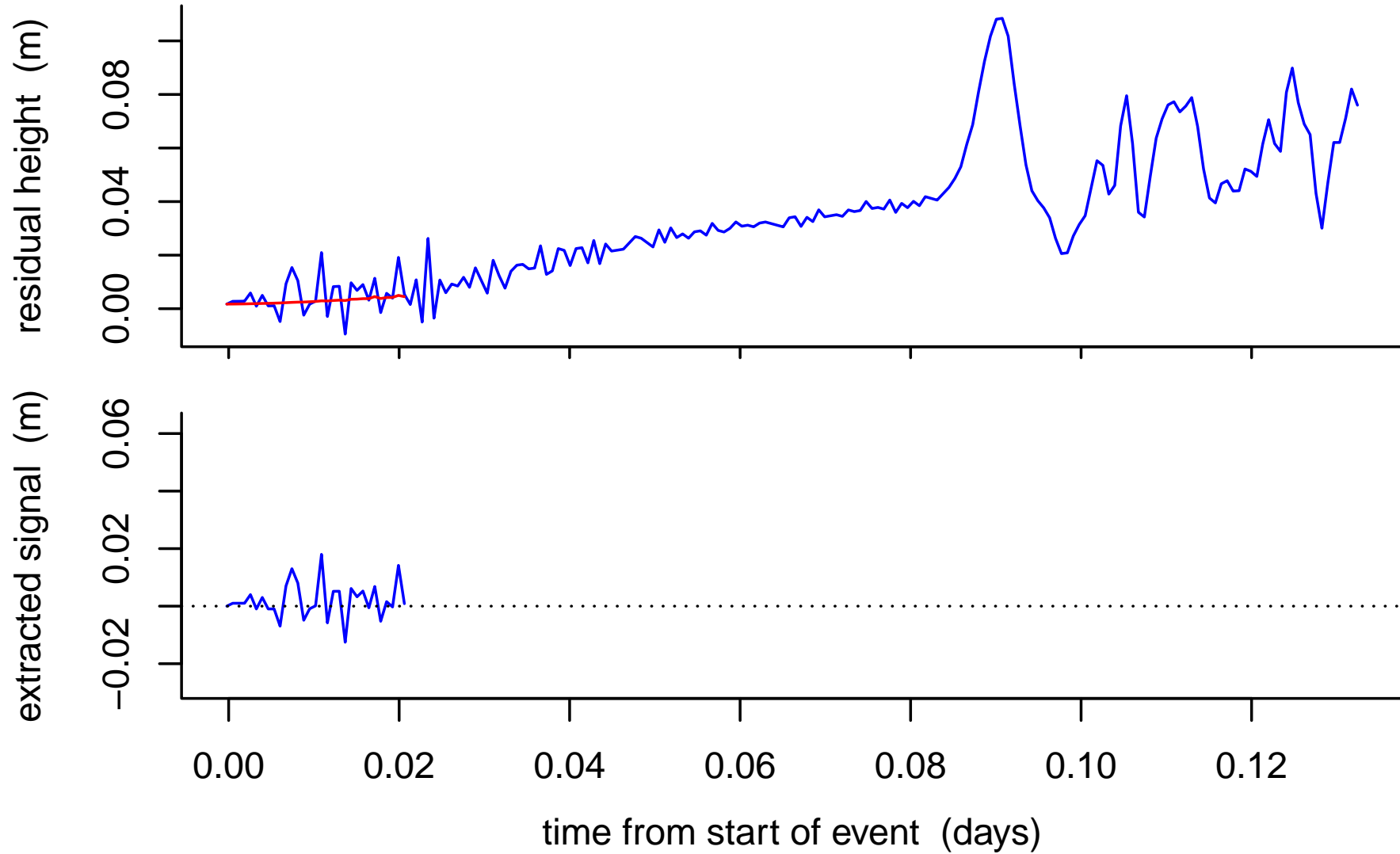
# Kalman Filtering/Smoothing (Local Level Model)



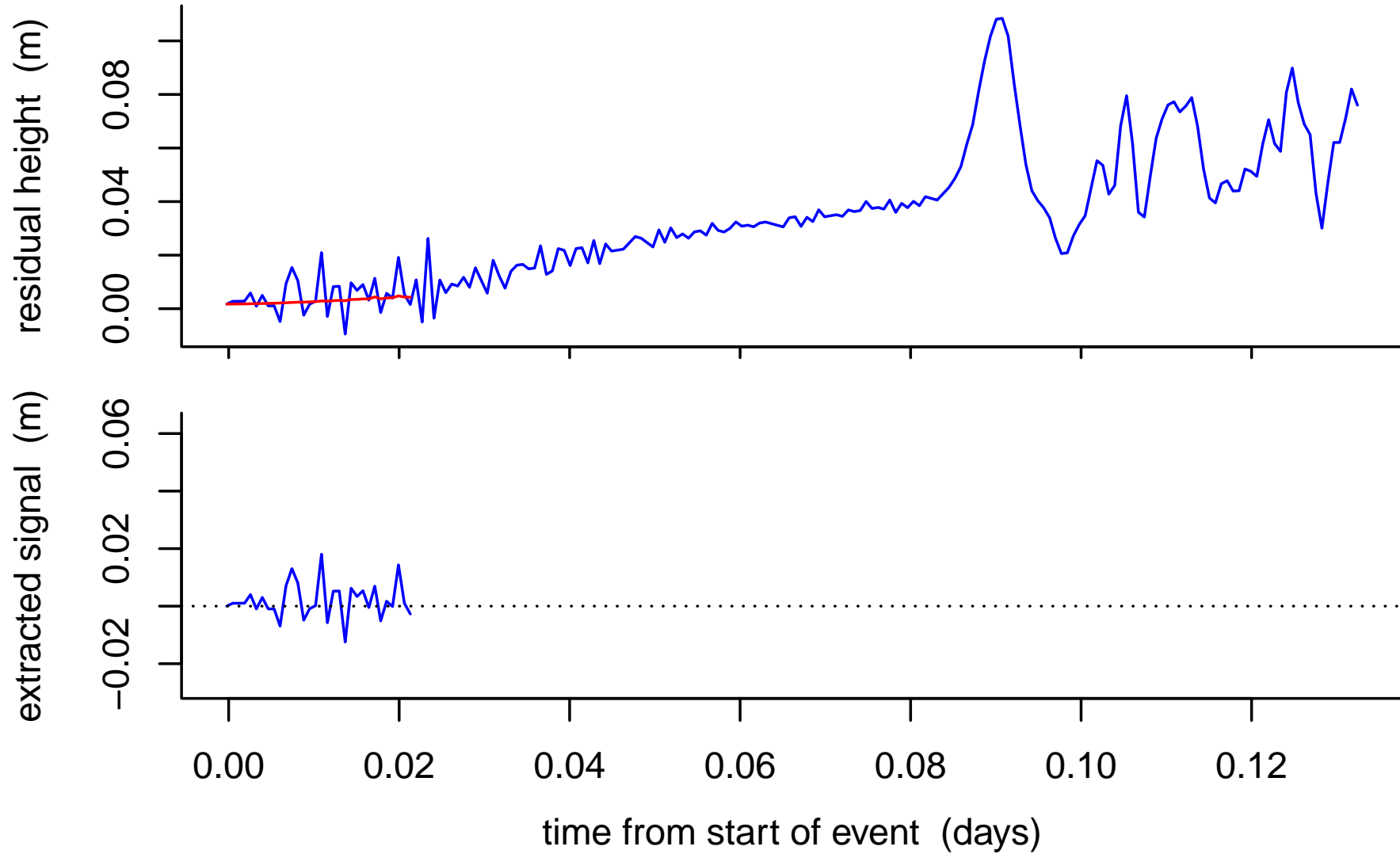
# Kalman Filtering/Smoothing (Local Level Model)



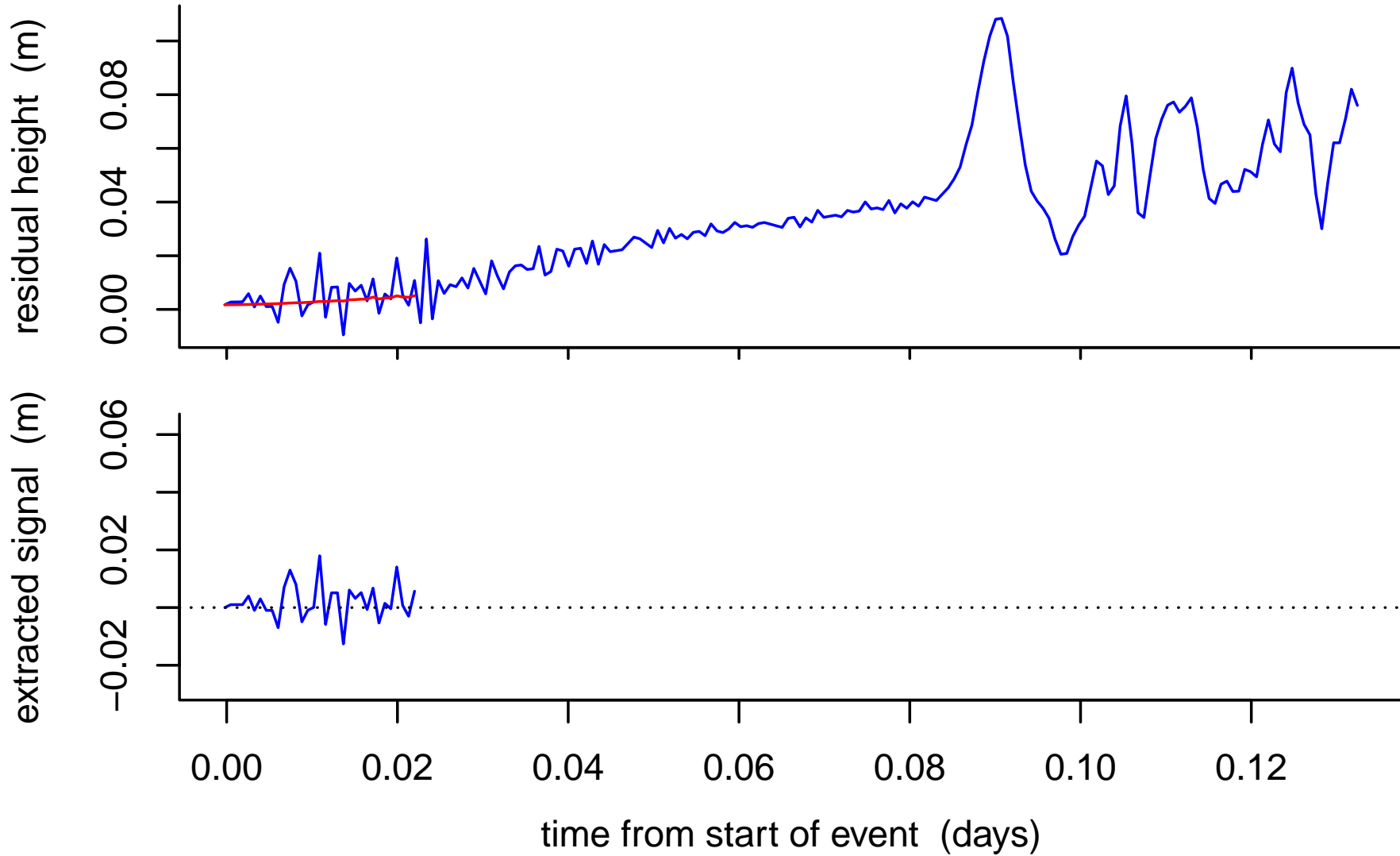
# Kalman Filtering/Smoothing (Local Level Model)



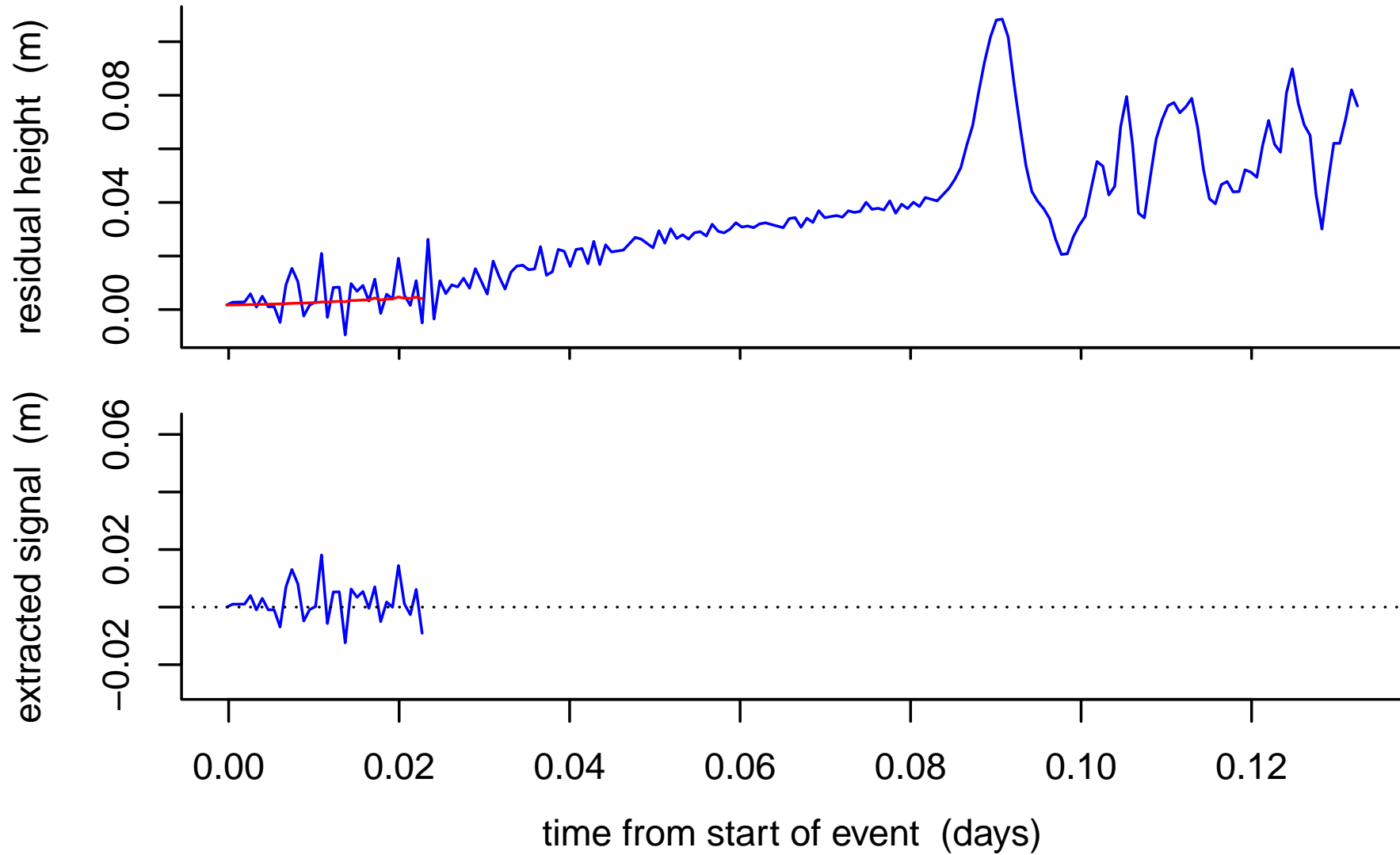
# Kalman Filtering/Smoothing (Local Level Model)



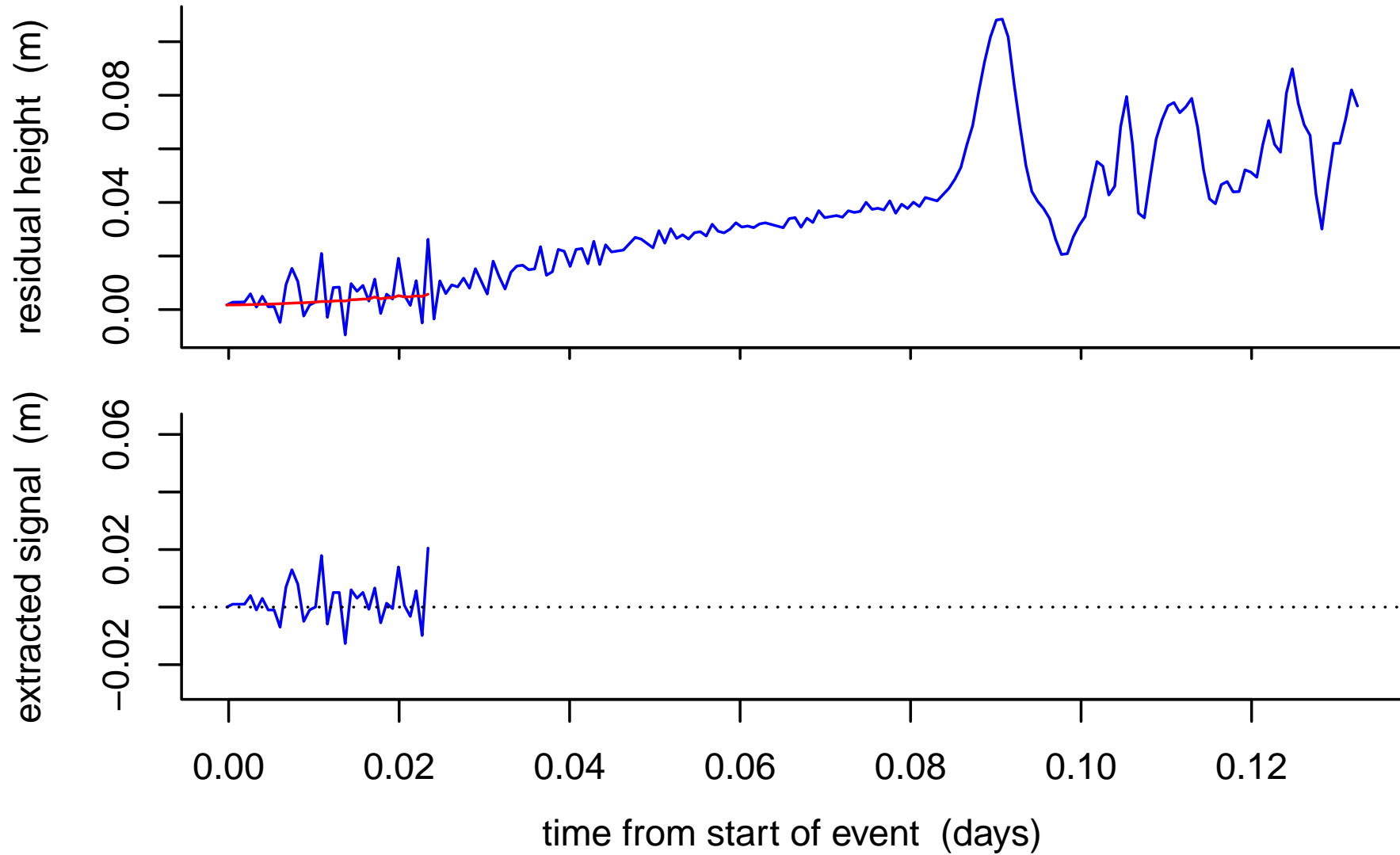
# Kalman Filtering/Smoothing (Local Level Model)



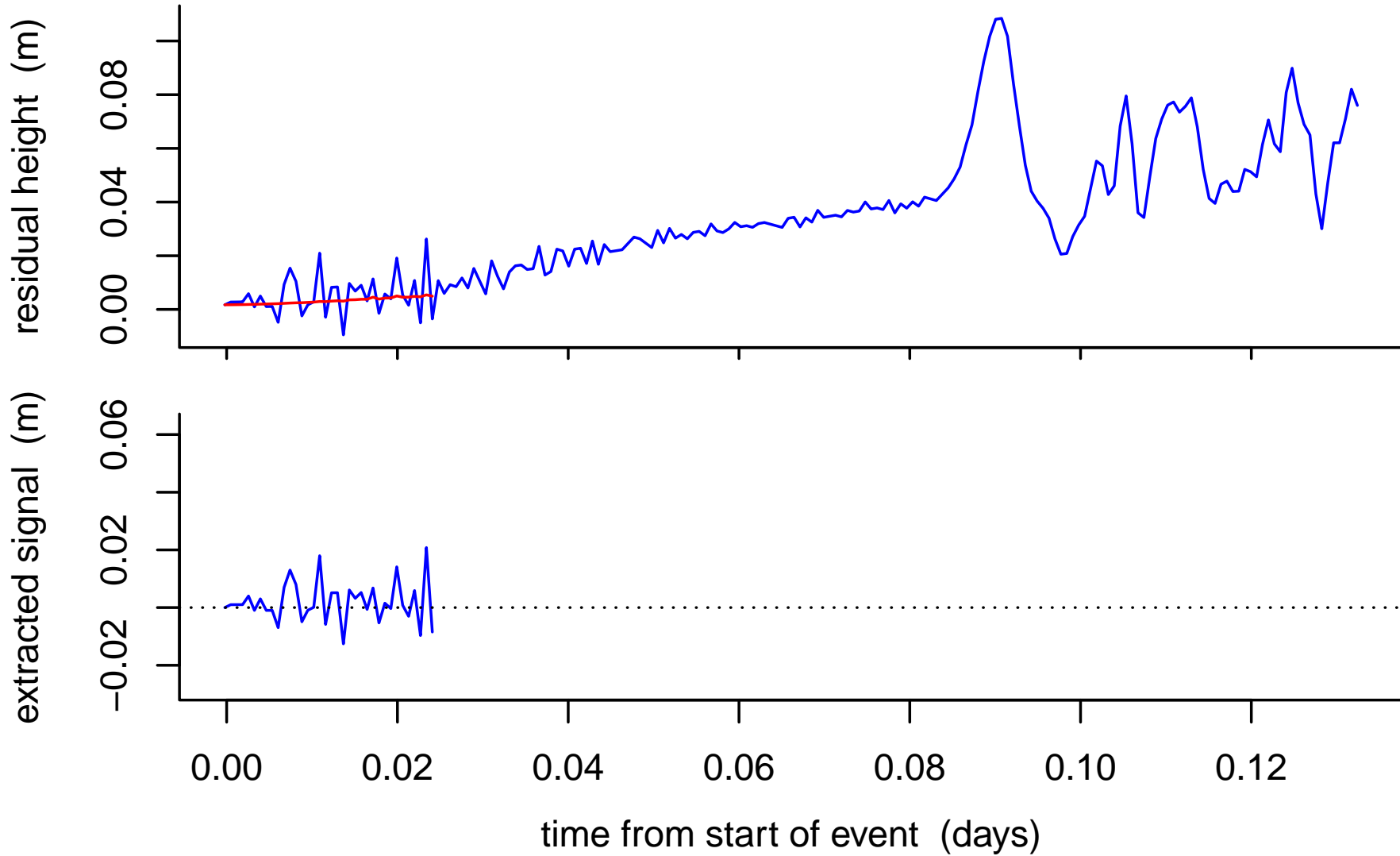
# Kalman Filtering/Smoothing (Local Level Model)



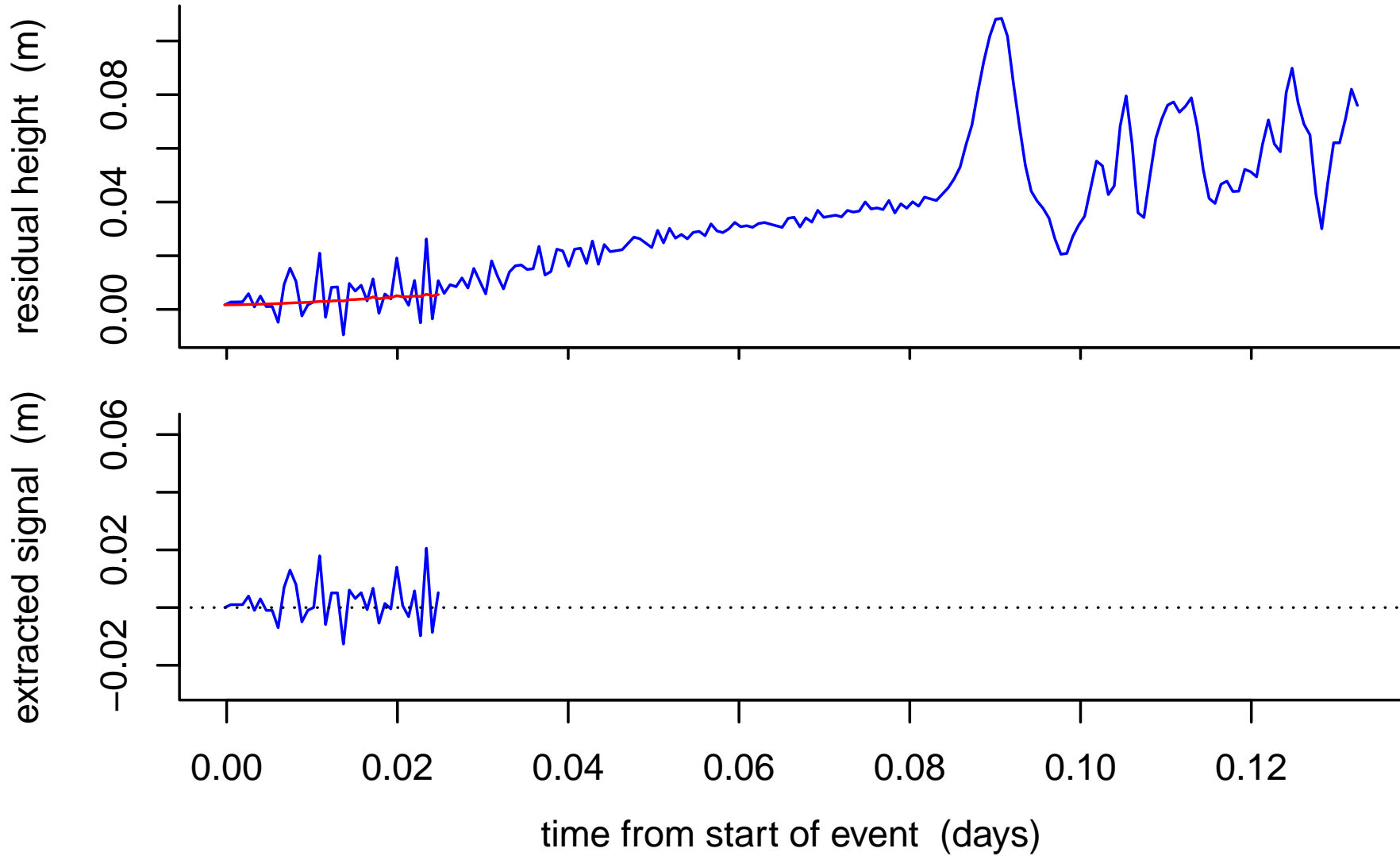
# Kalman Filtering/Smoothing (Local Level Model)



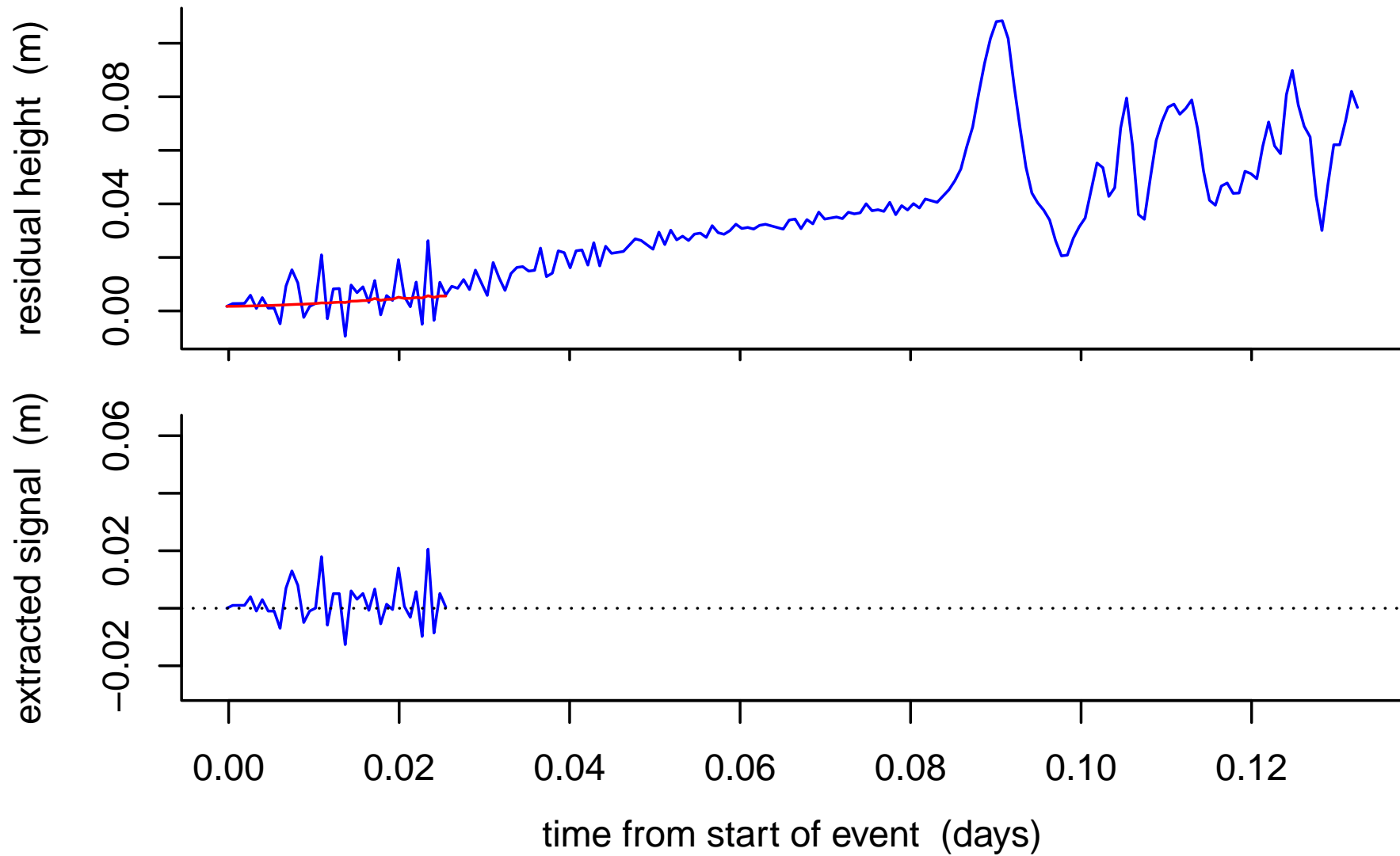
# Kalman Filtering/Smoothing (Local Level Model)



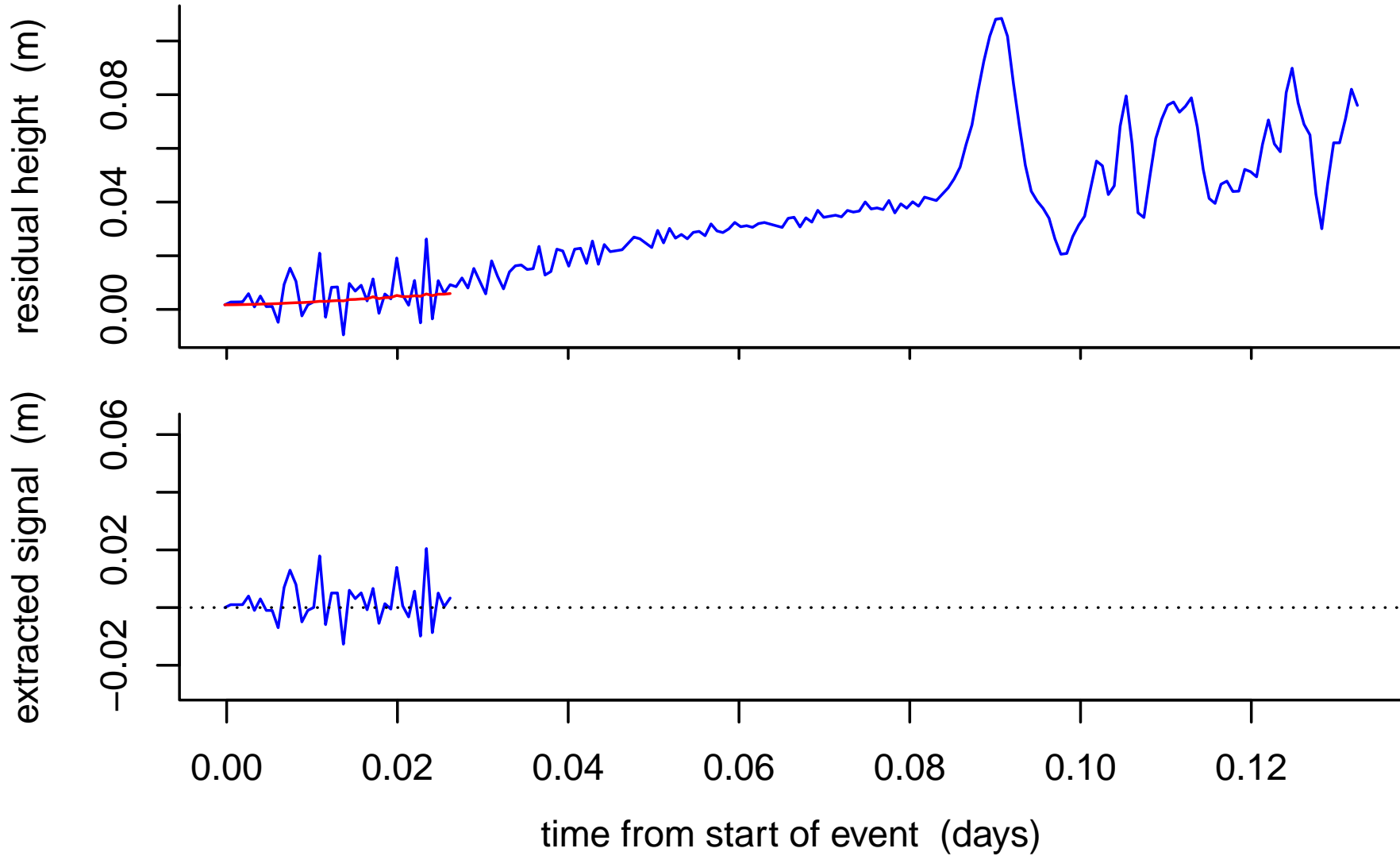
# Kalman Filtering/Smoothing (Local Level Model)



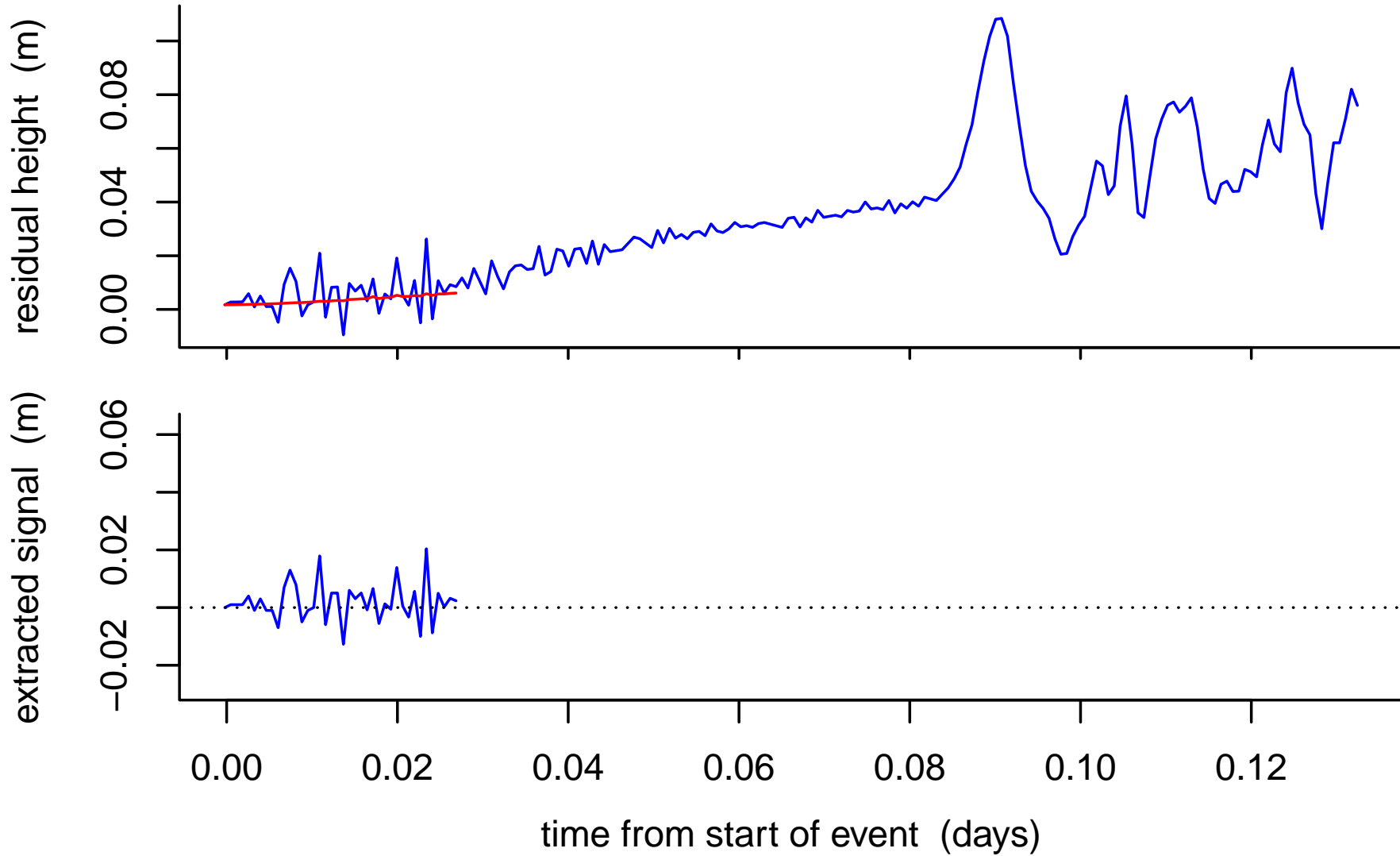
# Kalman Filtering/Smoothing (Local Level Model)



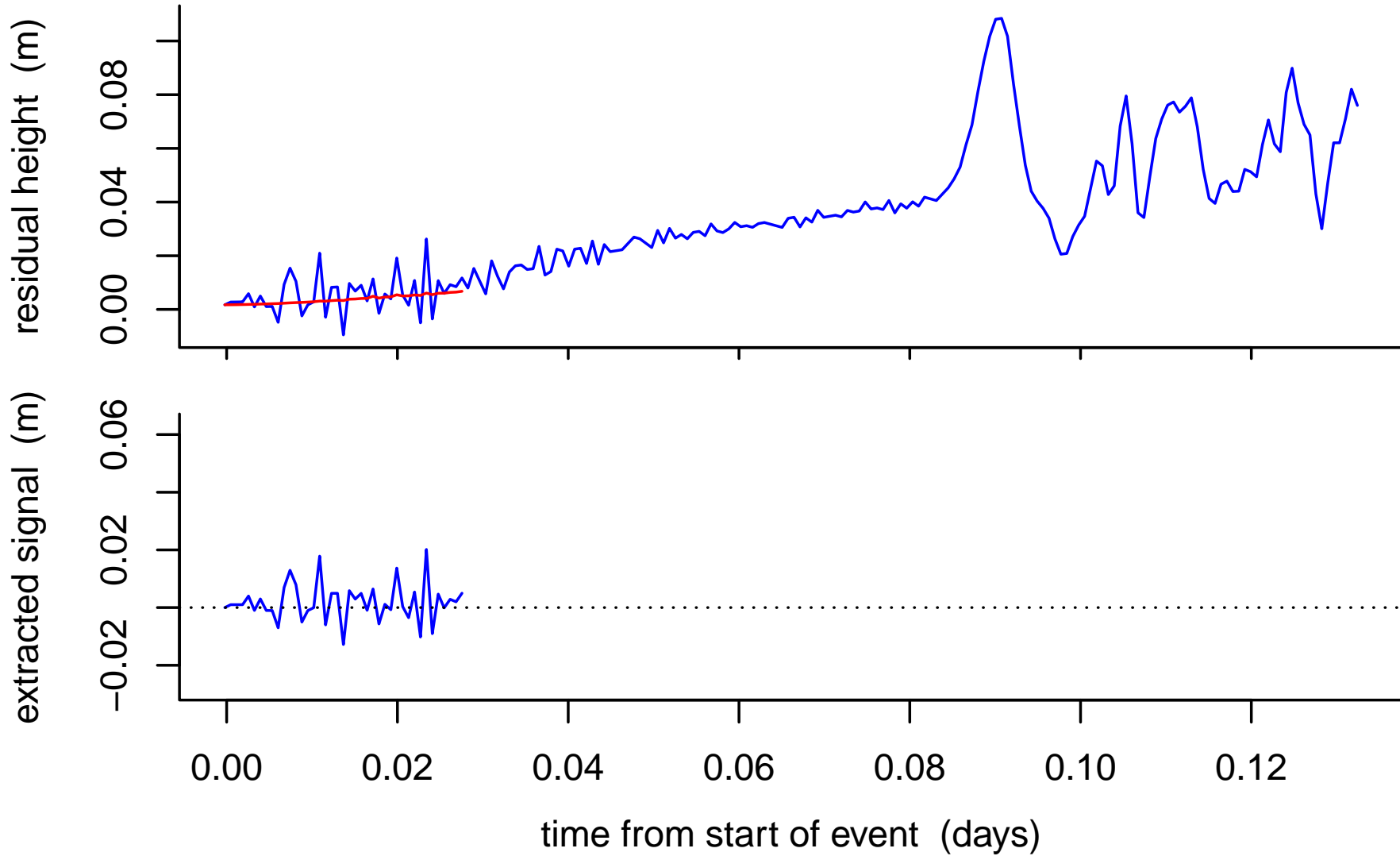
# Kalman Filtering/Smoothing (Local Level Model)



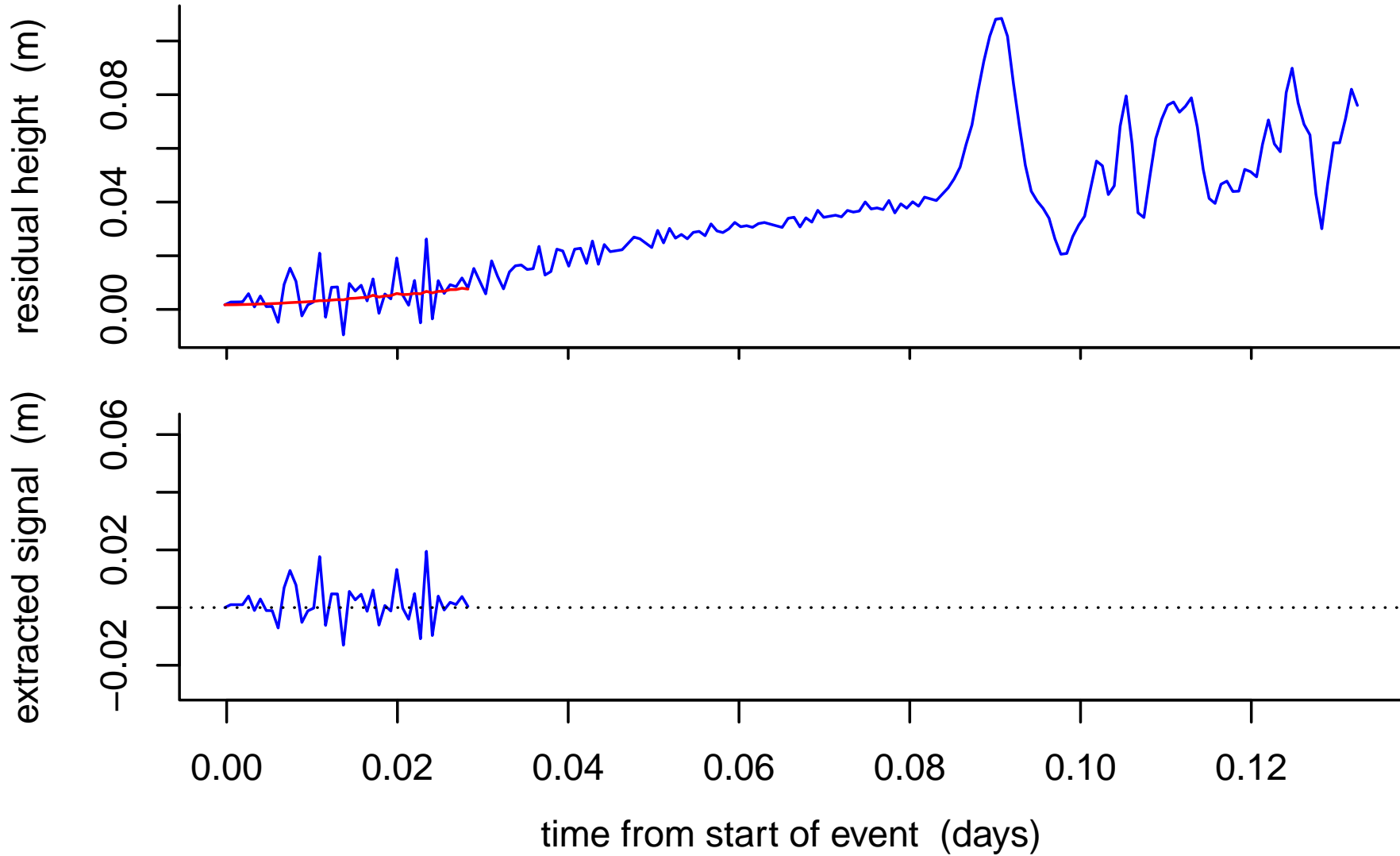
# Kalman Filtering/Smoothing (Local Level Model)



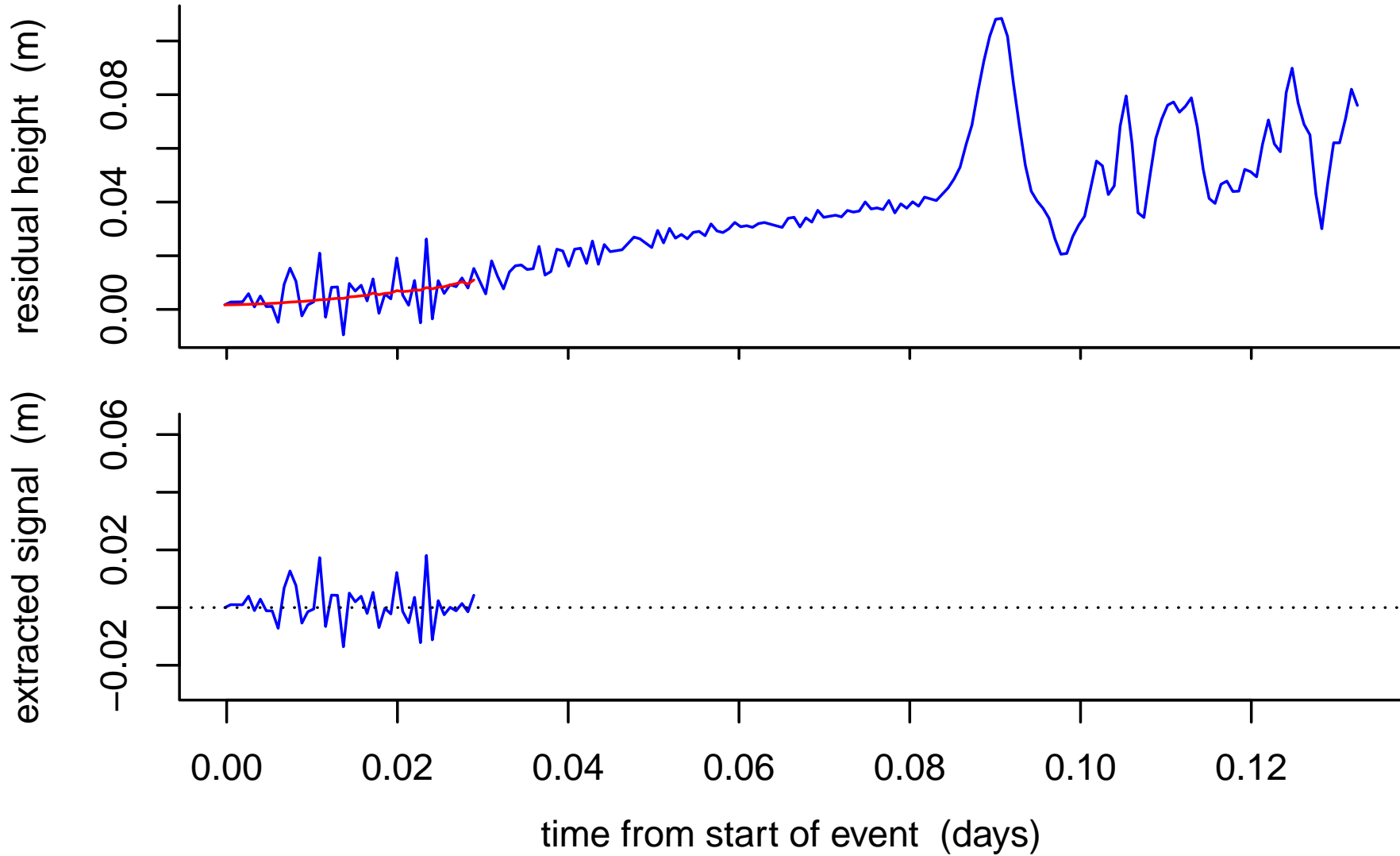
# Kalman Filtering/Smoothing (Local Level Model)



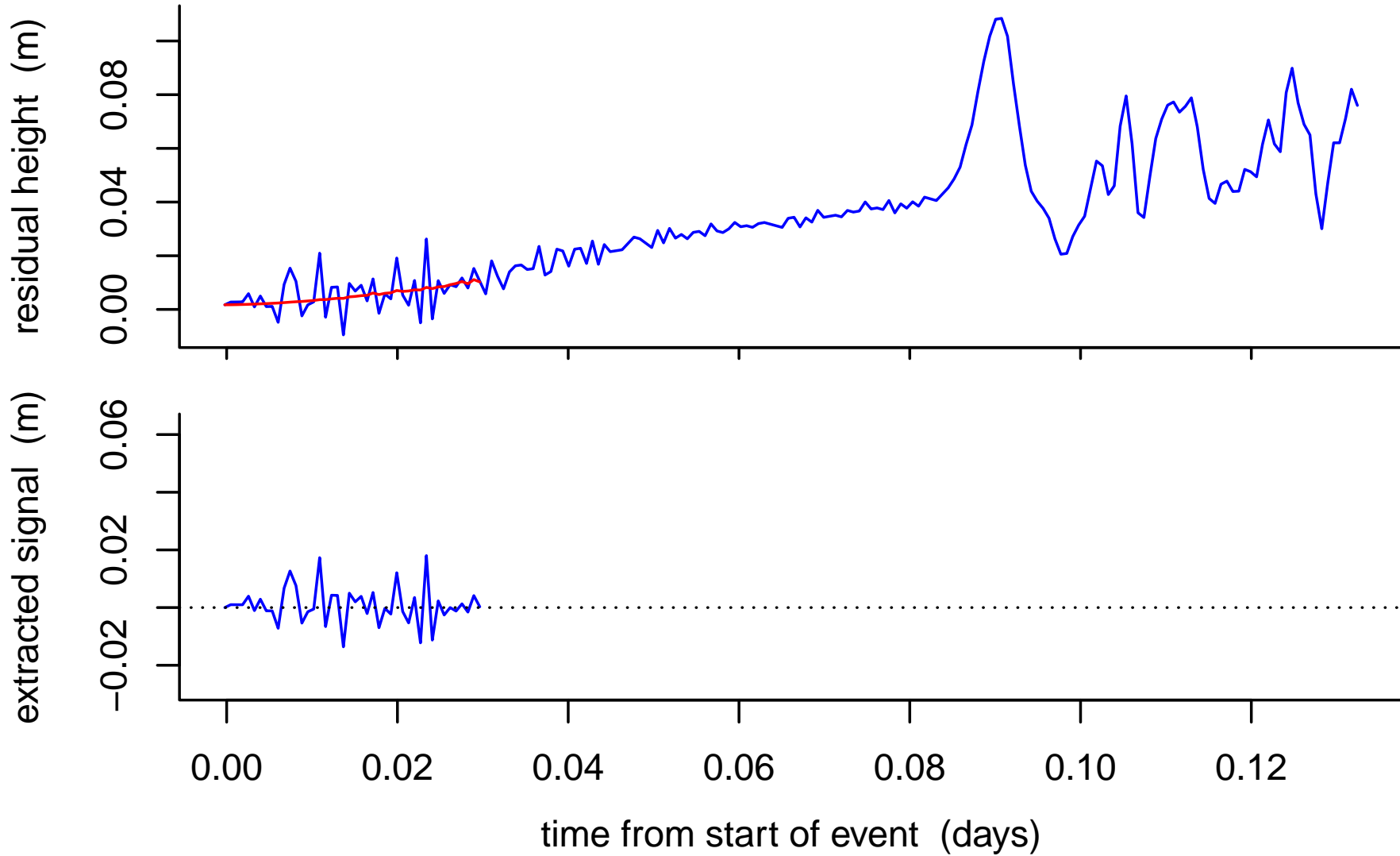
# Kalman Filtering/Smoothing (Local Level Model)



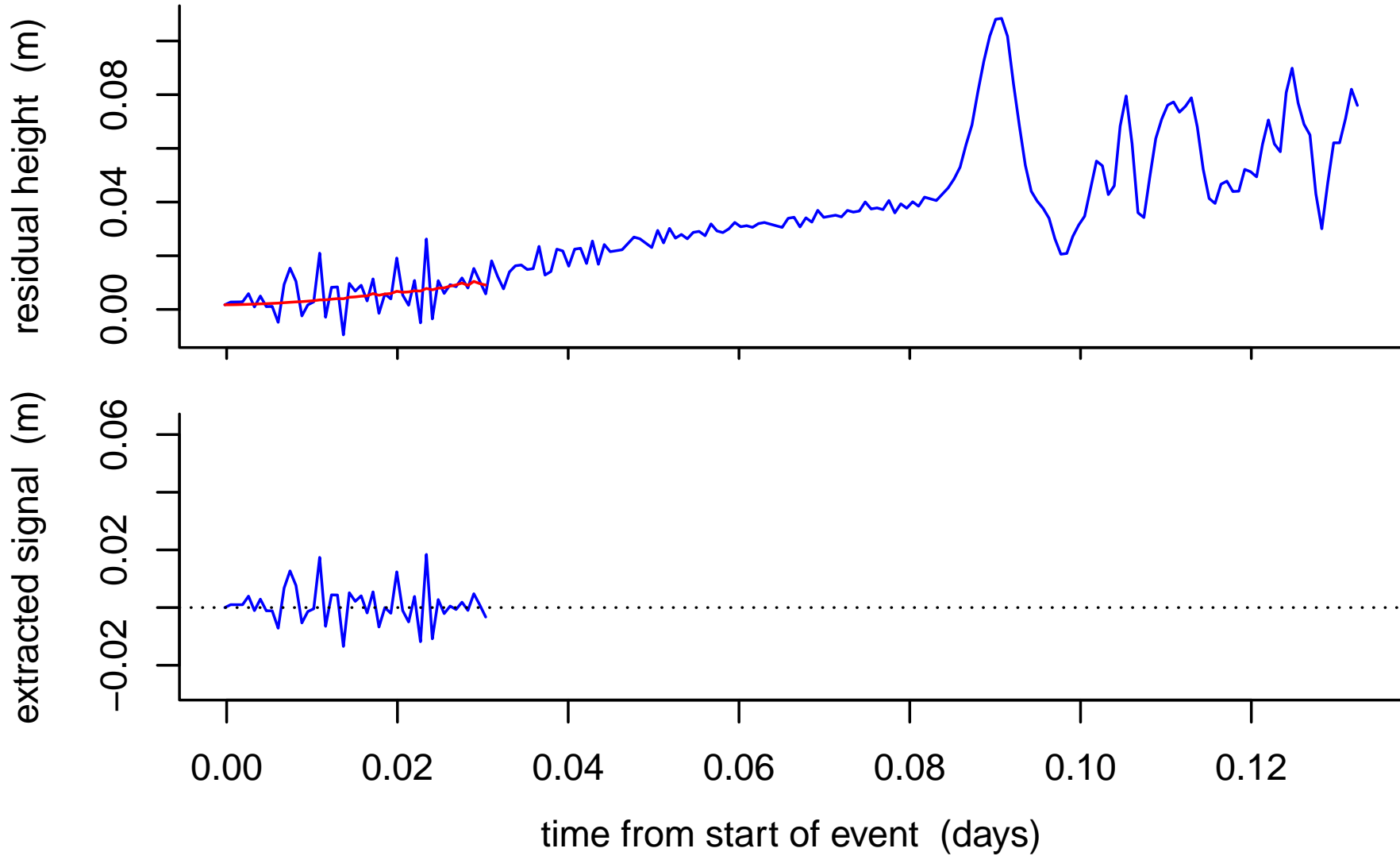
# Kalman Filtering/Smoothing (Local Level Model)



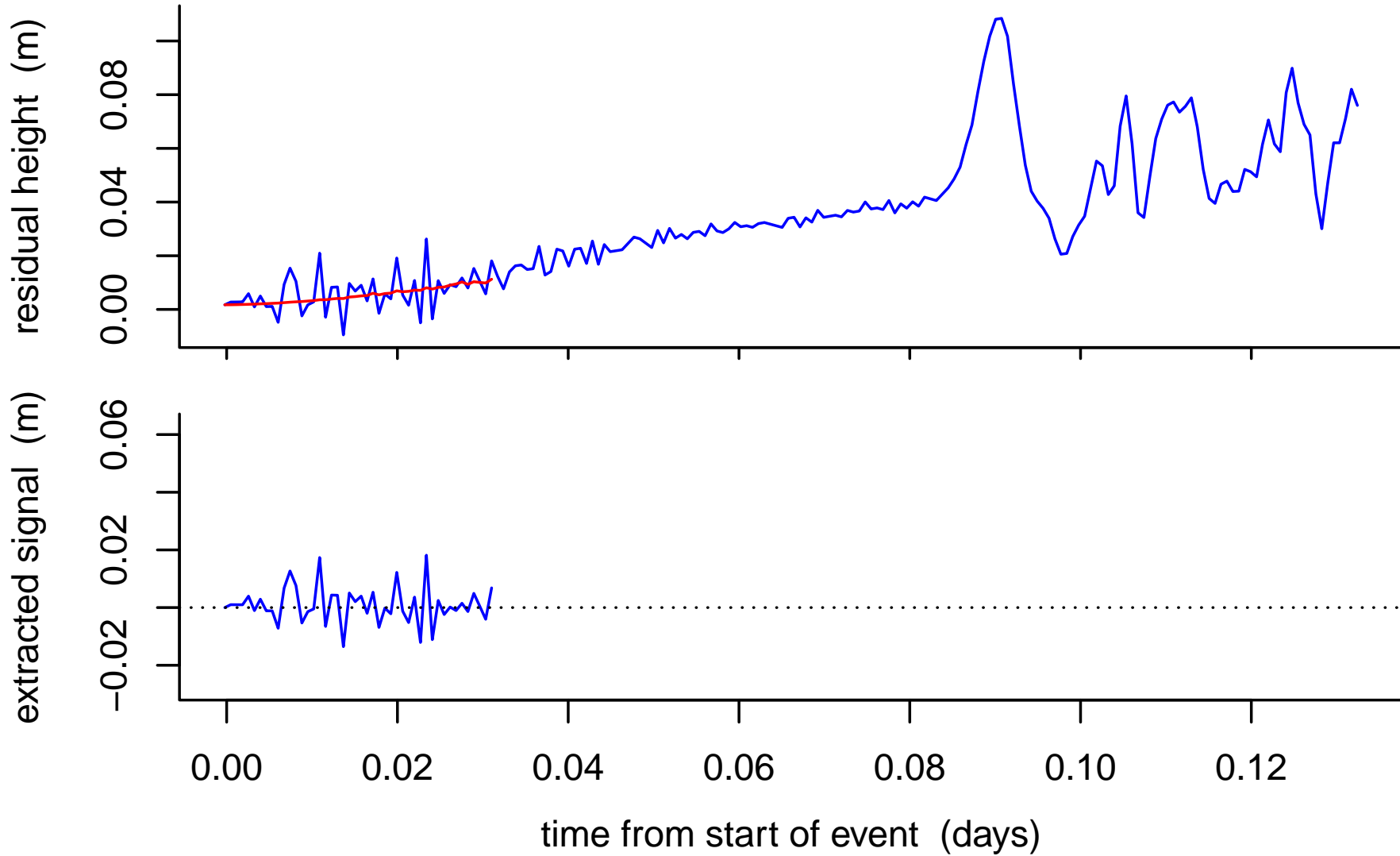
# Kalman Filtering/Smoothing (Local Level Model)



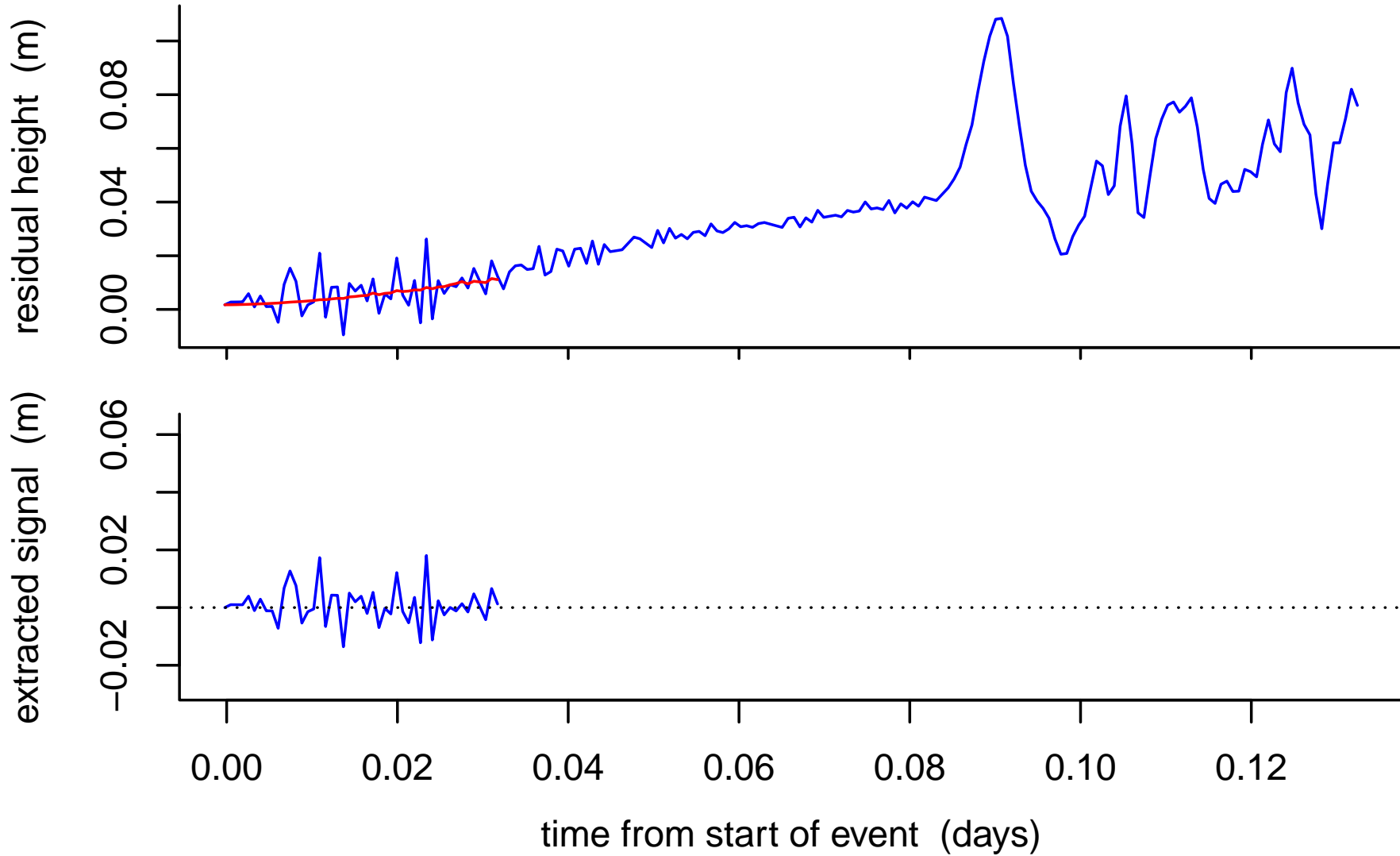
# Kalman Filtering/Smoothing (Local Level Model)



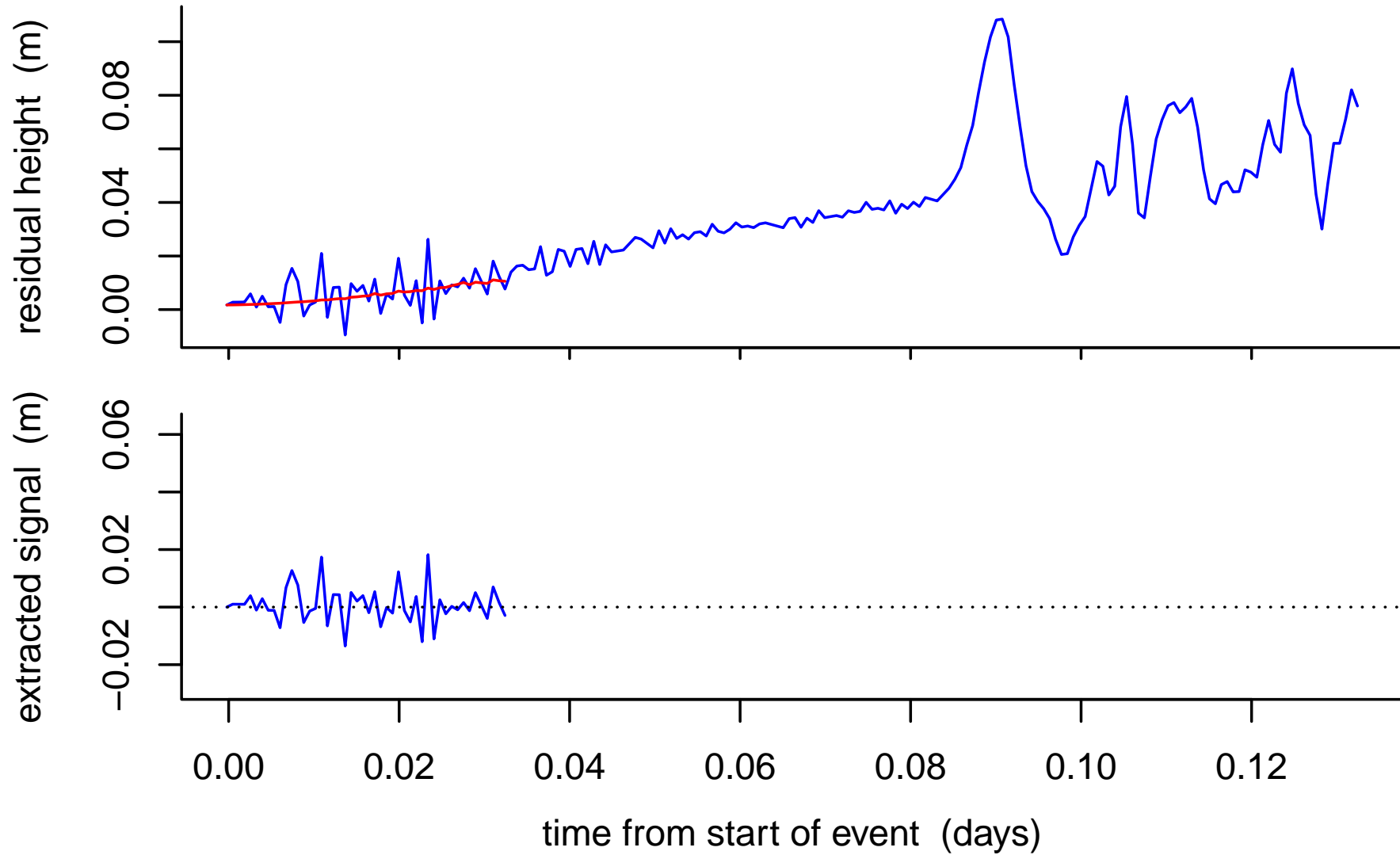
# Kalman Filtering/Smoothing (Local Level Model)



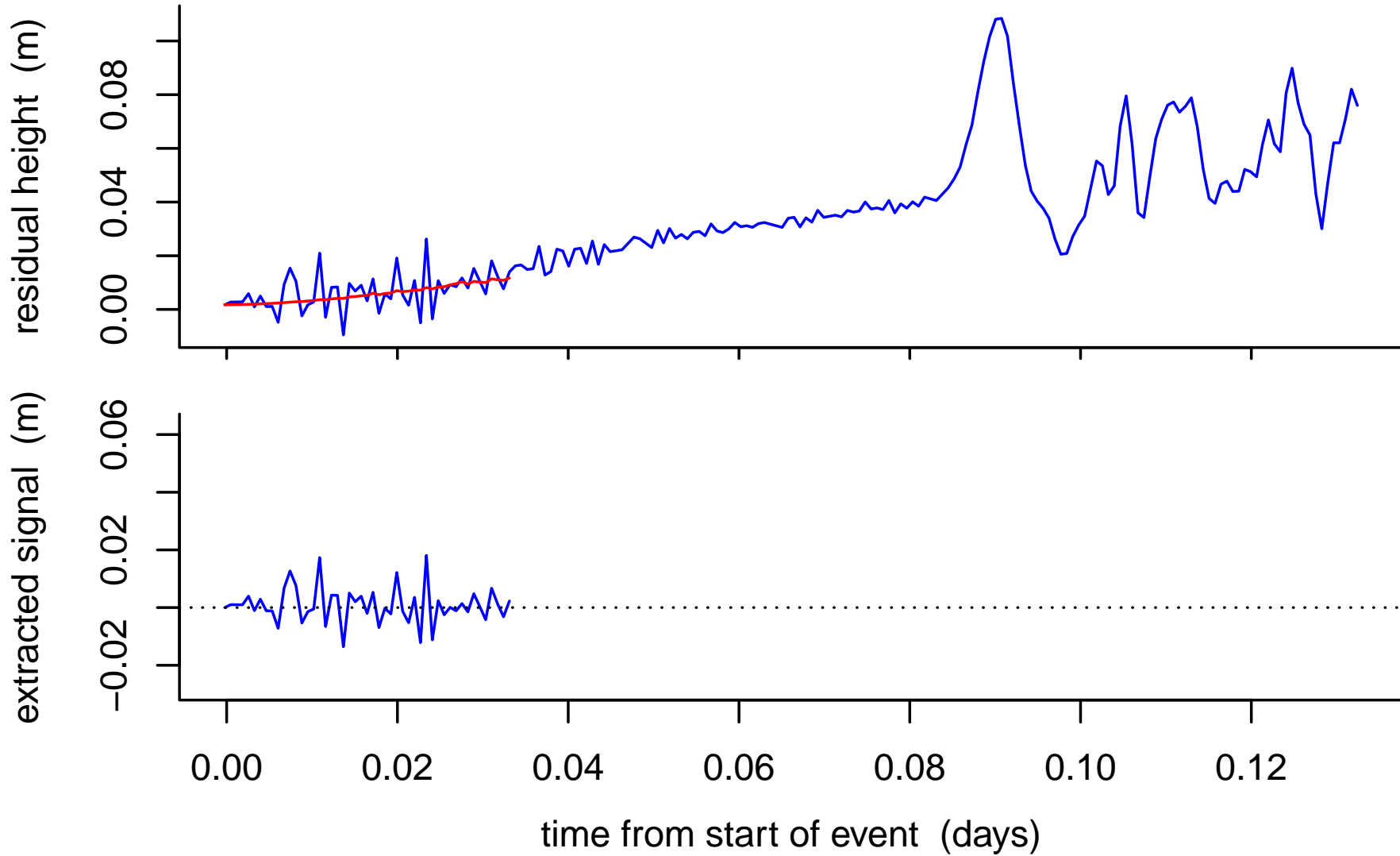
# Kalman Filtering/Smoothing (Local Level Model)



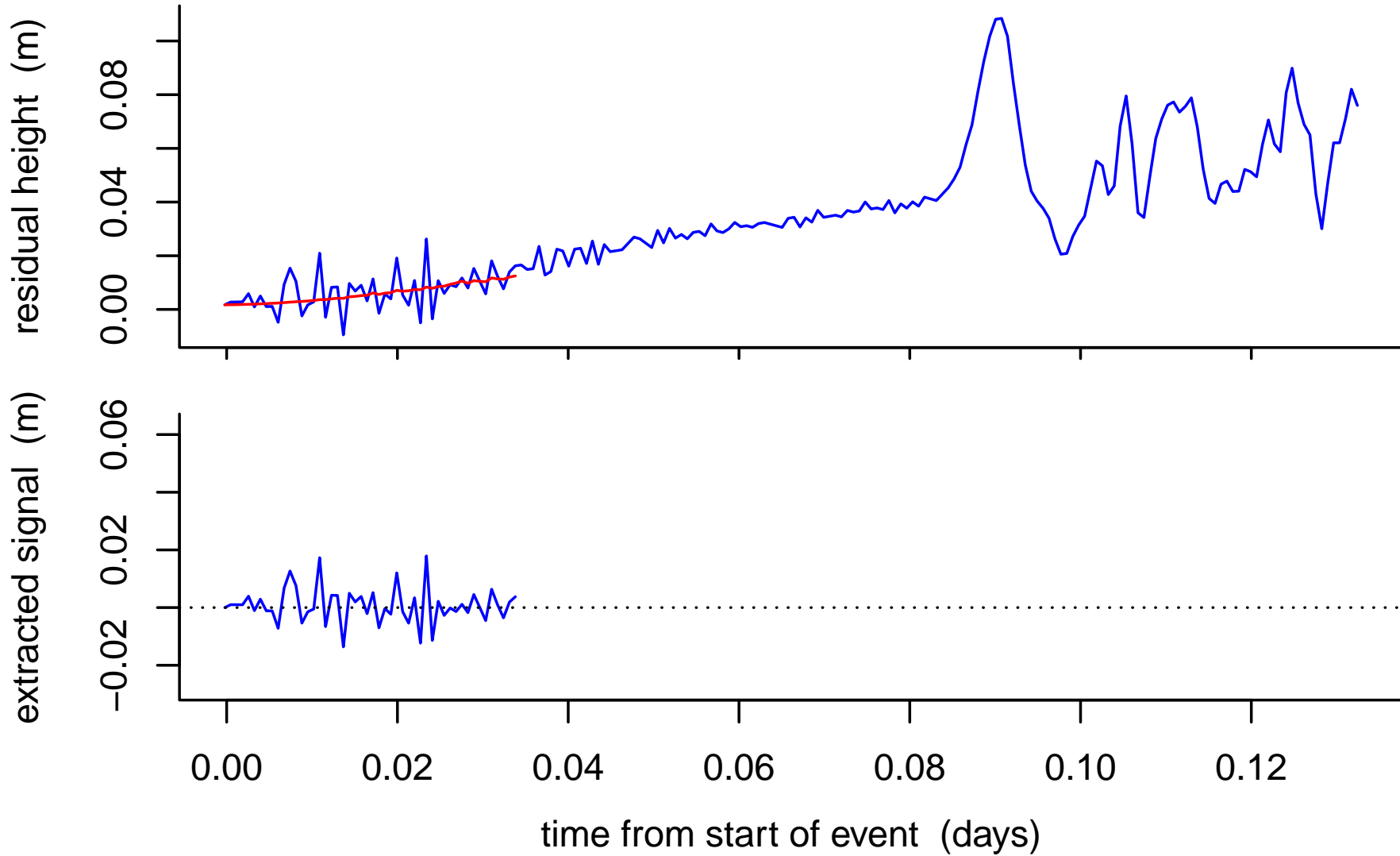
# Kalman Filtering/Smoothing (Local Level Model)



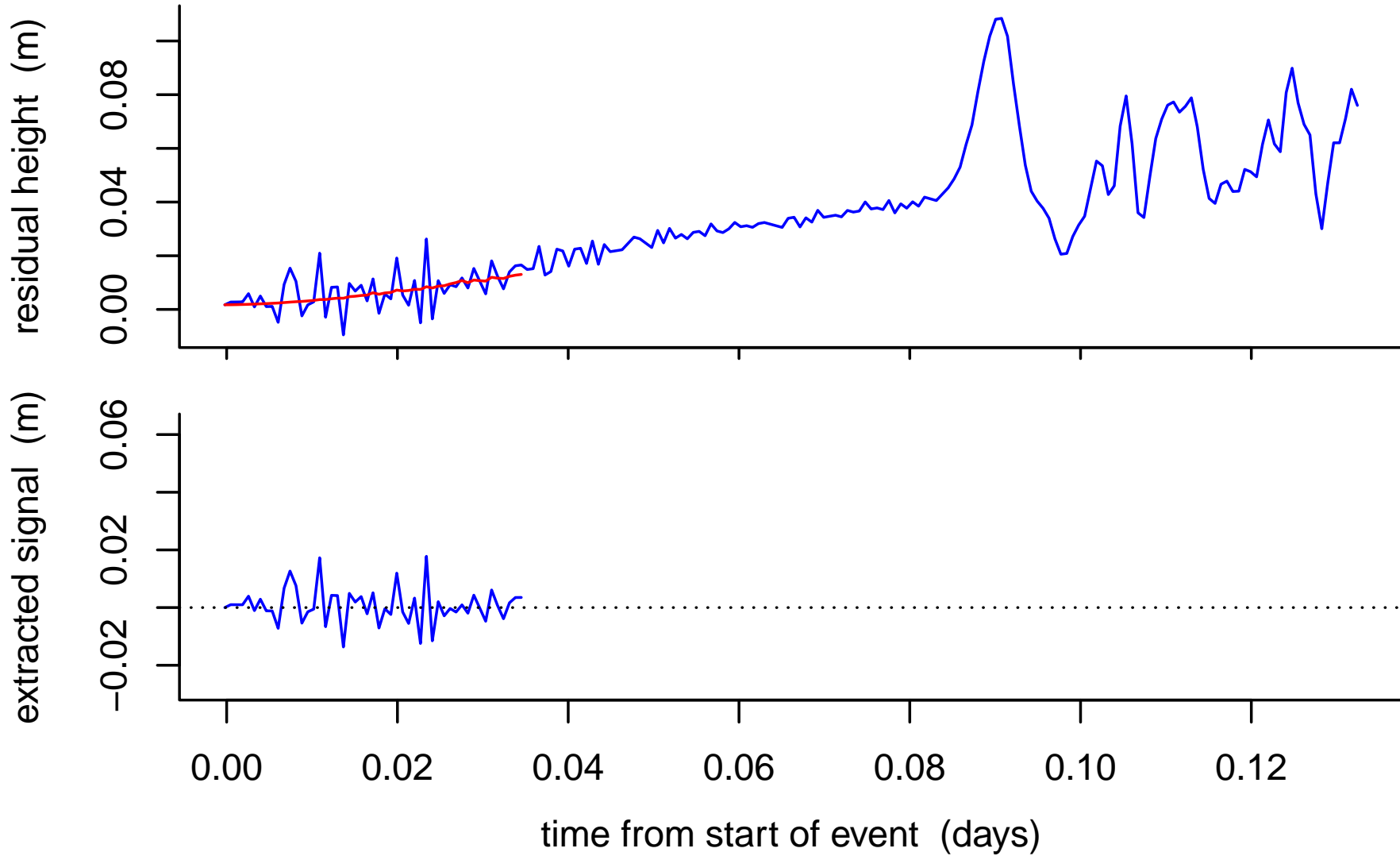
# Kalman Filtering/Smoothing (Local Level Model)



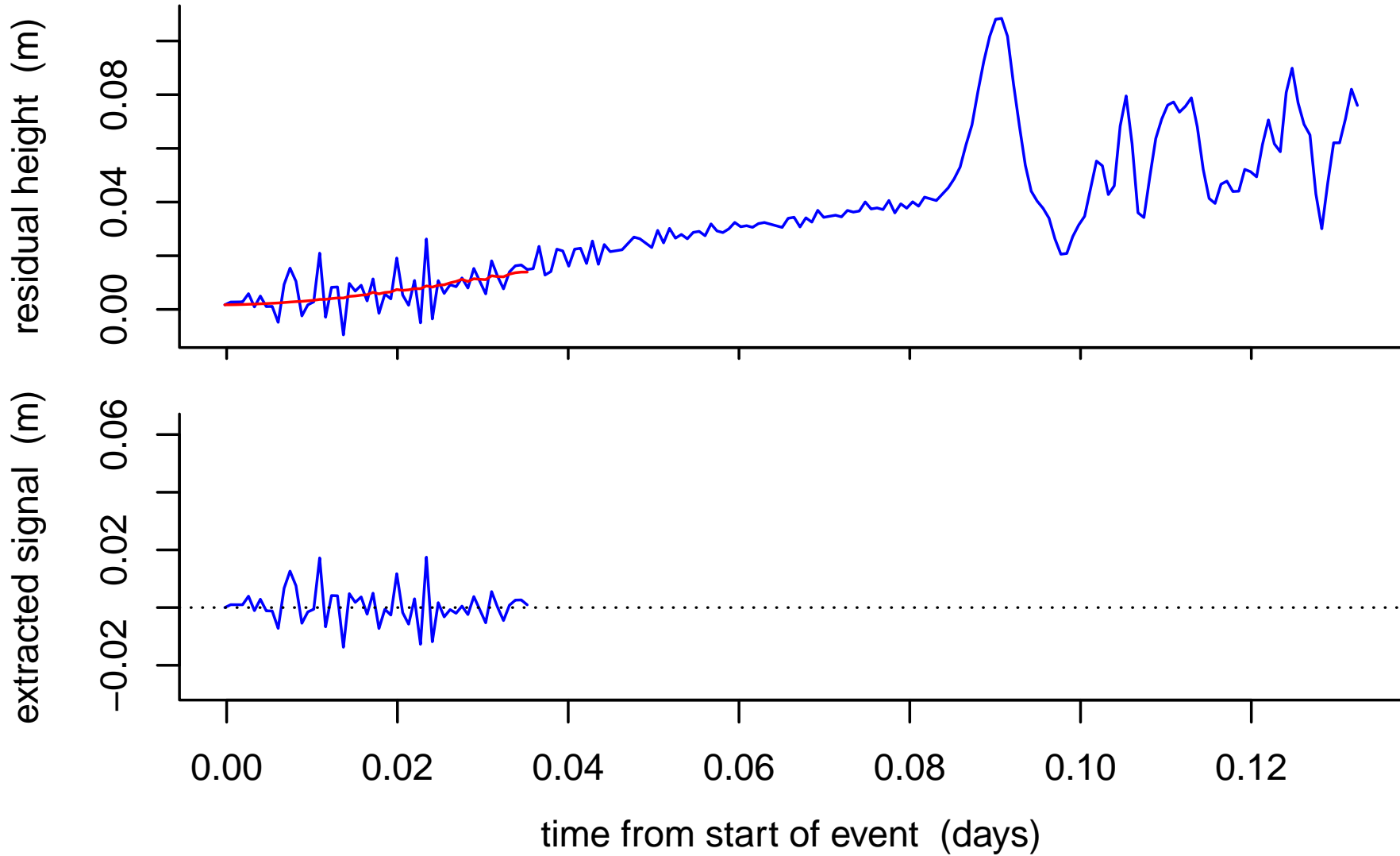
# Kalman Filtering/Smoothing (Local Level Model)



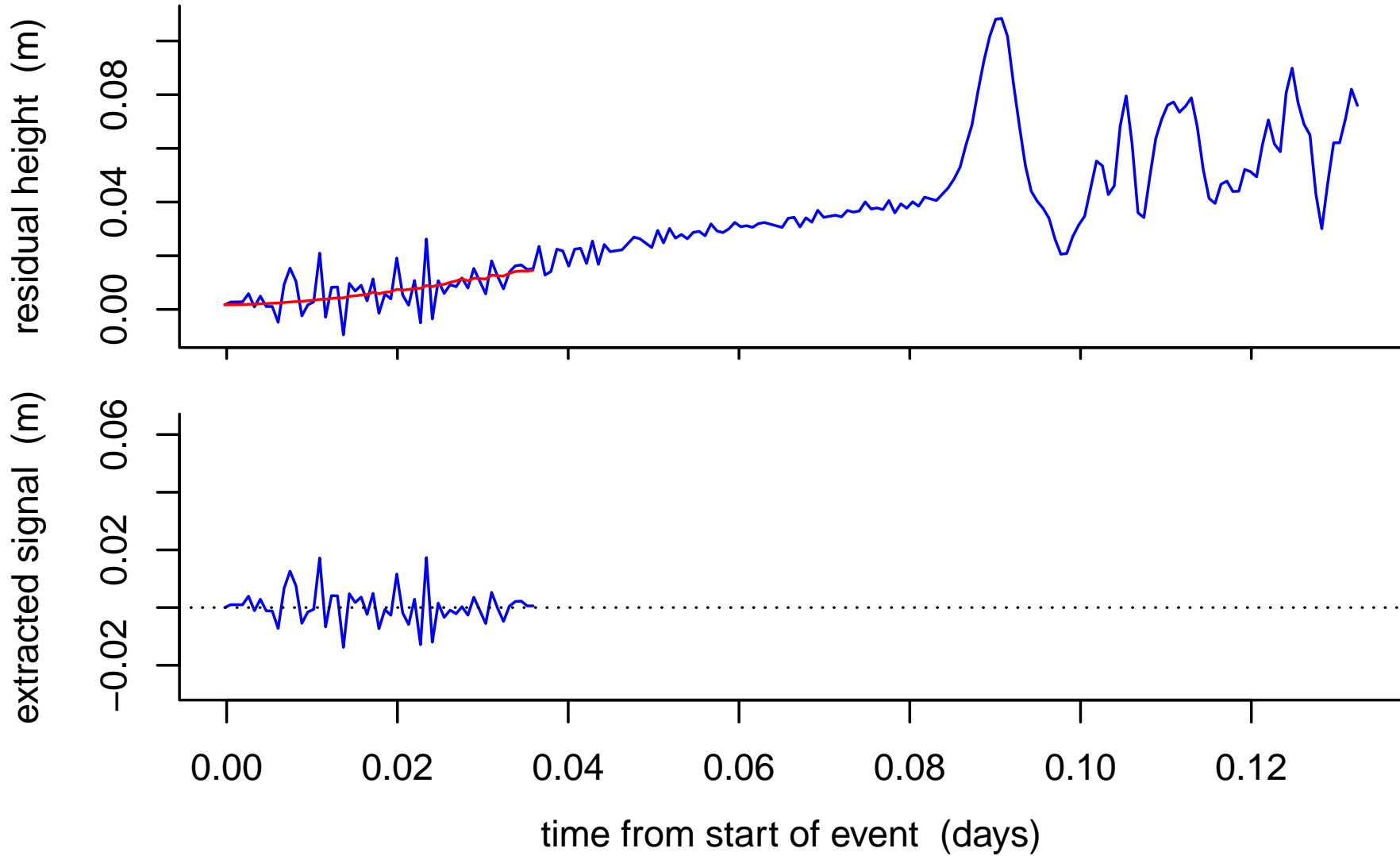
# Kalman Filtering/Smoothing (Local Level Model)



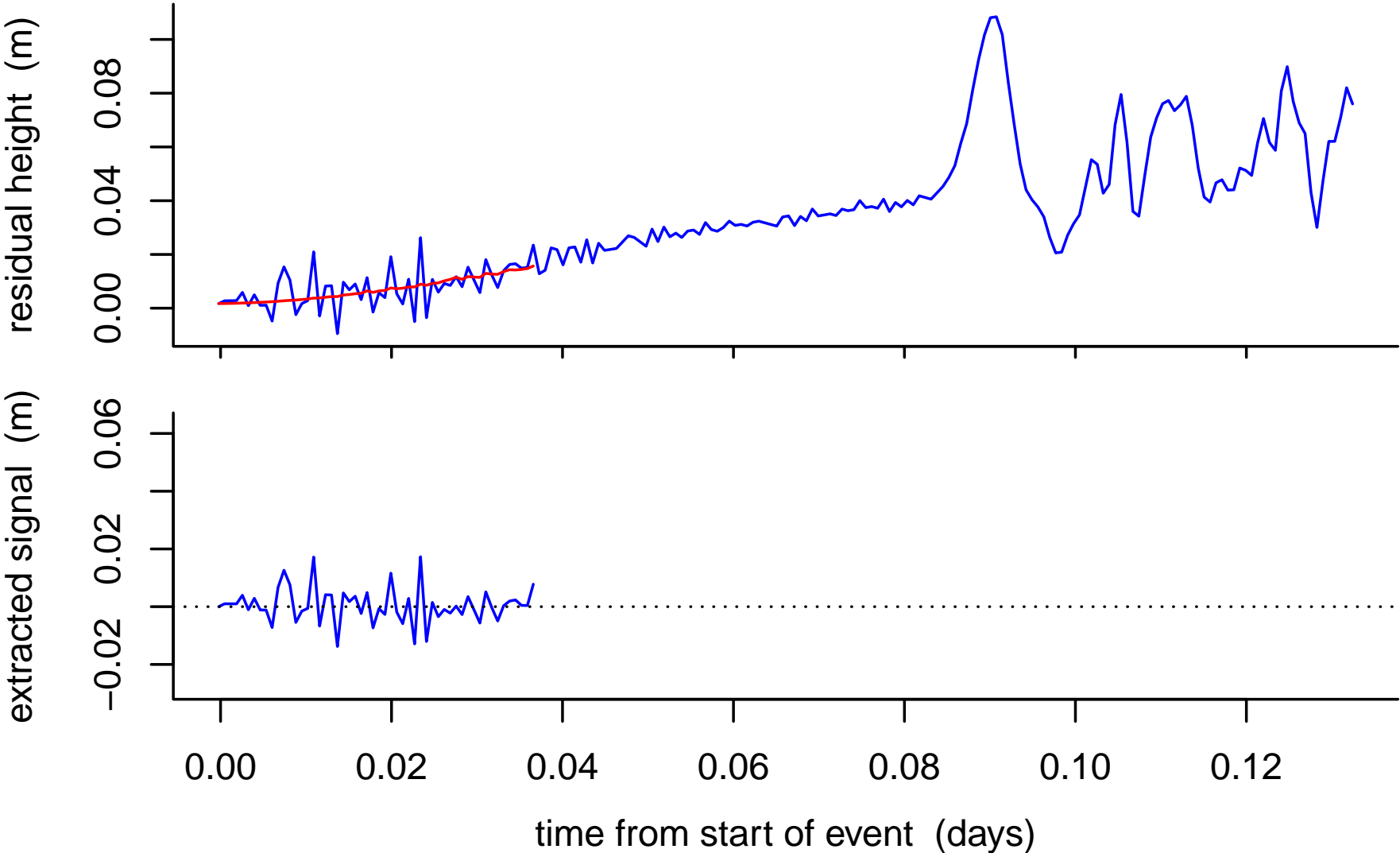
# Kalman Filtering/Smoothing (Local Level Model)



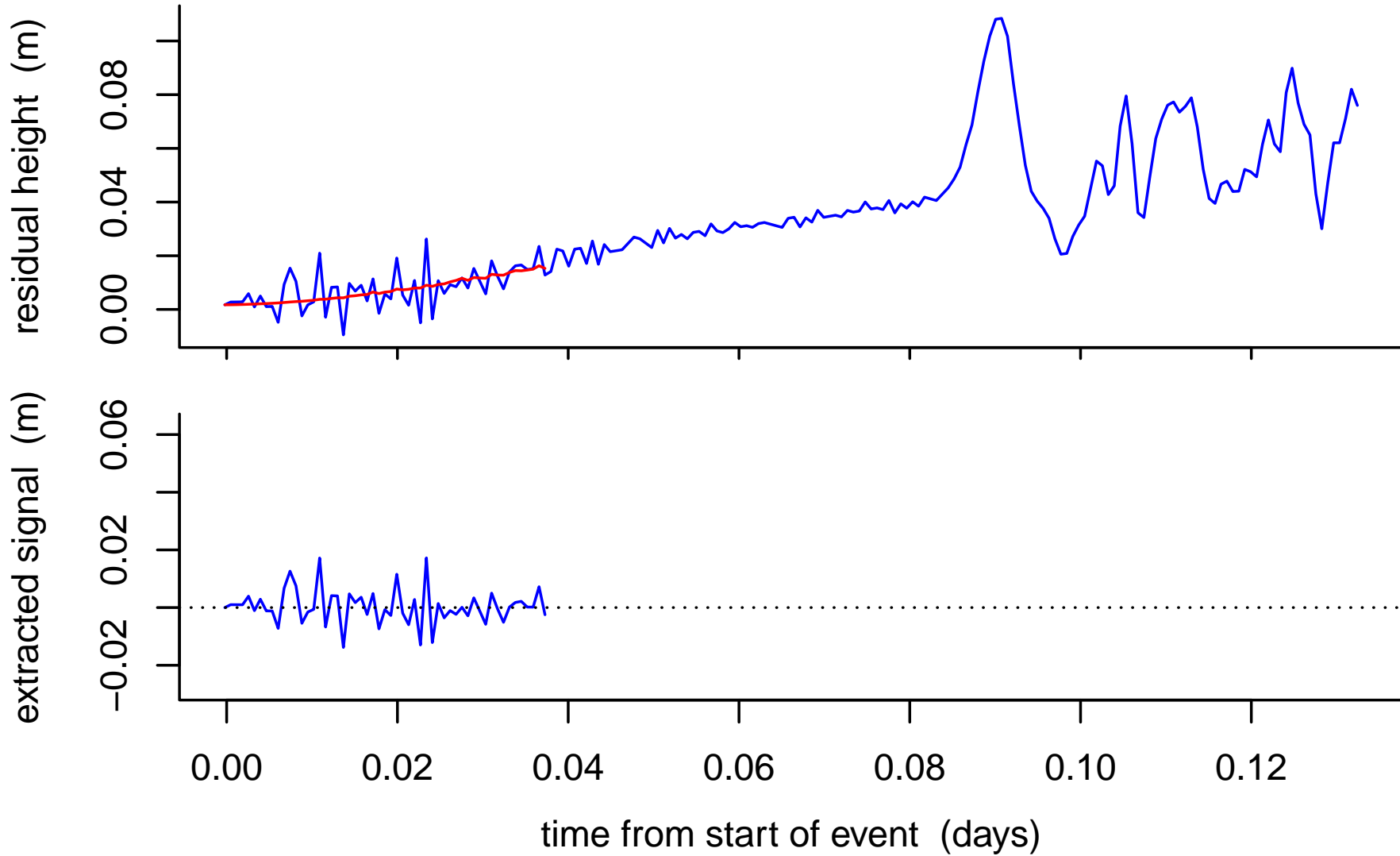
# Kalman Filtering/Smoothing (Local Level Model)



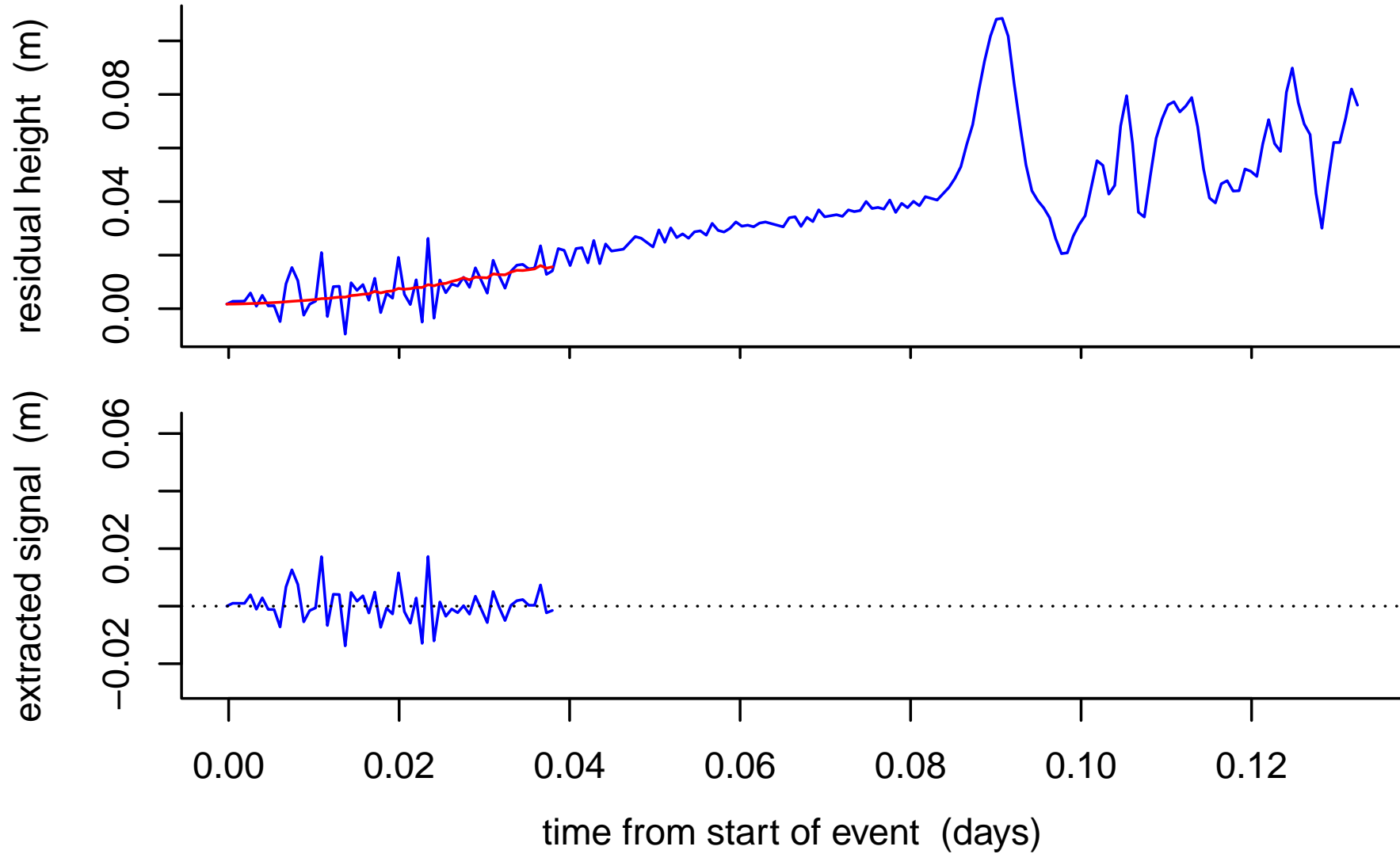
# Kalman Filtering/Smoothing (Local Level Model)



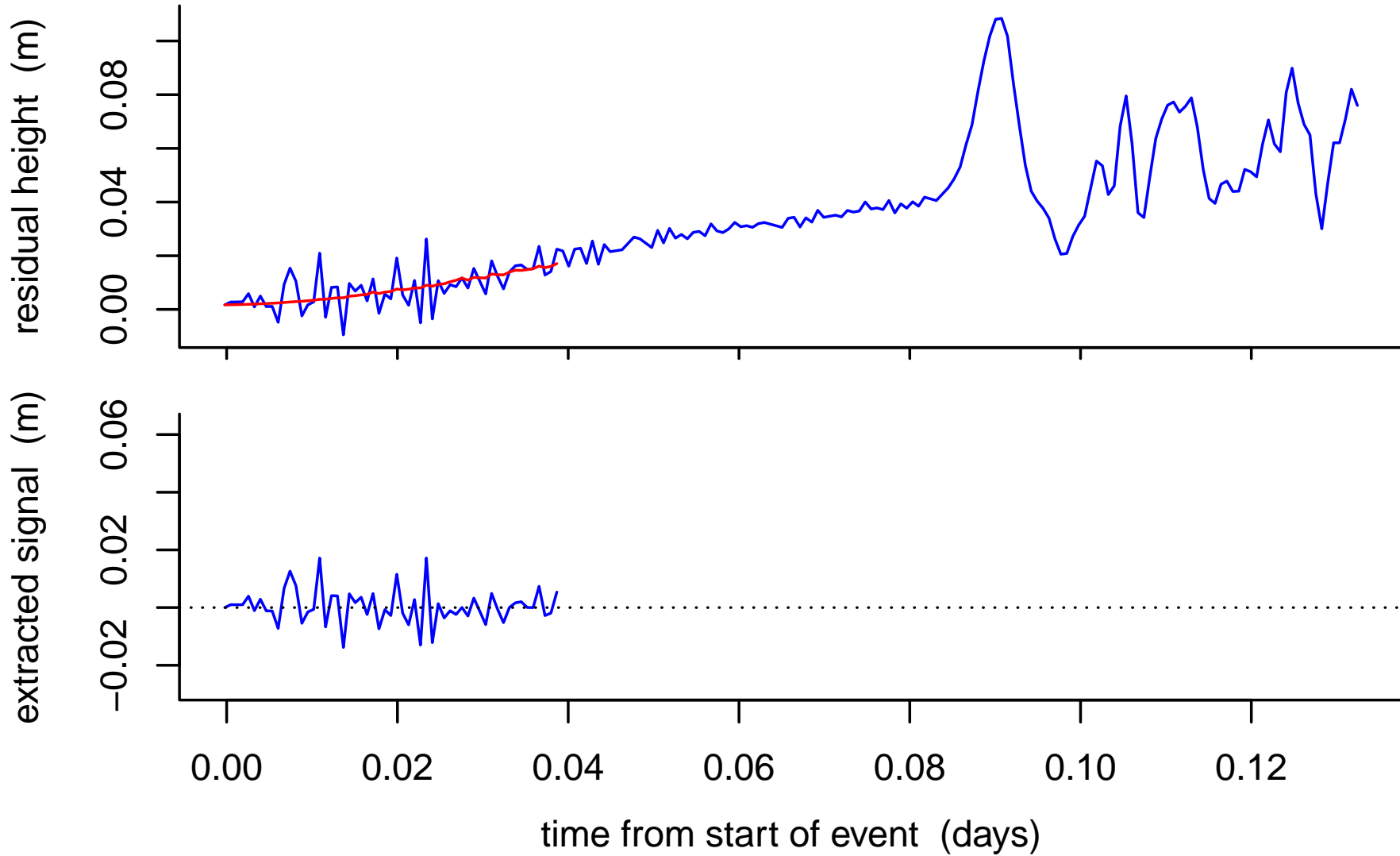
# Kalman Filtering/Smoothing (Local Level Model)



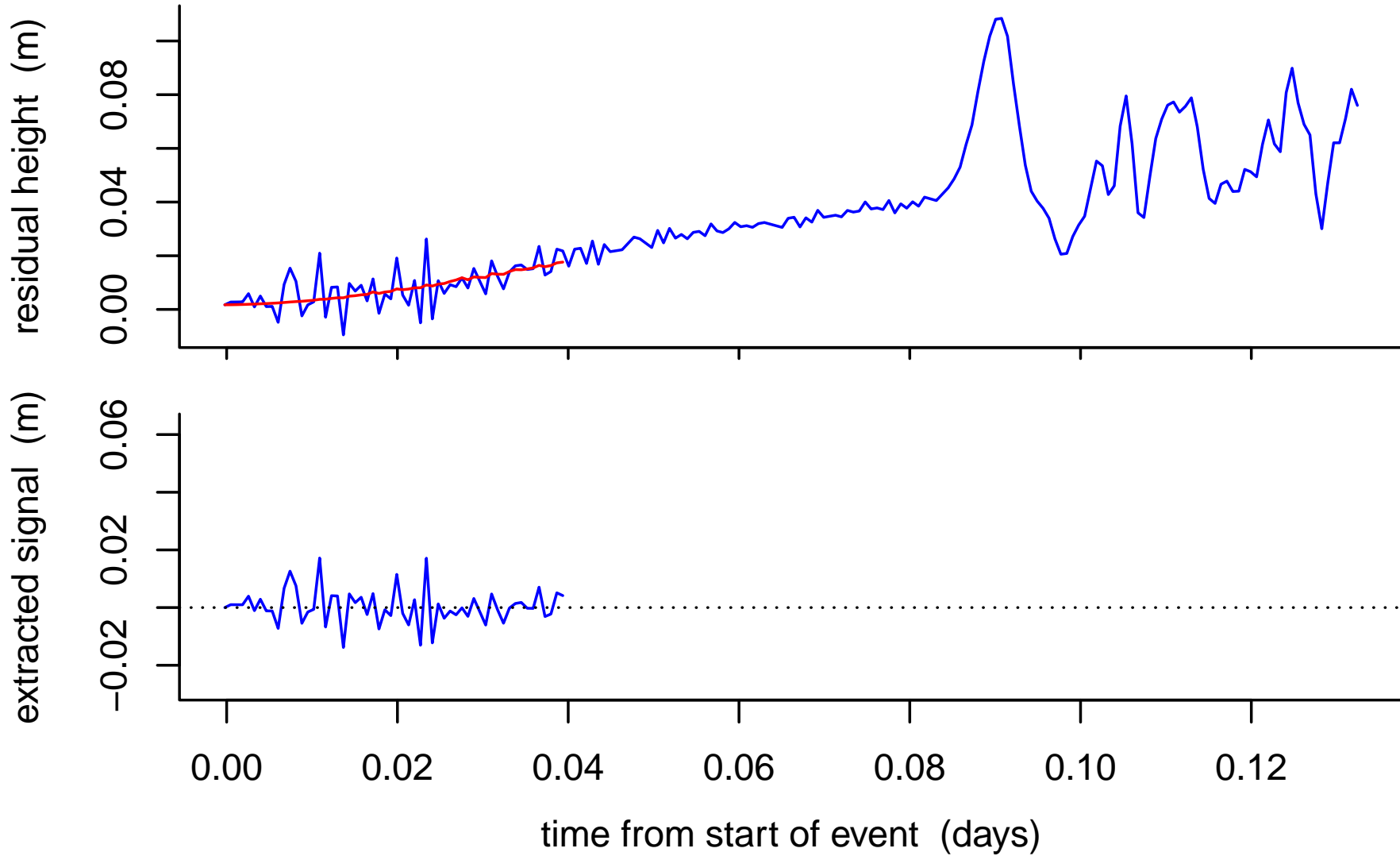
# Kalman Filtering/Smoothing (Local Level Model)



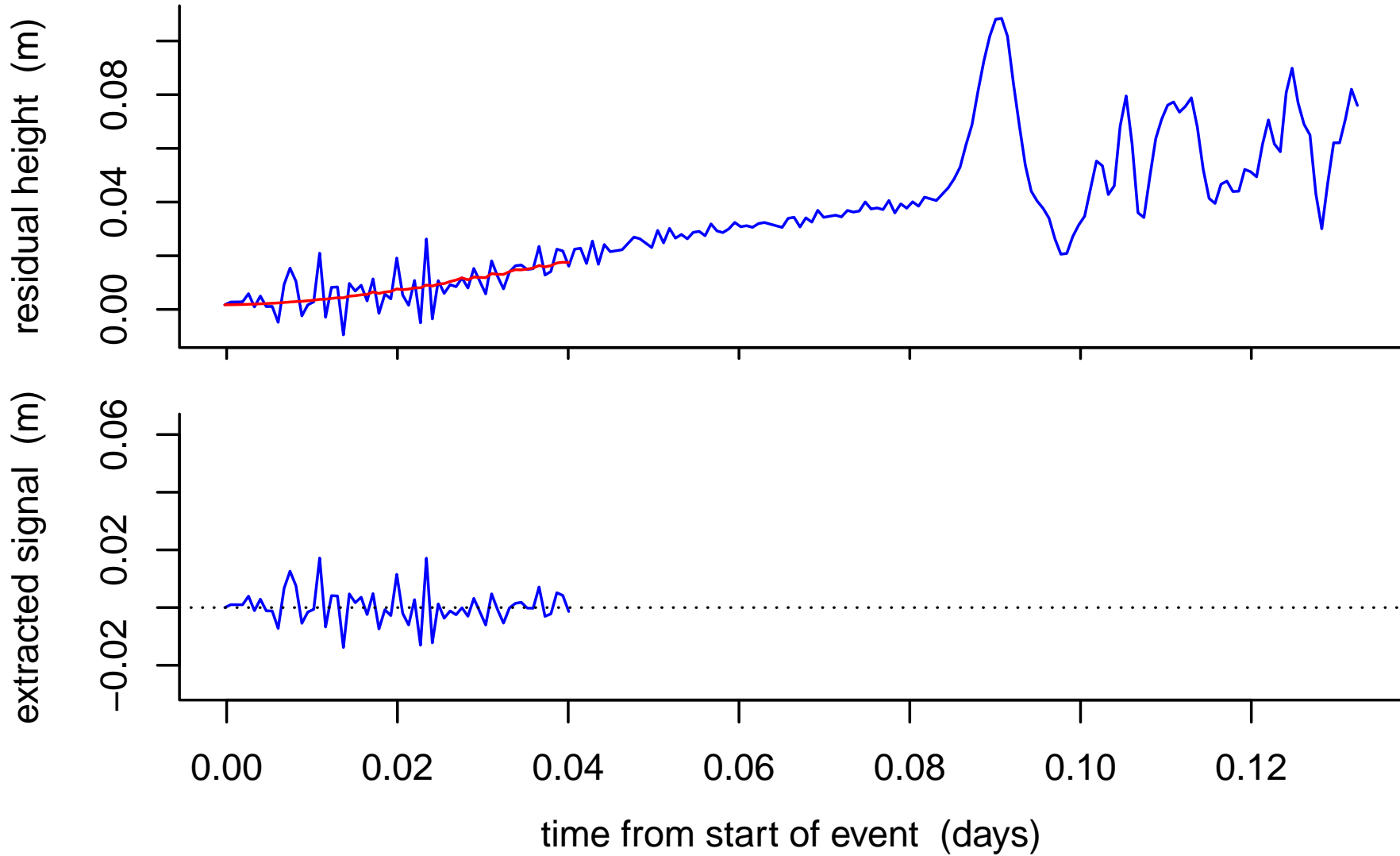
# Kalman Filtering/Smoothing (Local Level Model)



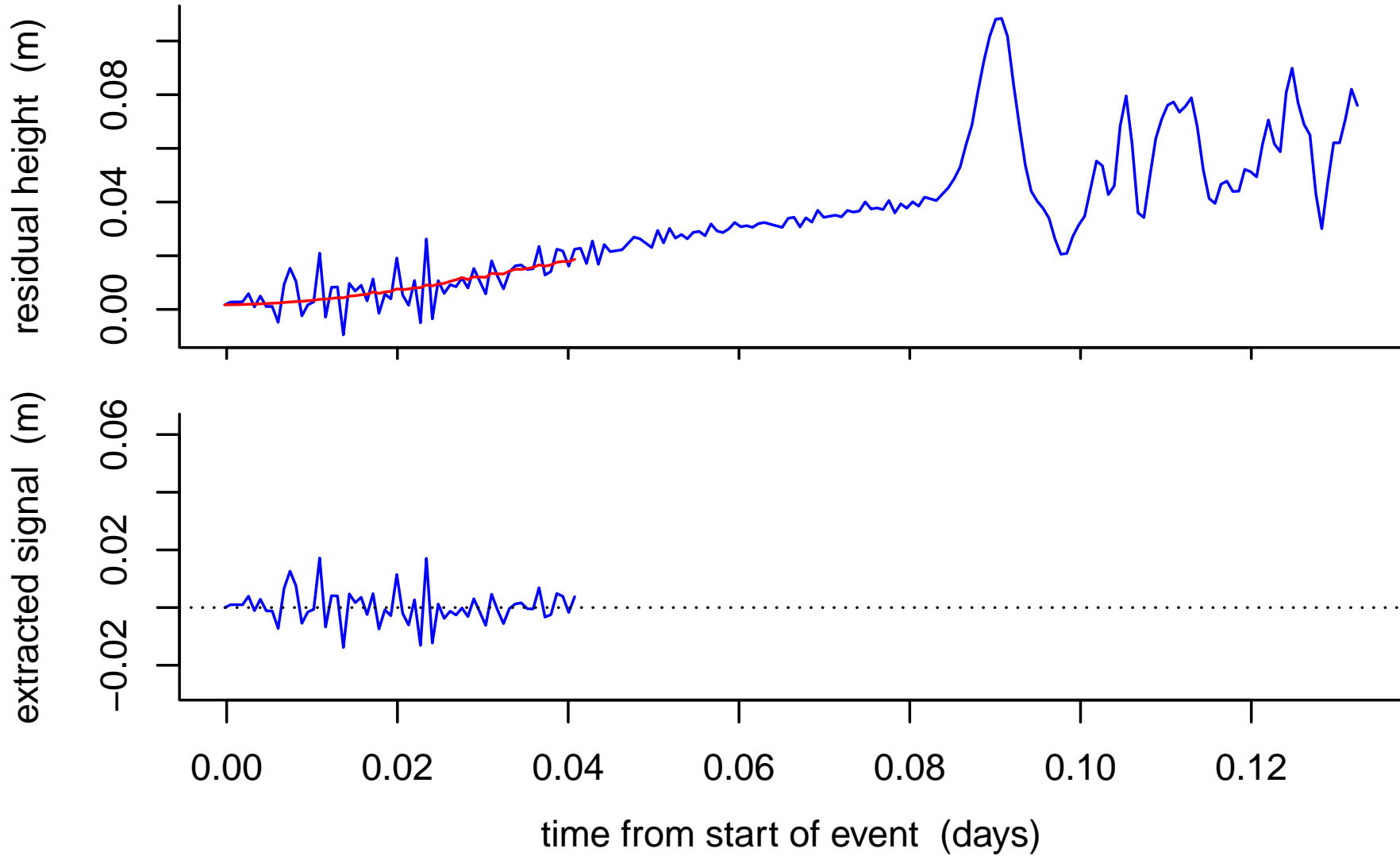
# Kalman Filtering/Smoothing (Local Level Model)



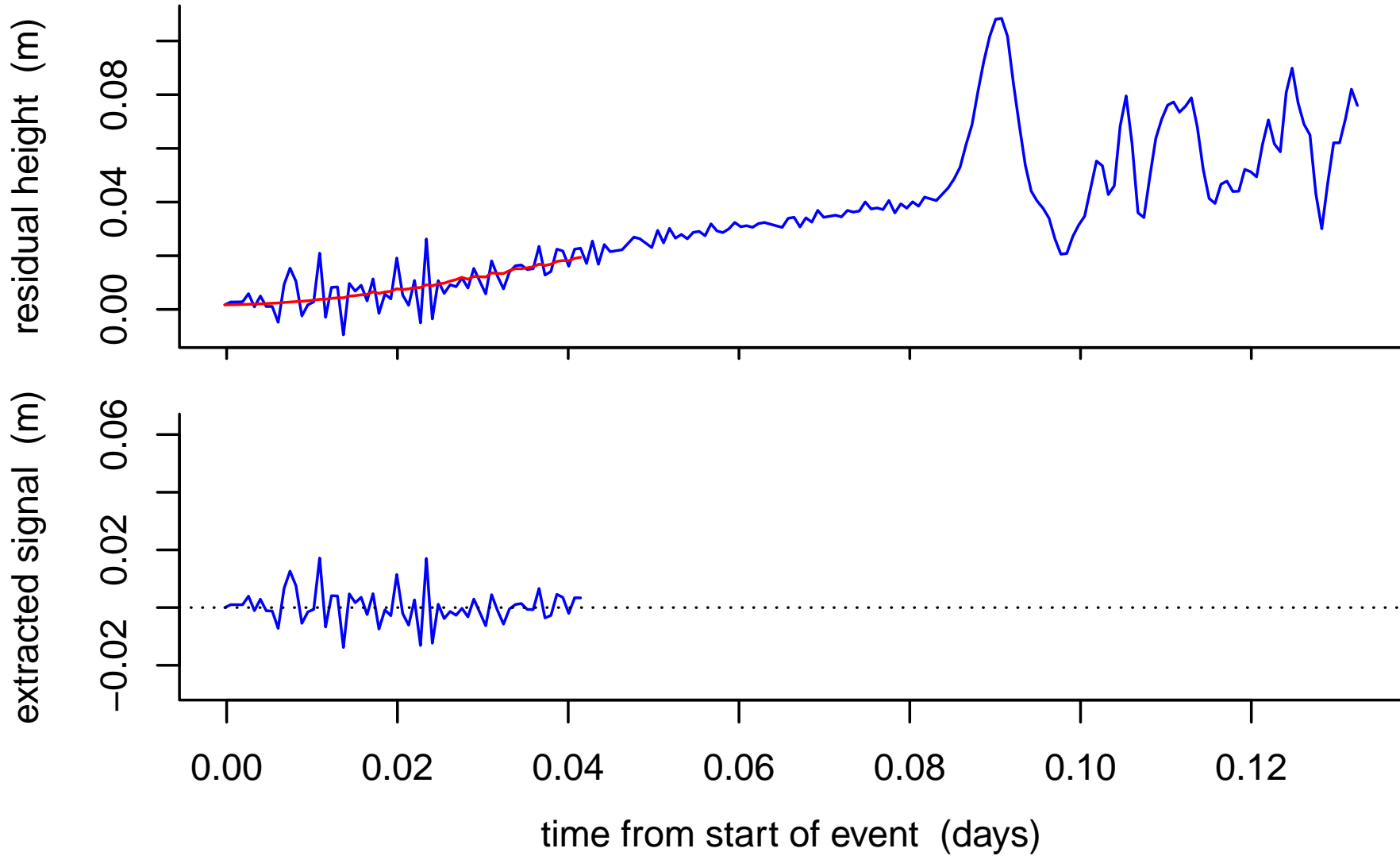
# Kalman Filtering/Smoothing (Local Level Model)



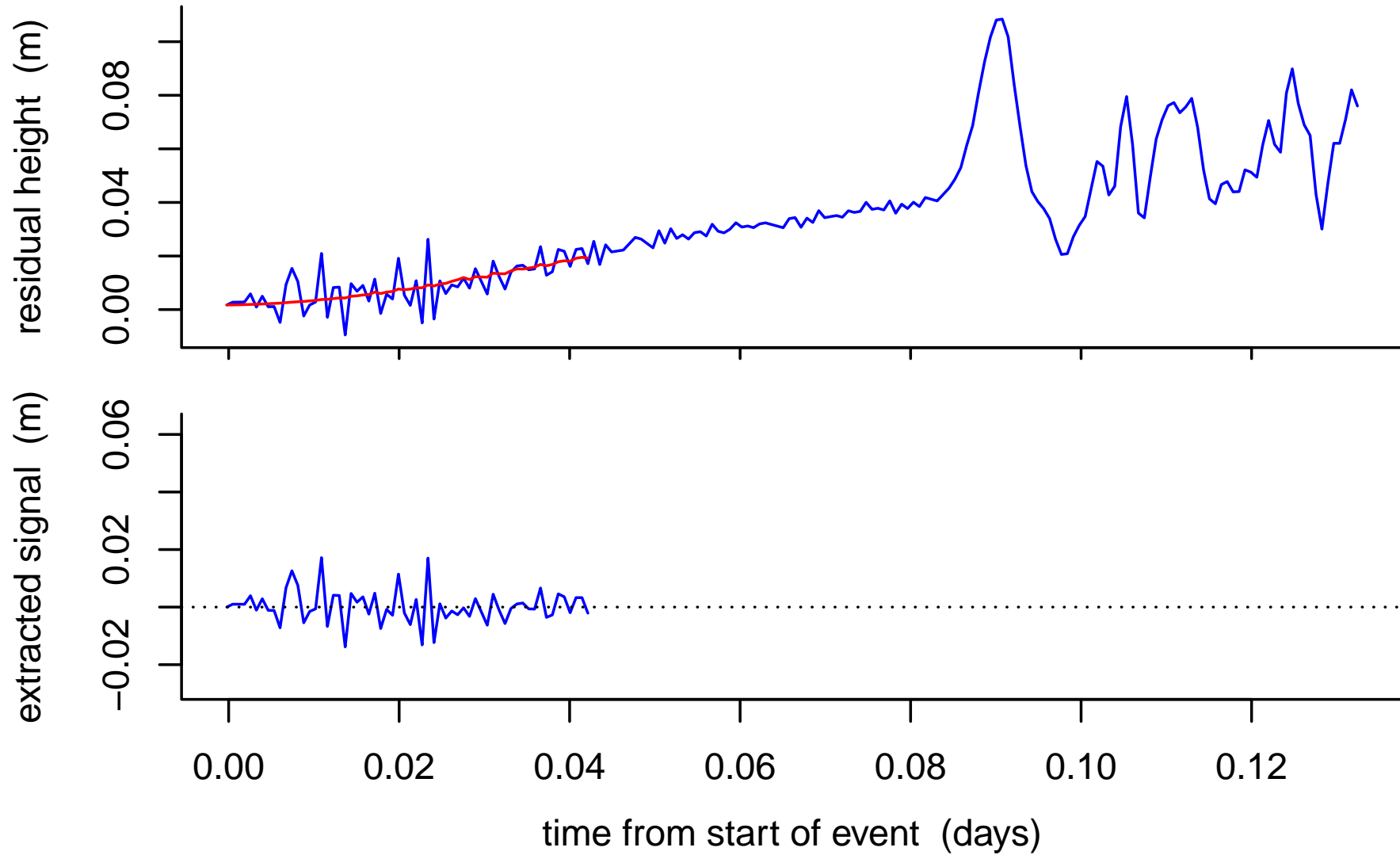
# Kalman Filtering/Smoothing (Local Level Model)



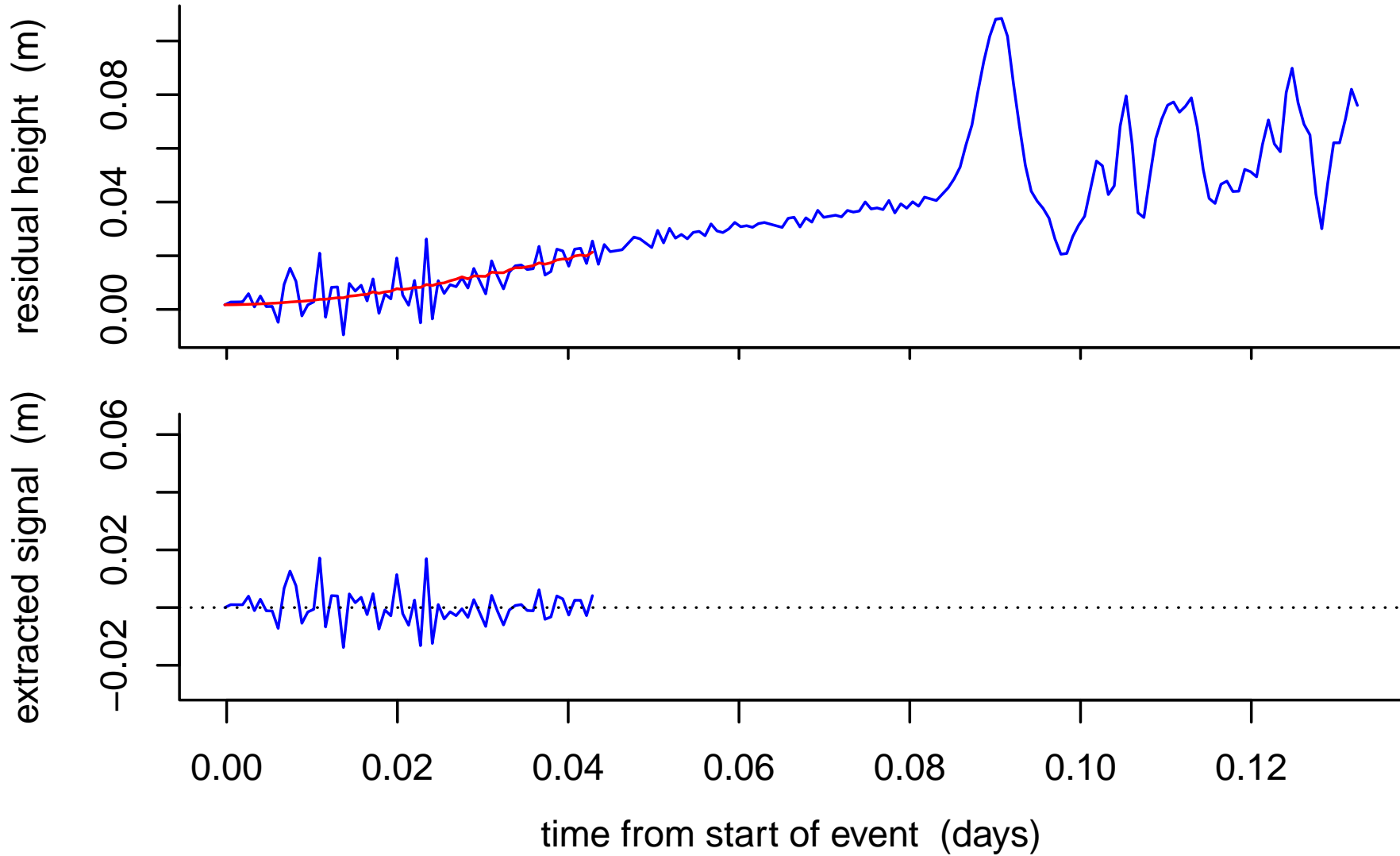
# Kalman Filtering/Smoothing (Local Level Model)



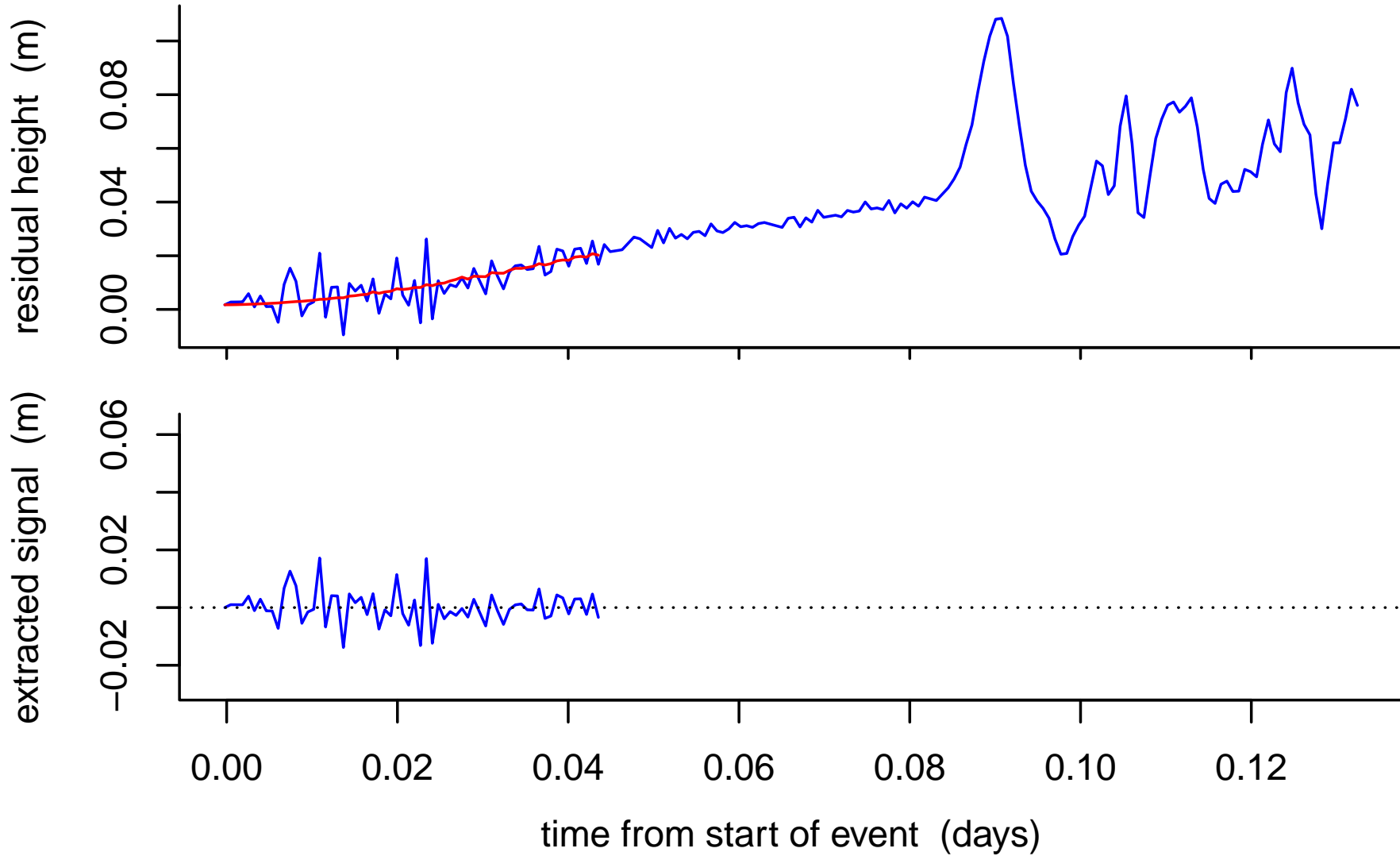
# Kalman Filtering/Smoothing (Local Level Model)



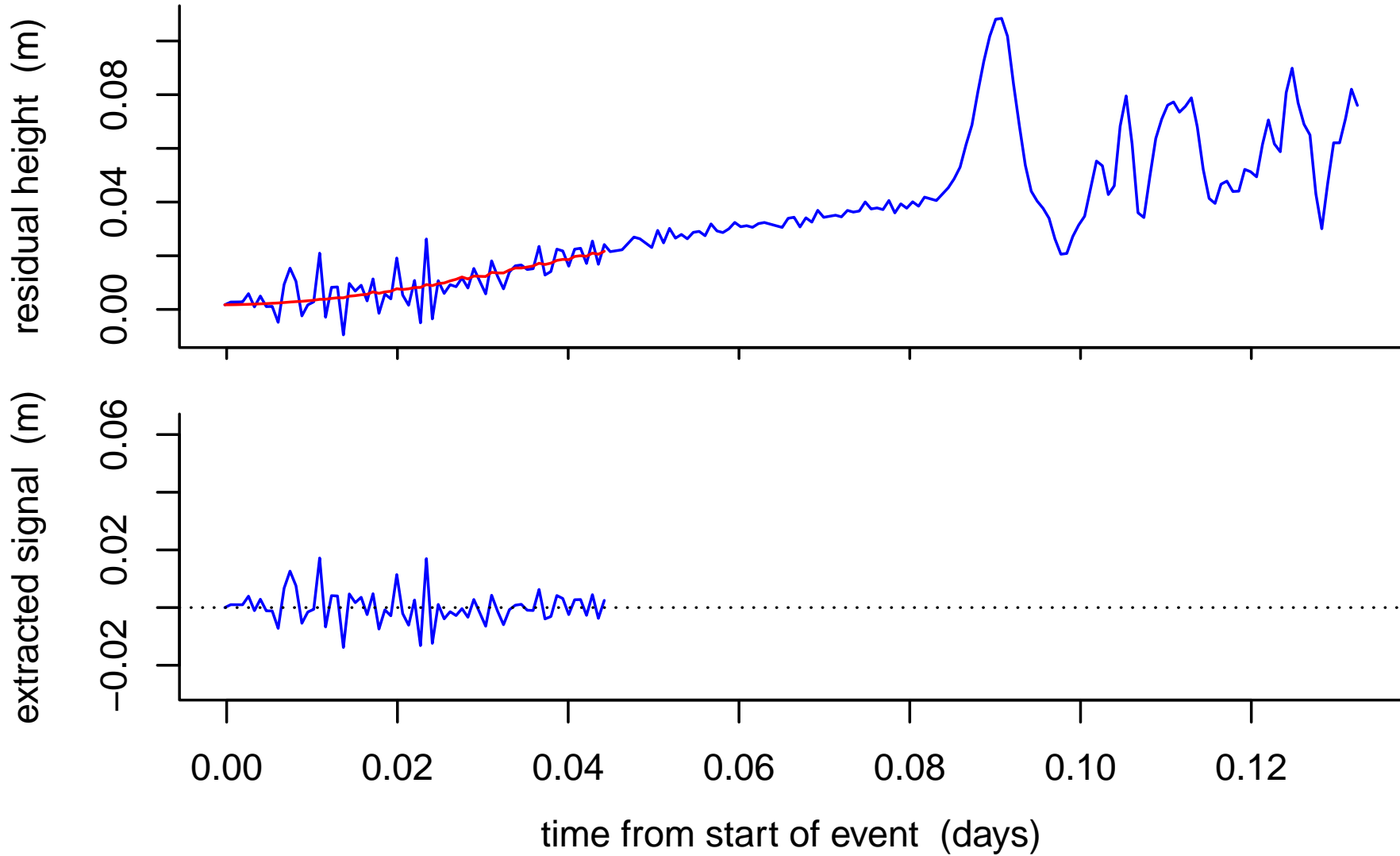
# Kalman Filtering/Smoothing (Local Level Model)



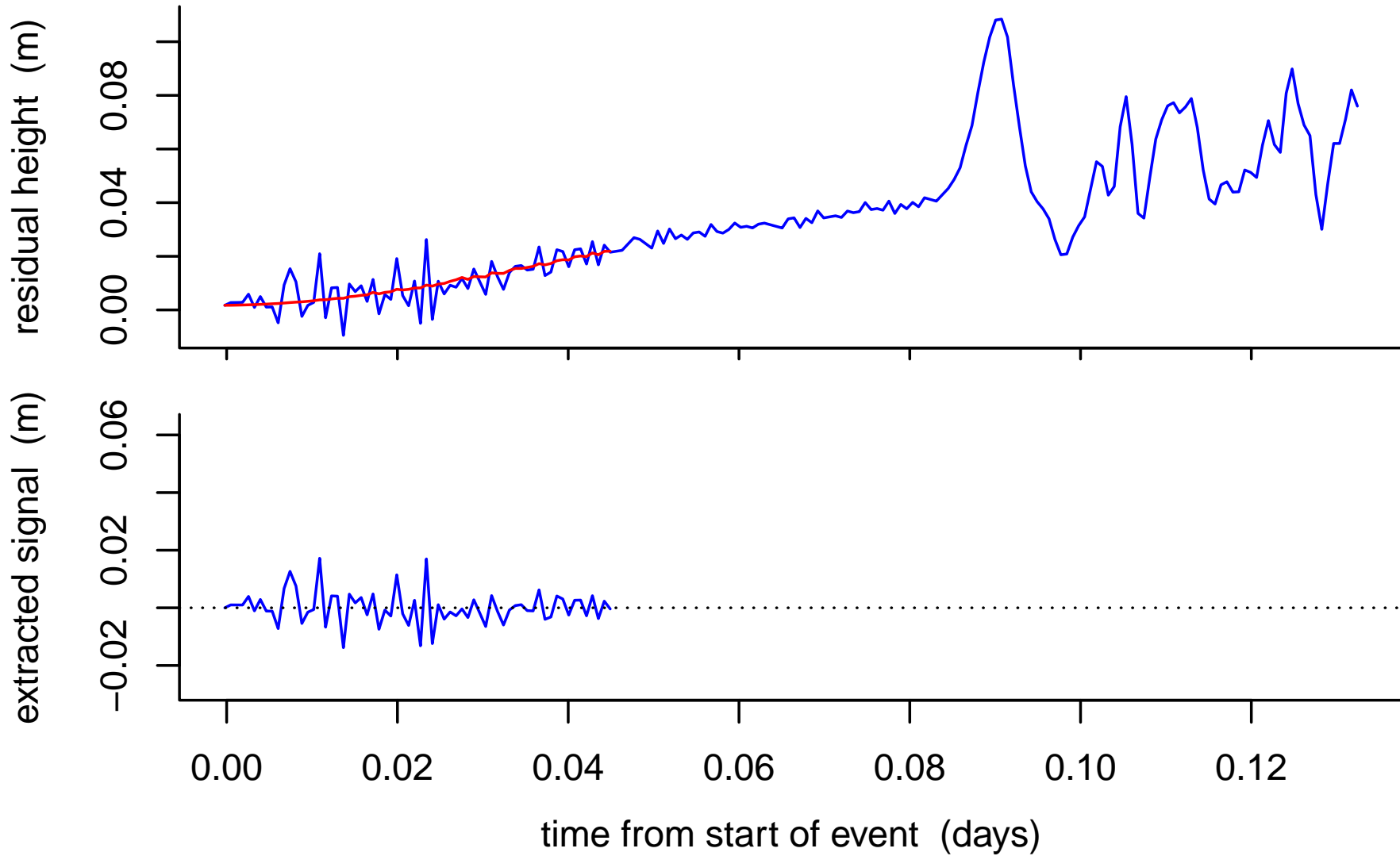
# Kalman Filtering/Smoothing (Local Level Model)



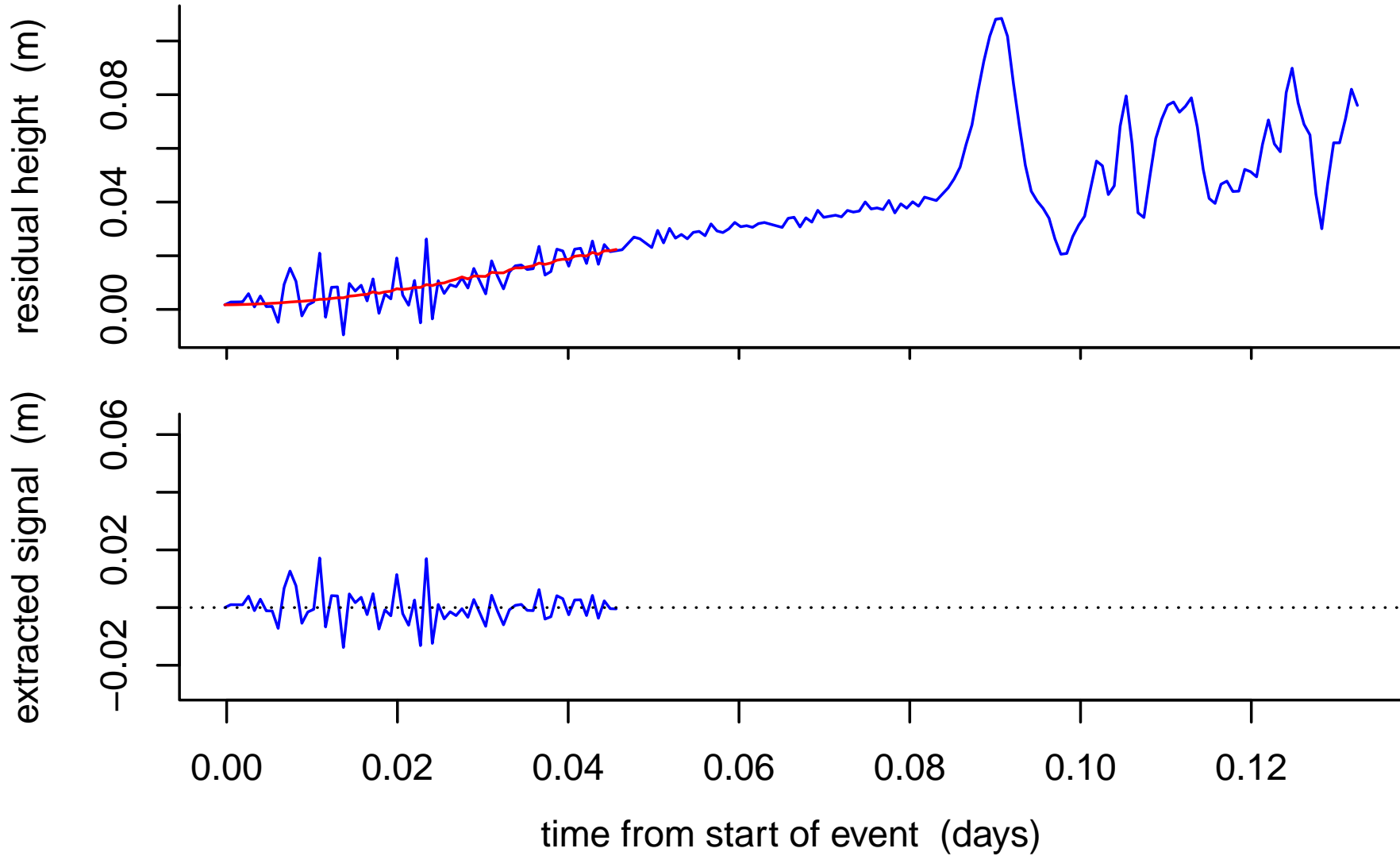
# Kalman Filtering/Smoothing (Local Level Model)



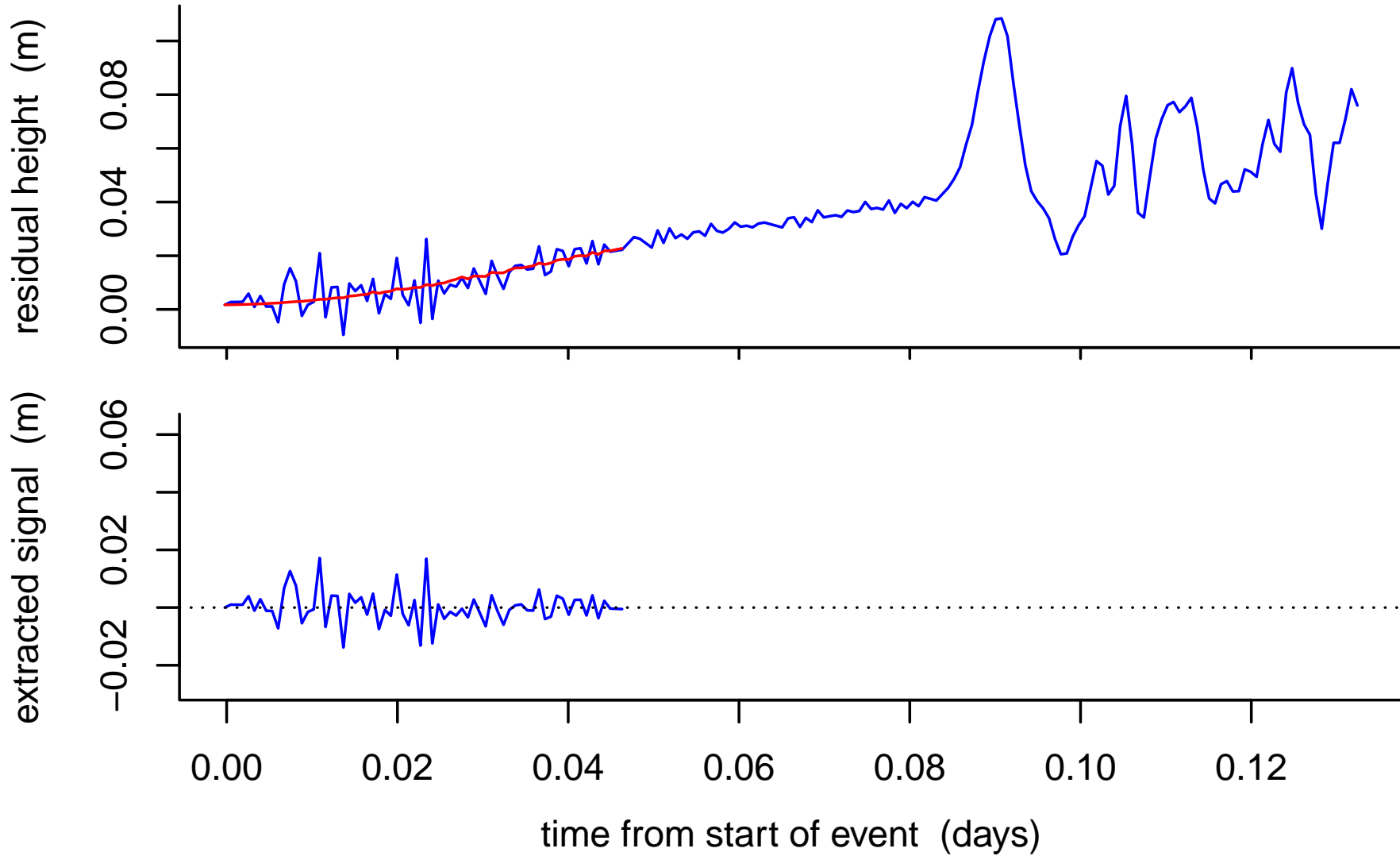
# Kalman Filtering/Smoothing (Local Level Model)



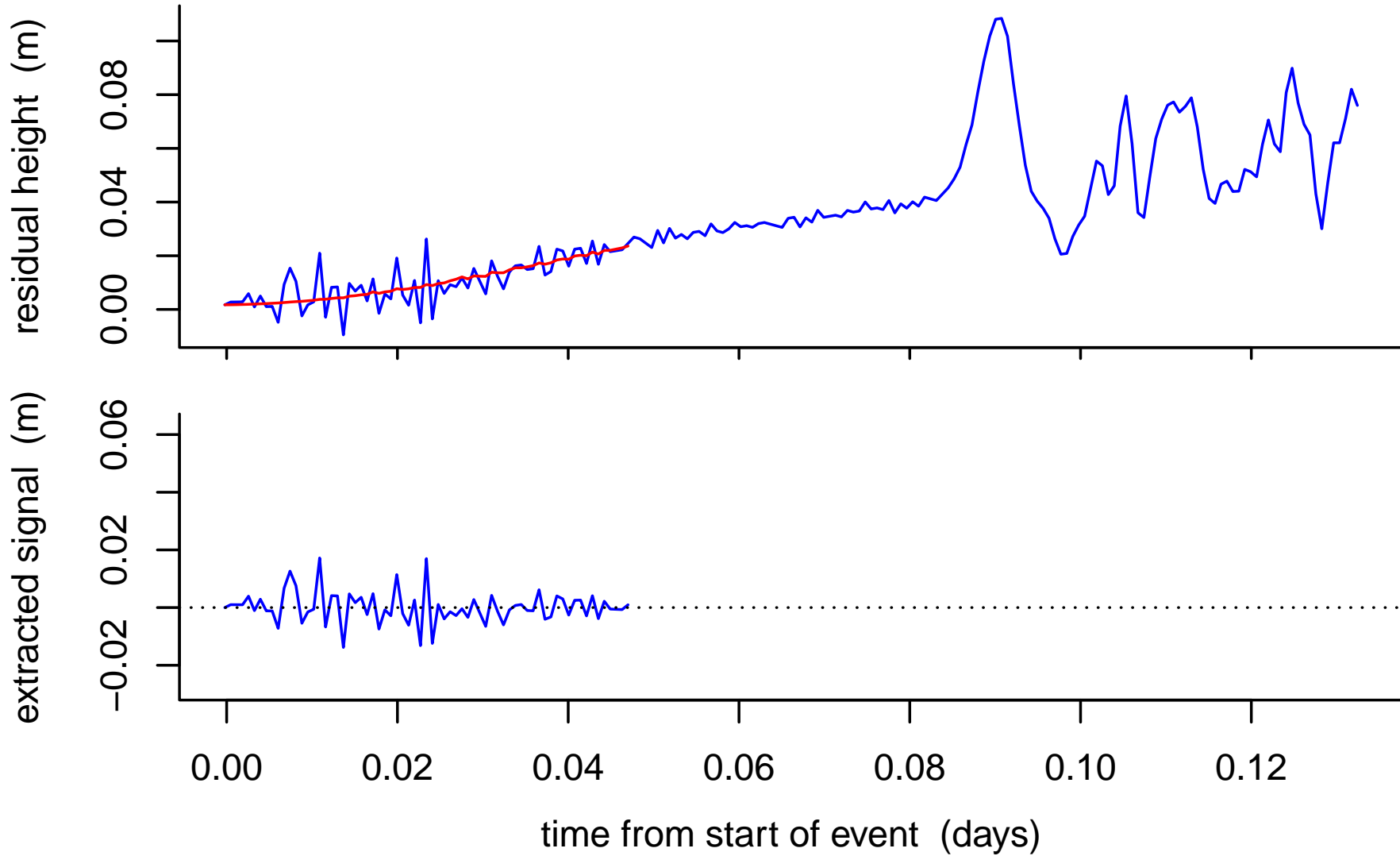
# Kalman Filtering/Smoothing (Local Level Model)



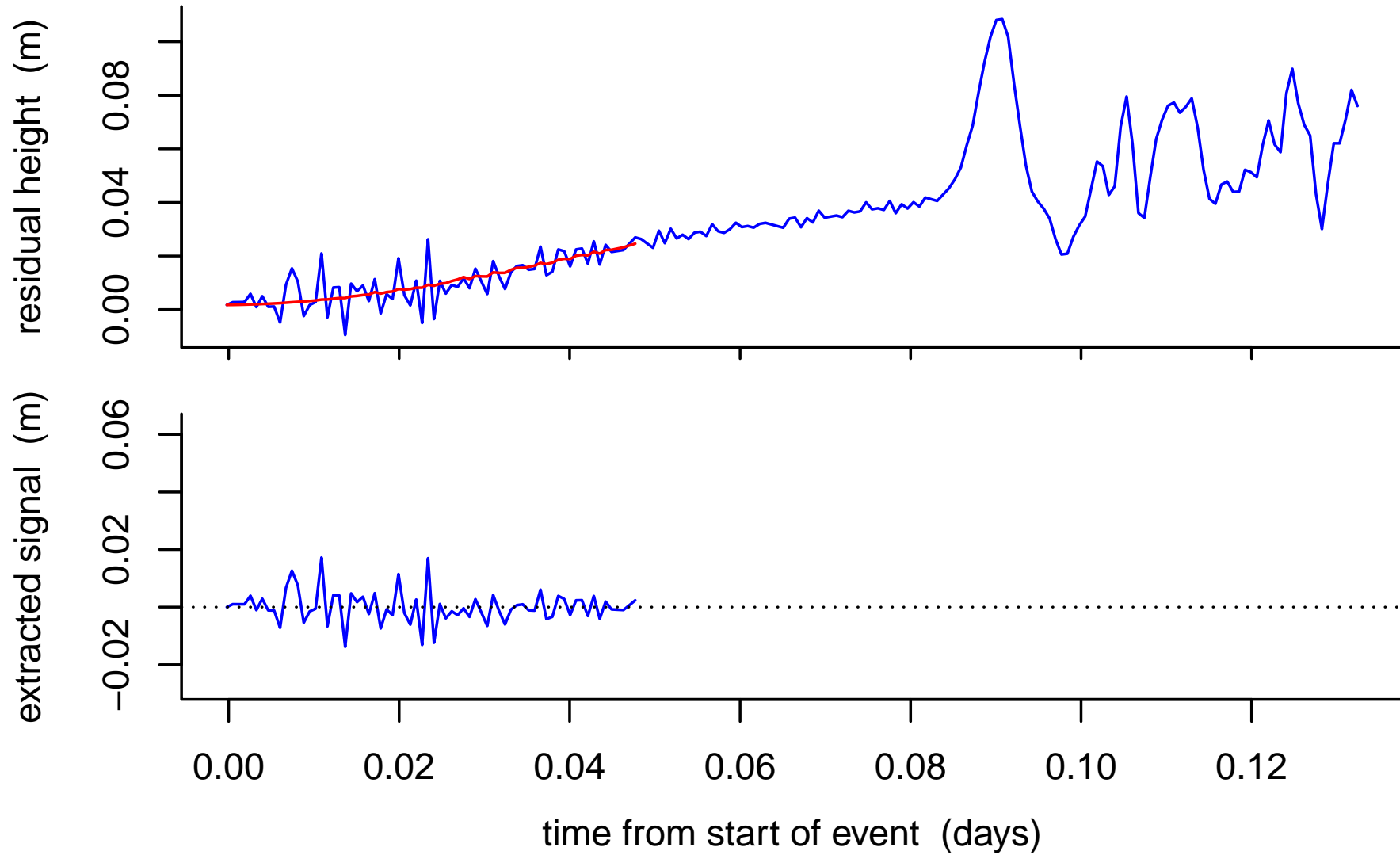
# Kalman Filtering/Smoothing (Local Level Model)



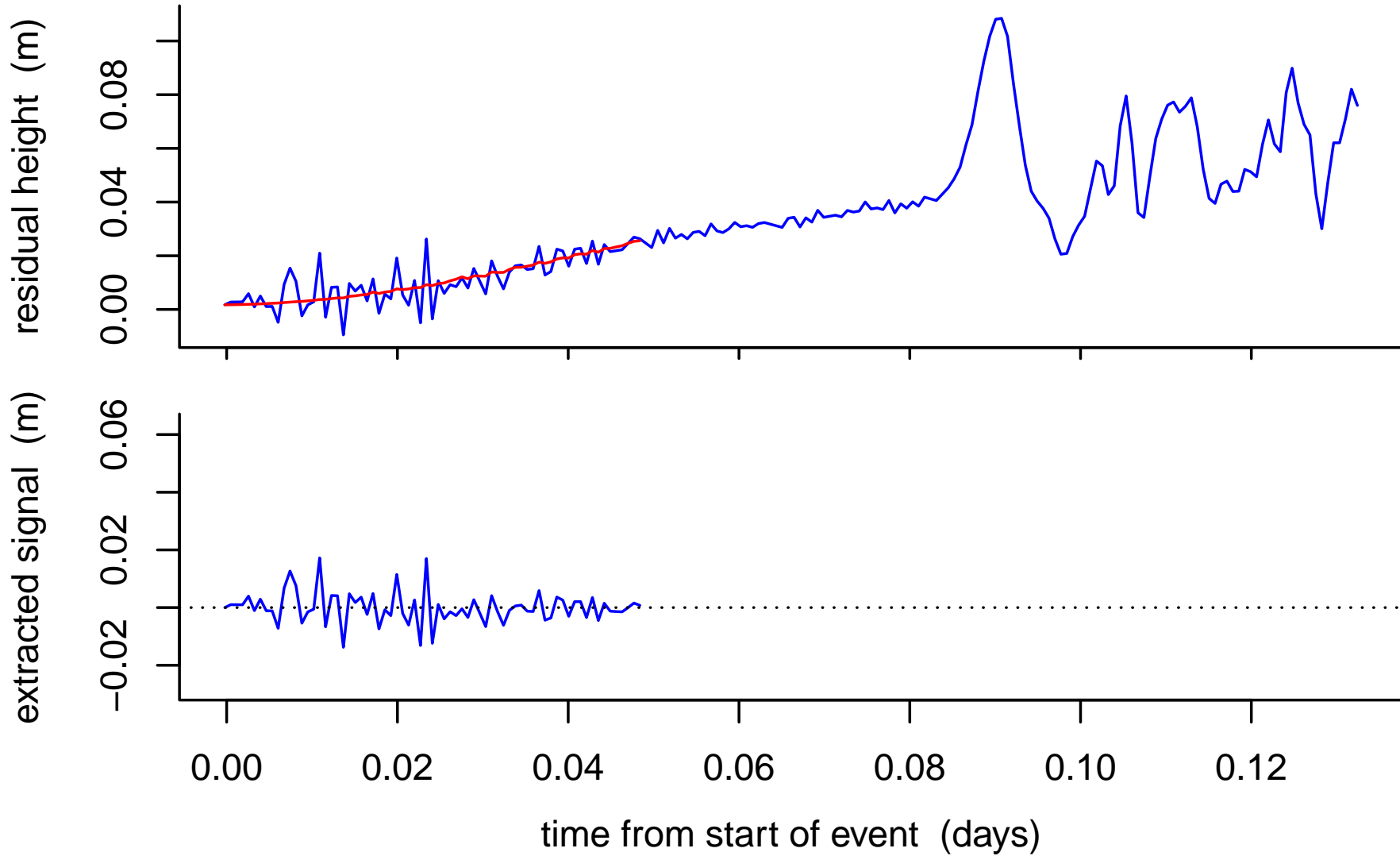
# Kalman Filtering/Smoothing (Local Level Model)



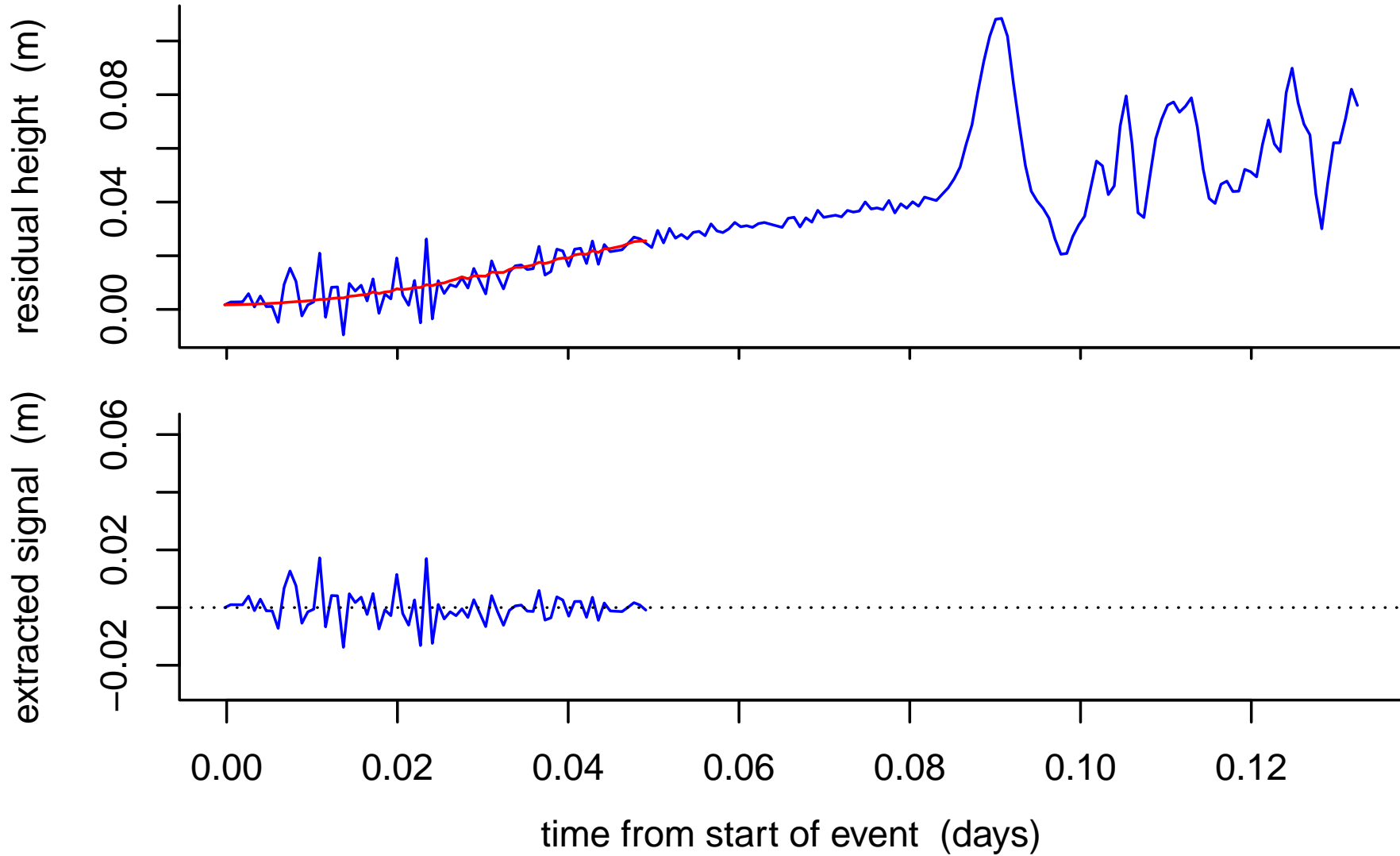
# Kalman Filtering/Smoothing (Local Level Model)



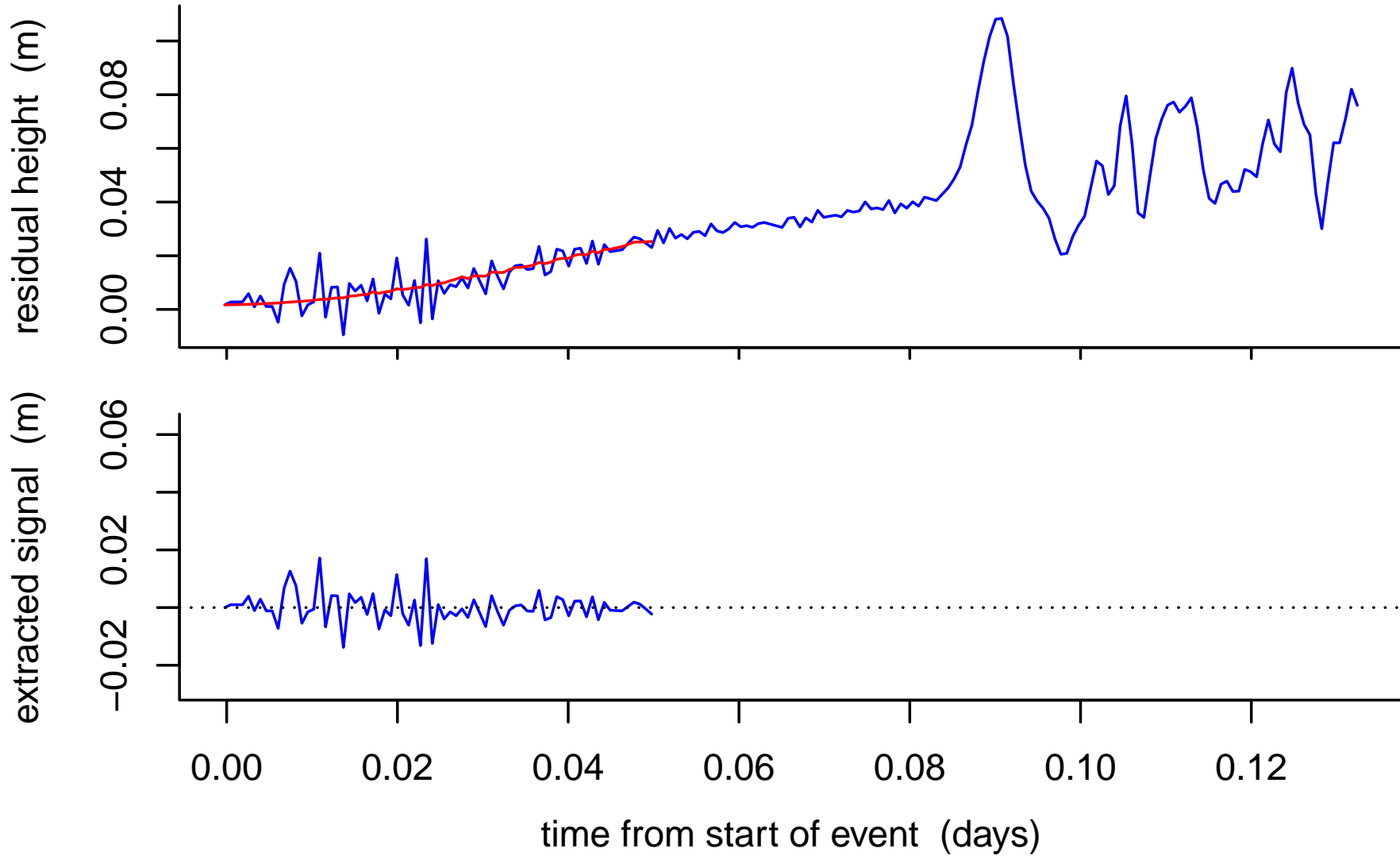
# Kalman Filtering/Smoothing (Local Level Model)



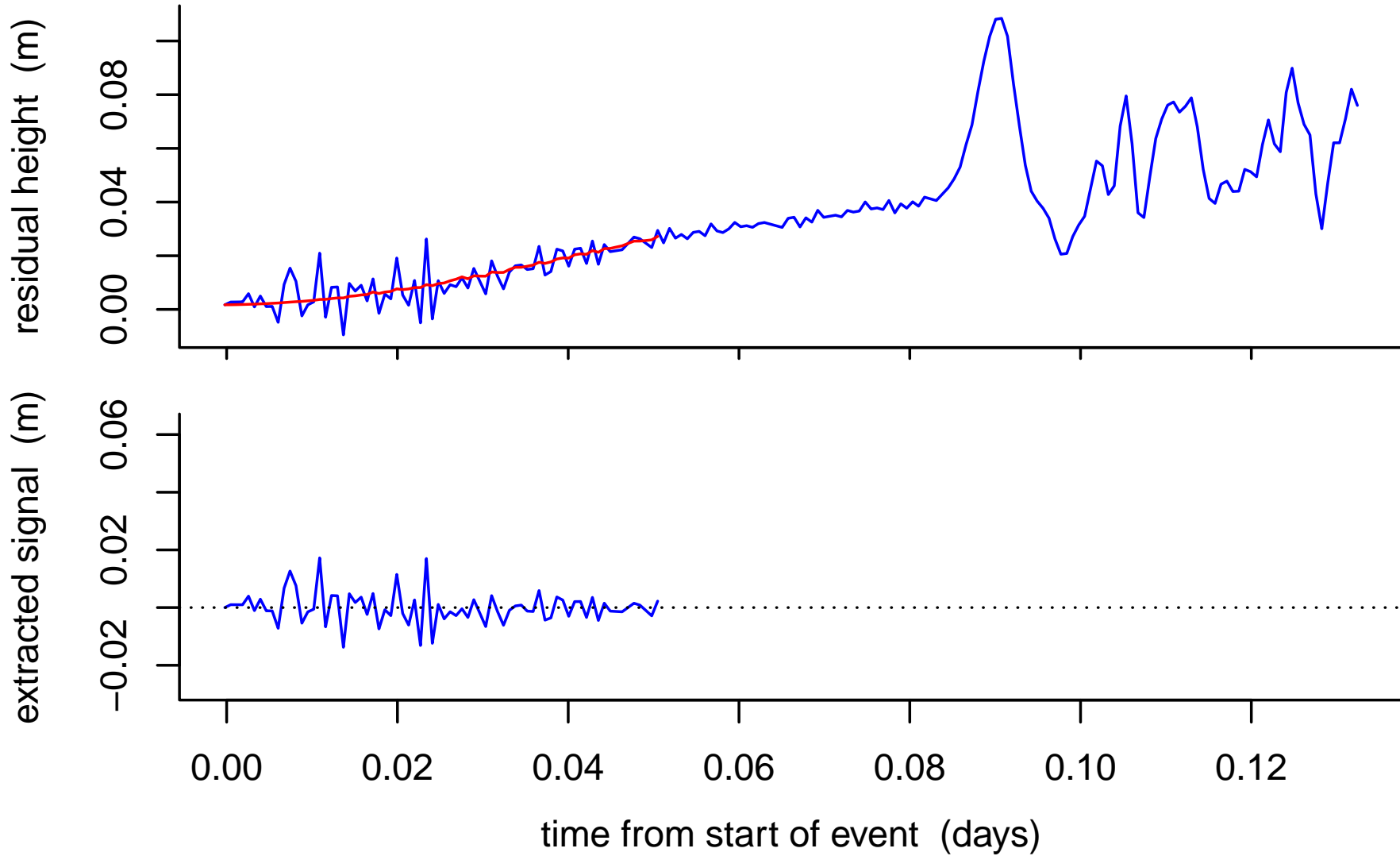
# Kalman Filtering/Smoothing (Local Level Model)



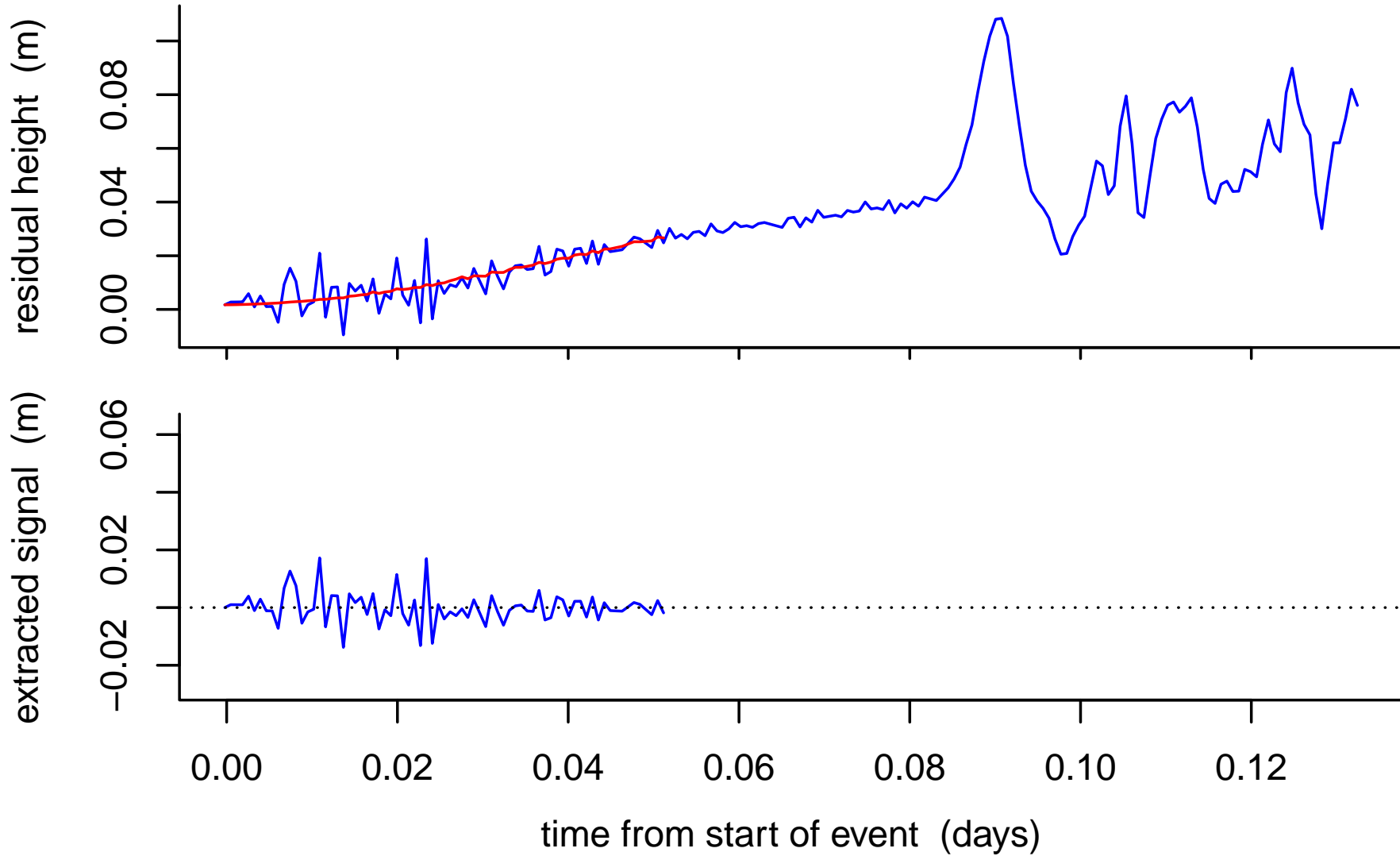
# Kalman Filtering/Smoothing (Local Level Model)



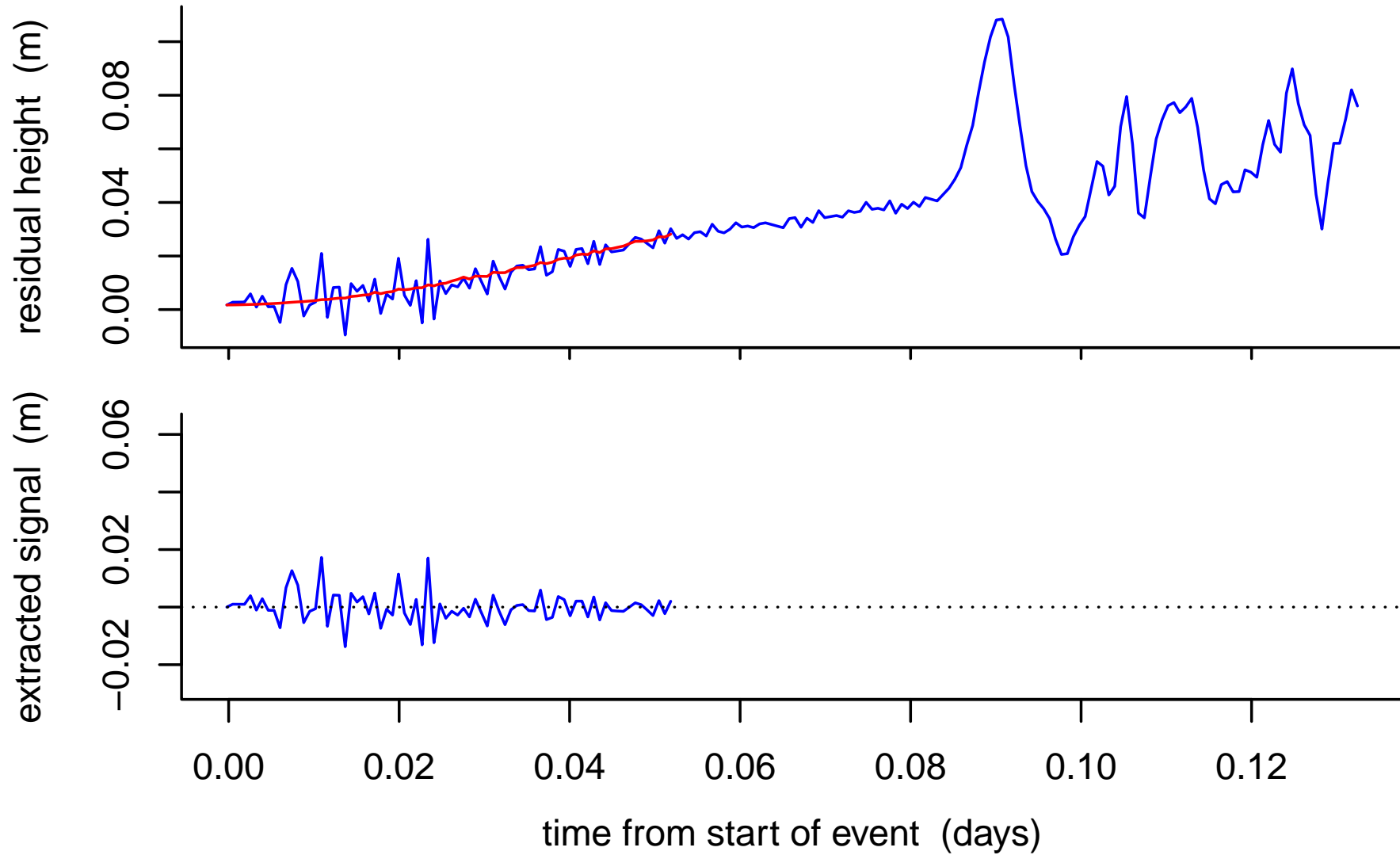
# Kalman Filtering/Smoothing (Local Level Model)



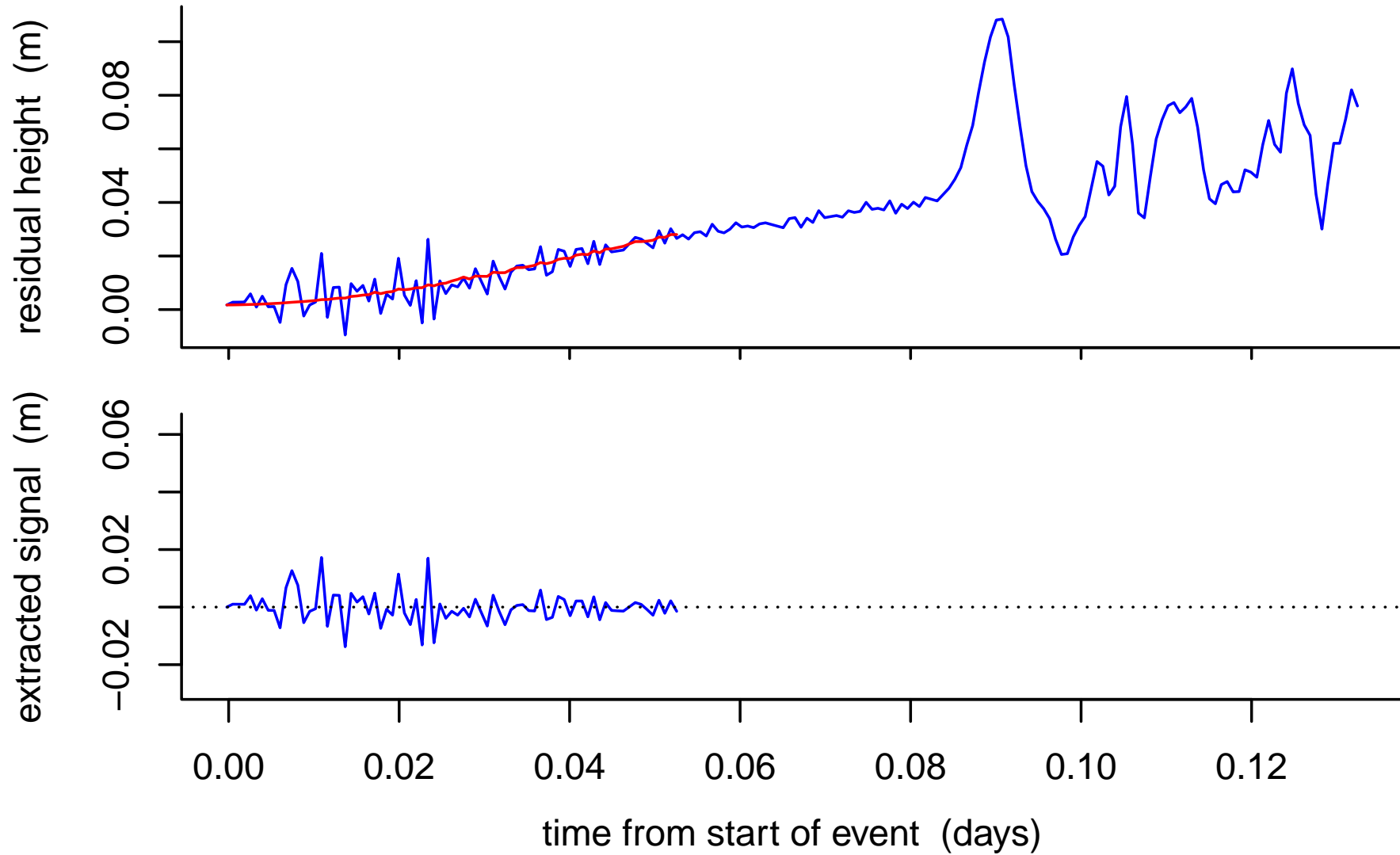
# Kalman Filtering/Smoothing (Local Level Model)



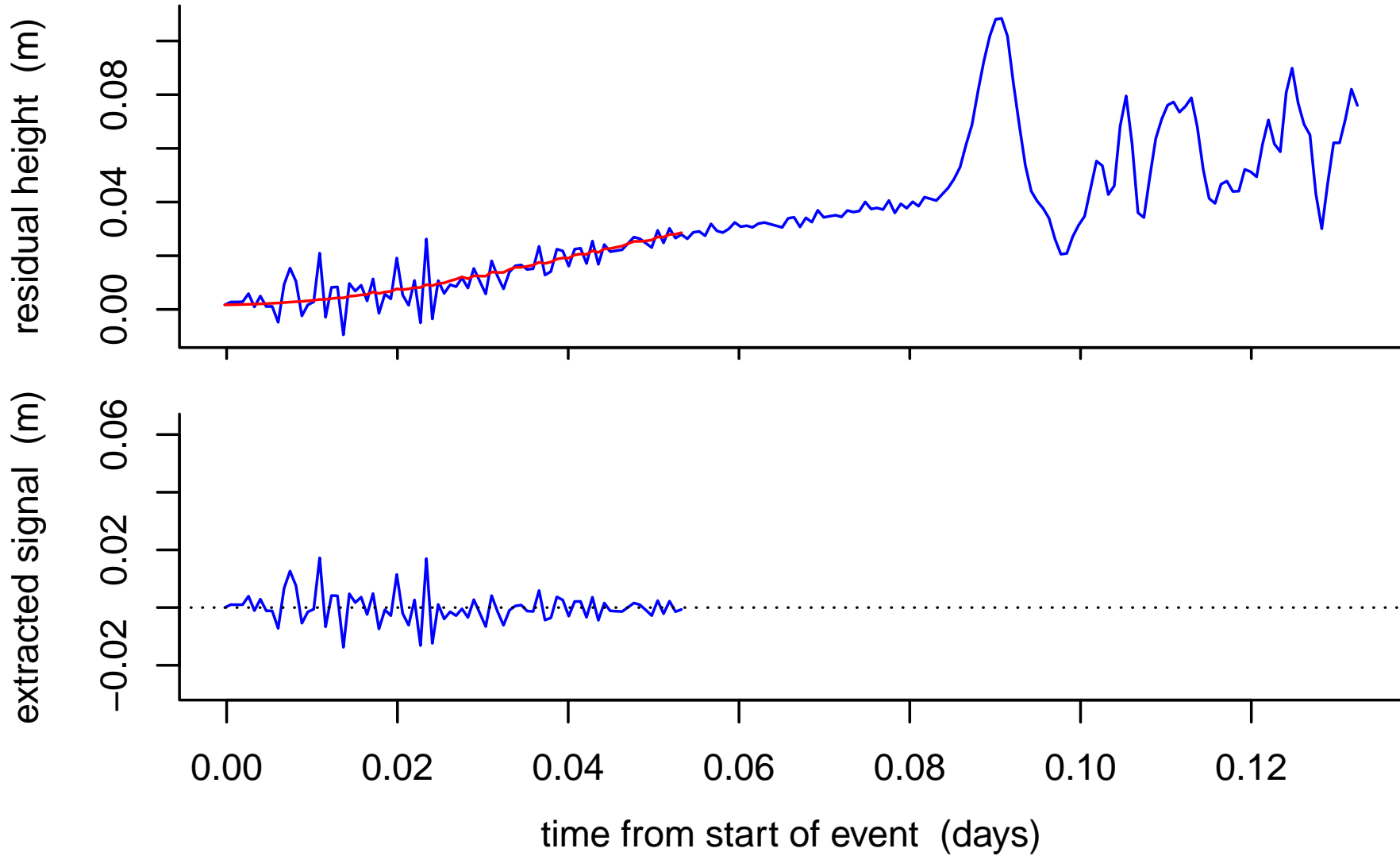
# Kalman Filtering/Smoothing (Local Level Model)



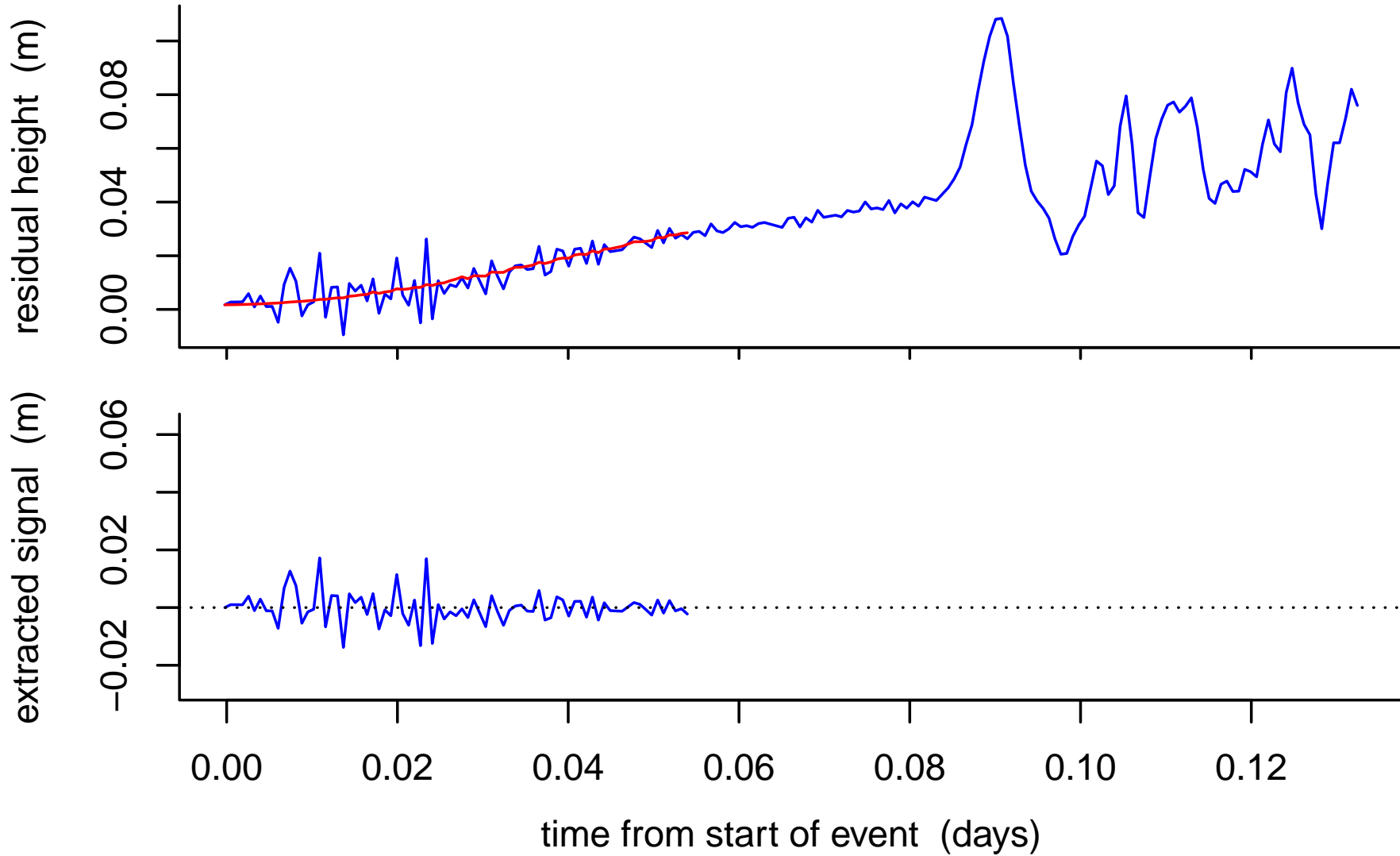
# Kalman Filtering/Smoothing (Local Level Model)



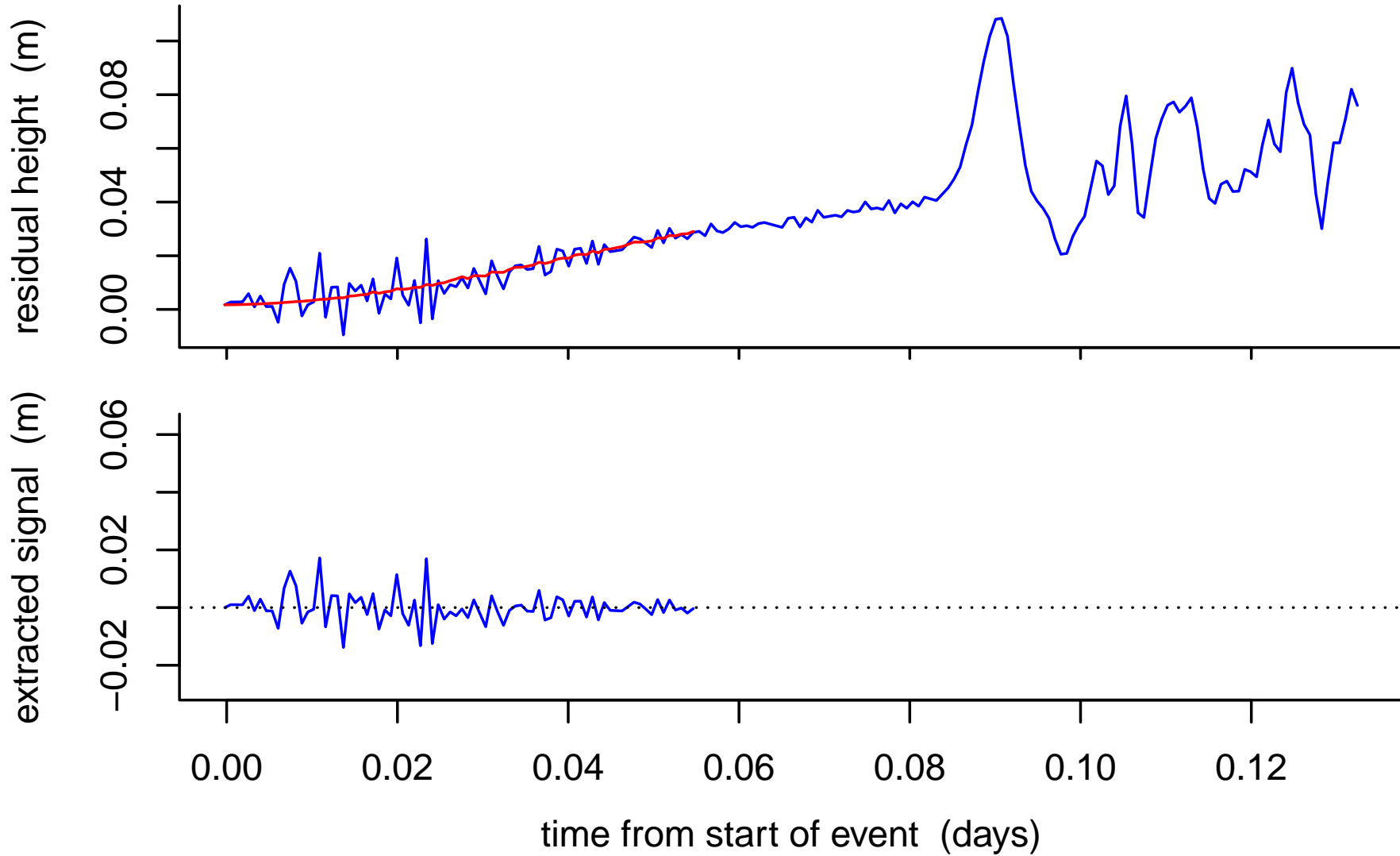
# Kalman Filtering/Smoothing (Local Level Model)



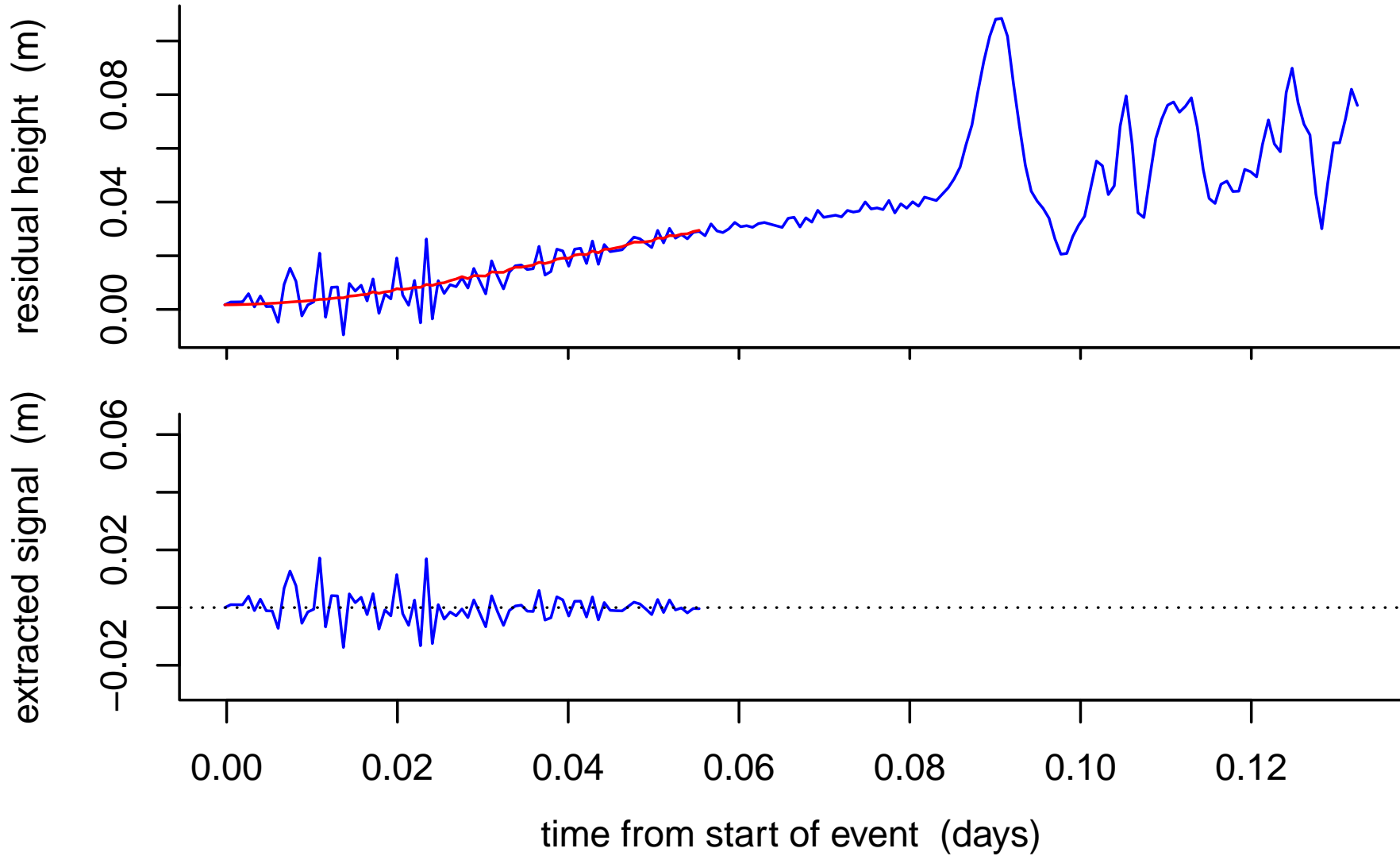
# Kalman Filtering/Smoothing (Local Level Model)



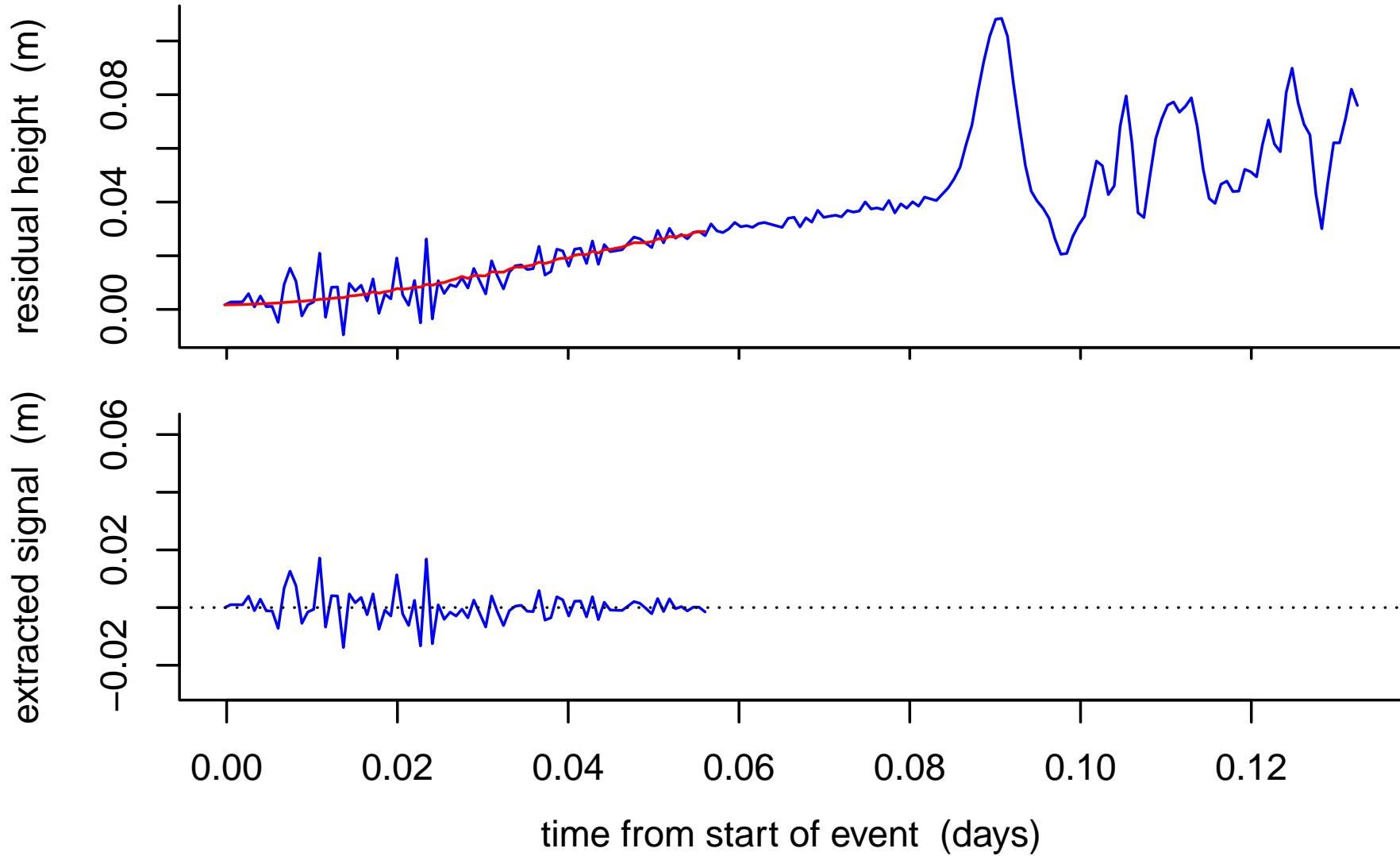
# Kalman Filtering/Smoothing (Local Level Model)



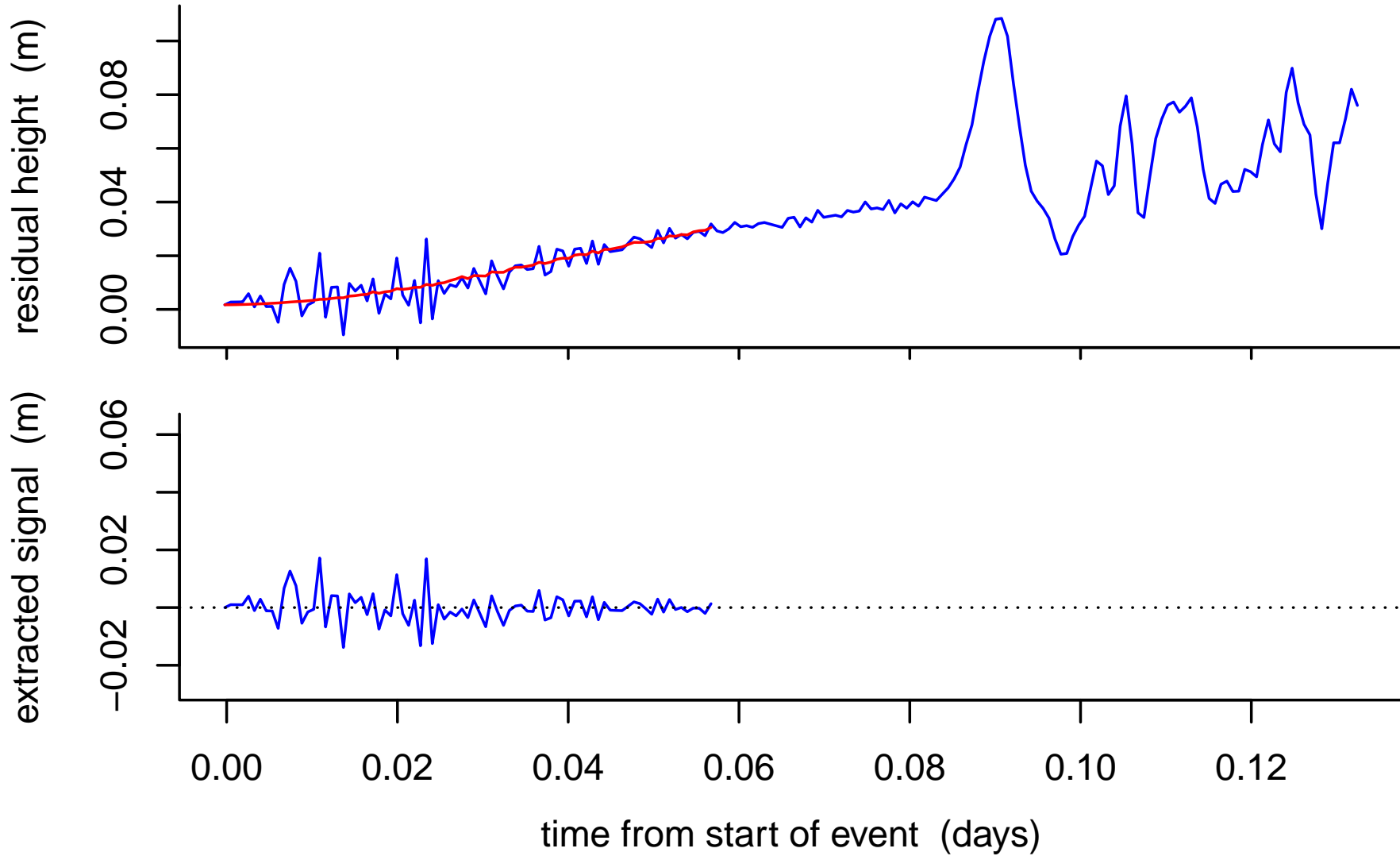
# Kalman Filtering/Smoothing (Local Level Model)



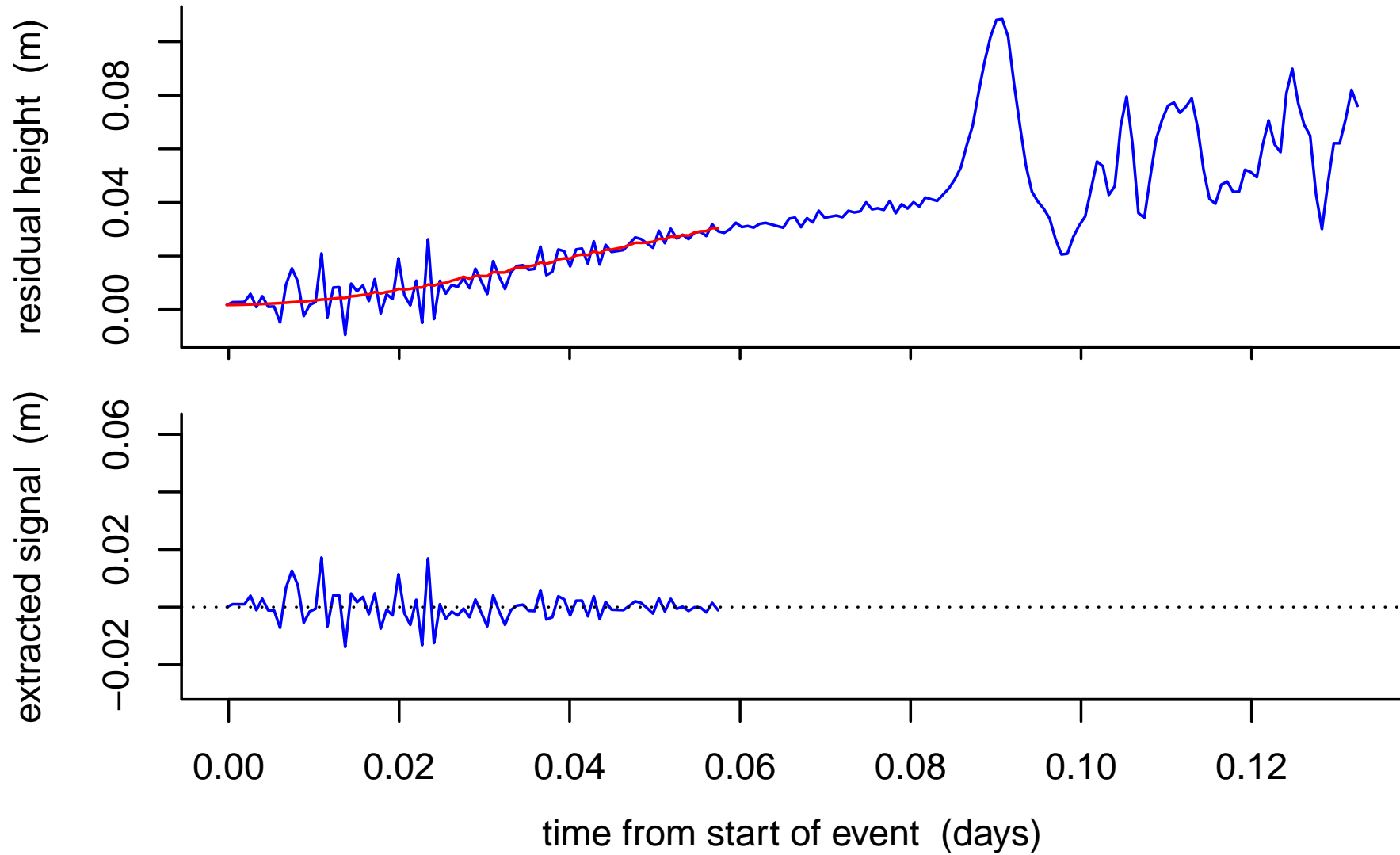
# Kalman Filtering/Smoothing (Local Level Model)



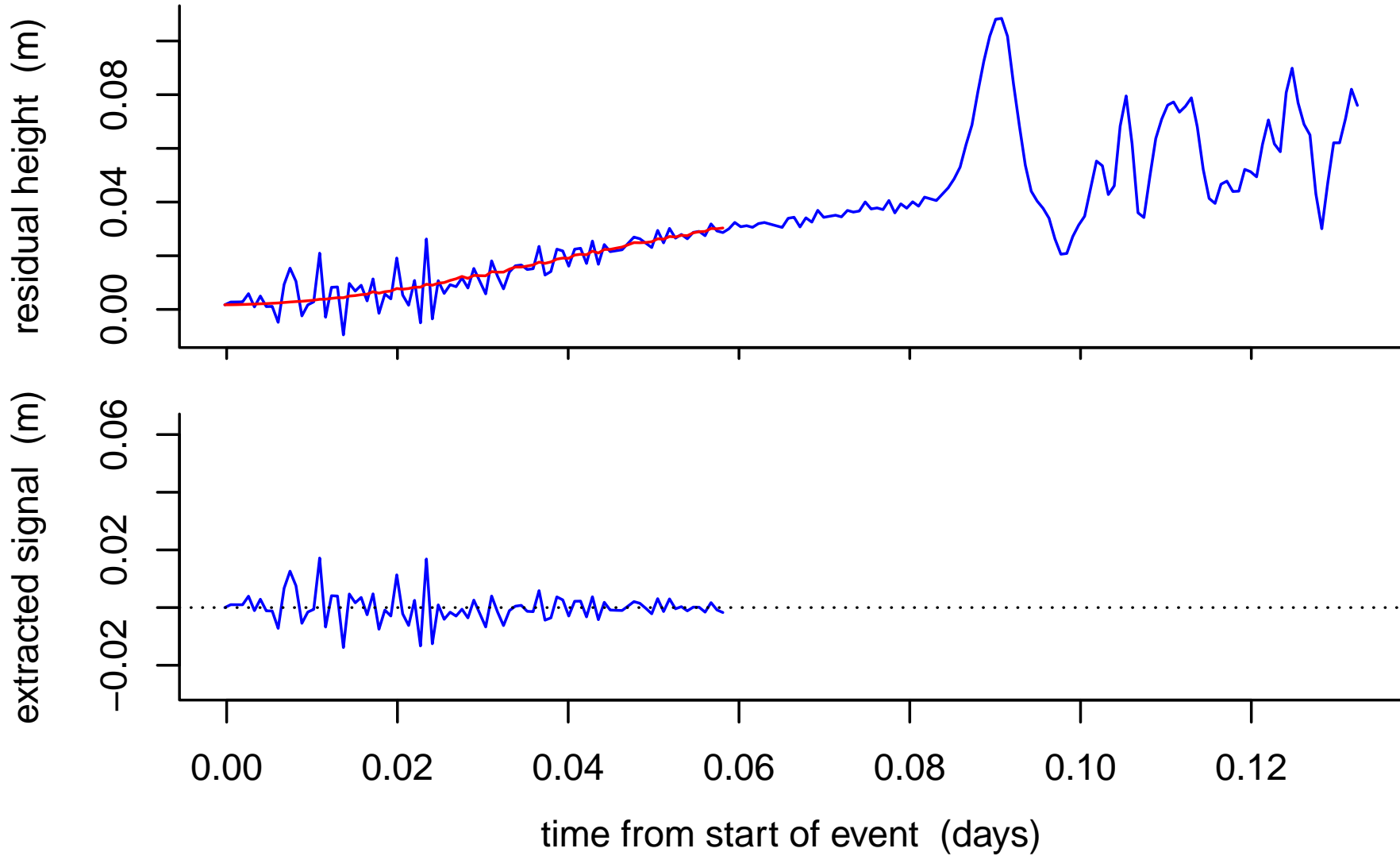
# Kalman Filtering/Smoothing (Local Level Model)



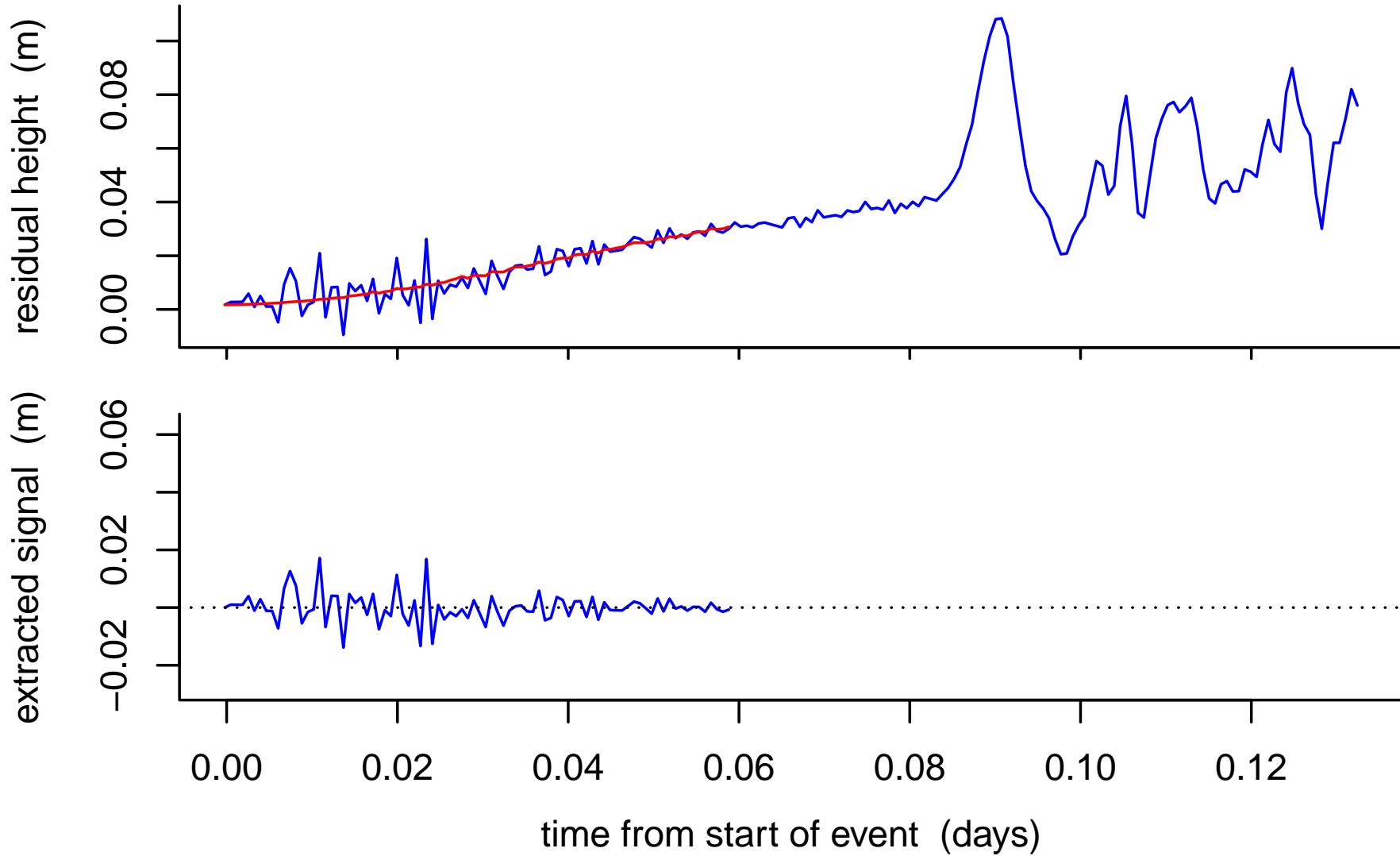
# Kalman Filtering/Smoothing (Local Level Model)



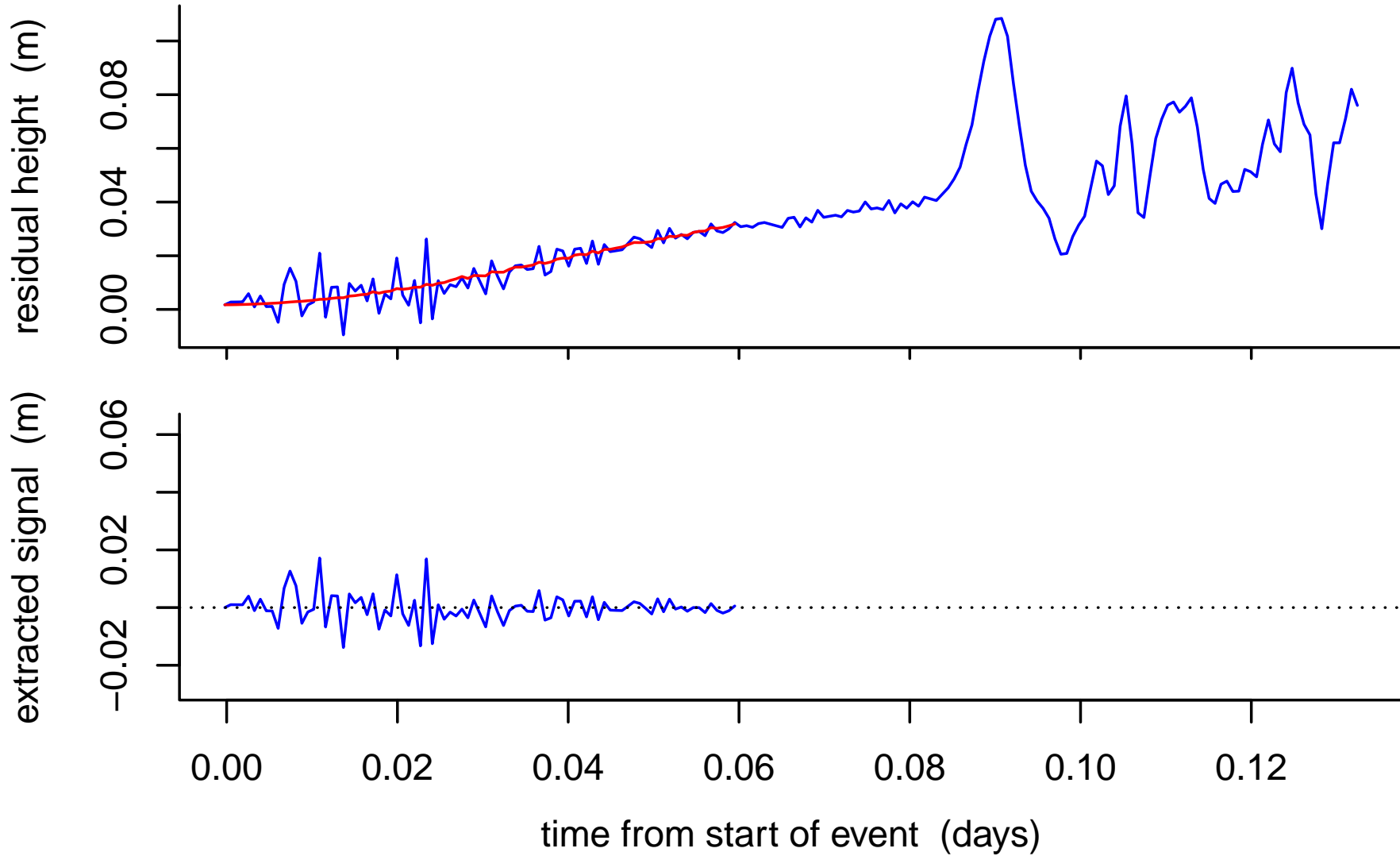
# Kalman Filtering/Smoothing (Local Level Model)



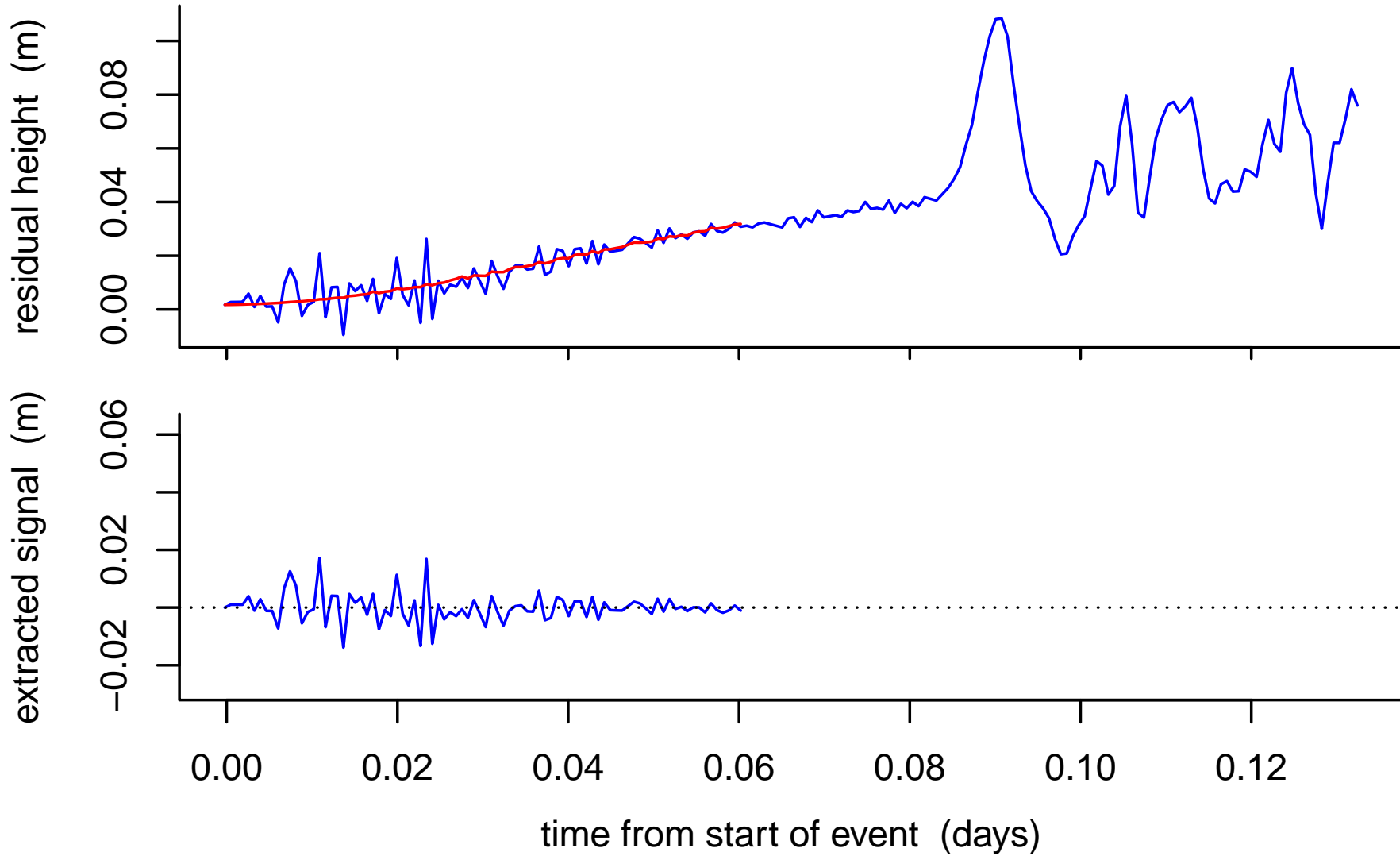
# Kalman Filtering/Smoothing (Local Level Model)



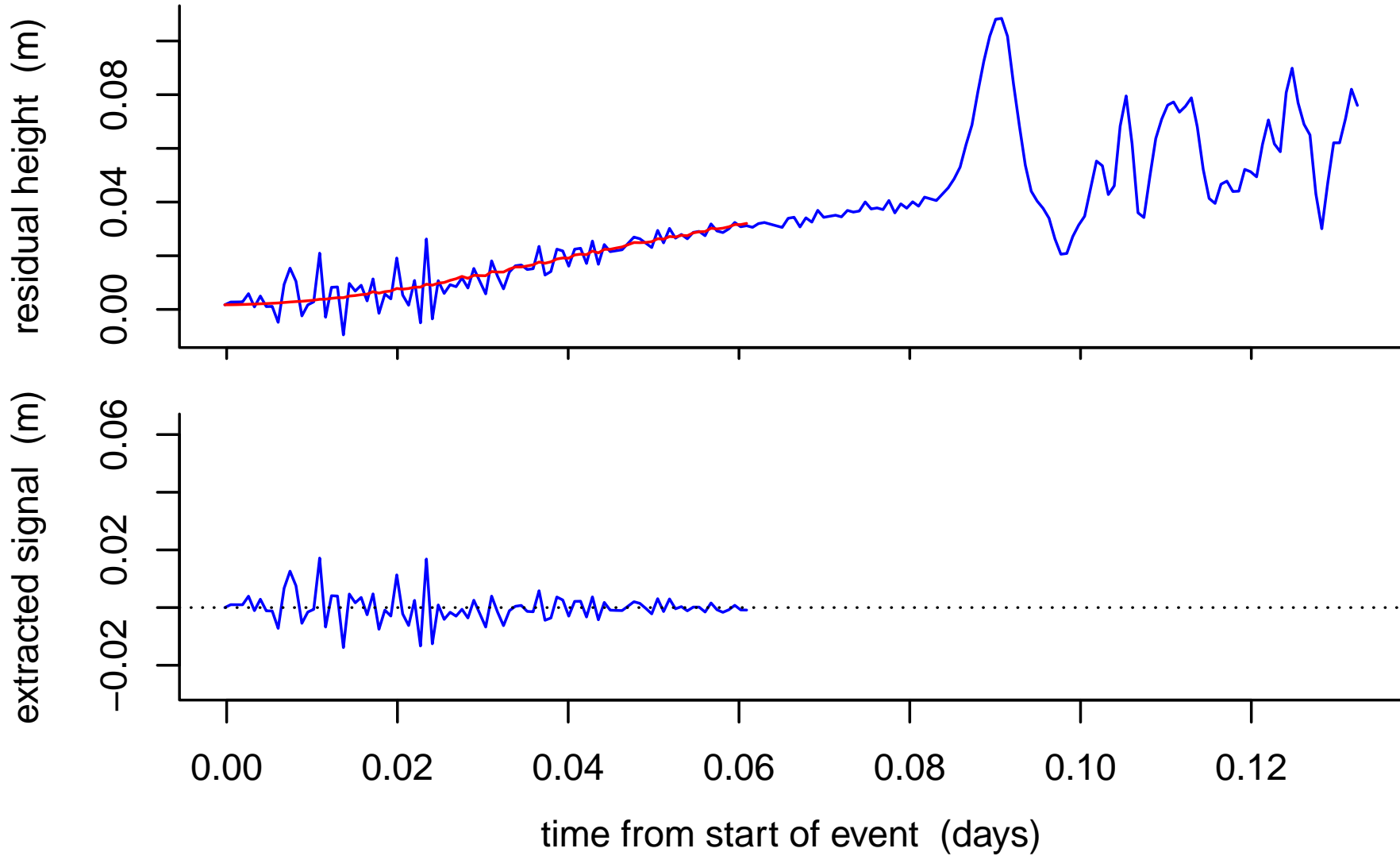
# Kalman Filtering/Smoothing (Local Level Model)



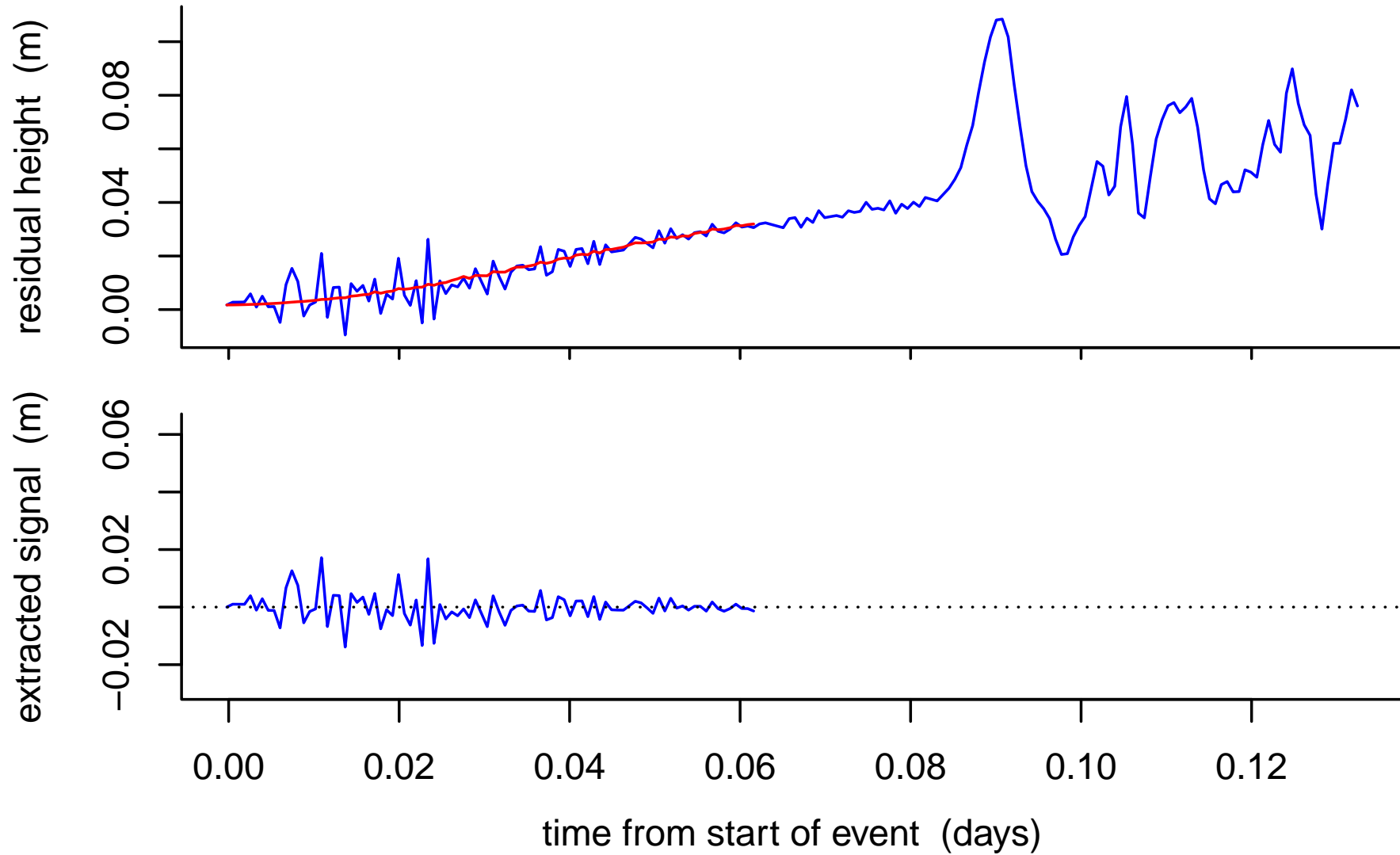
# Kalman Filtering/Smoothing (Local Level Model)



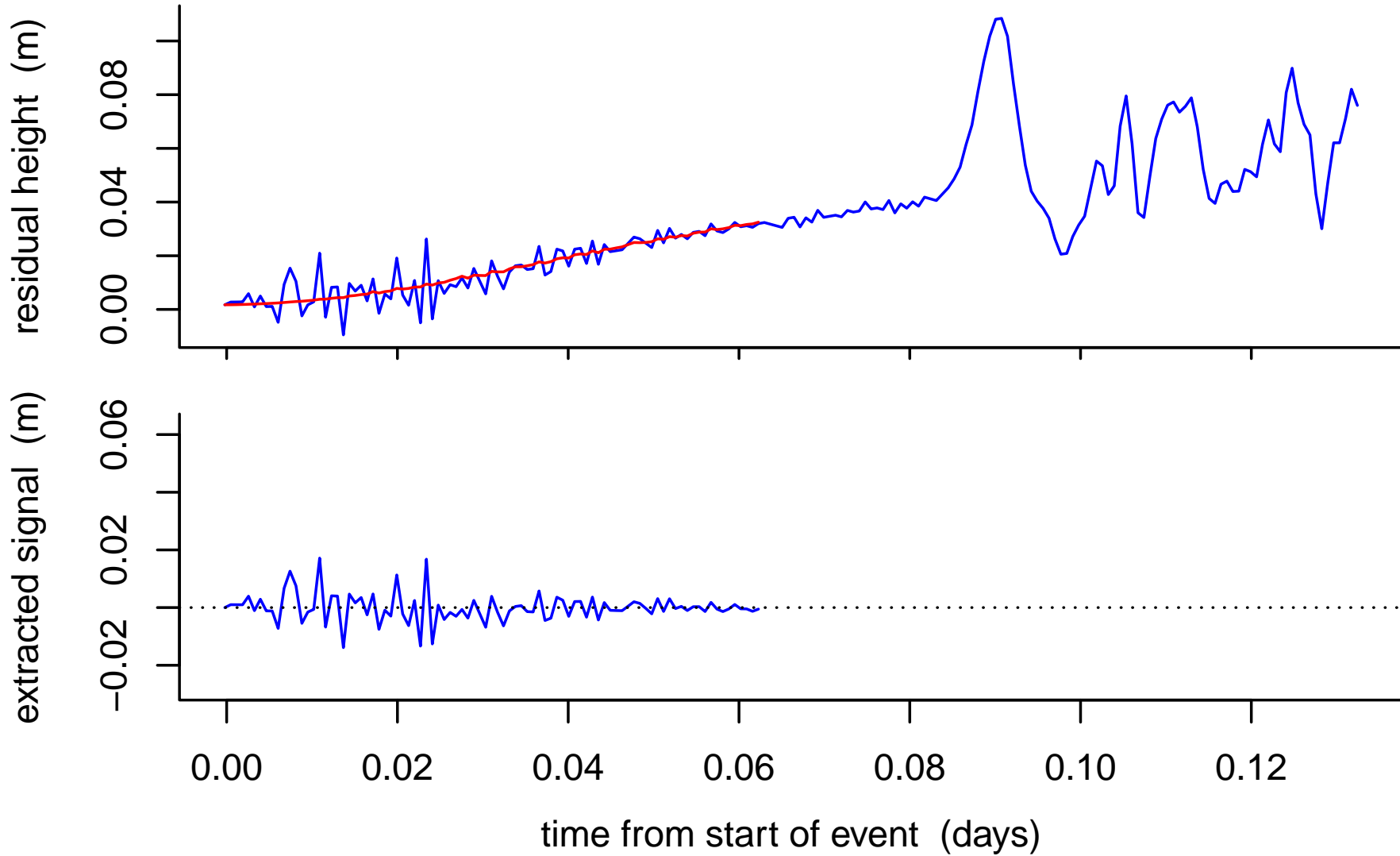
# Kalman Filtering/Smoothing (Local Level Model)



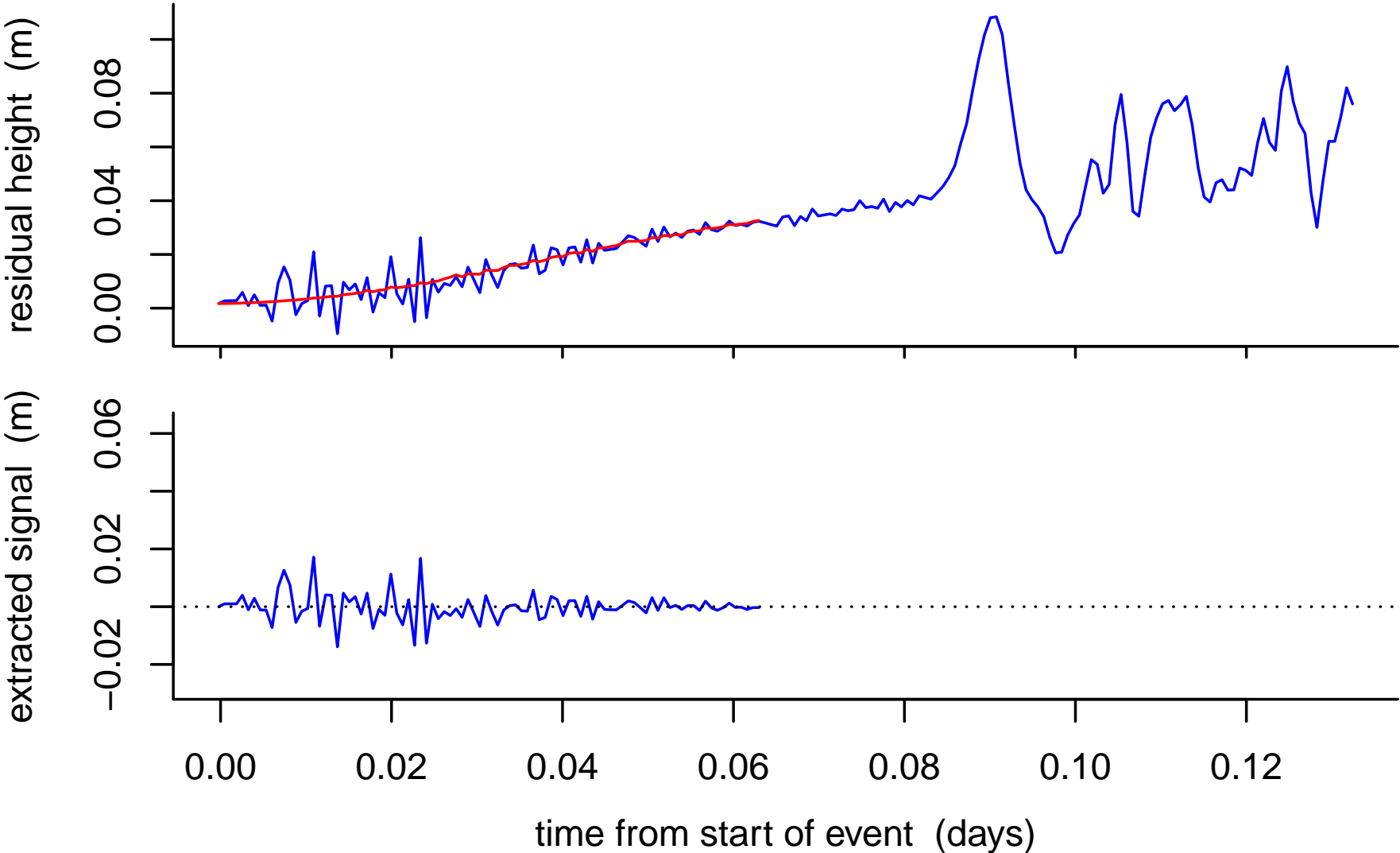
# Kalman Filtering/Smoothing (Local Level Model)



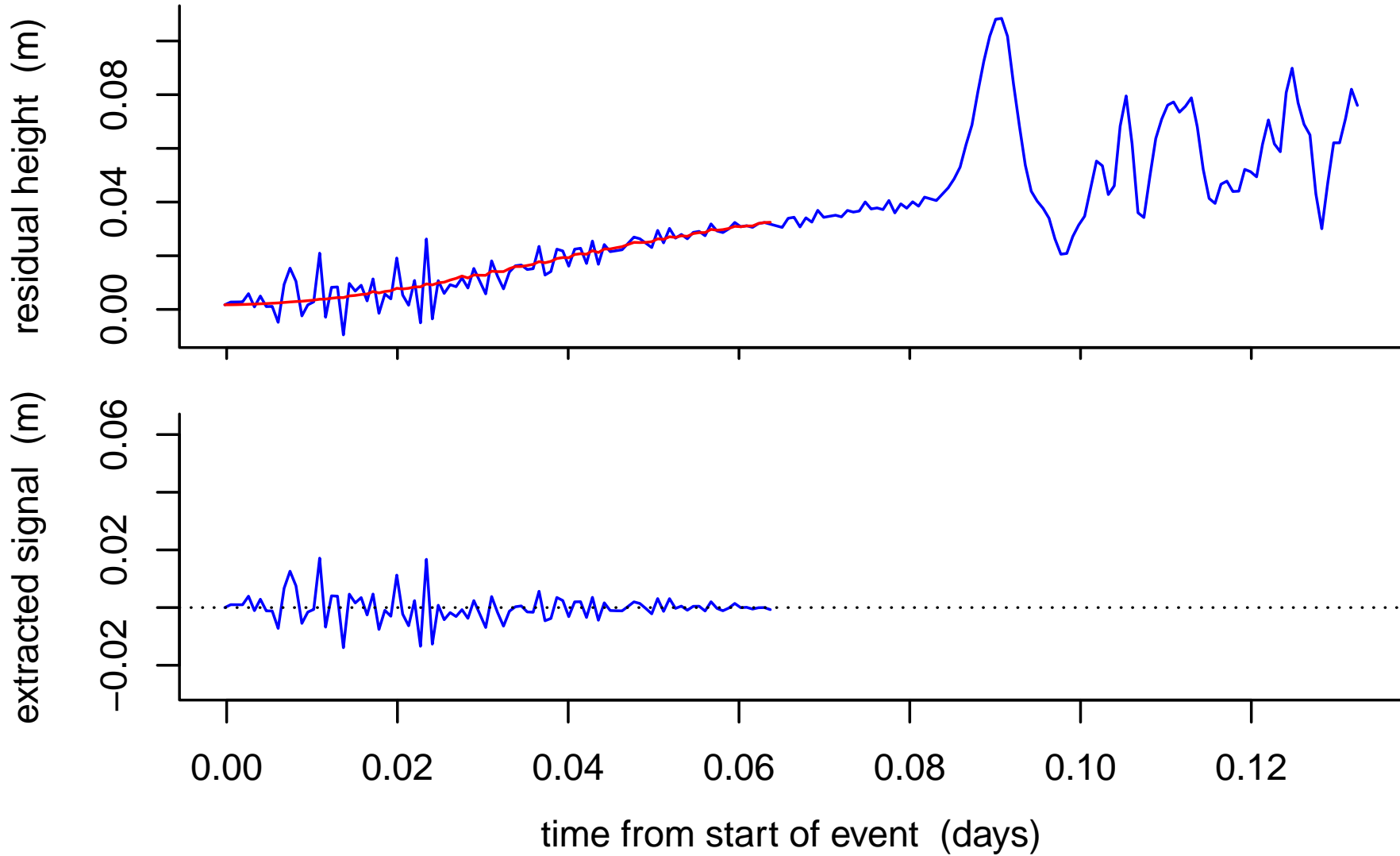
# Kalman Filtering/Smoothing (Local Level Model)



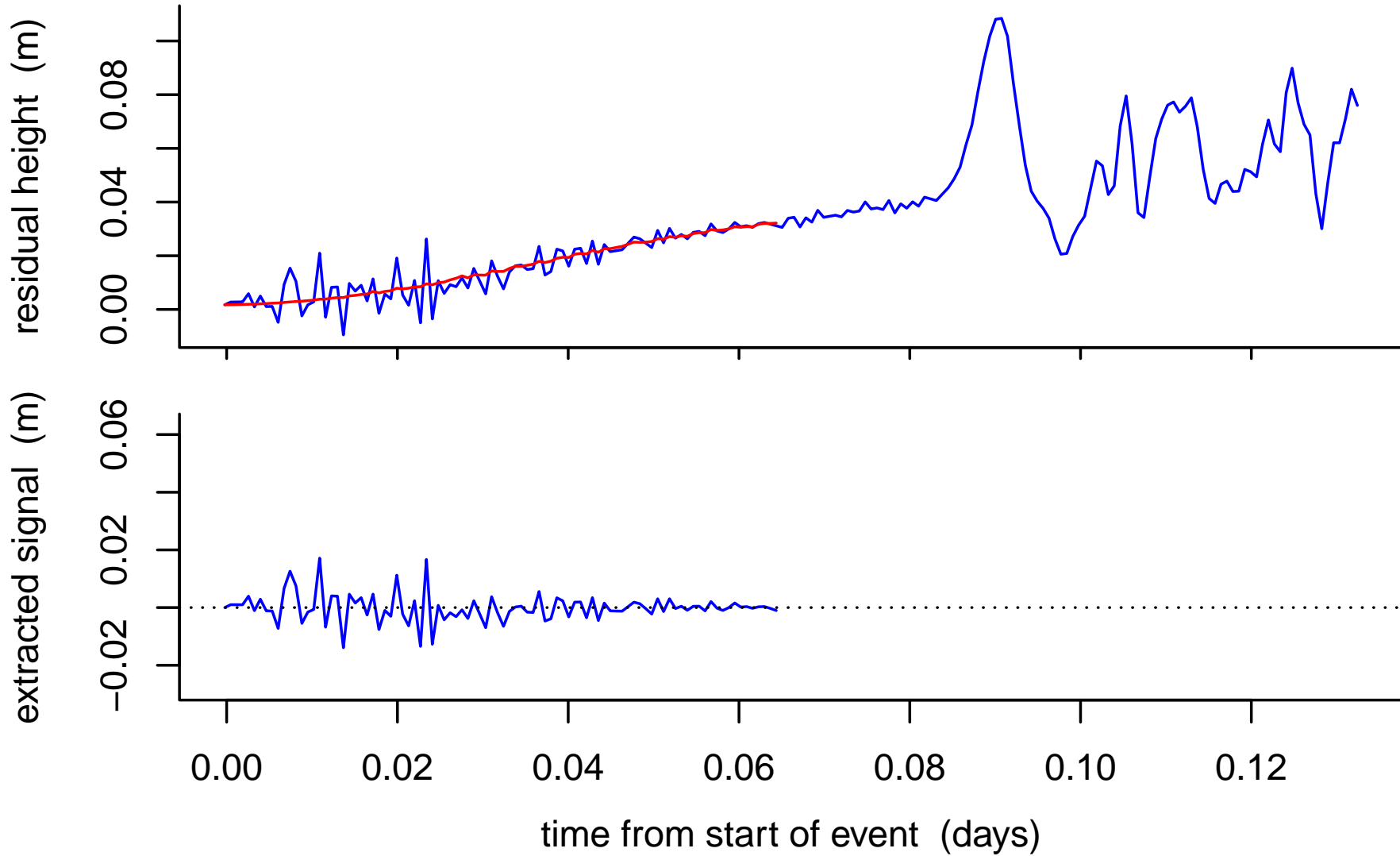
# Kalman Filtering/Smoothing (Local Level Model)



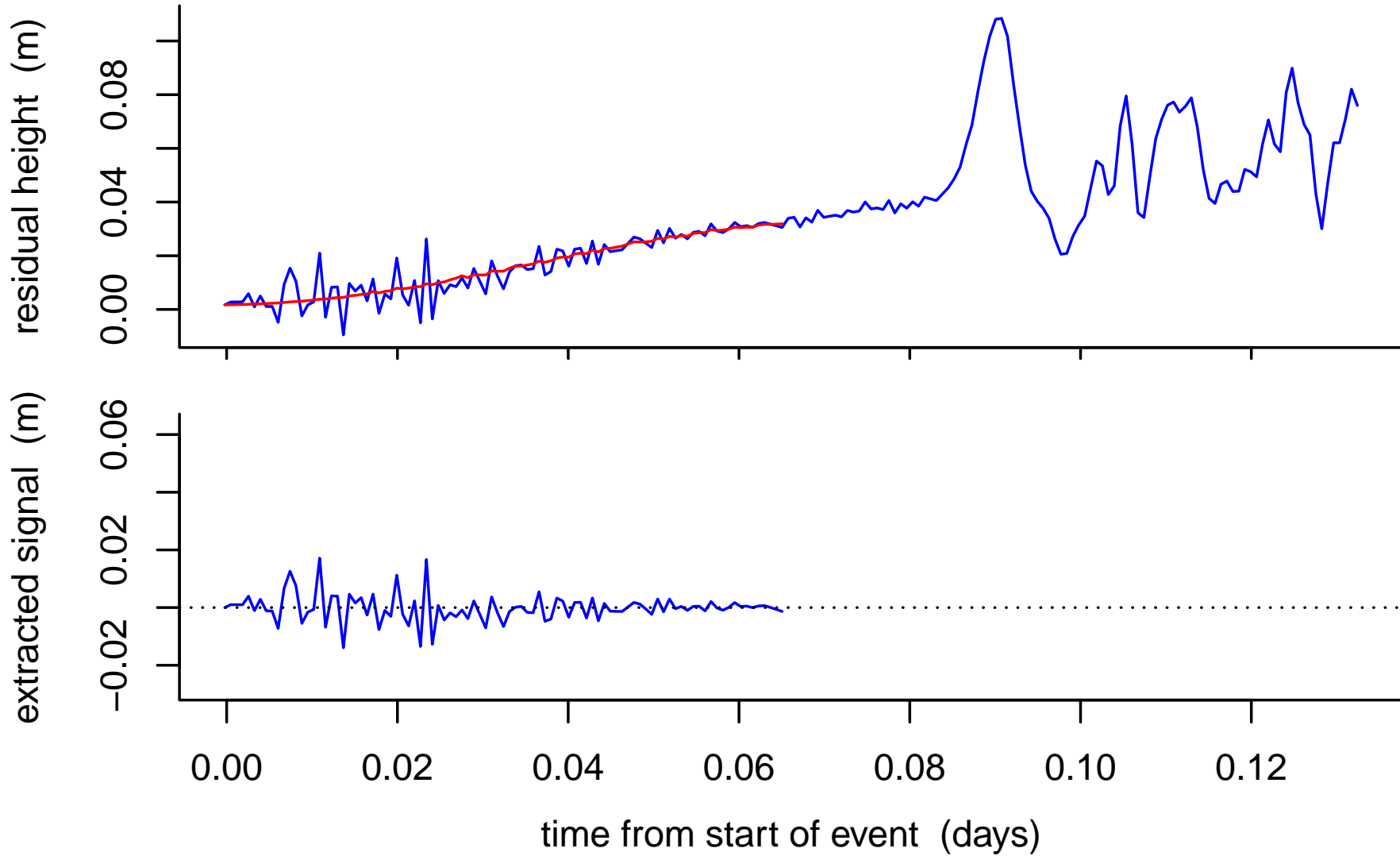
# Kalman Filtering/Smoothing (Local Level Model)



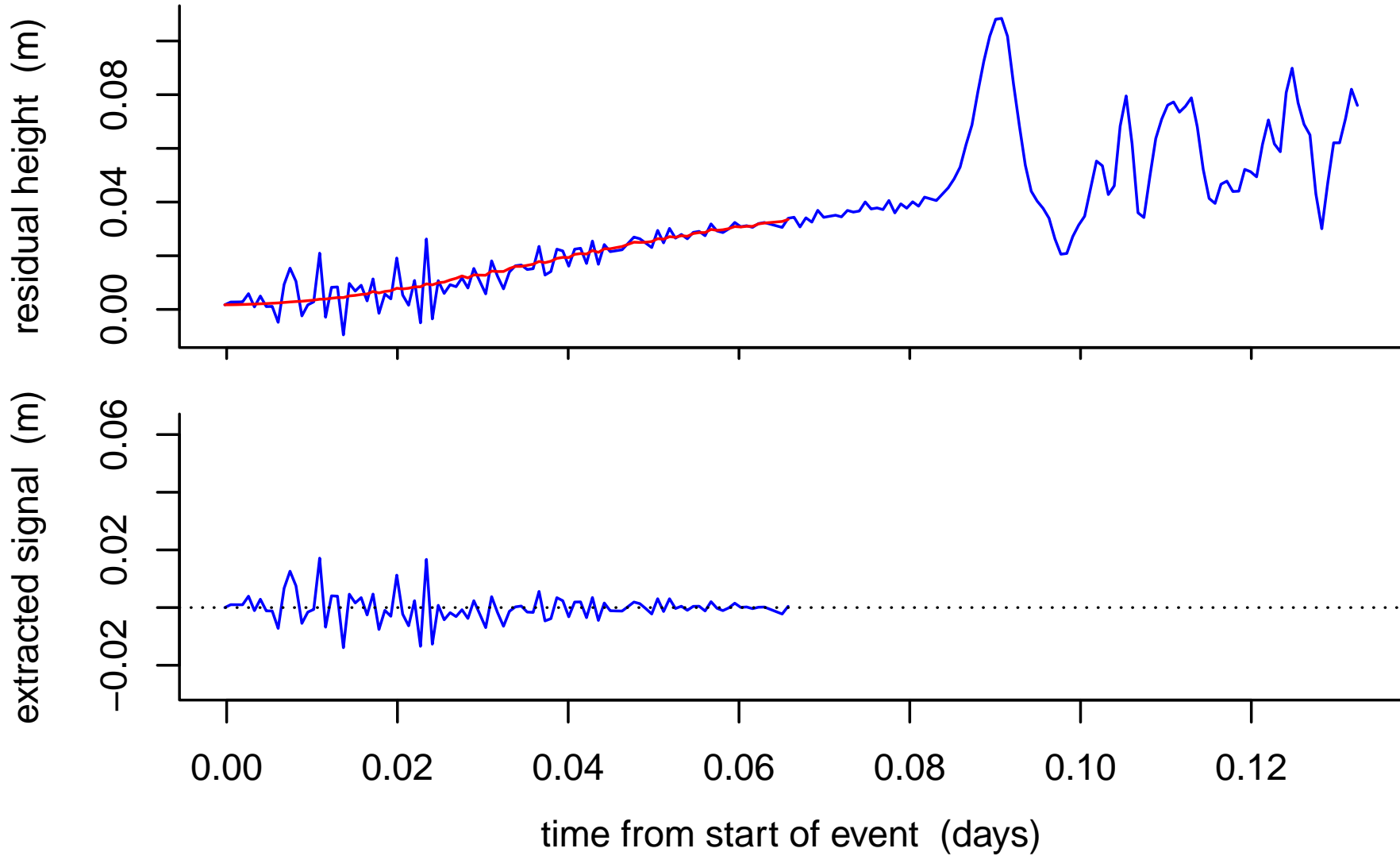
# Kalman Filtering/Smoothing (Local Level Model)



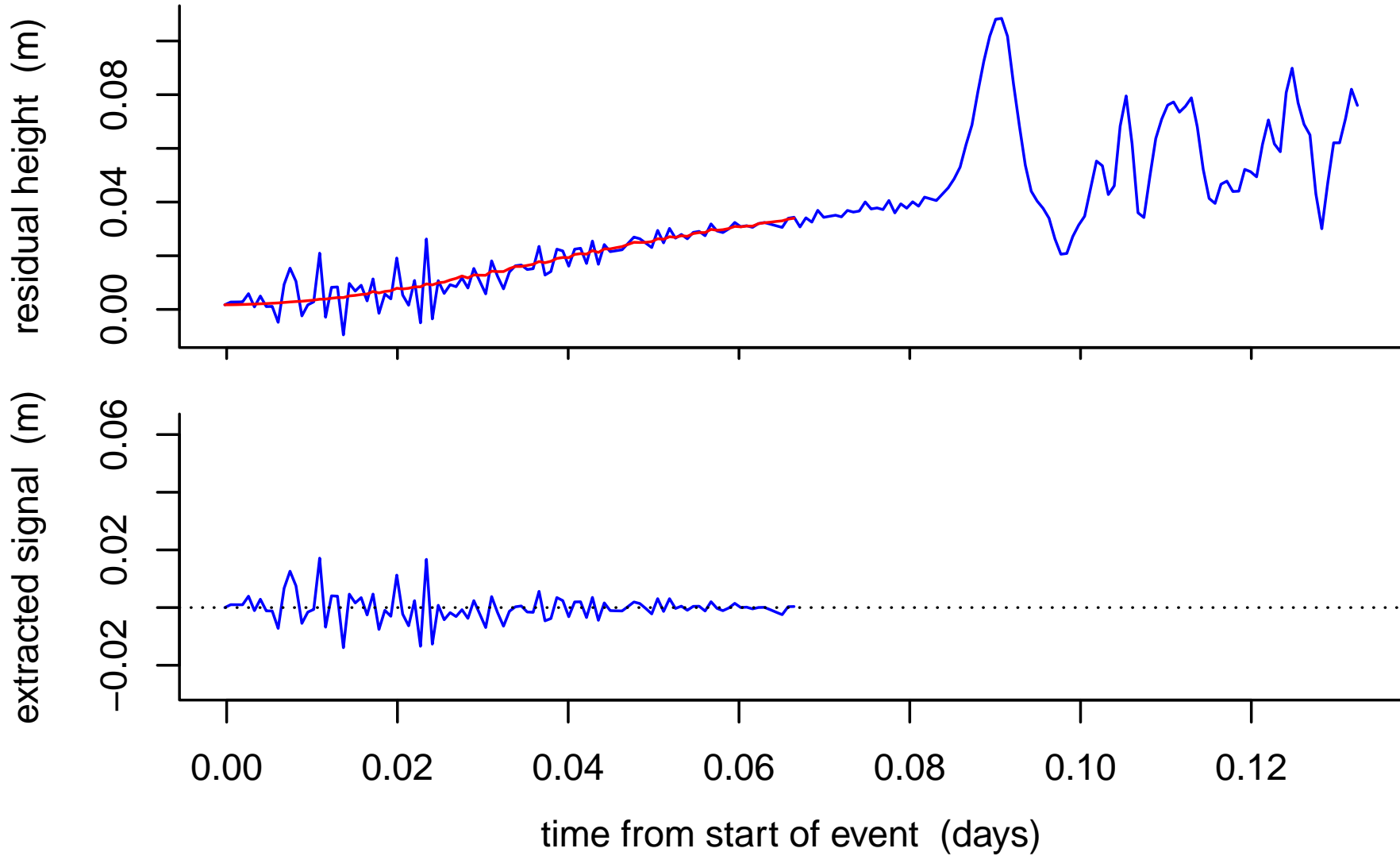
# Kalman Filtering/Smoothing (Local Level Model)



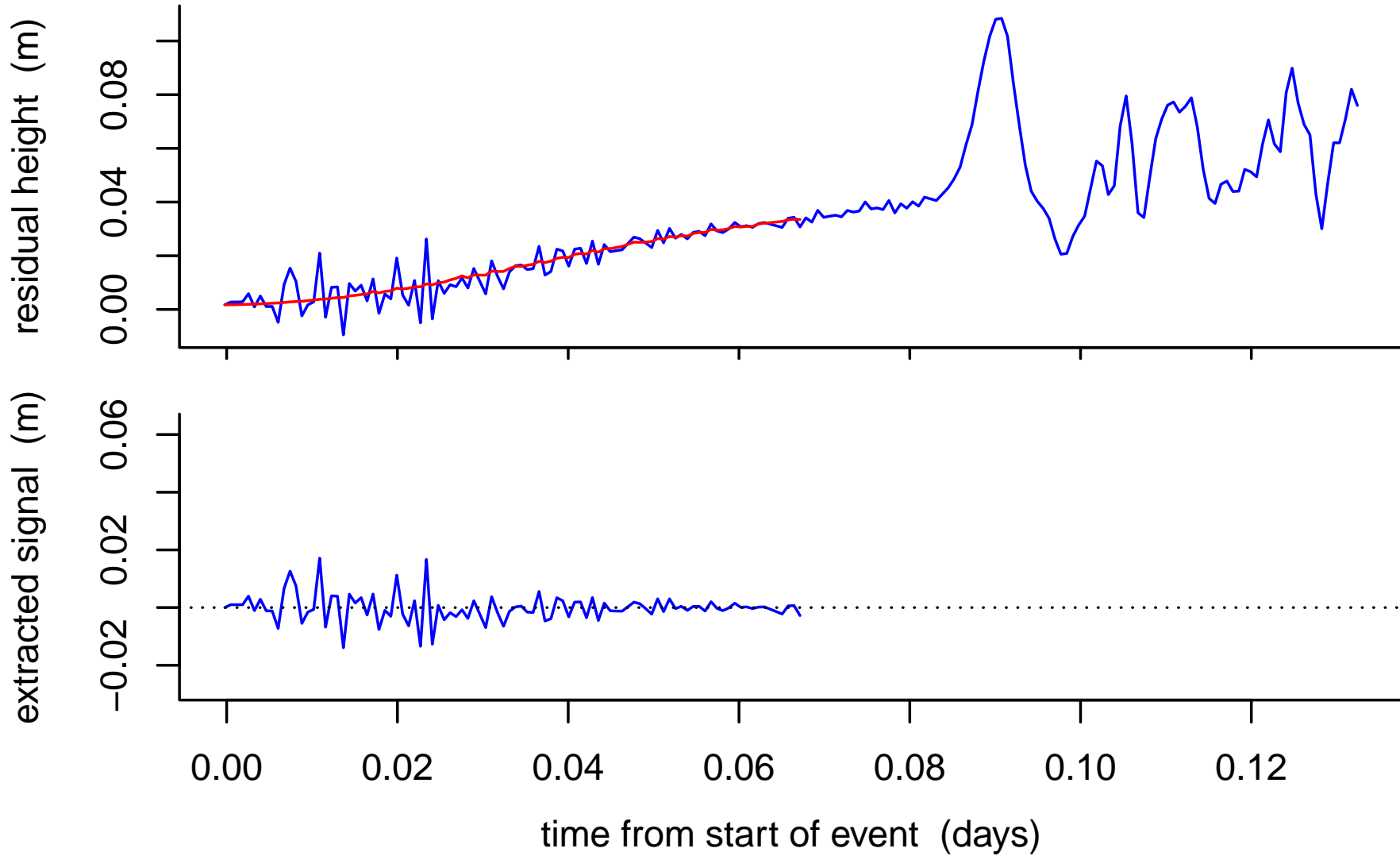
# Kalman Filtering/Smoothing (Local Level Model)



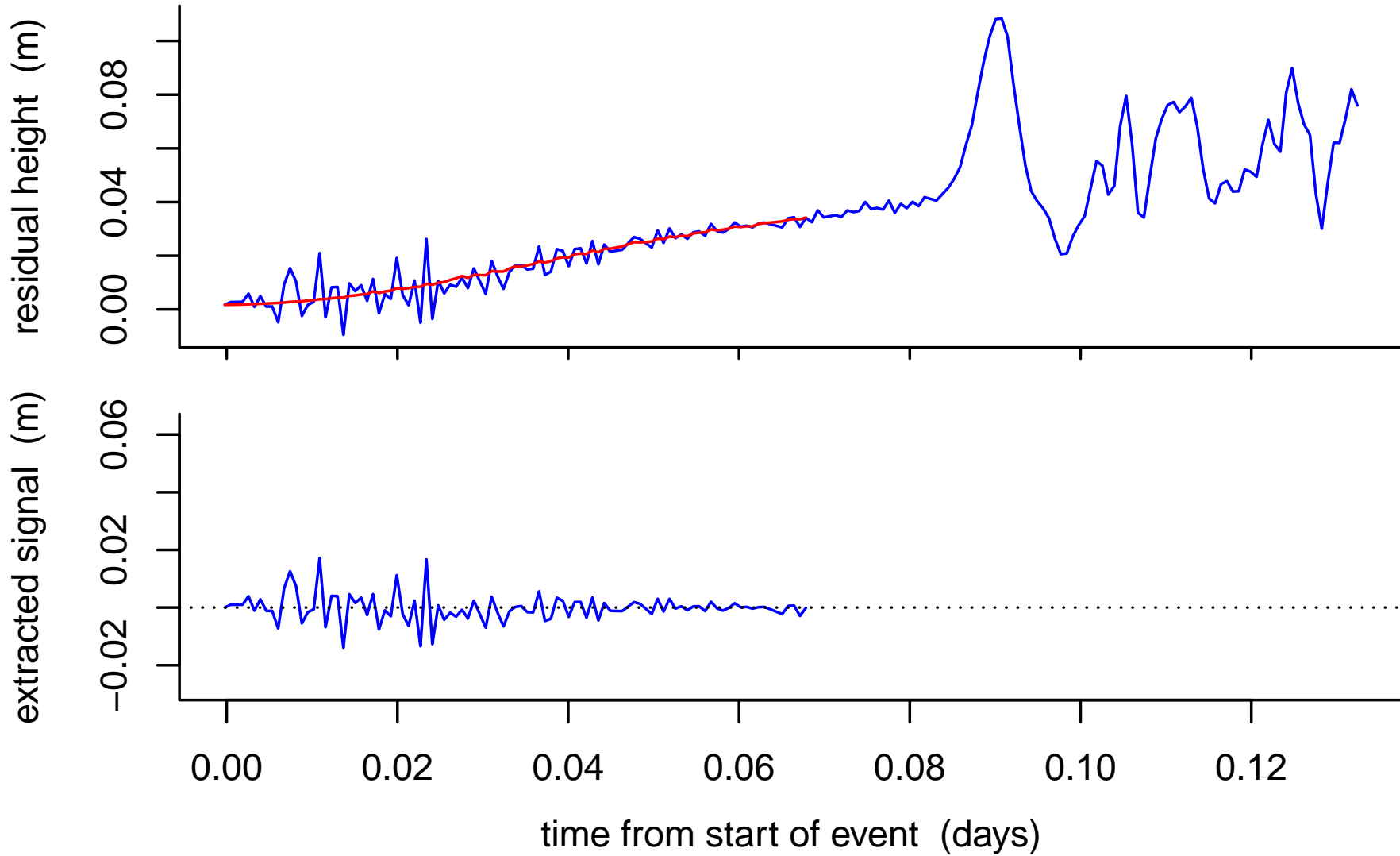
# Kalman Filtering/Smoothing (Local Level Model)



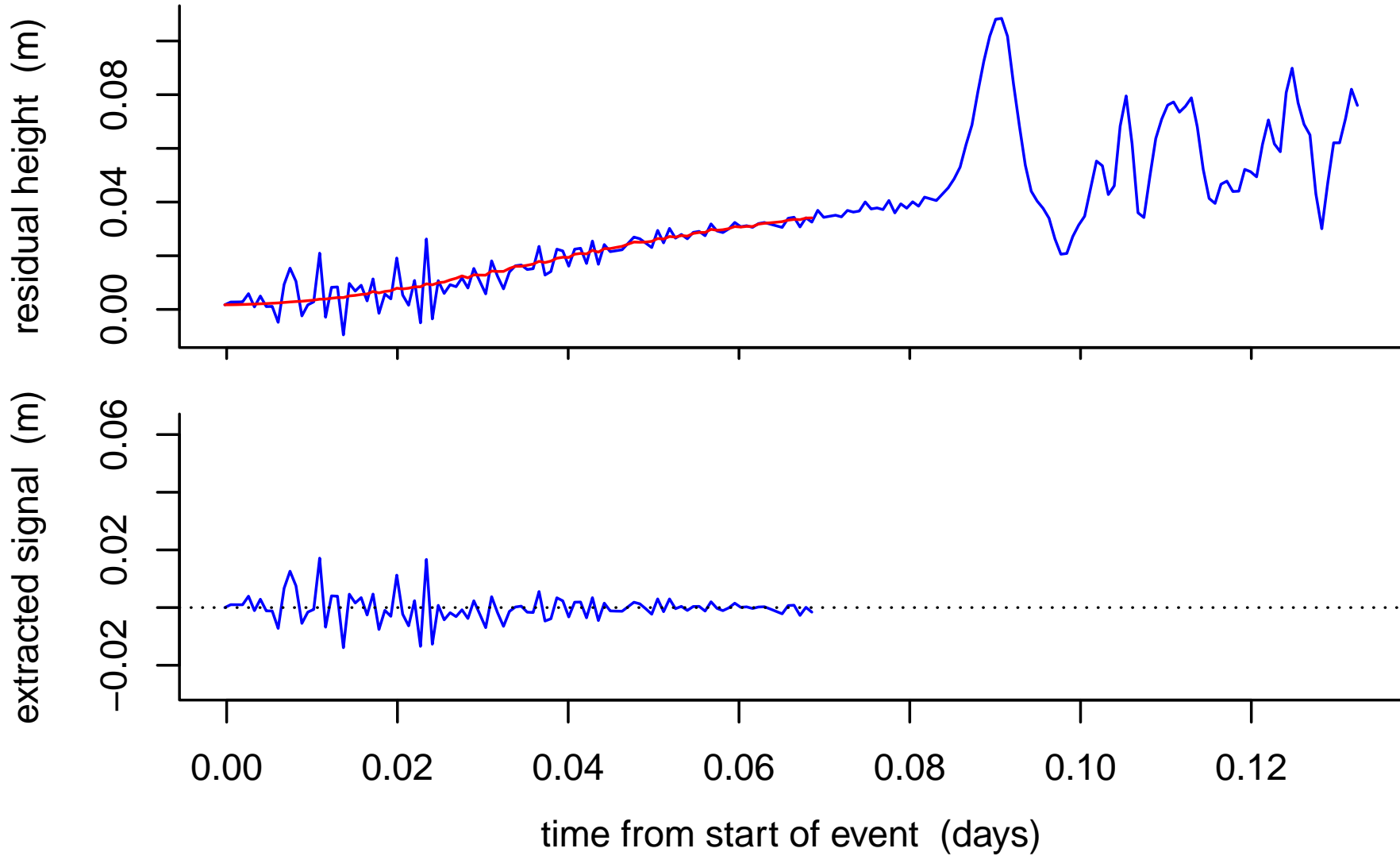
# Kalman Filtering/Smoothing (Local Level Model)



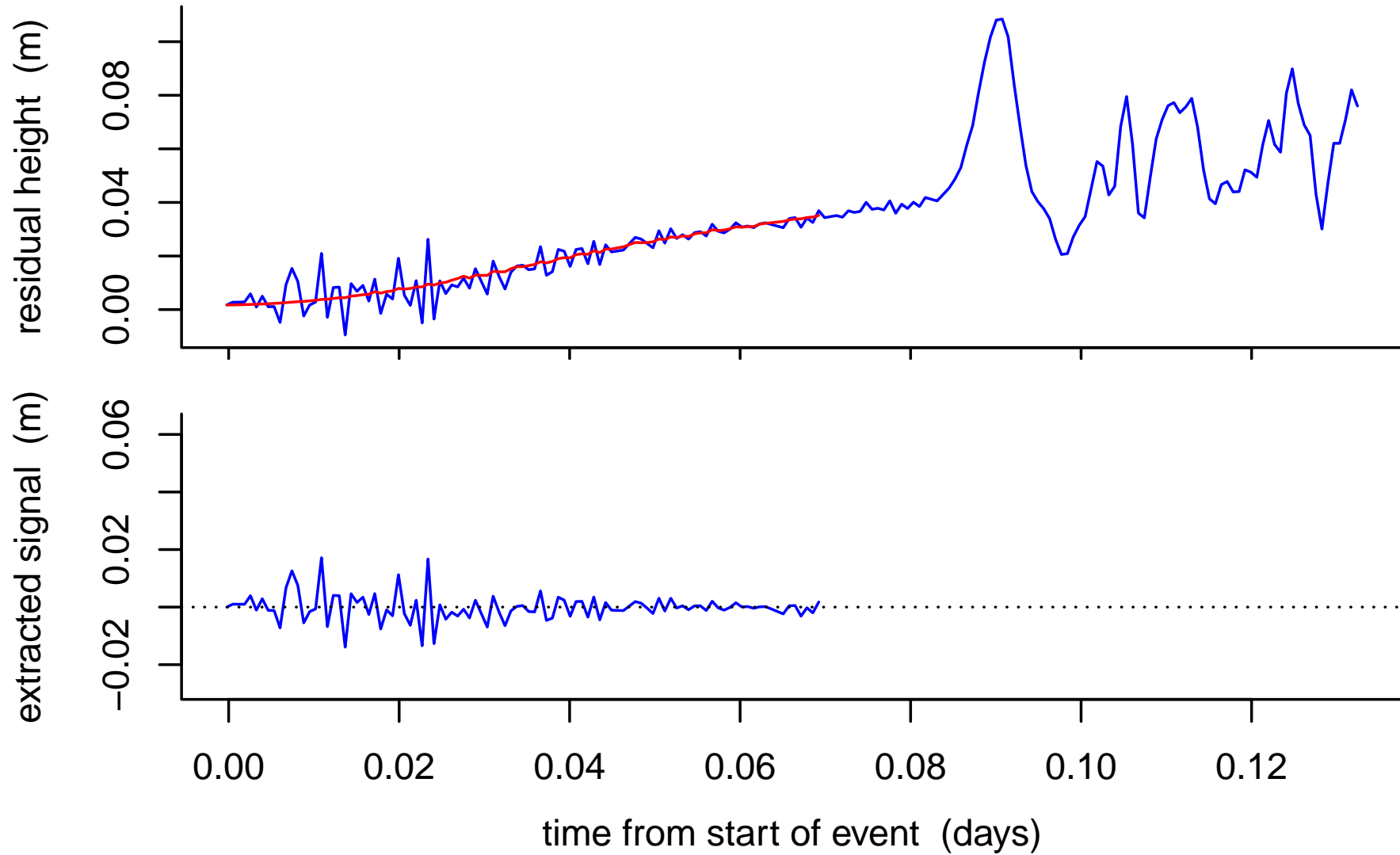
# Kalman Filtering/Smoothing (Local Level Model)



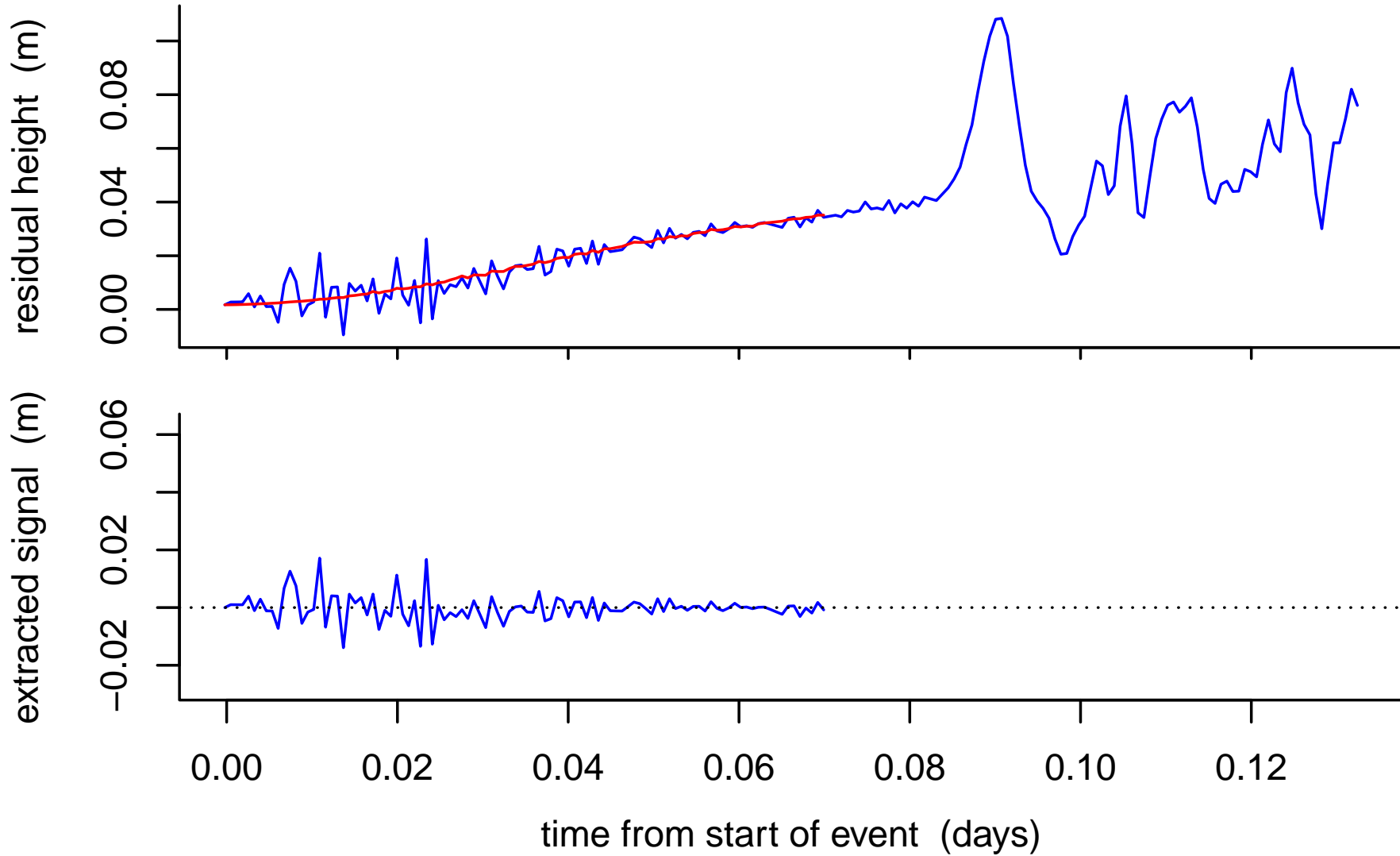
# Kalman Filtering/Smoothing (Local Level Model)



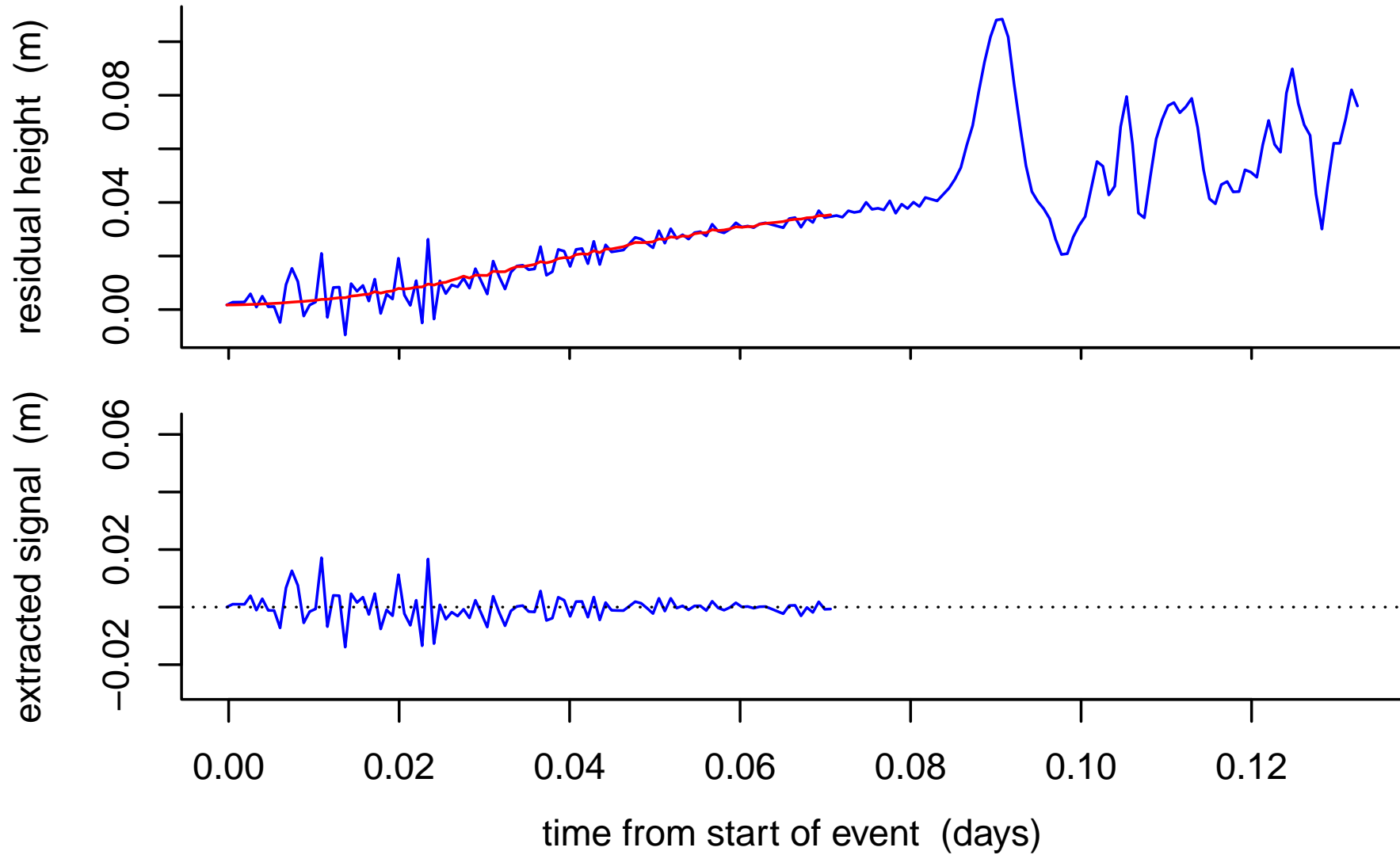
# Kalman Filtering/Smoothing (Local Level Model)



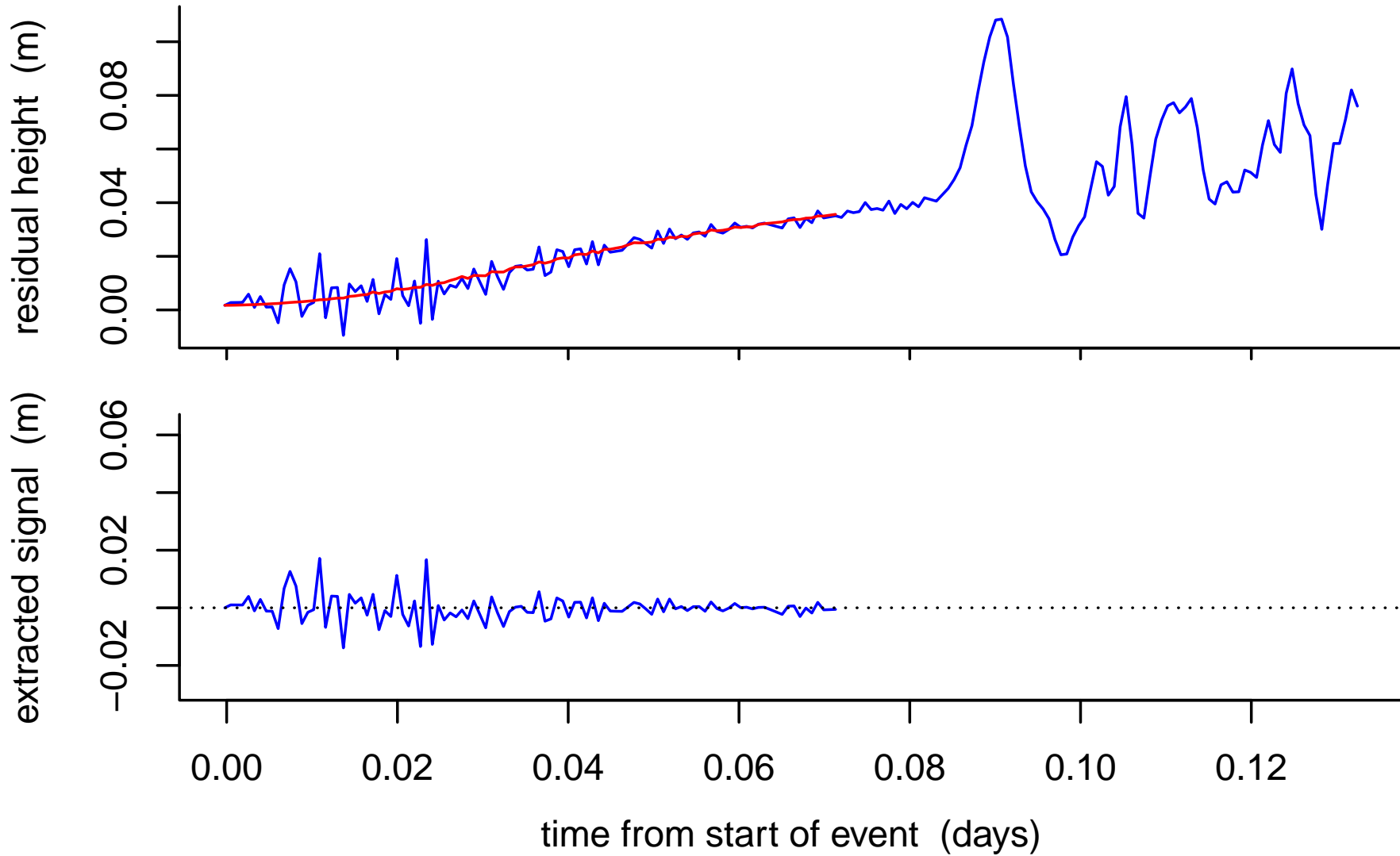
# Kalman Filtering/Smoothing (Local Level Model)



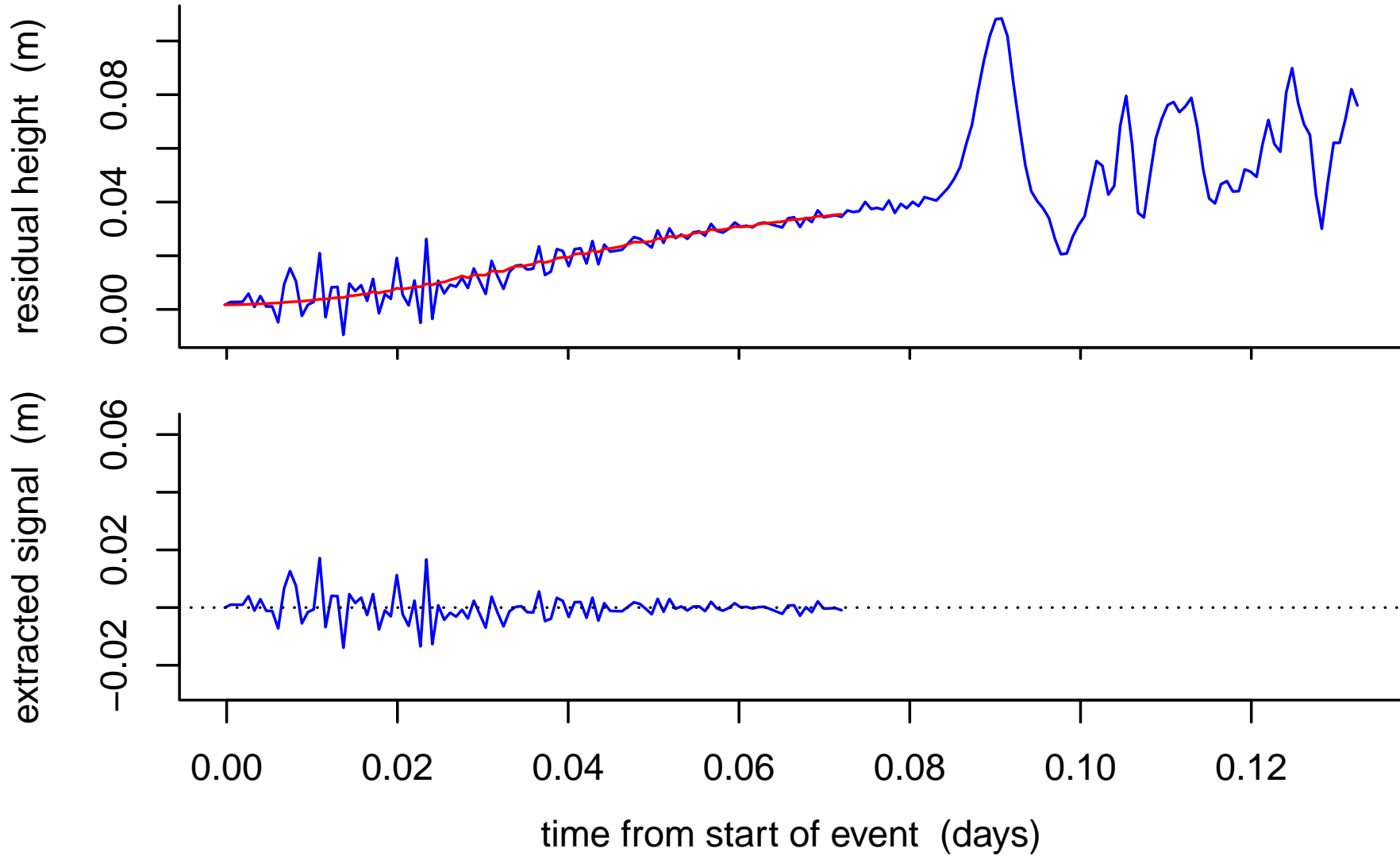
# Kalman Filtering/Smoothing (Local Level Model)



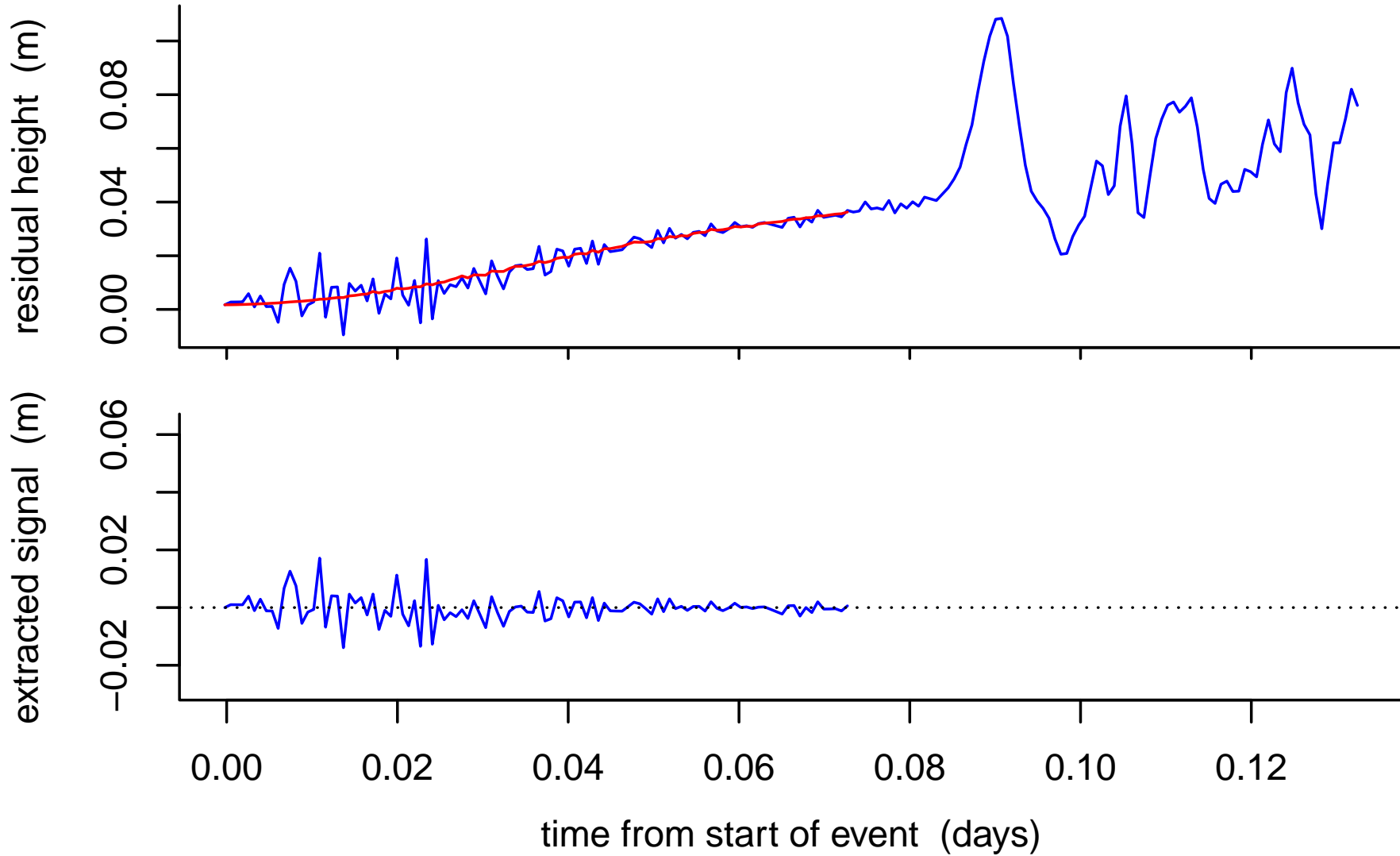
# Kalman Filtering/Smoothing (Local Level Model)



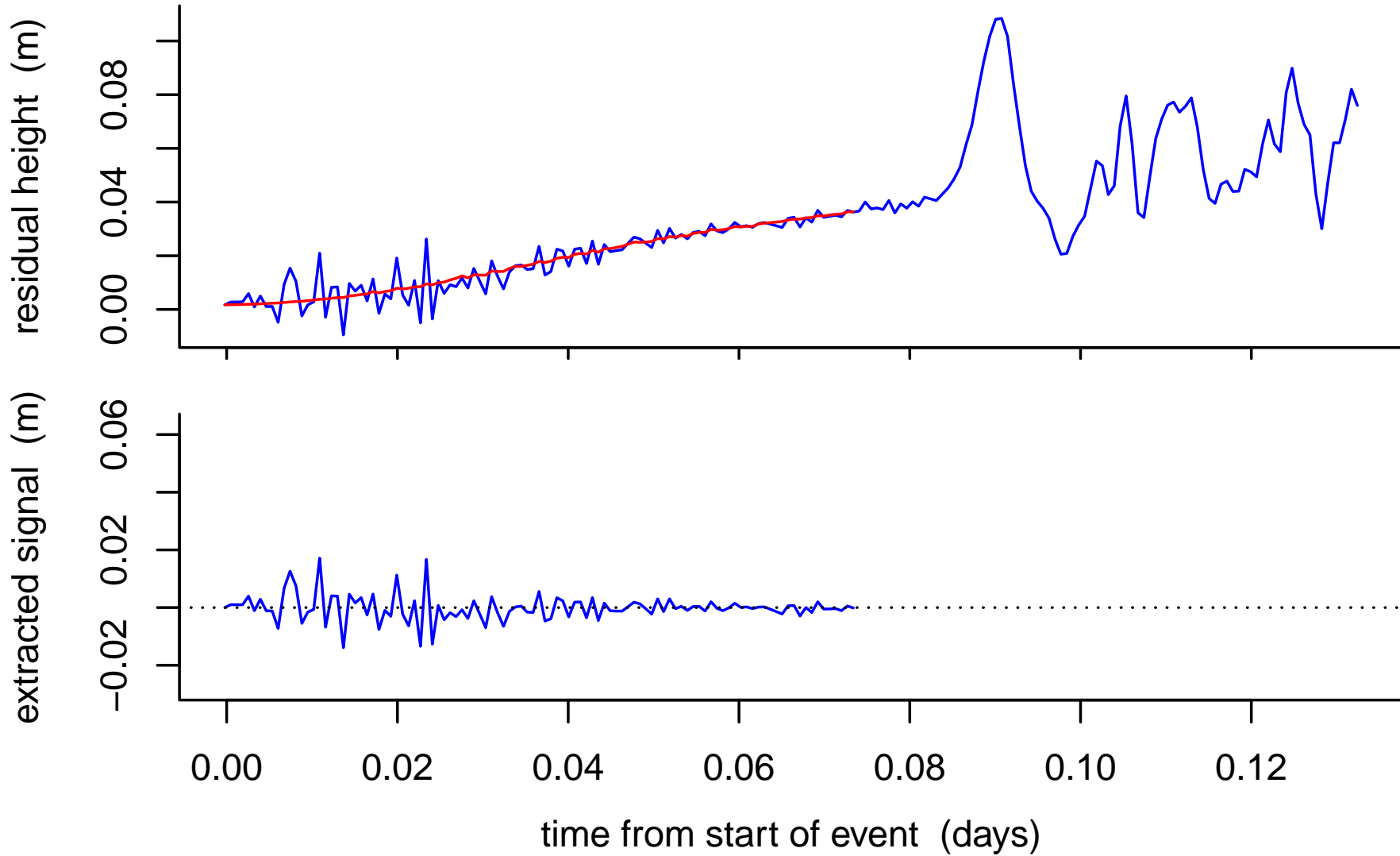
# Kalman Filtering/Smoothing (Local Level Model)



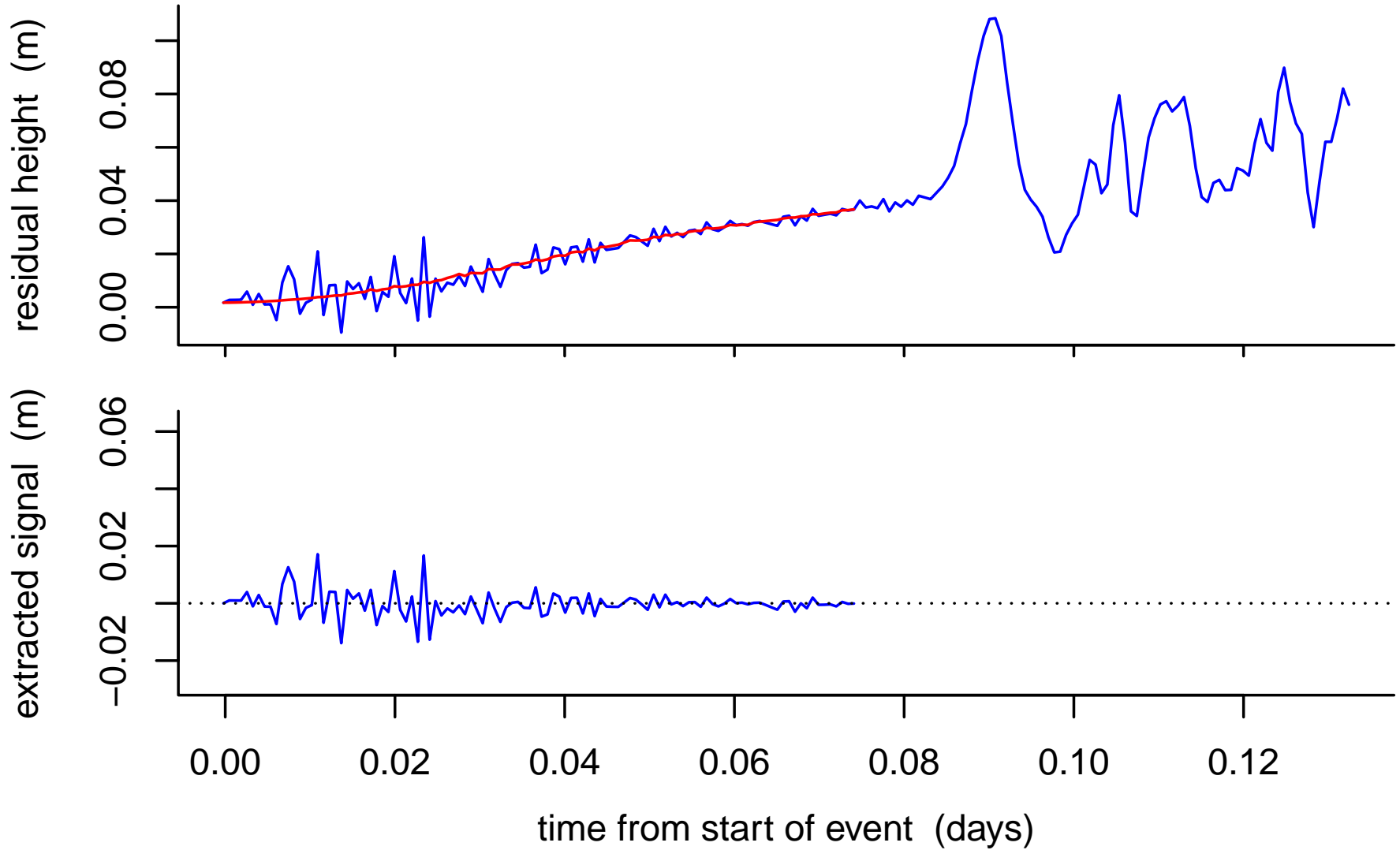
# Kalman Filtering/Smoothing (Local Level Model)



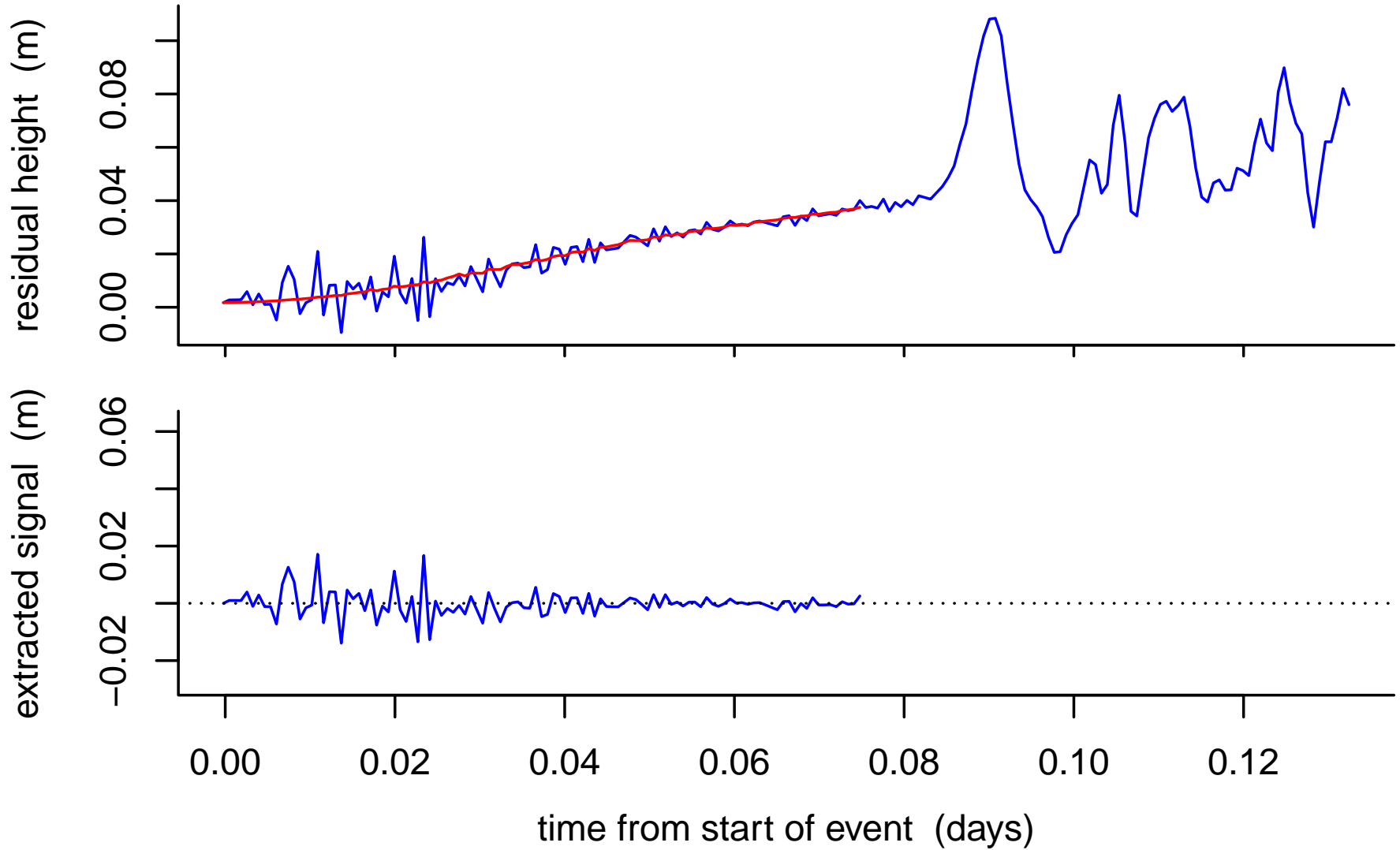
# Kalman Filtering/Smoothing (Local Level Model)



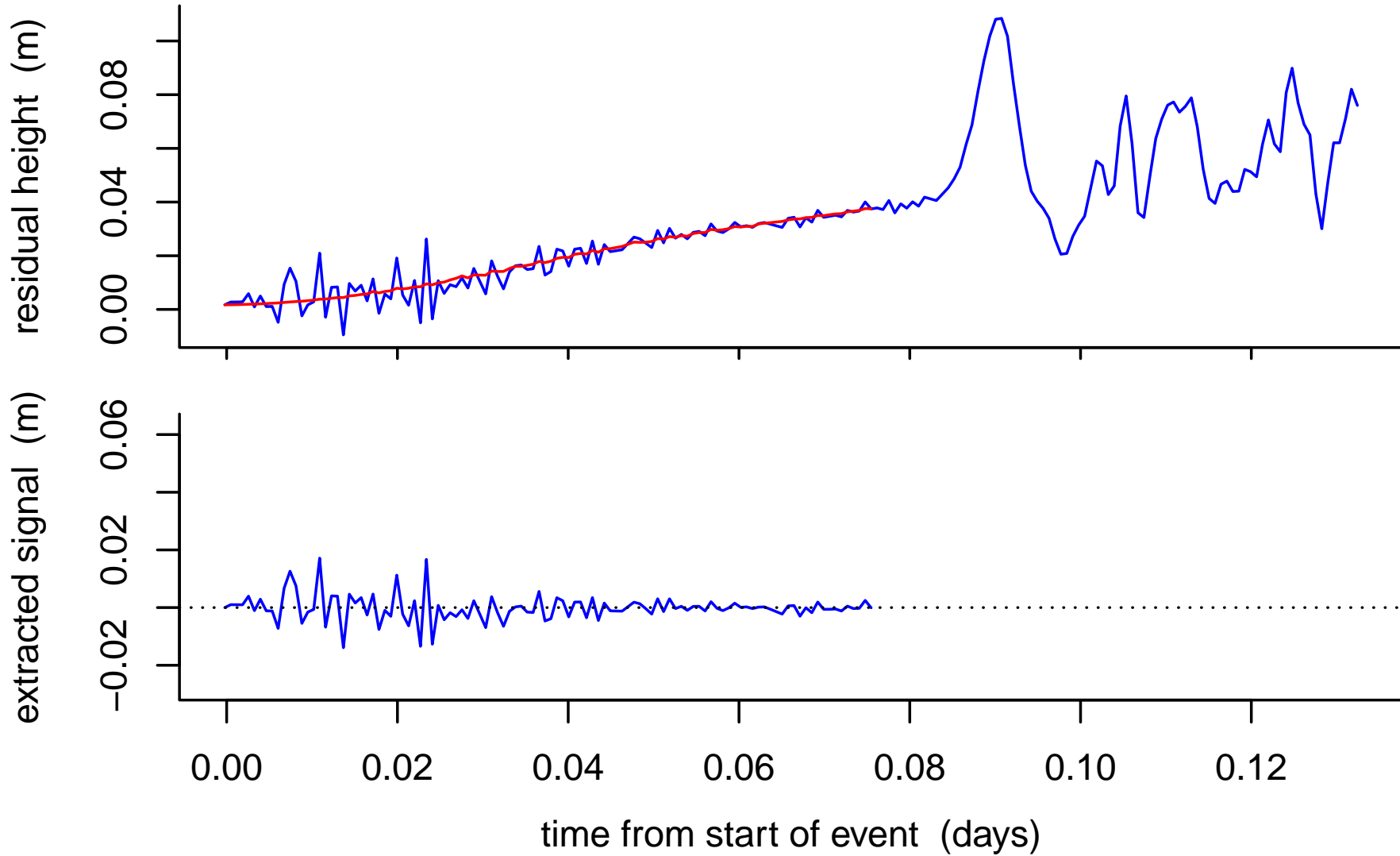
# Kalman Filtering/Smoothing (Local Level Model)



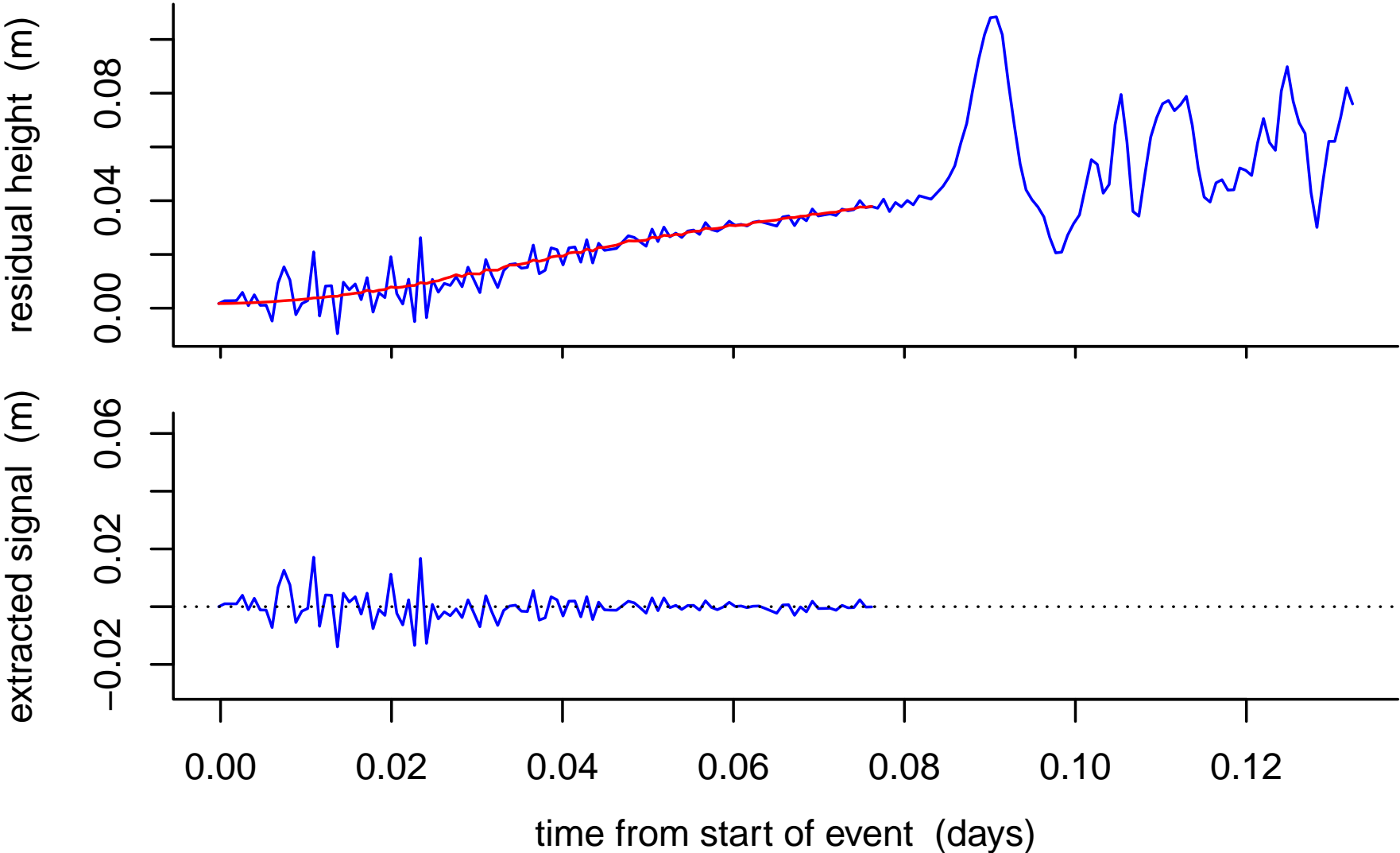
# Kalman Filtering/Smoothing (Local Level Model)



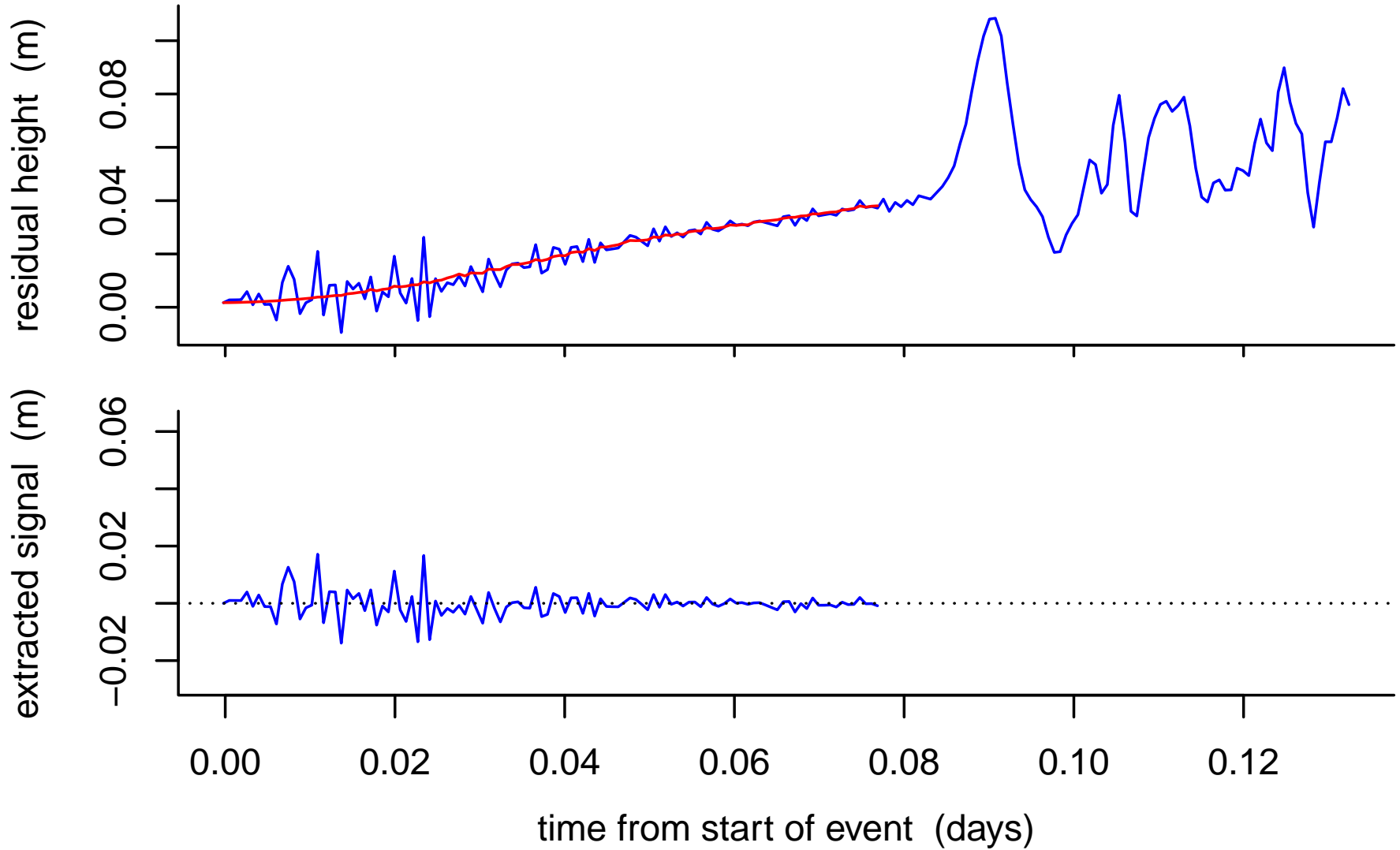
# Kalman Filtering/Smoothing (Local Level Model)



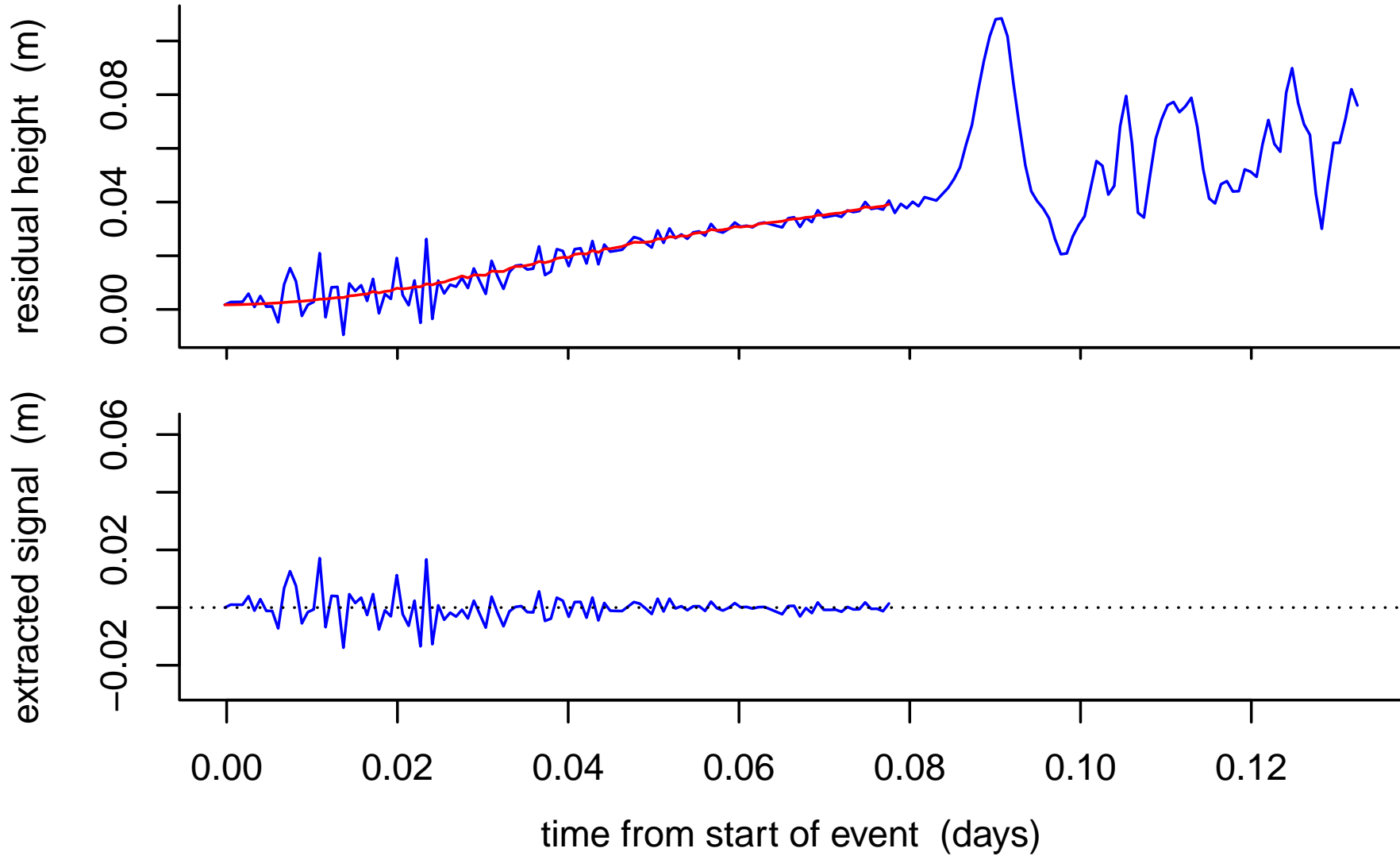
# Kalman Filtering/Smoothing (Local Level Model)



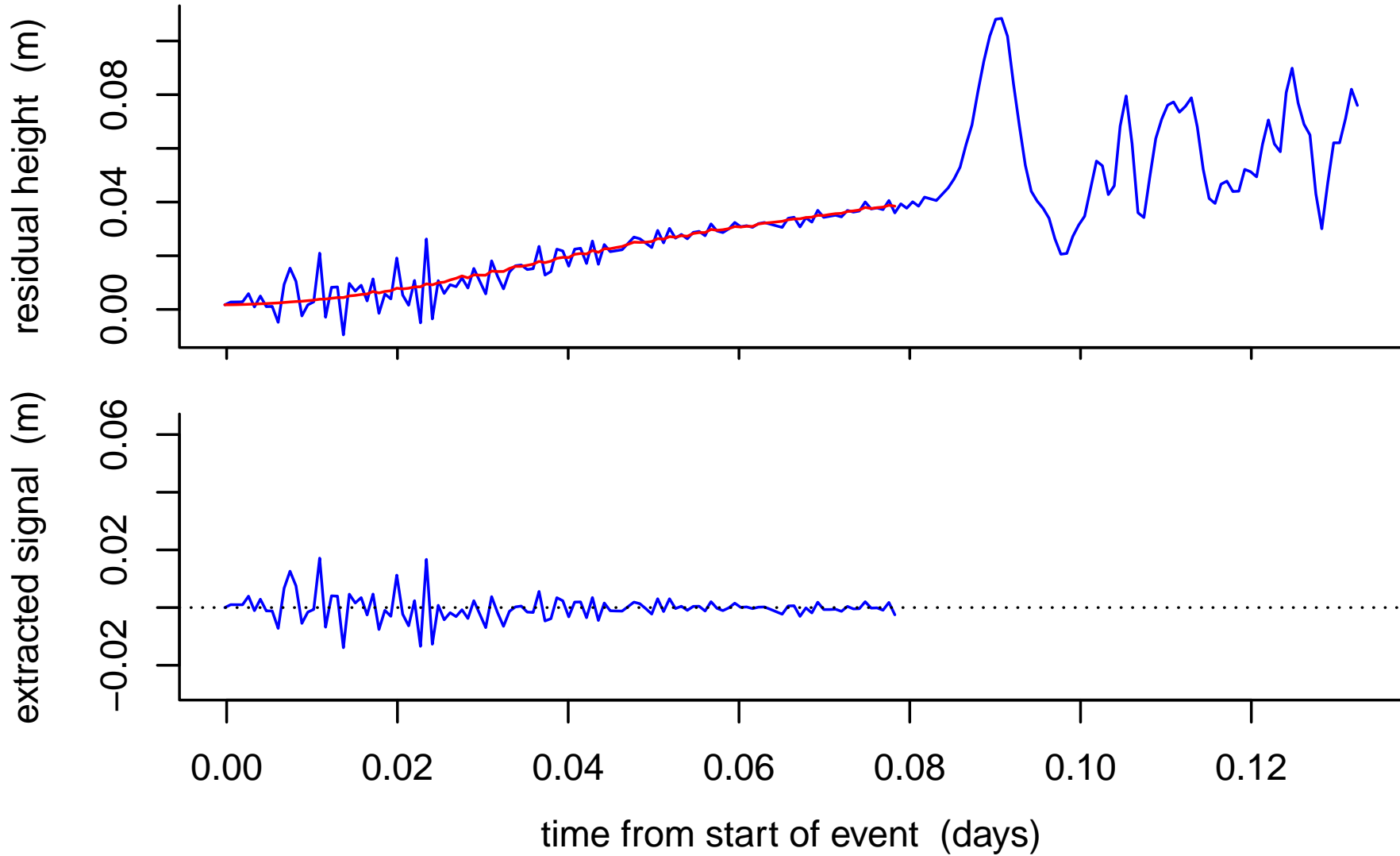
# Kalman Filtering/Smoothing (Local Level Model)



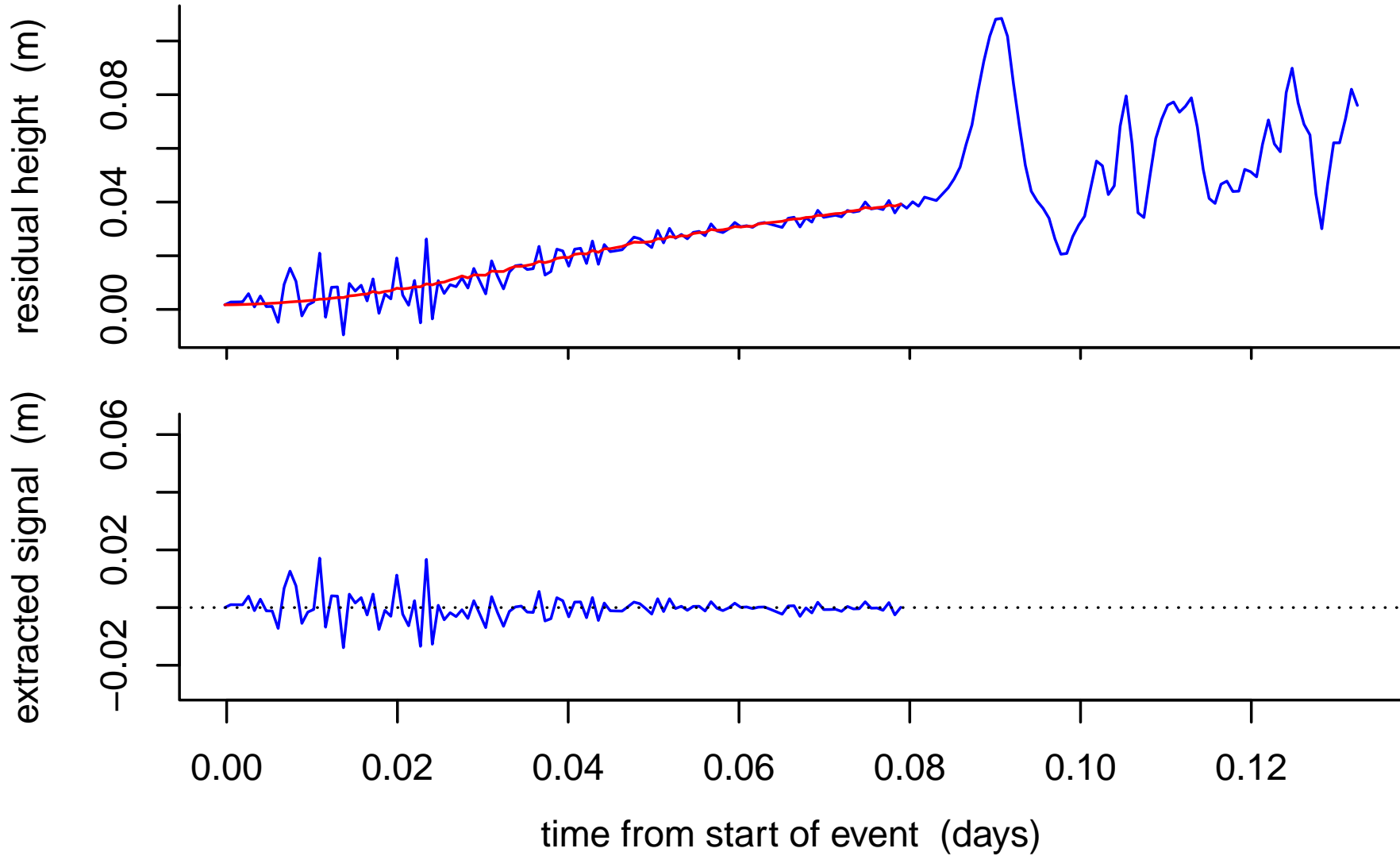
# Kalman Filtering/Smoothing (Local Level Model)



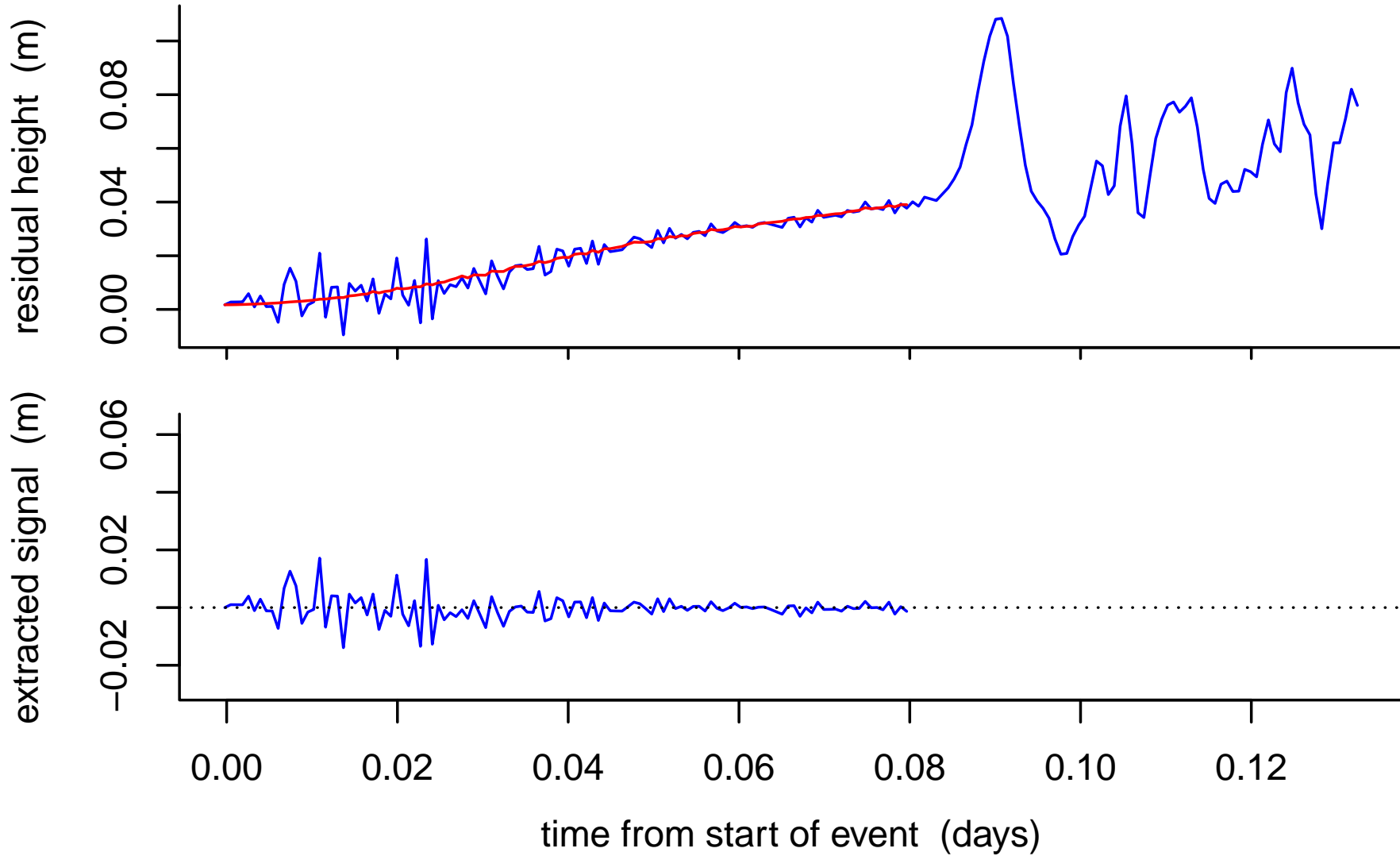
# Kalman Filtering/Smoothing (Local Level Model)



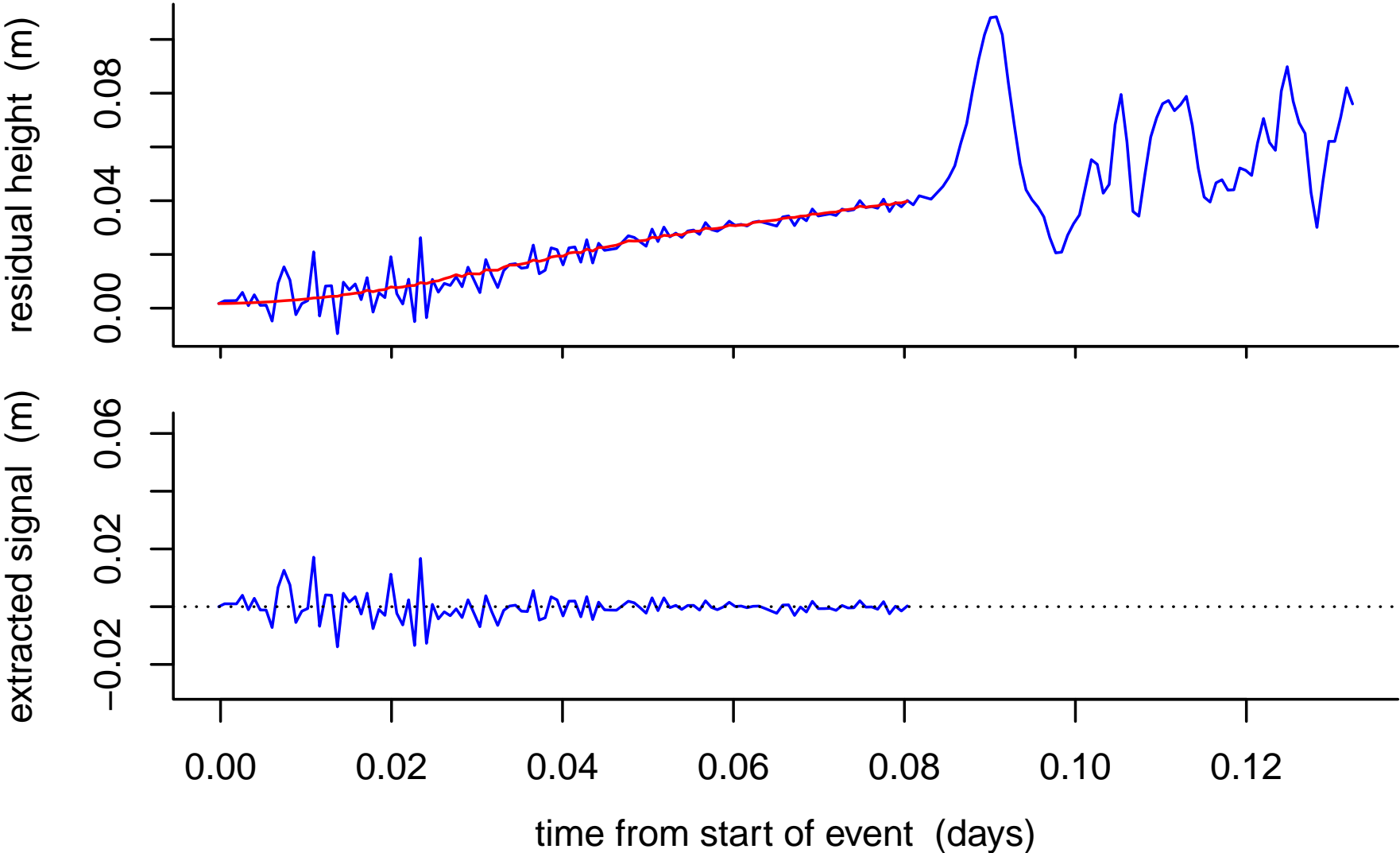
# Kalman Filtering/Smoothing (Local Level Model)



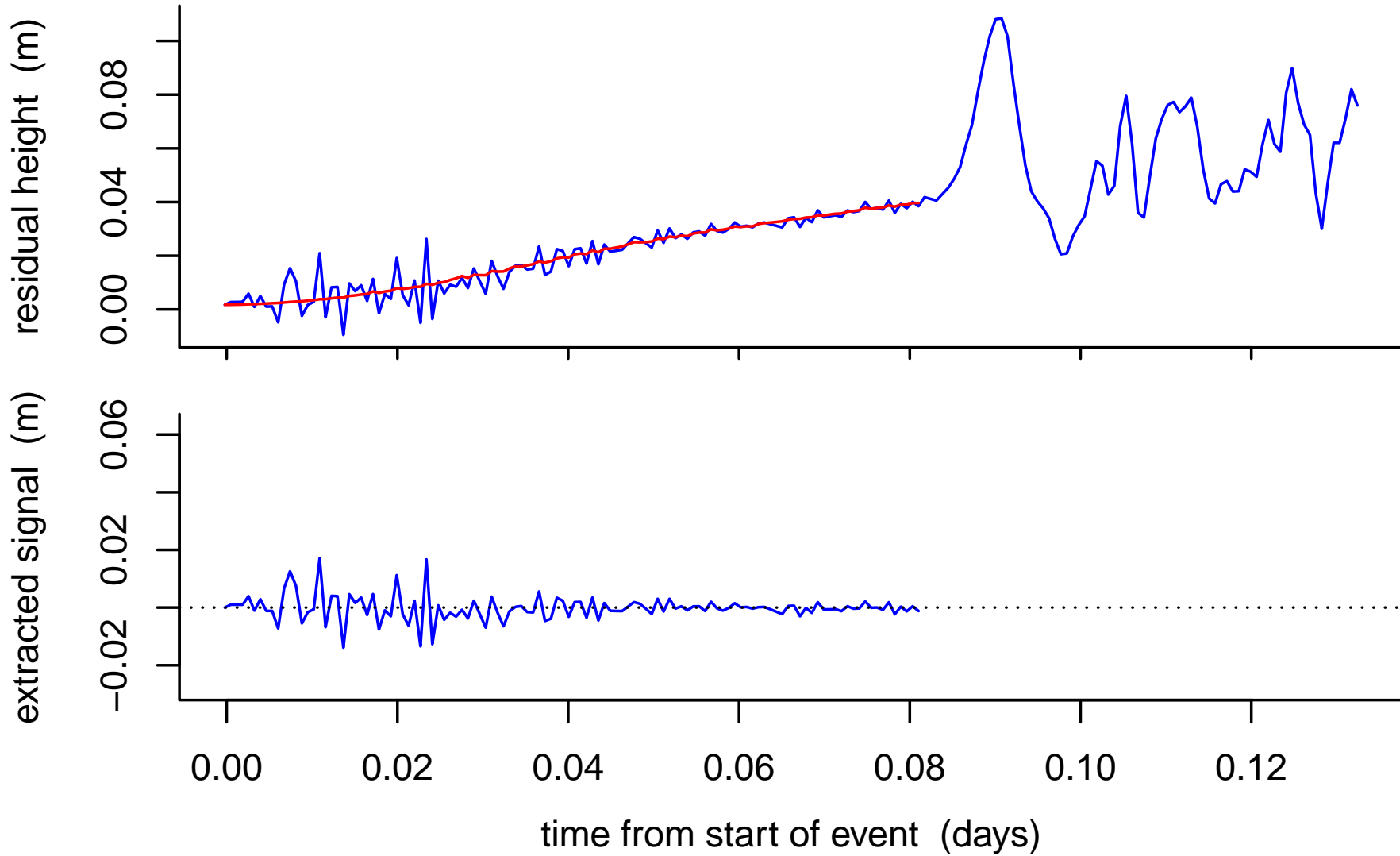
# Kalman Filtering/Smoothing (Local Level Model)



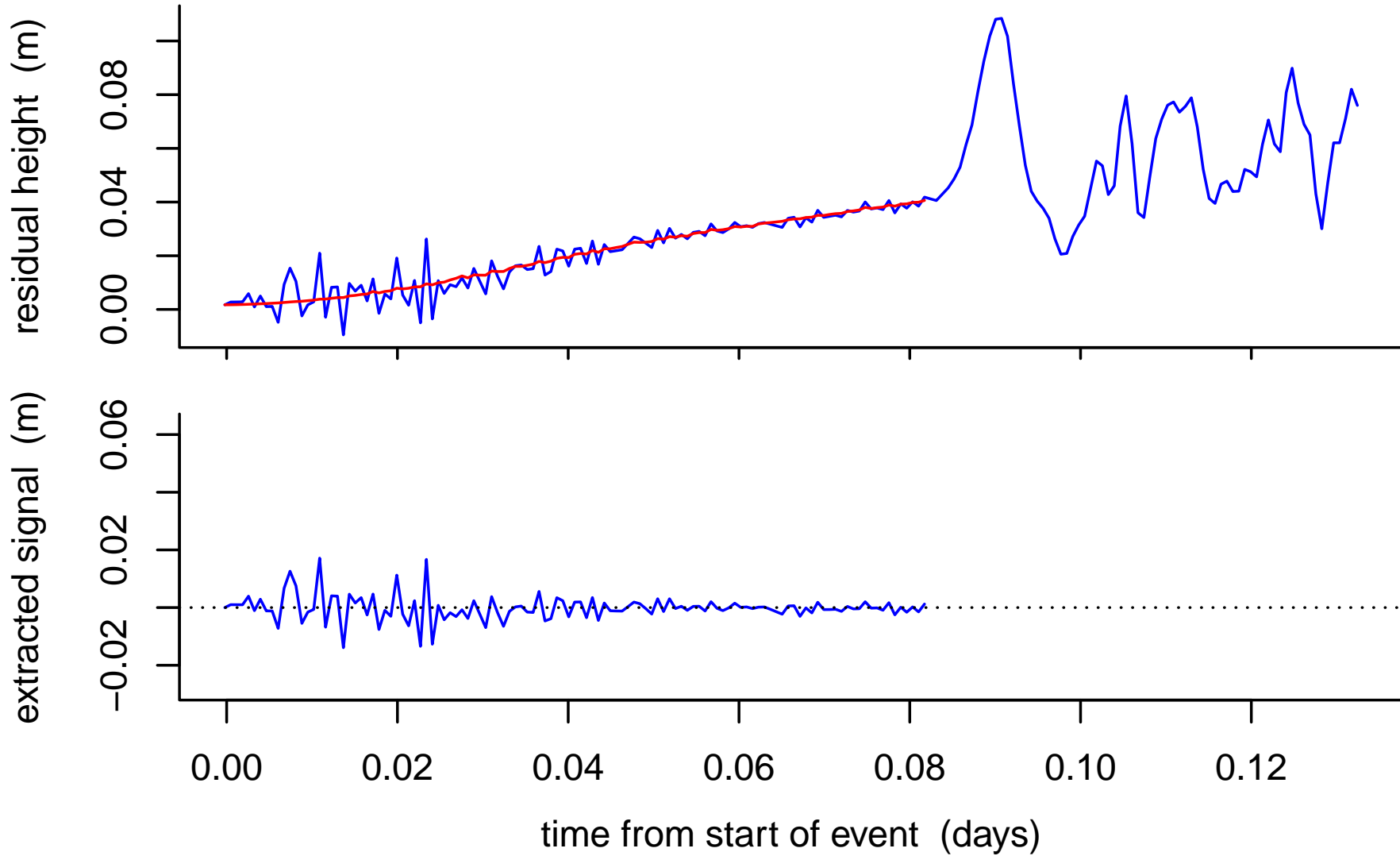
# Kalman Filtering/Smoothing (Local Level Model)



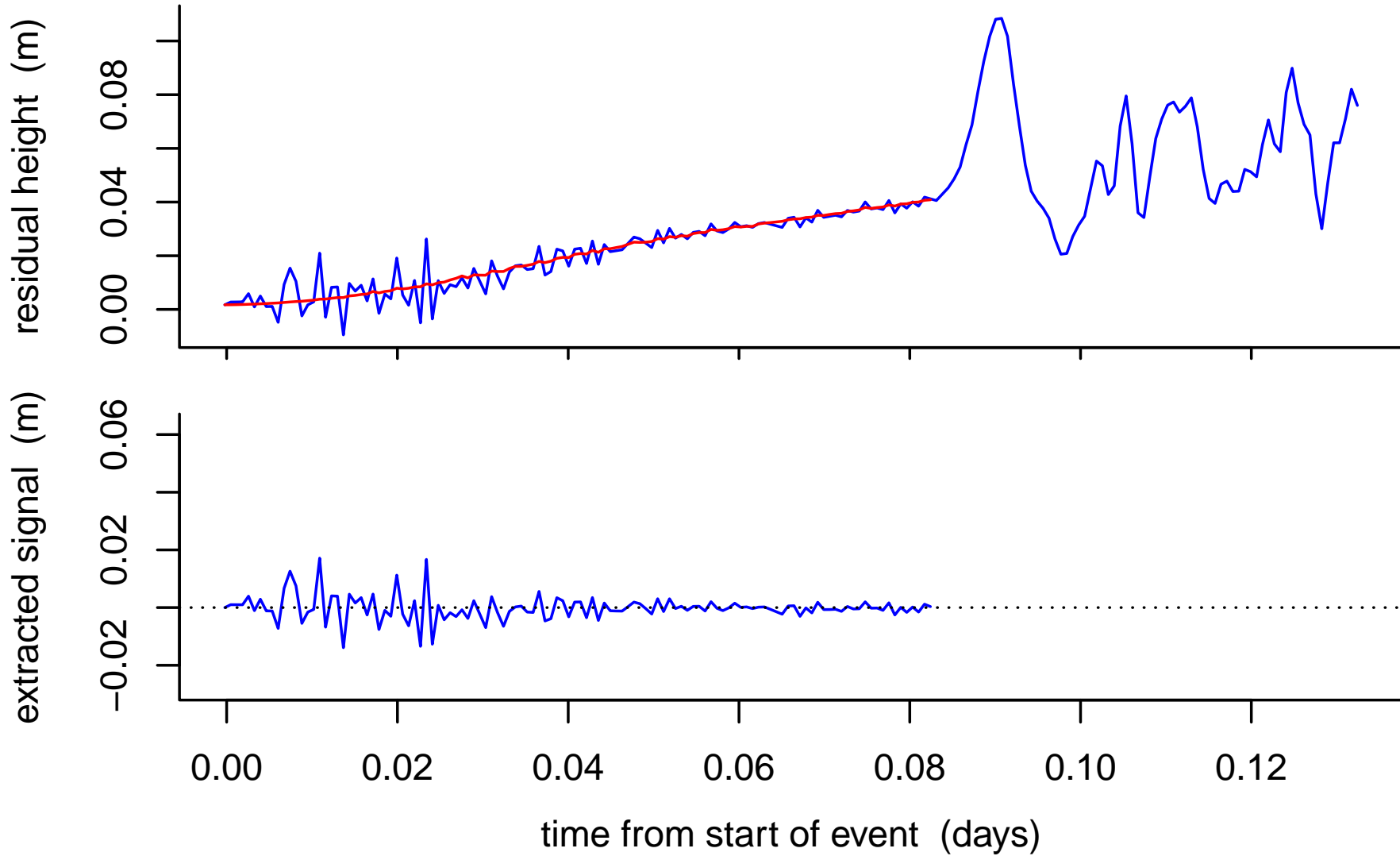
# Kalman Filtering/Smoothing (Local Level Model)



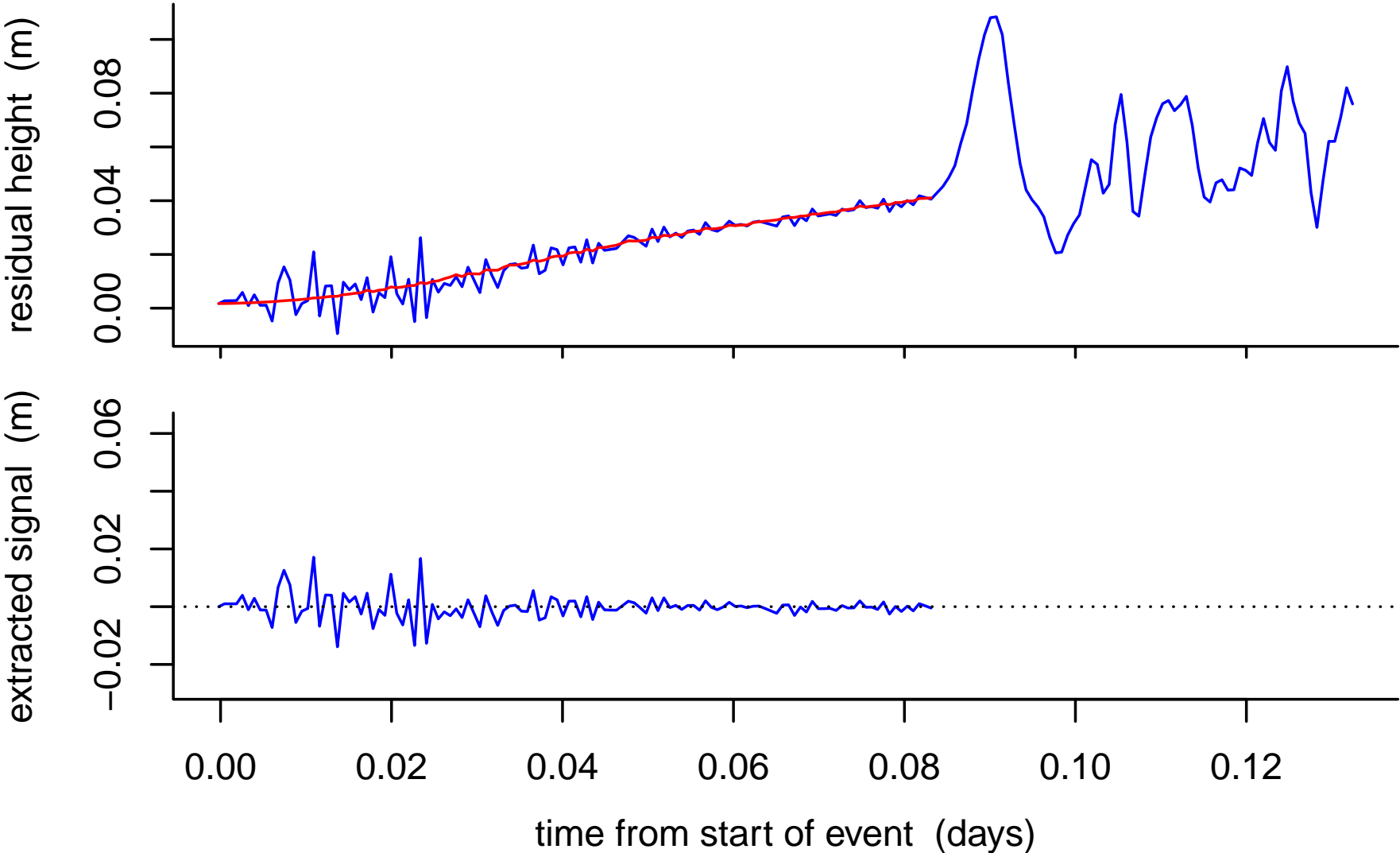
# Kalman Filtering/Smoothing (Local Level Model)



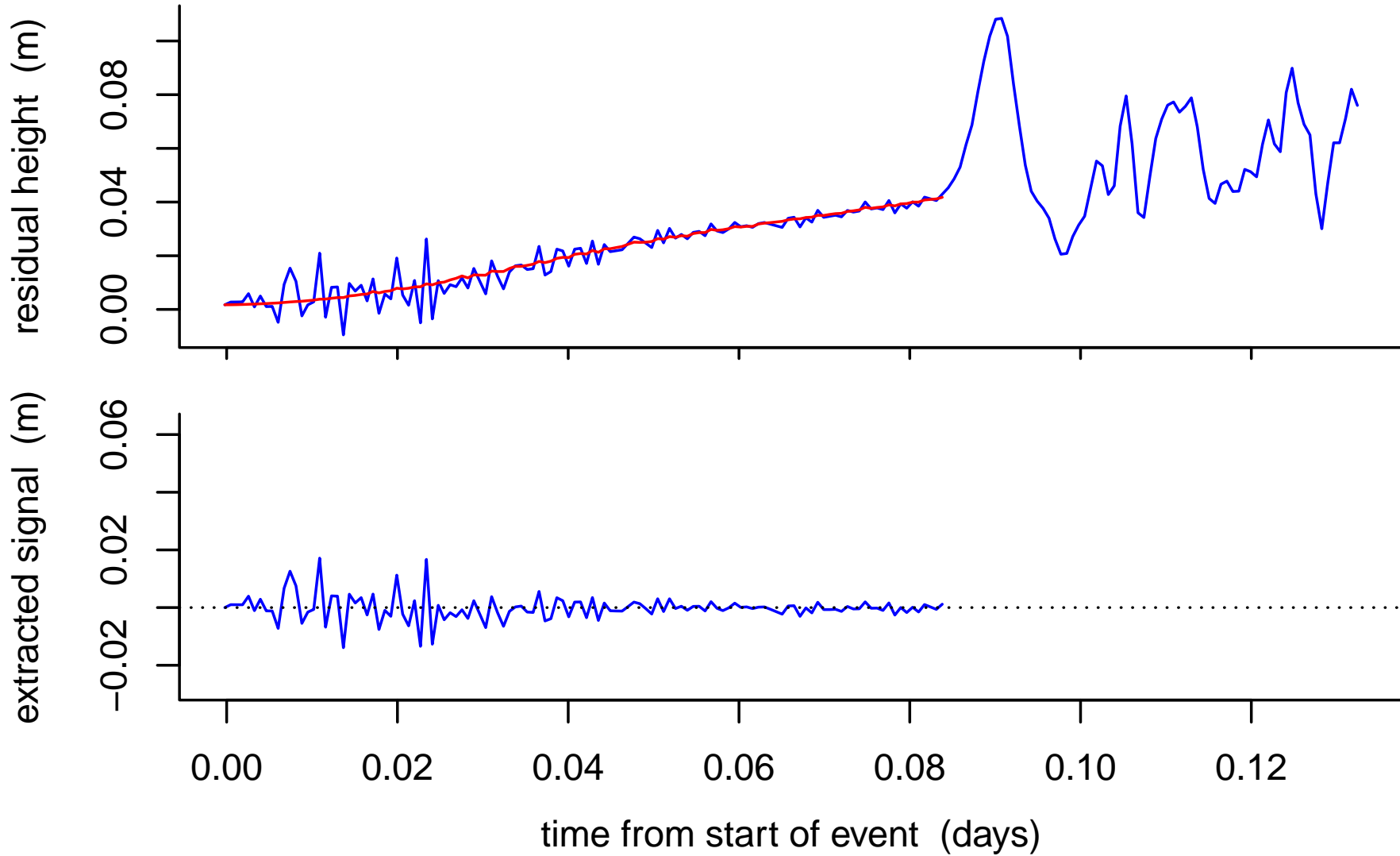
# Kalman Filtering/Smoothing (Local Level Model)



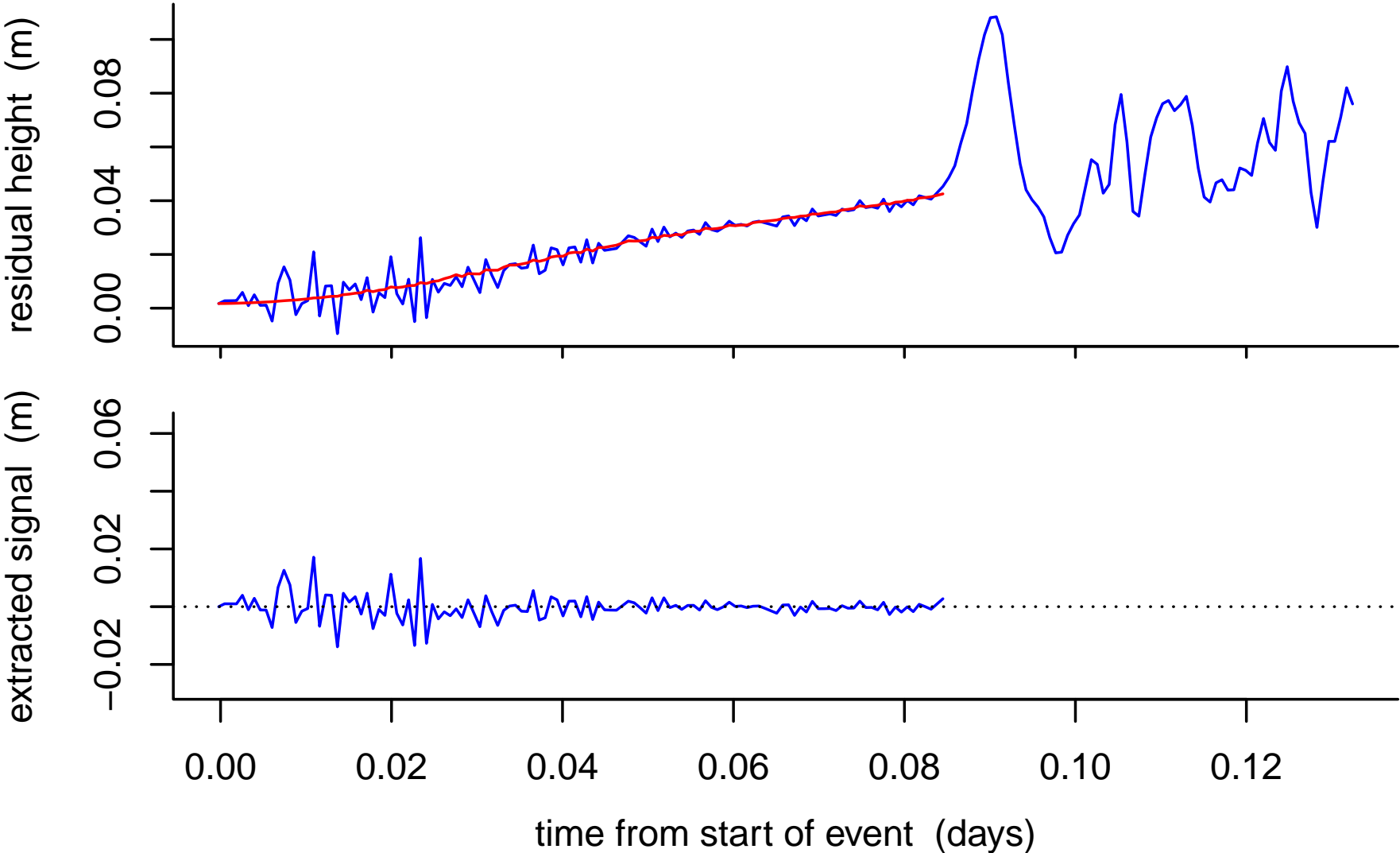
# Kalman Filtering/Smoothing (Local Level Model)



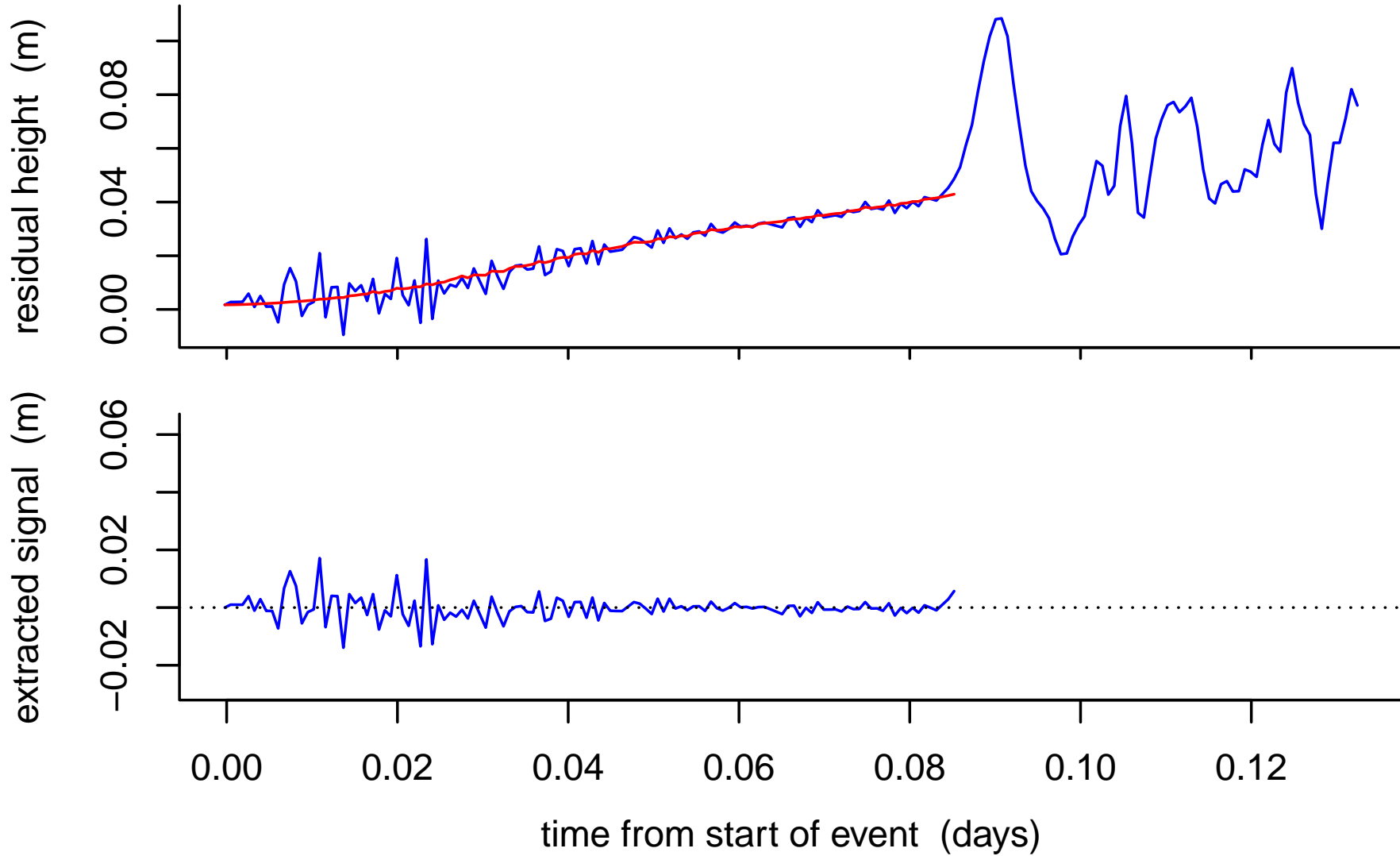
# Kalman Filtering/Smoothing (Local Level Model)



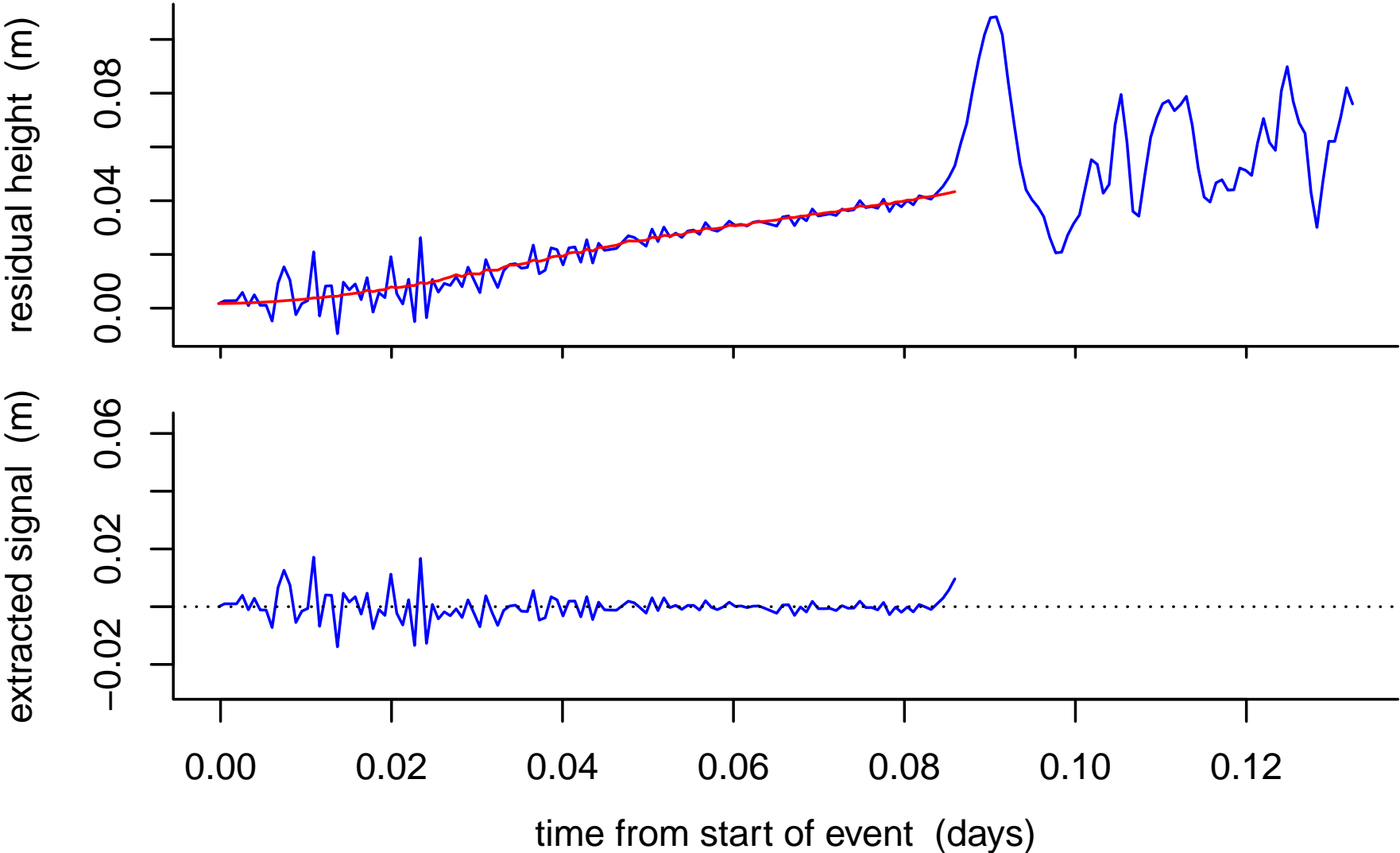
# Kalman Filtering/Smoothing (Local Level Model)



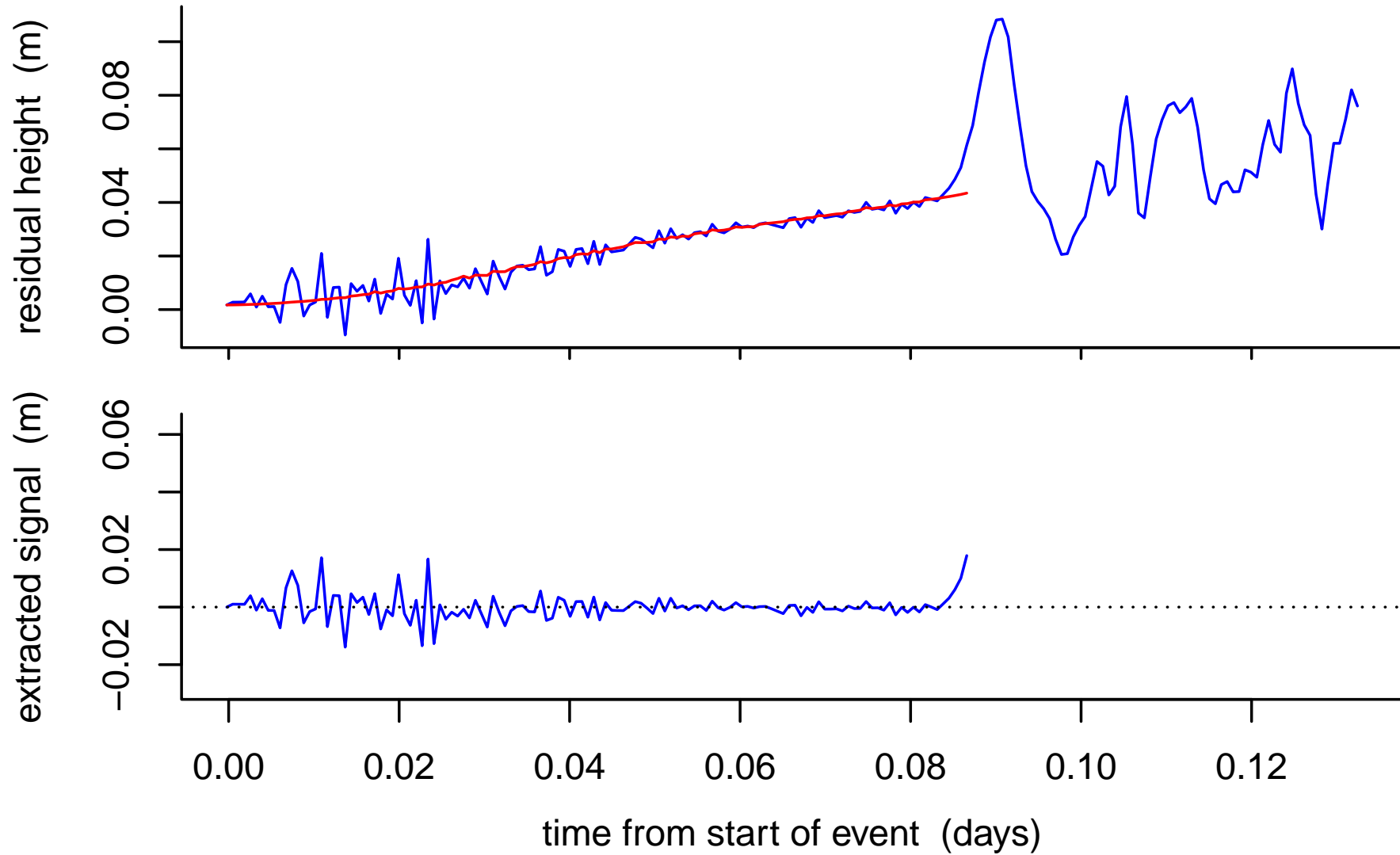
# Kalman Filtering/Smoothing (Local Level Model)



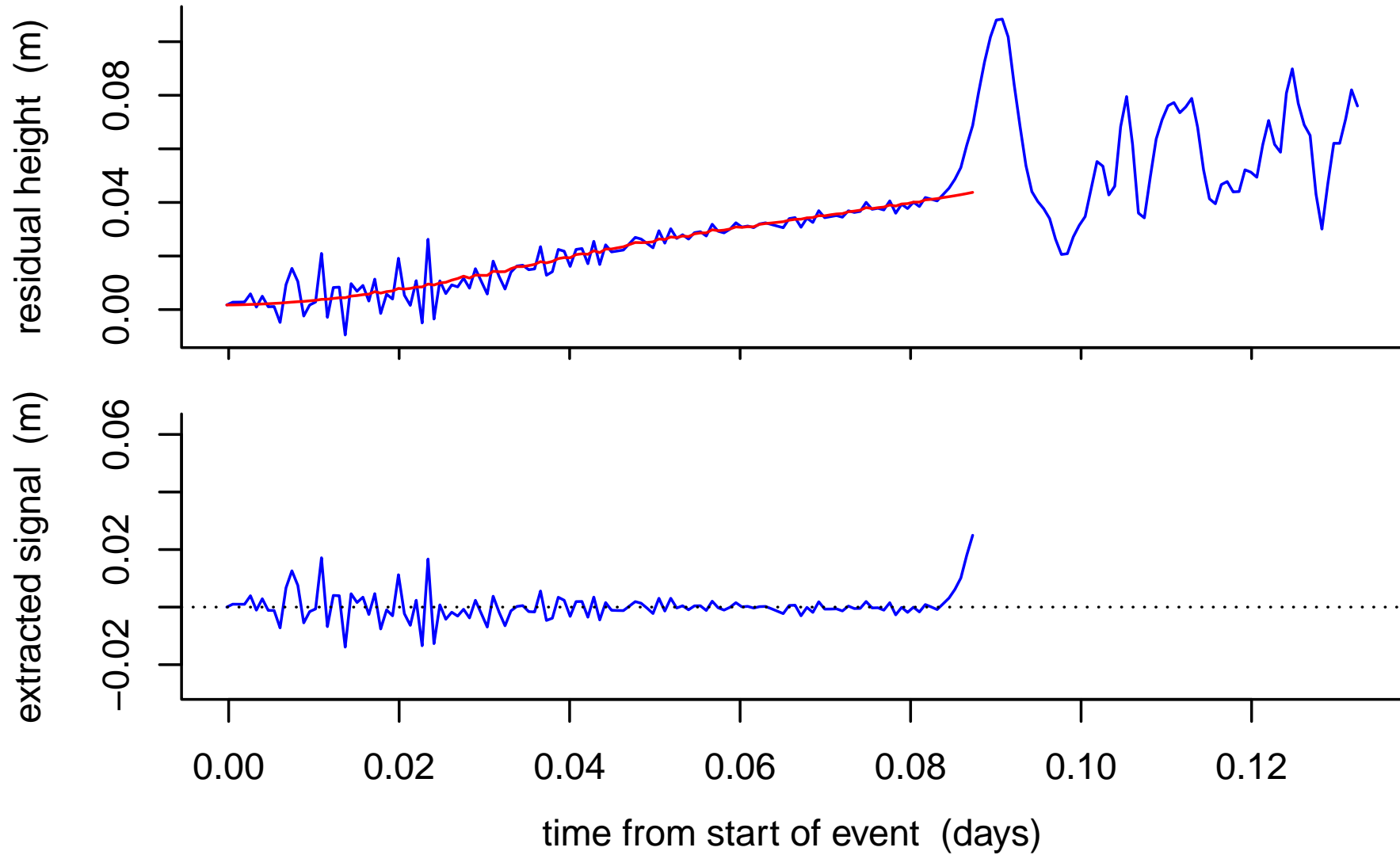
# Kalman Filtering/Smoothing (Local Level Model)



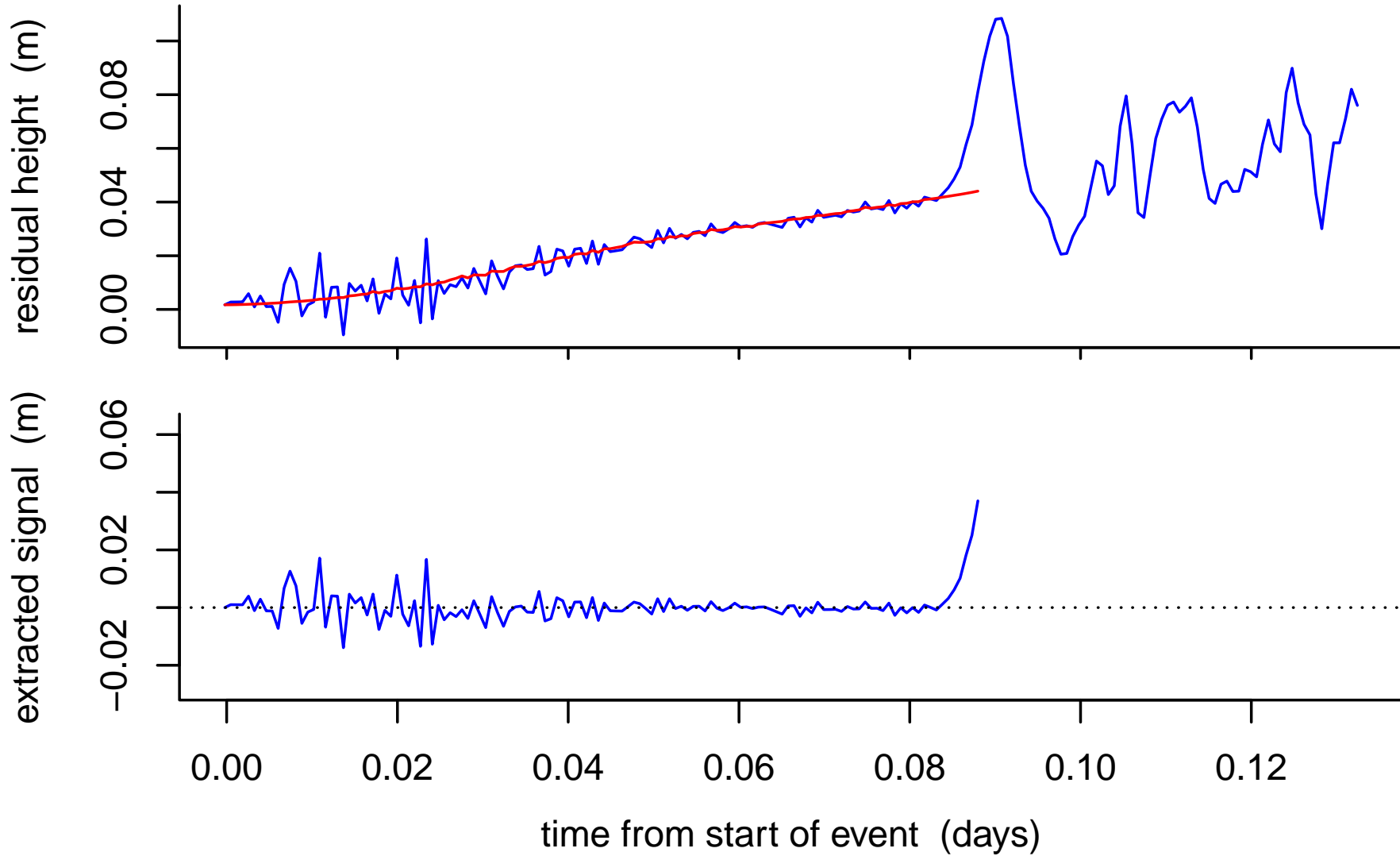
# Kalman Filtering/Smoothing (Local Level Model)



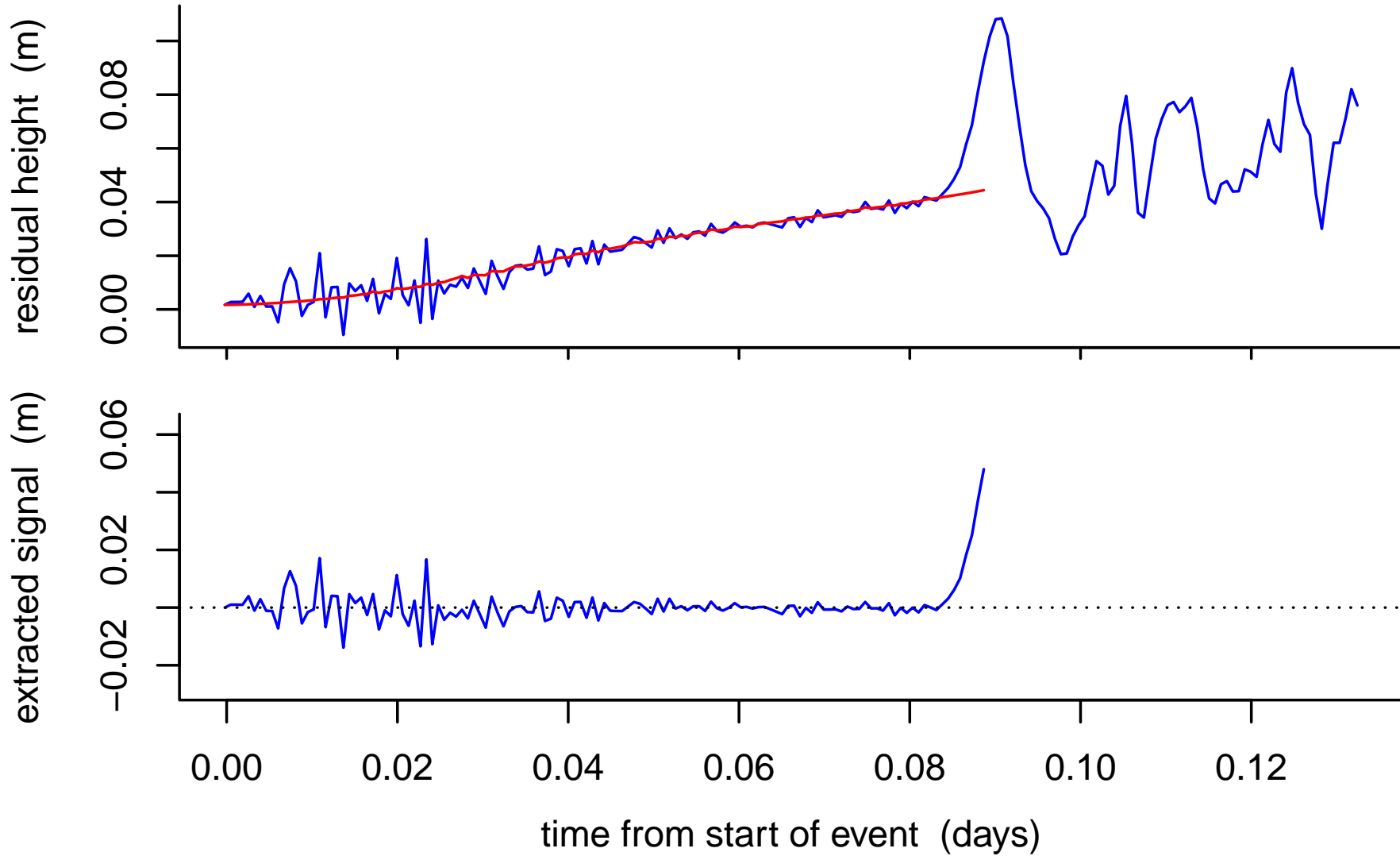
# Kalman Filtering/Smoothing (Local Level Model)



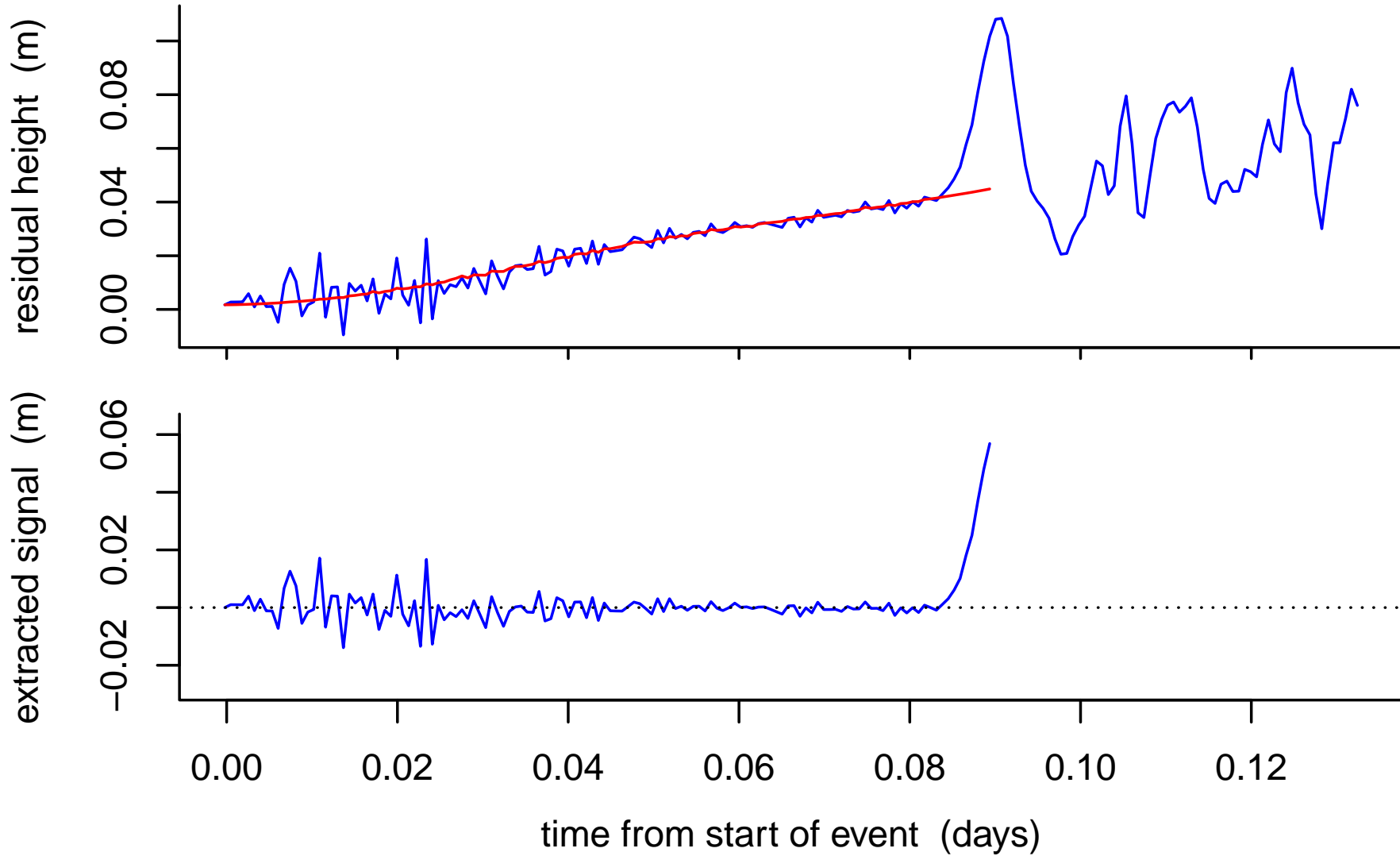
# Kalman Filtering/Smoothing (Local Level Model)



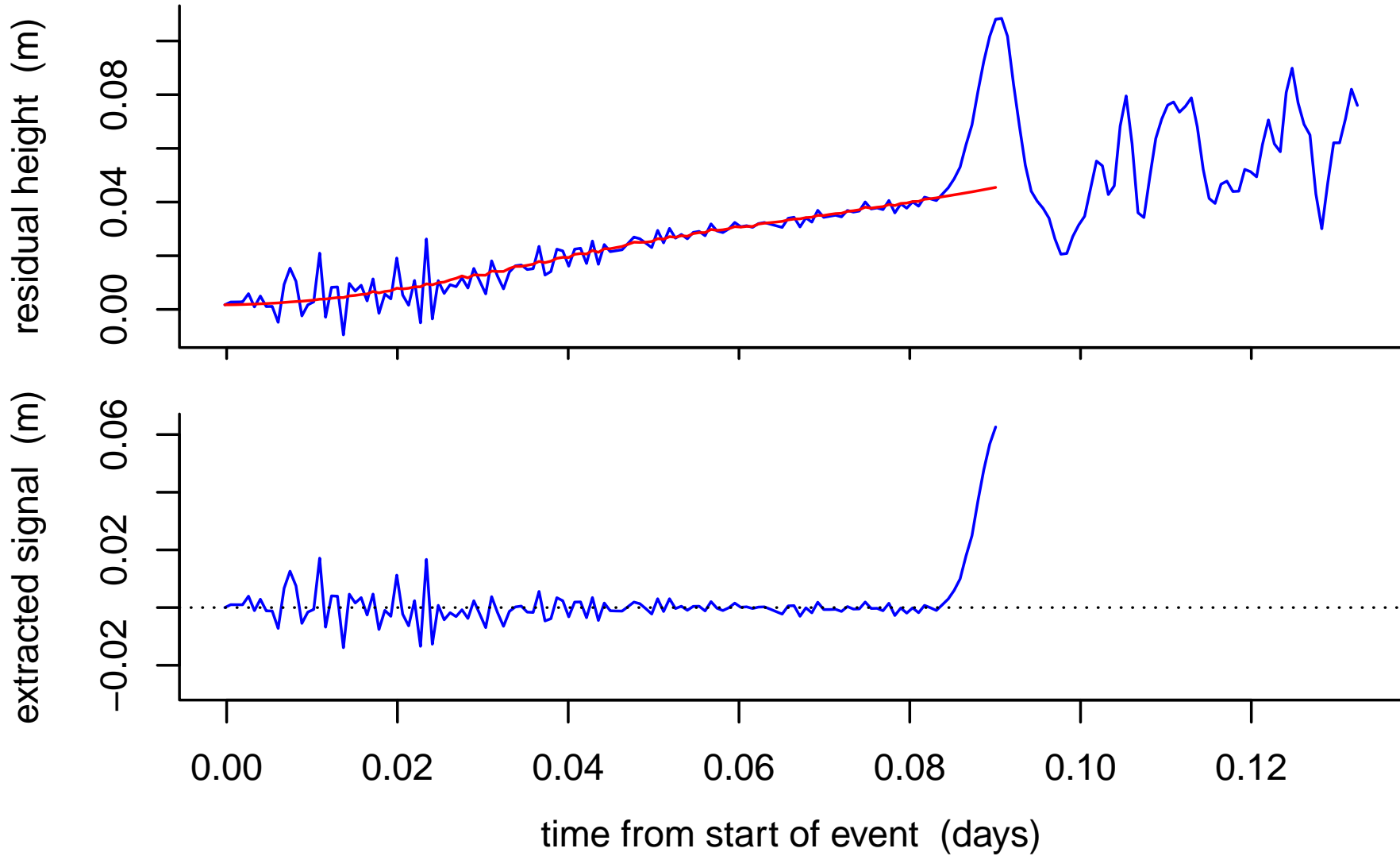
# Kalman Filtering/Smoothing (Local Level Model)



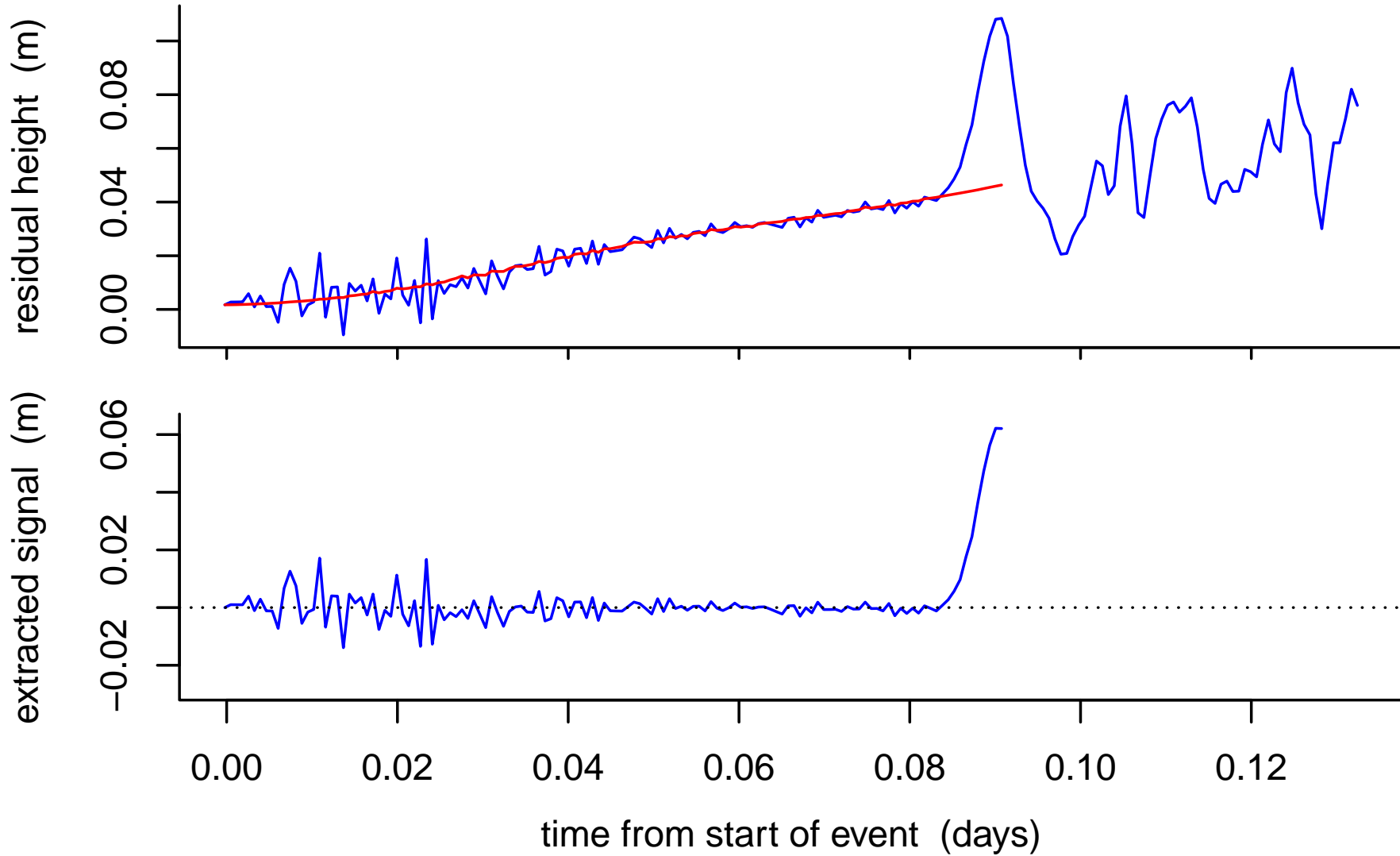
# Kalman Filtering/Smoothing (Local Level Model)



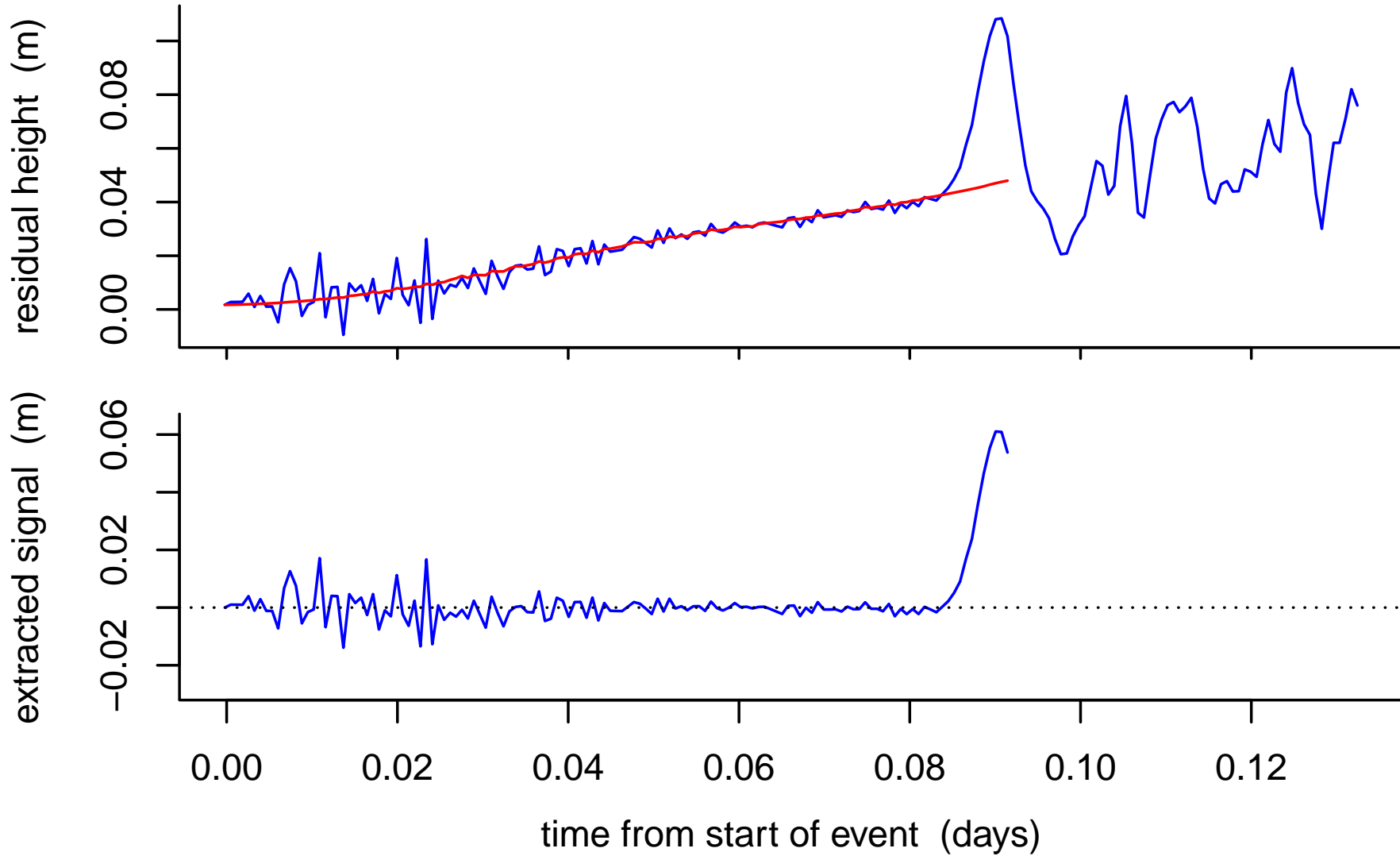
# Kalman Filtering/Smoothing (Local Level Model)



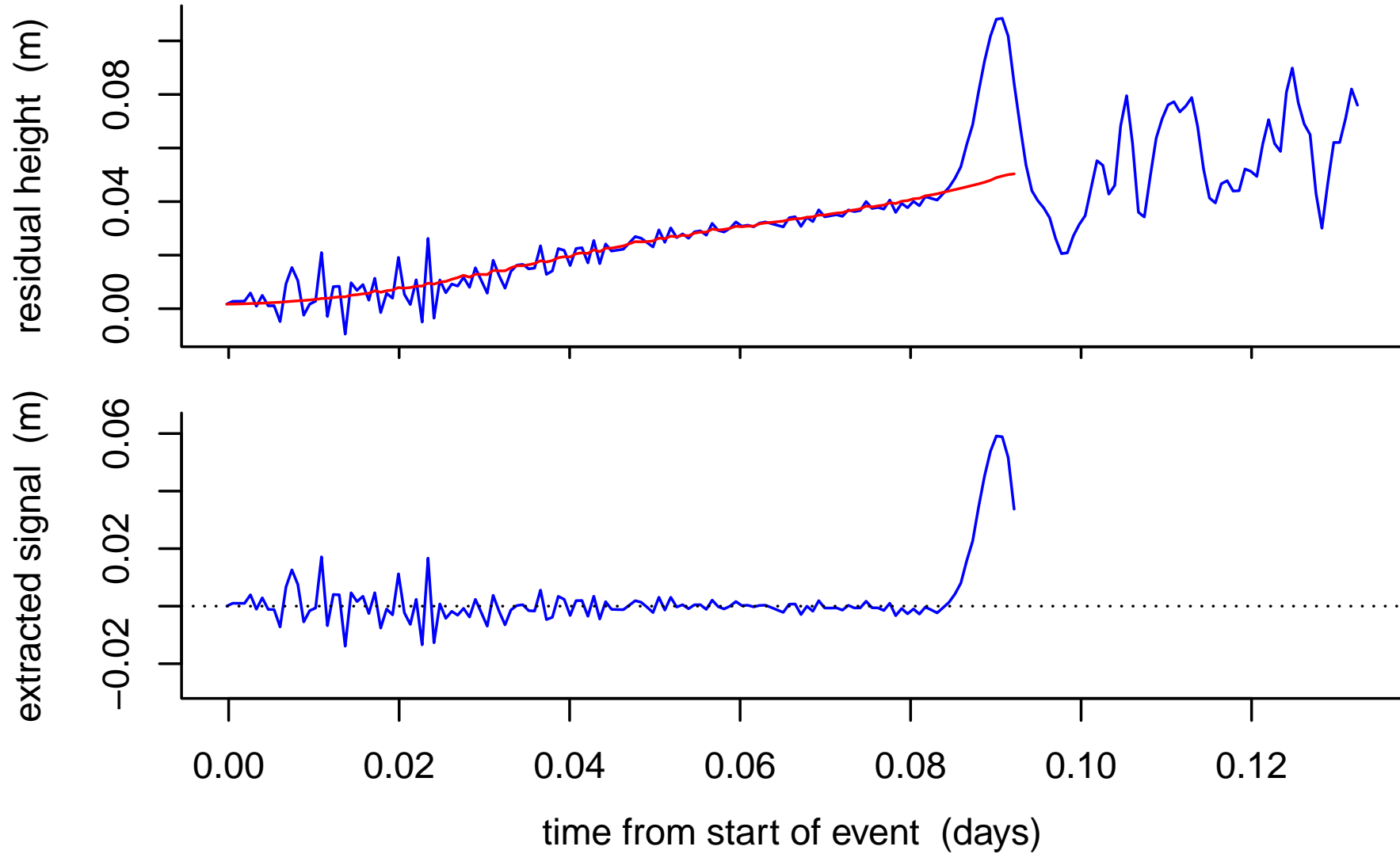
# Kalman Filtering/Smoothing (Local Level Model)



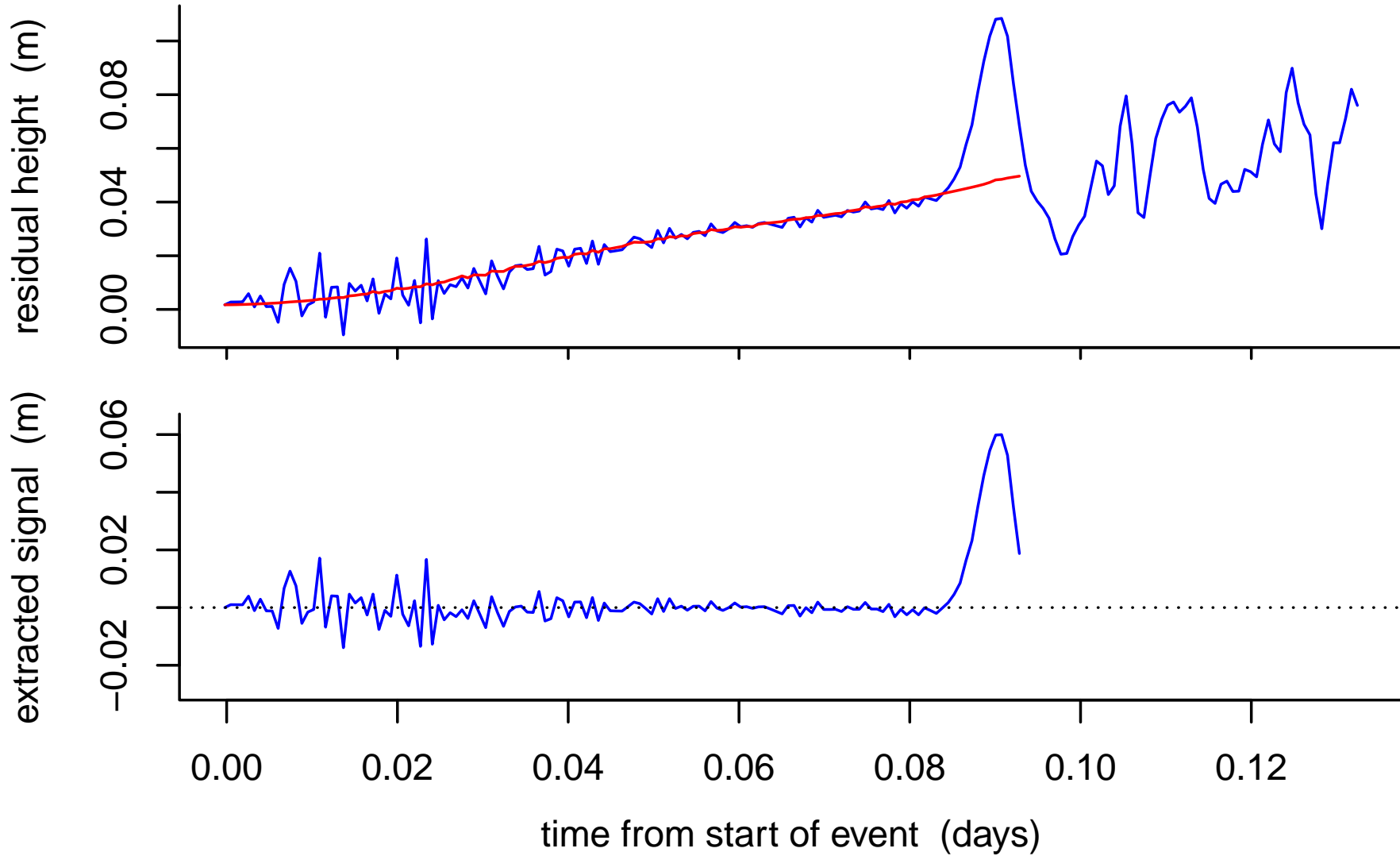
# Kalman Filtering/Smoothing (Local Level Model)



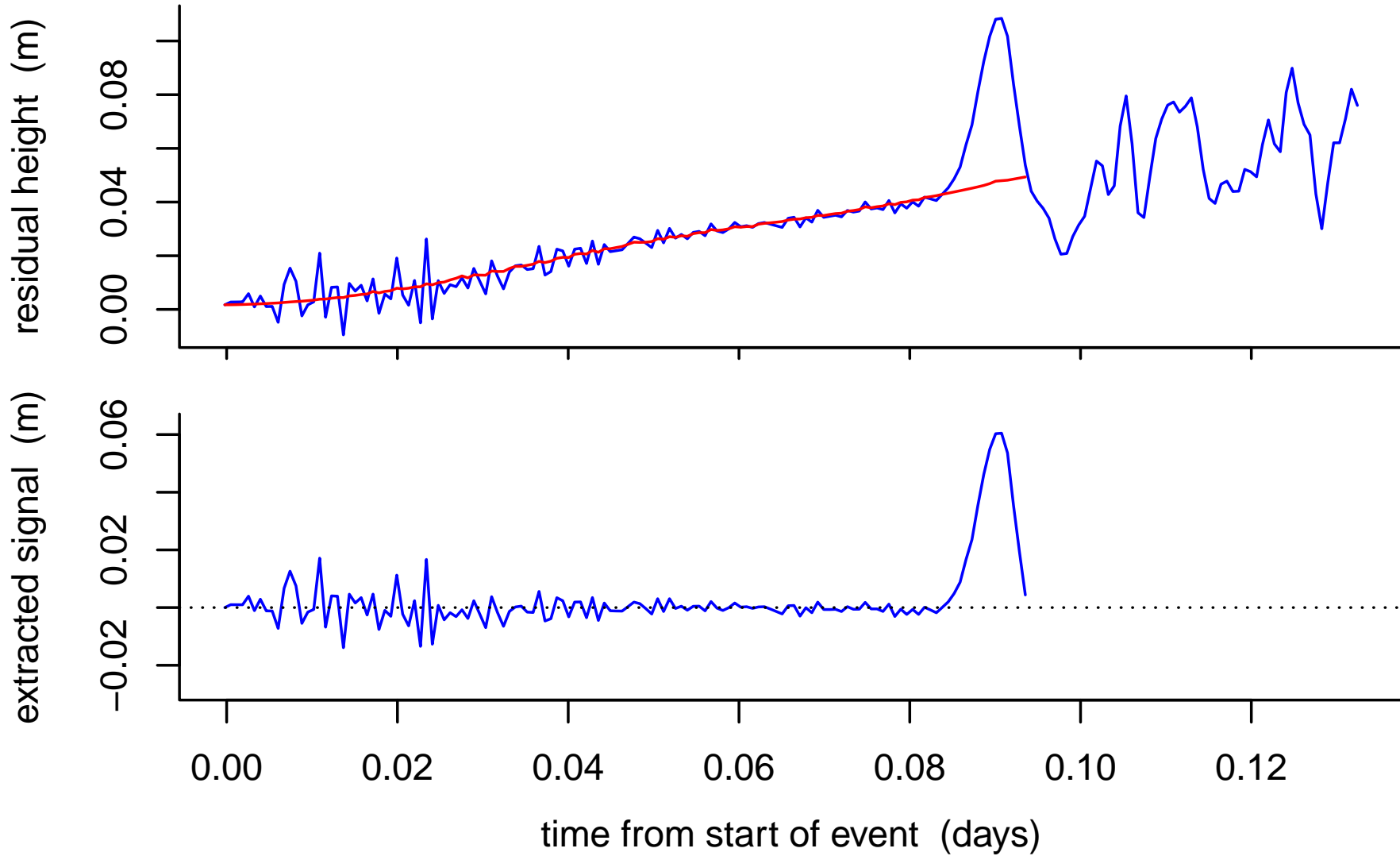
# Kalman Filtering/Smoothing (Local Level Model)



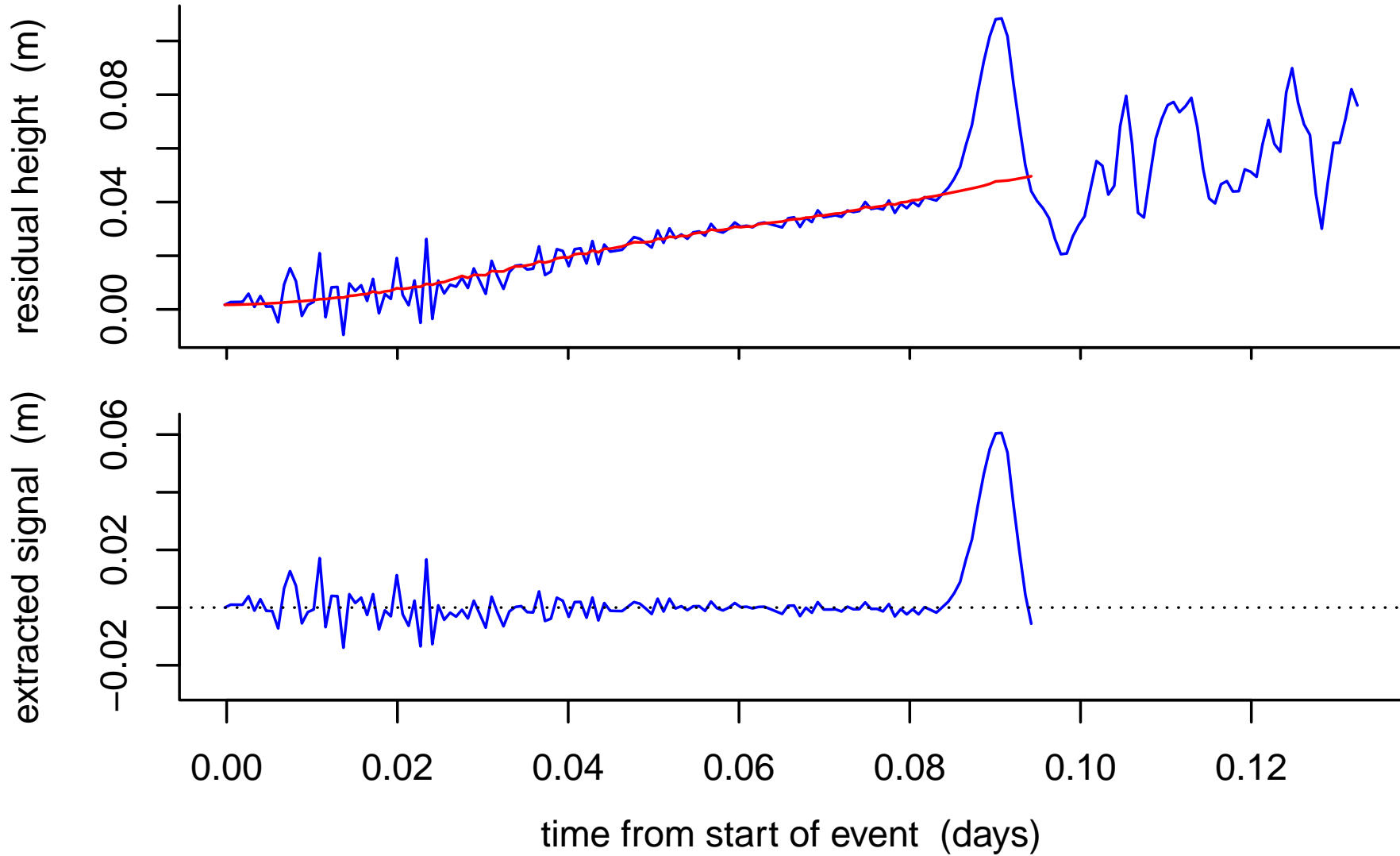
# Kalman Filtering/Smoothing (Local Level Model)



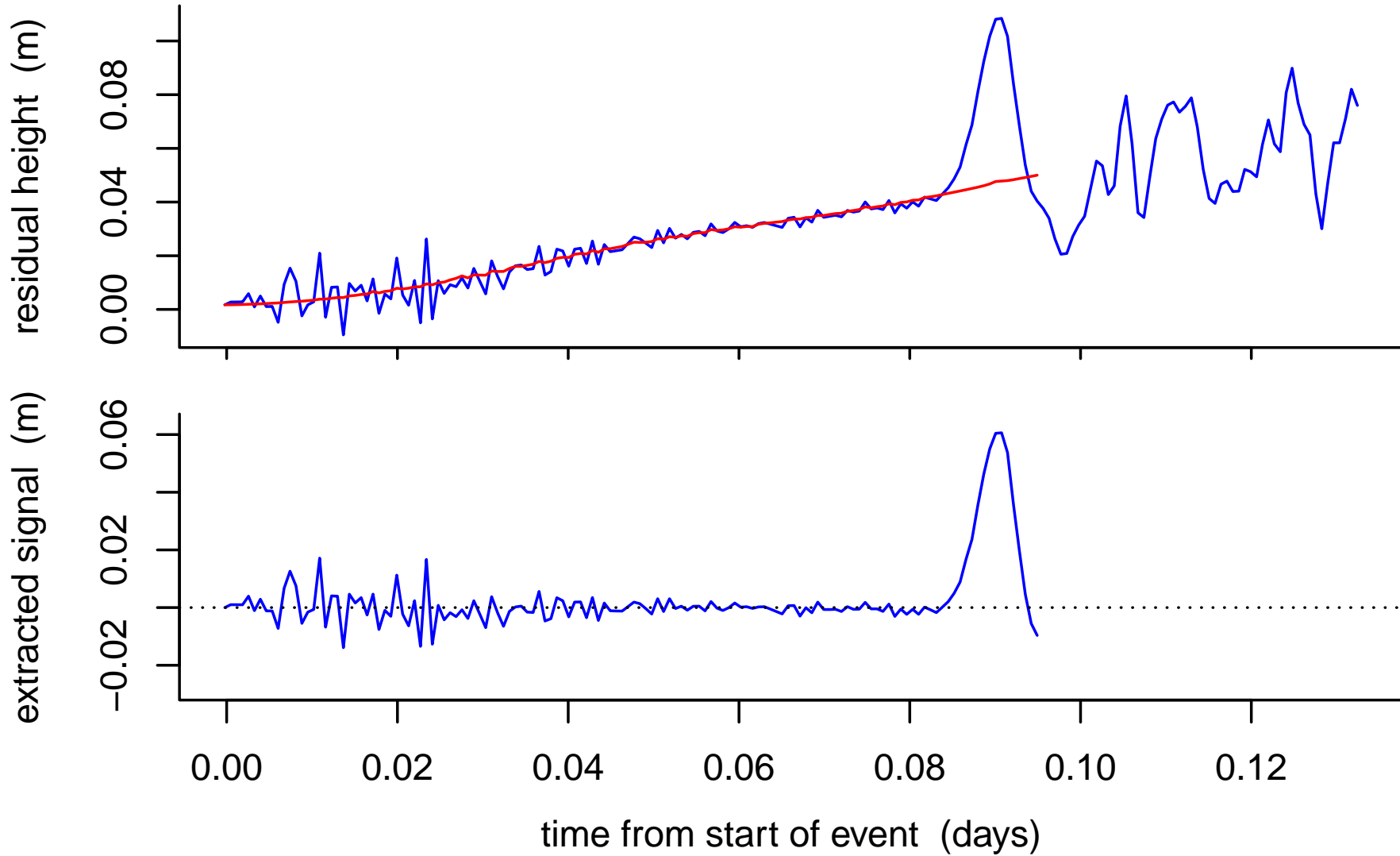
# Kalman Filtering/Smoothing (Local Level Model)



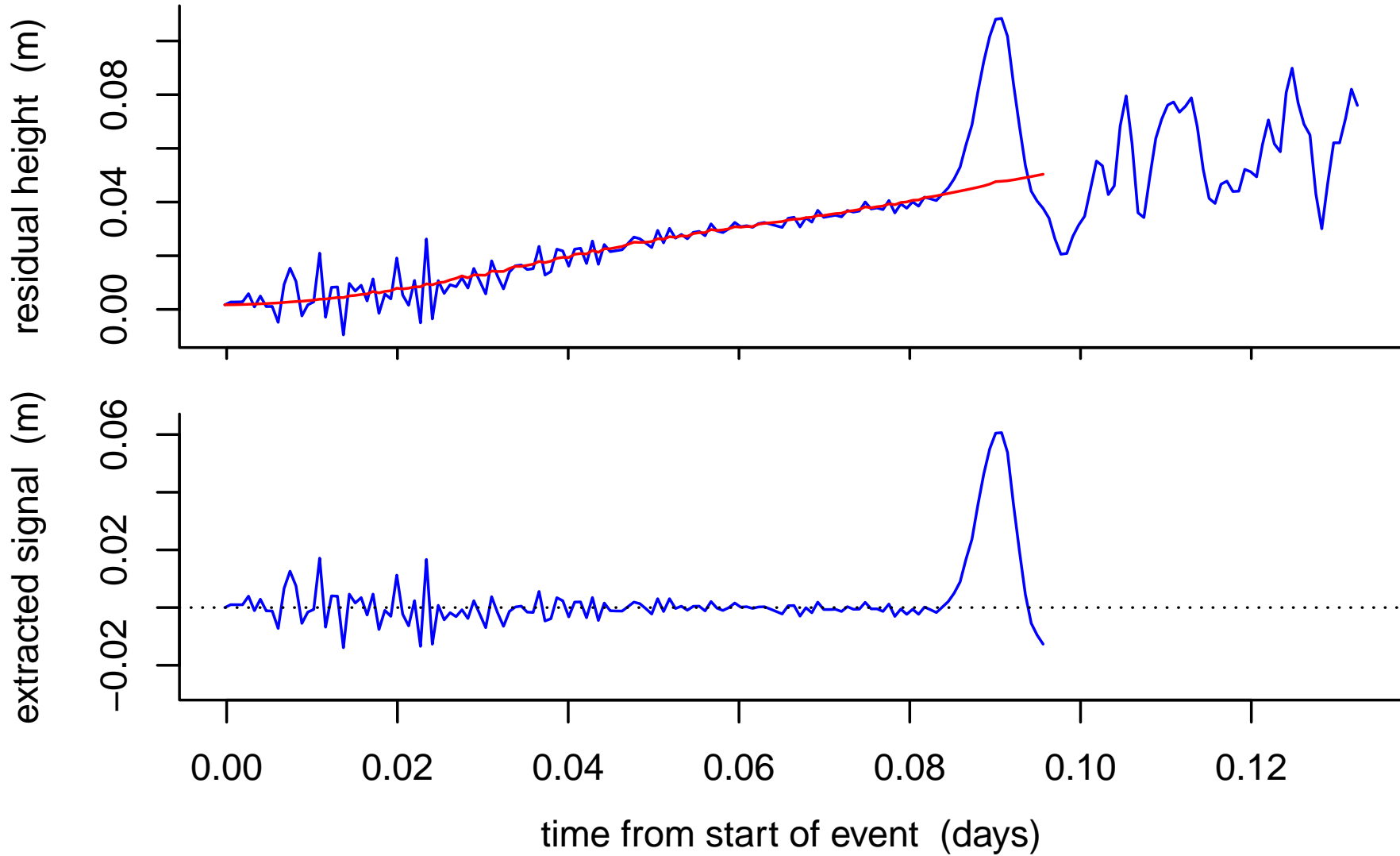
# Kalman Filtering/Smoothing (Local Level Model)



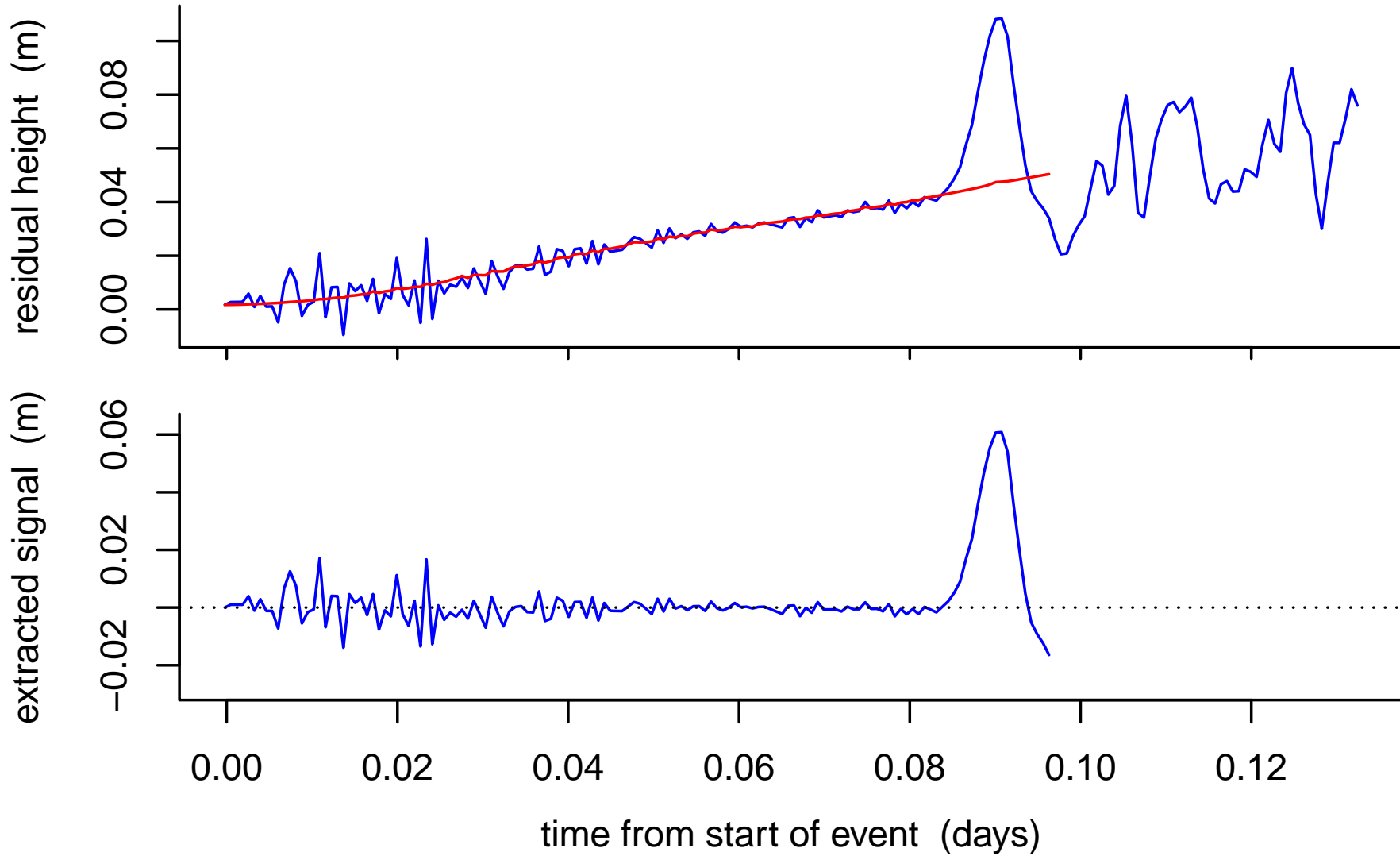
# Kalman Filtering/Smoothing (Local Level Model)



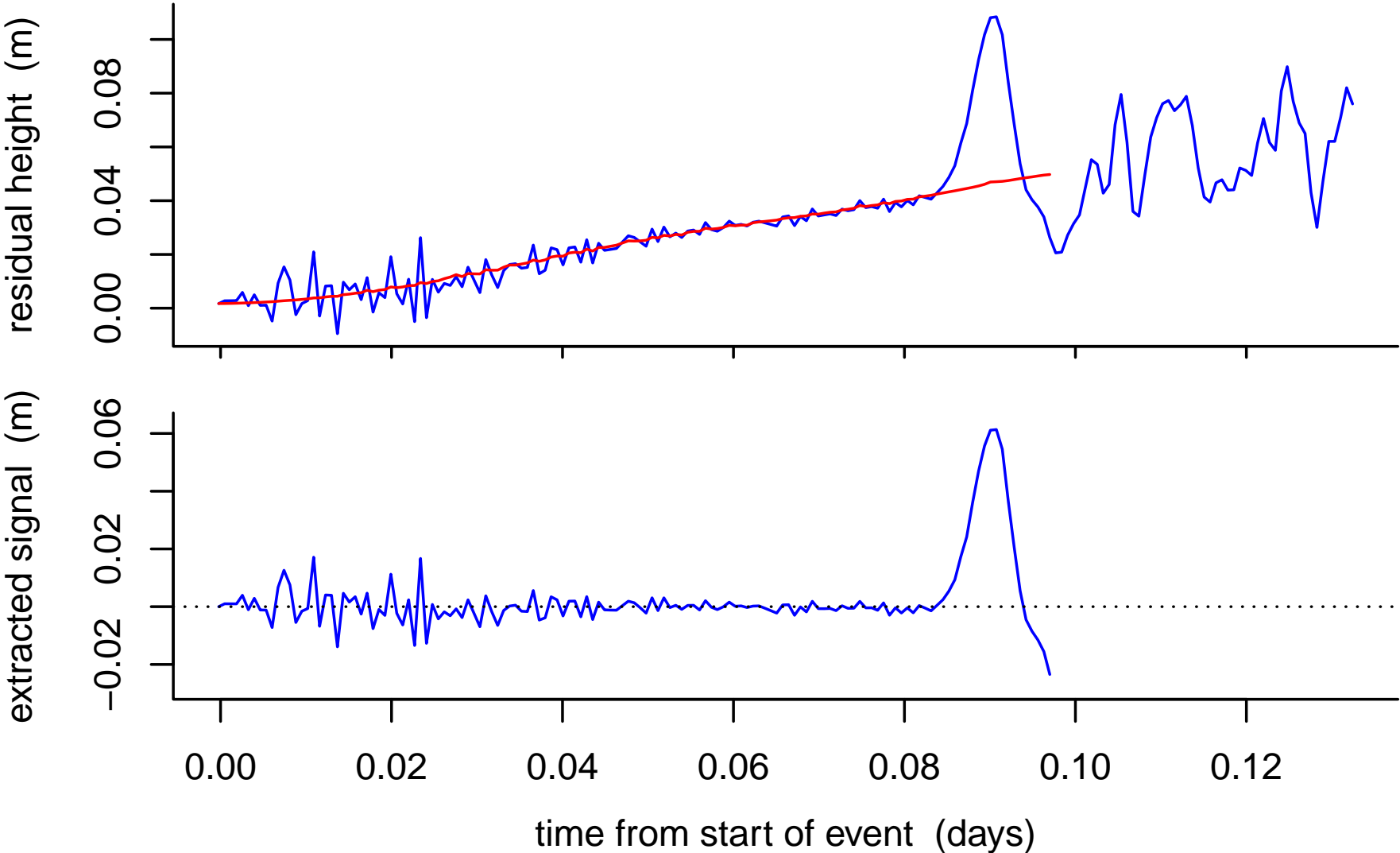
# Kalman Filtering/Smoothing (Local Level Model)



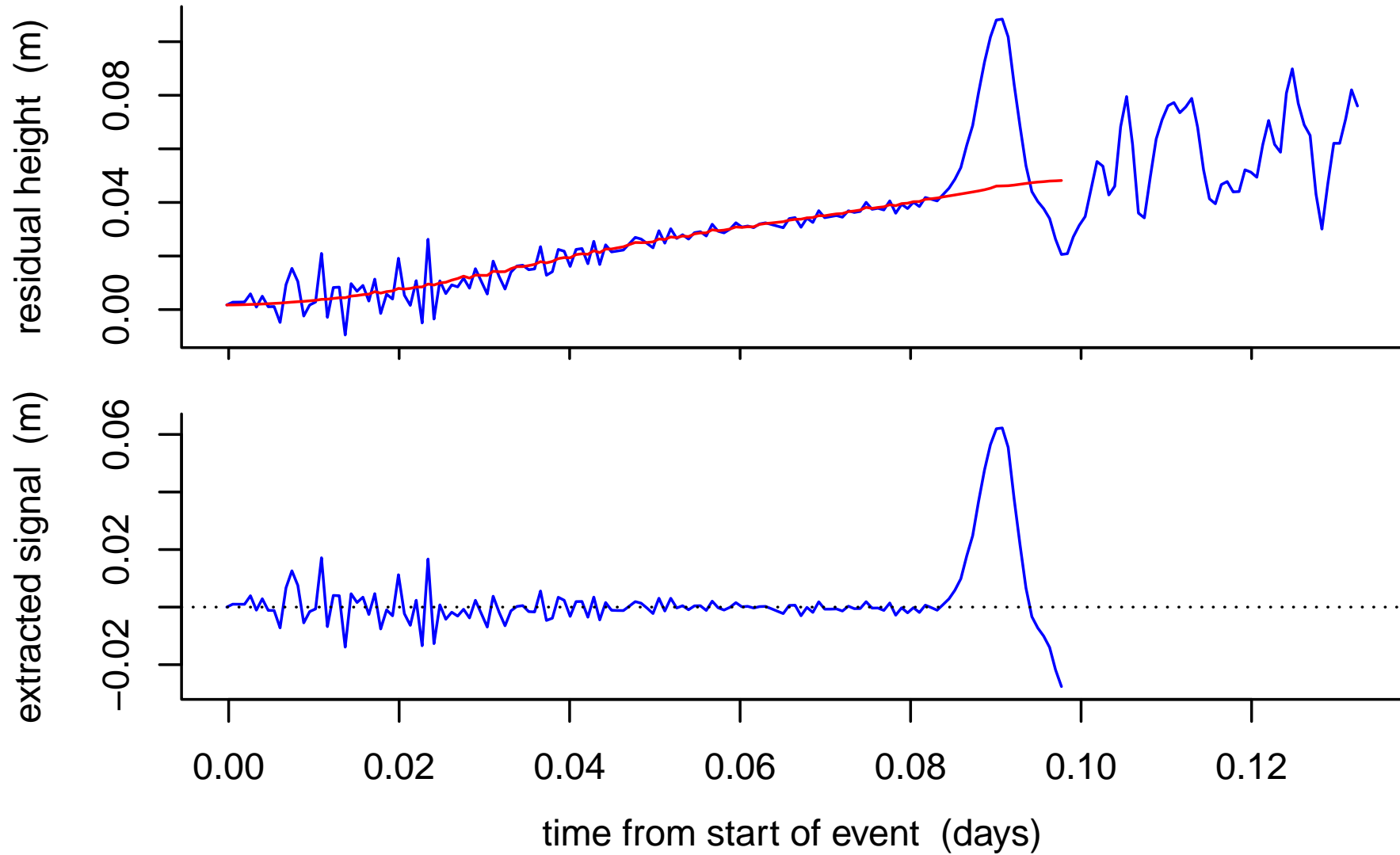
# Kalman Filtering/Smoothing (Local Level Model)



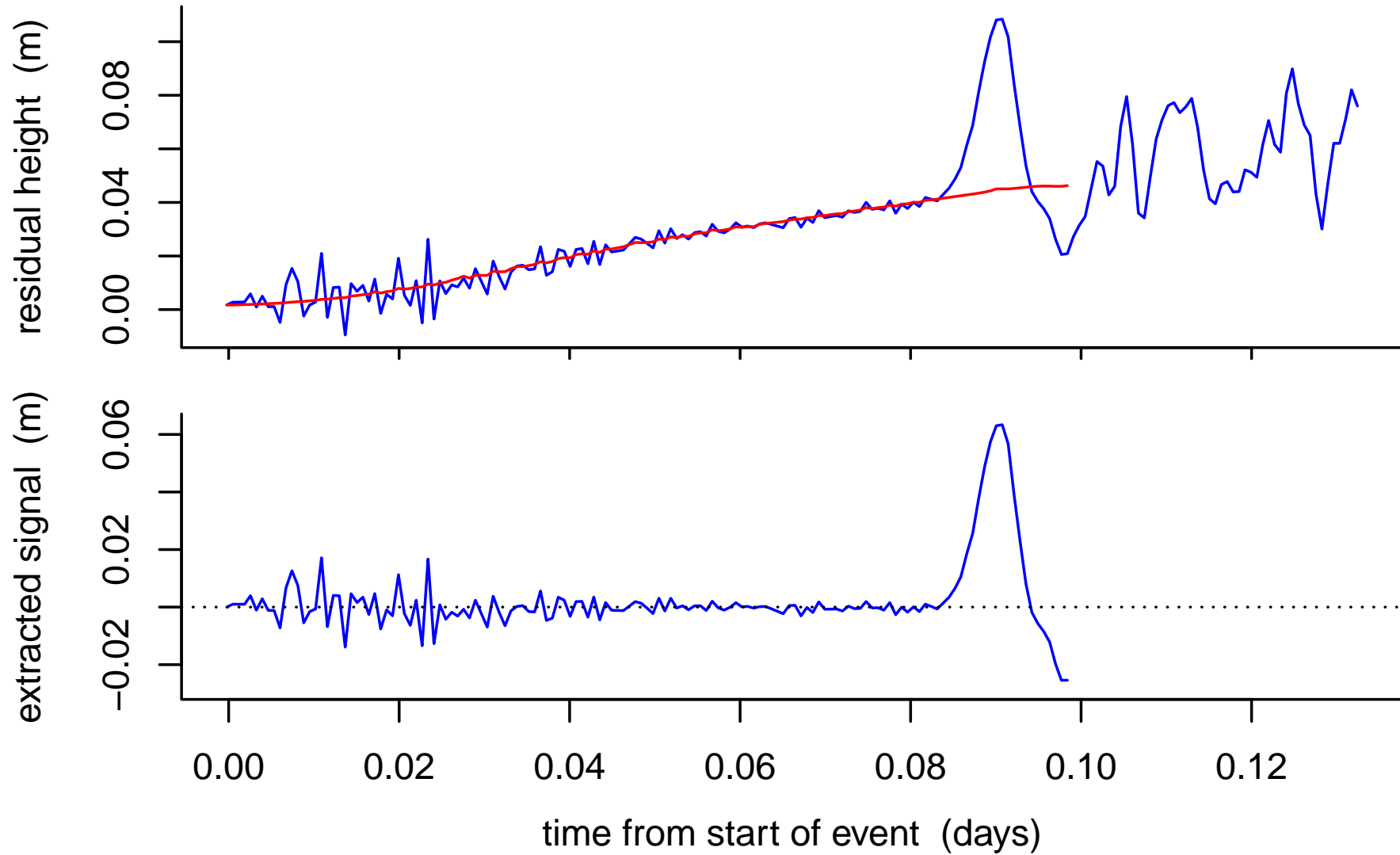
# Kalman Filtering/Smoothing (Local Level Model)



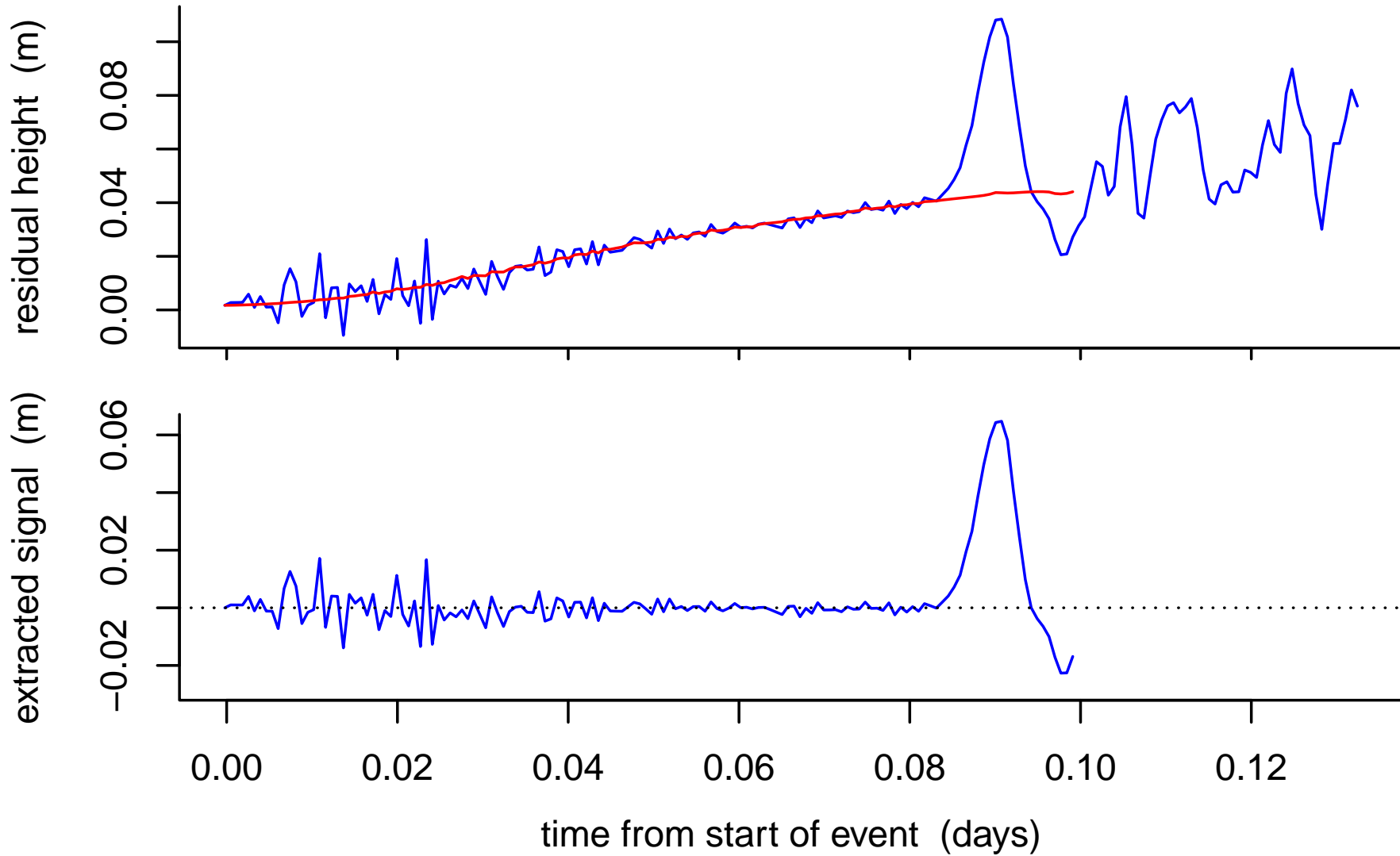
# Kalman Filtering/Smoothing (Local Level Model)



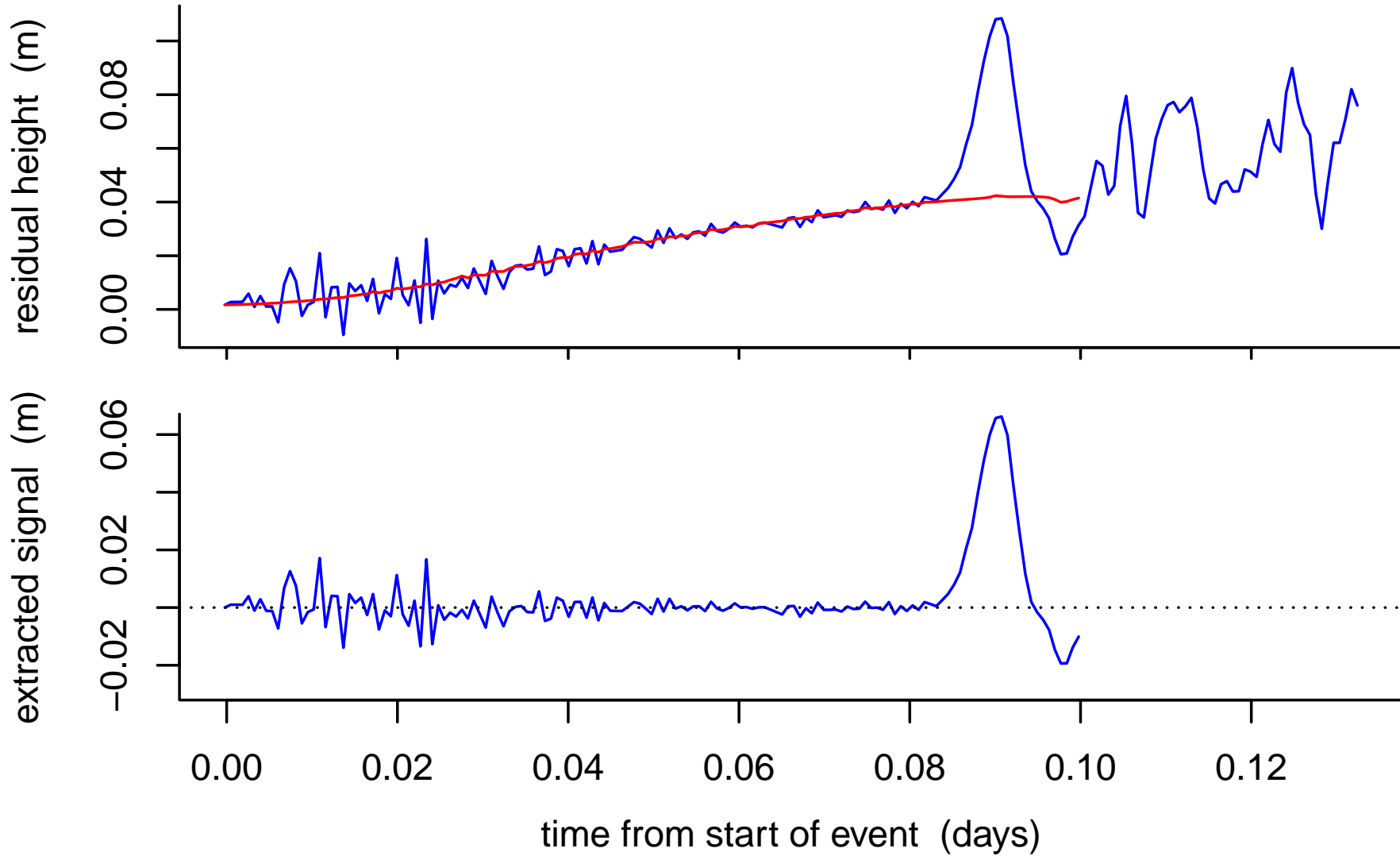
# Kalman Filtering/Smoothing (Local Level Model)



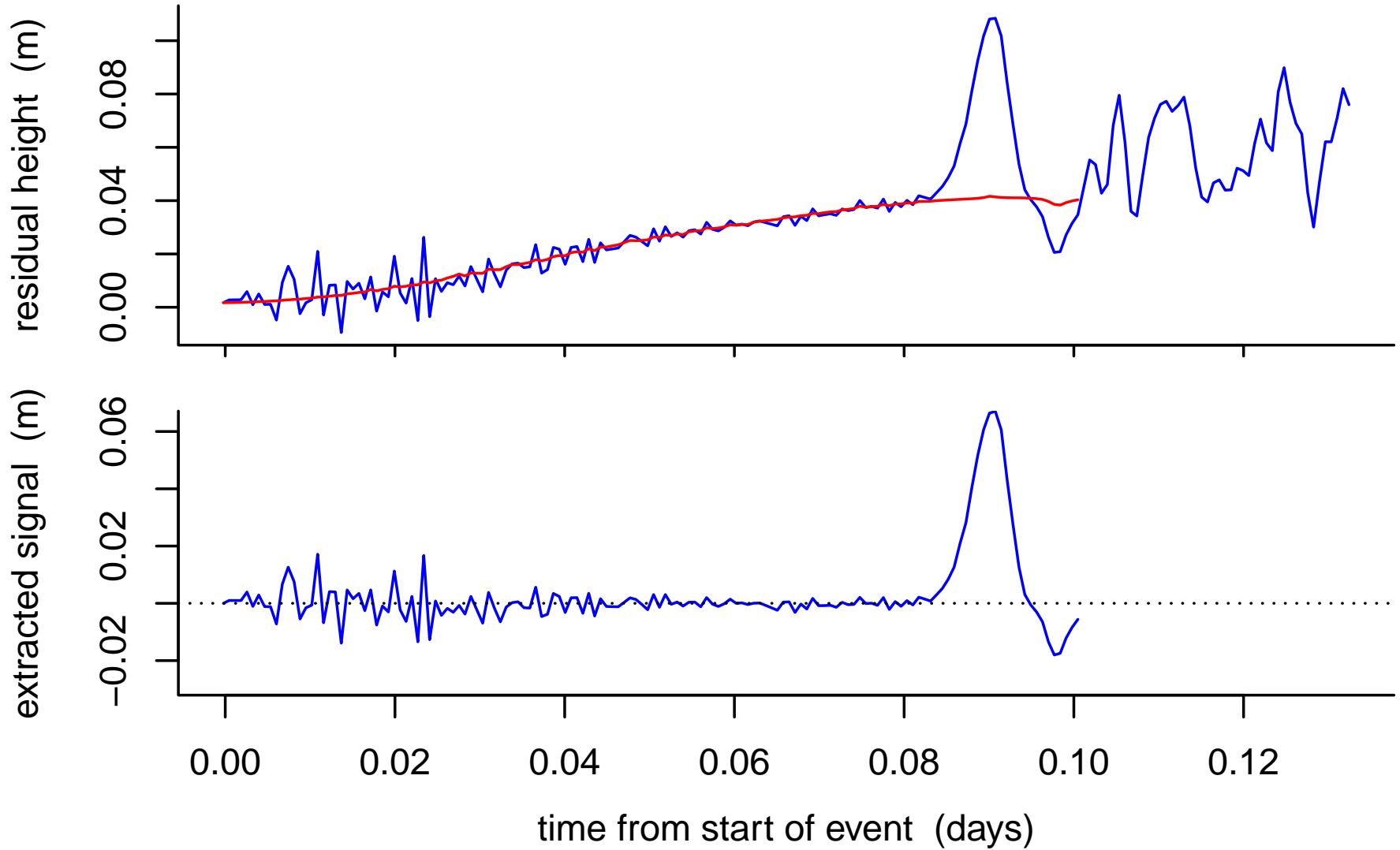
# Kalman Filtering/Smoothing (Local Level Model)



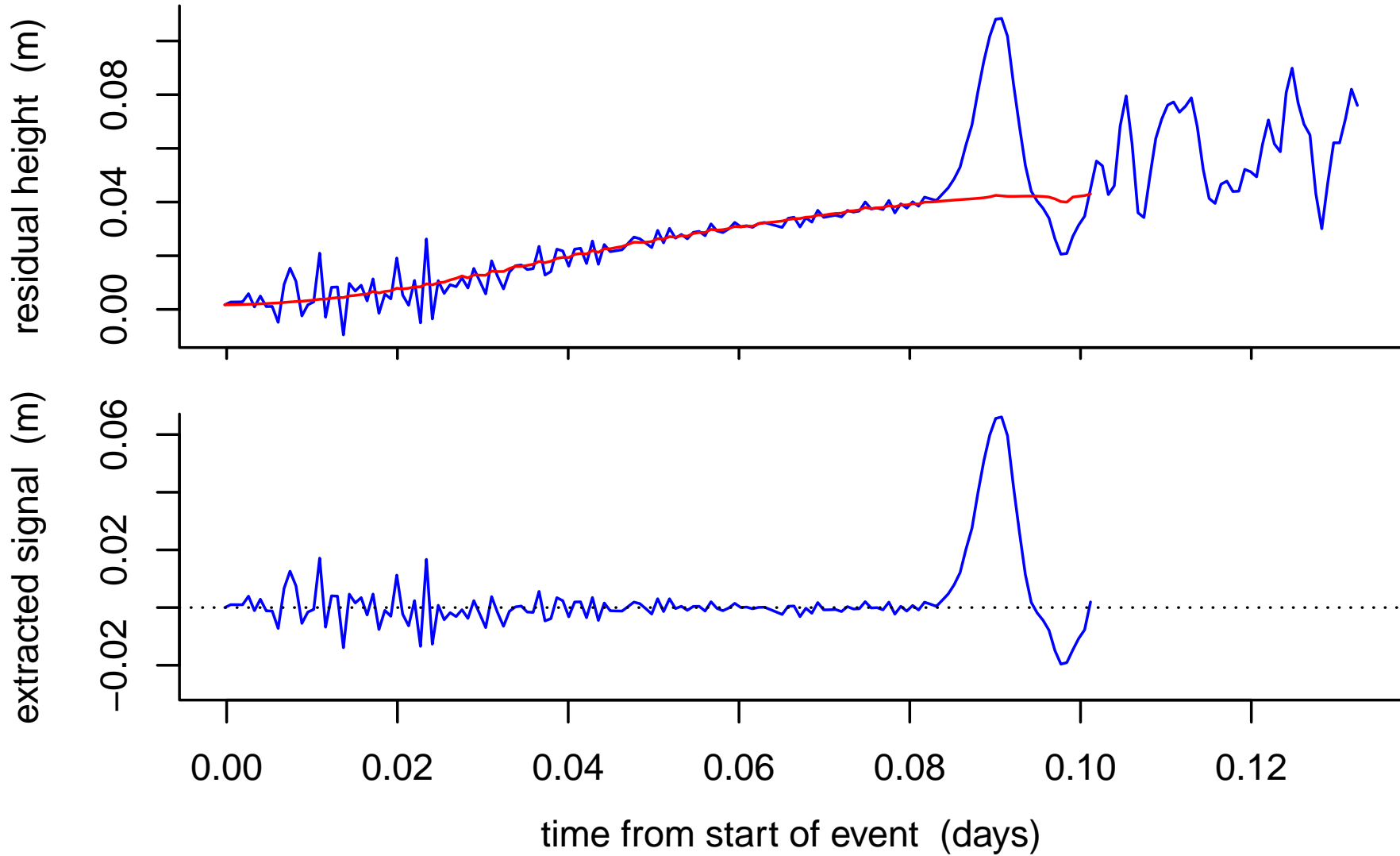
# Kalman Filtering/Smoothing (Local Level Model)



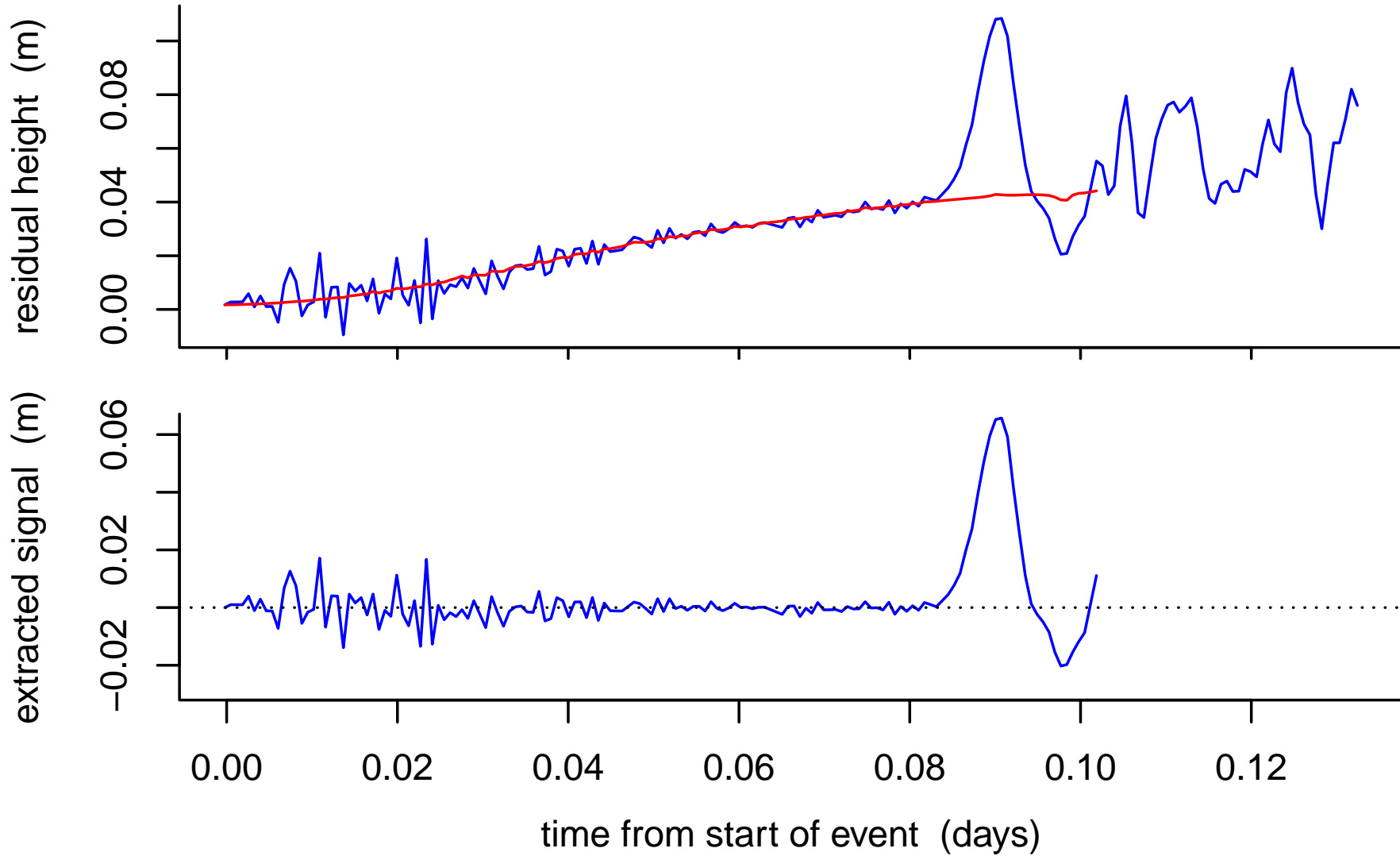
# Kalman Filtering/Smoothing (Local Level Model)



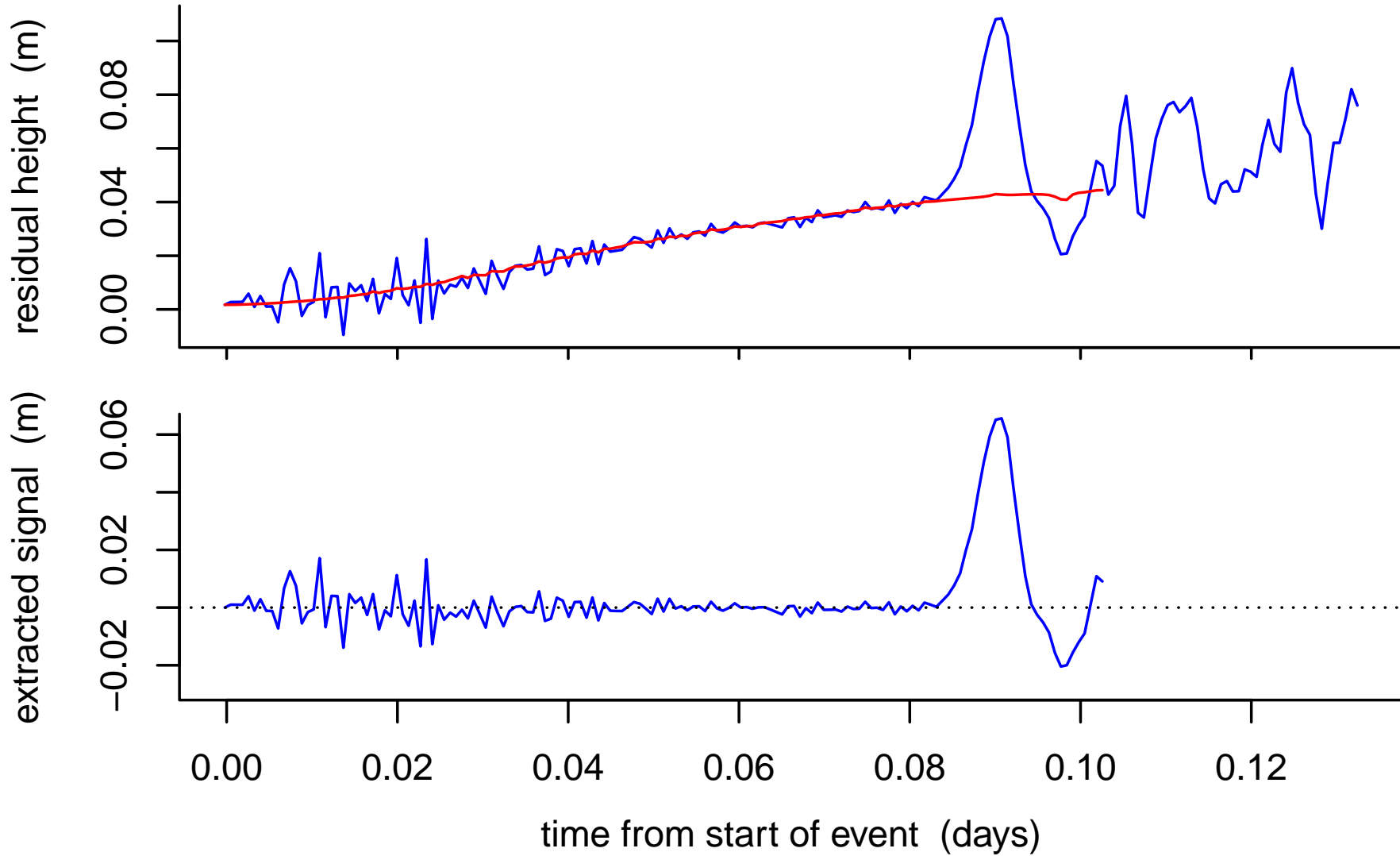
# Kalman Filtering/Smoothing (Local Level Model)



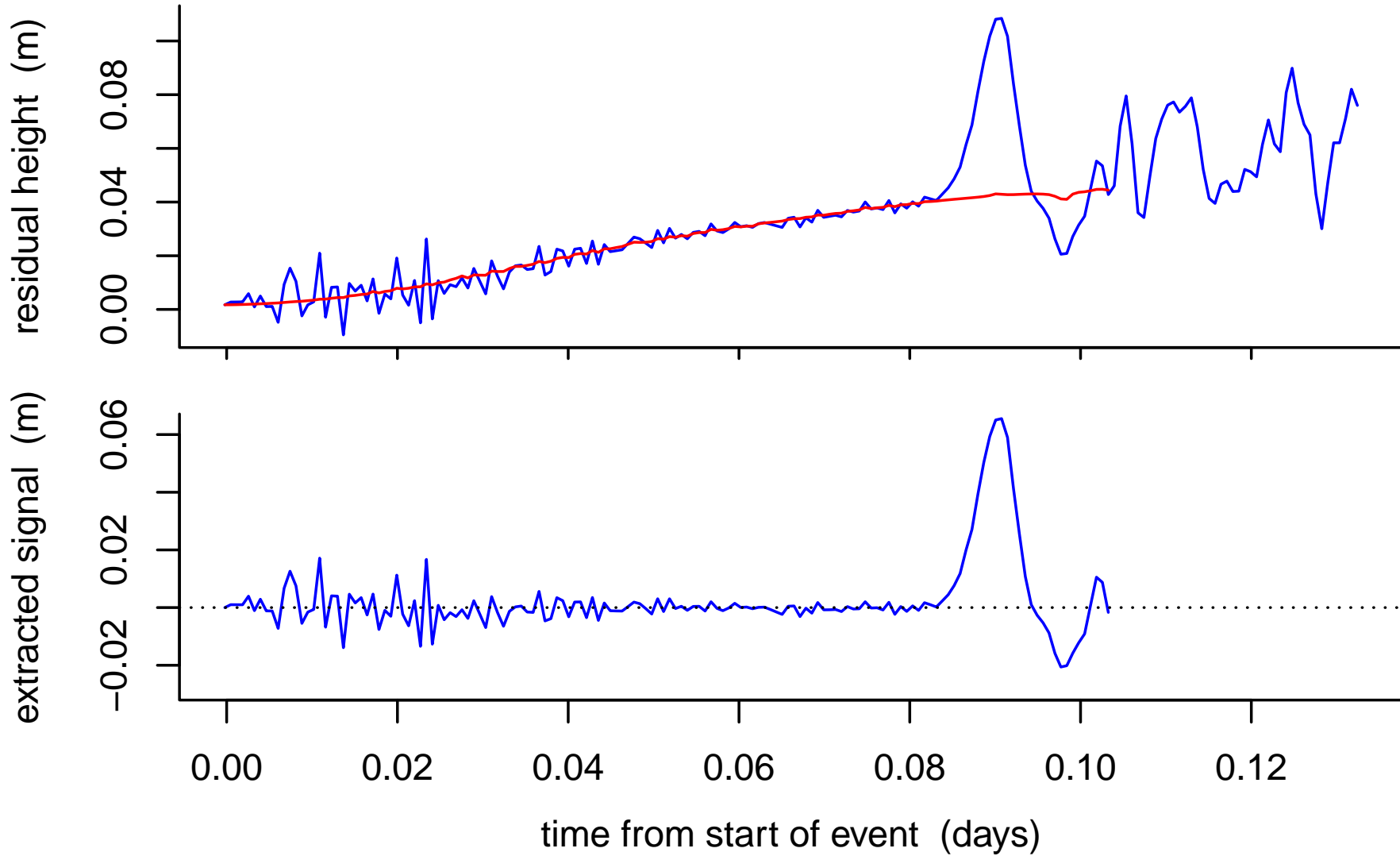
# Kalman Filtering/Smoothing (Local Level Model)



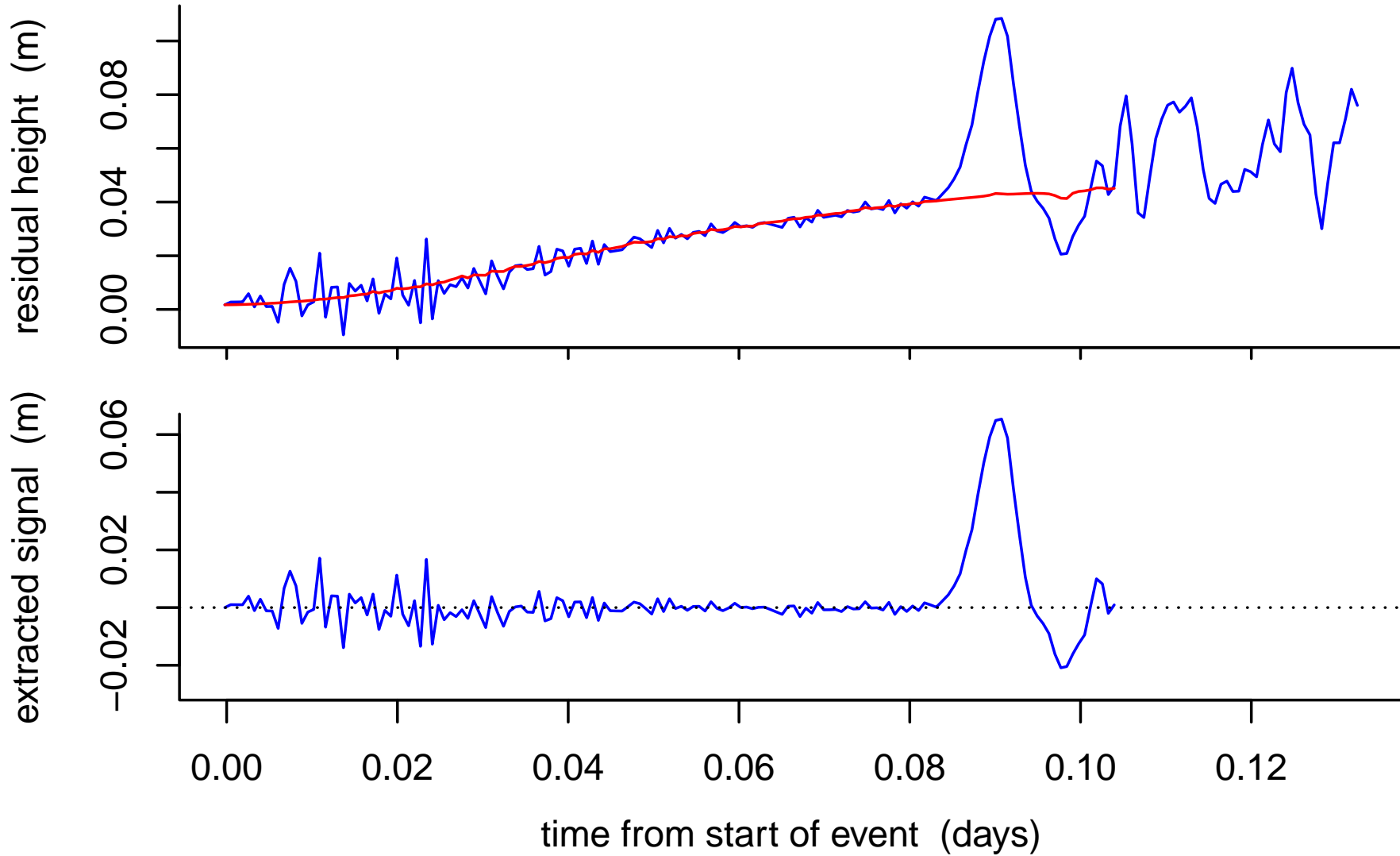
# Kalman Filtering/Smoothing (Local Level Model)



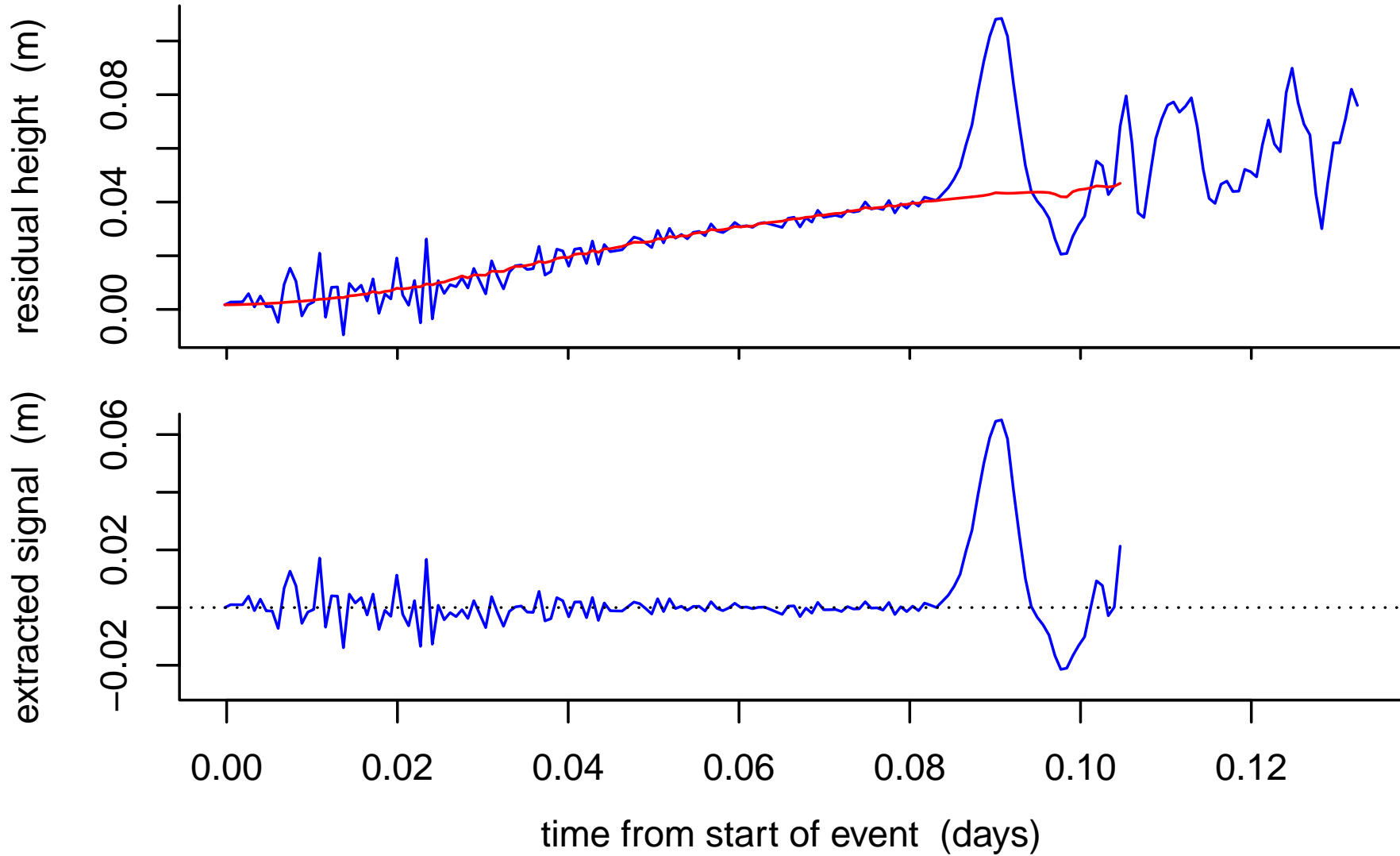
# Kalman Filtering/Smoothing (Local Level Model)



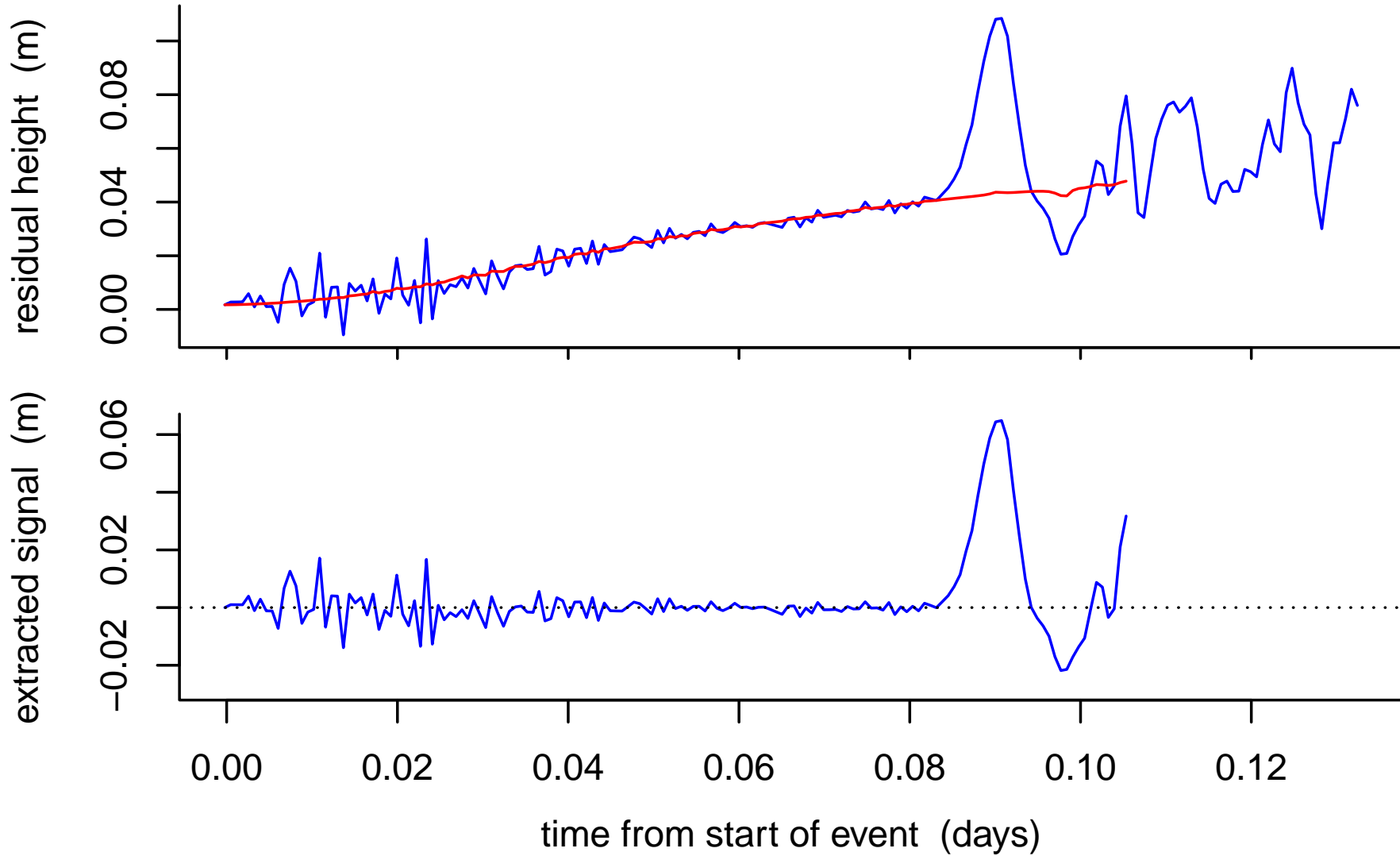
# Kalman Filtering/Smoothing (Local Level Model)



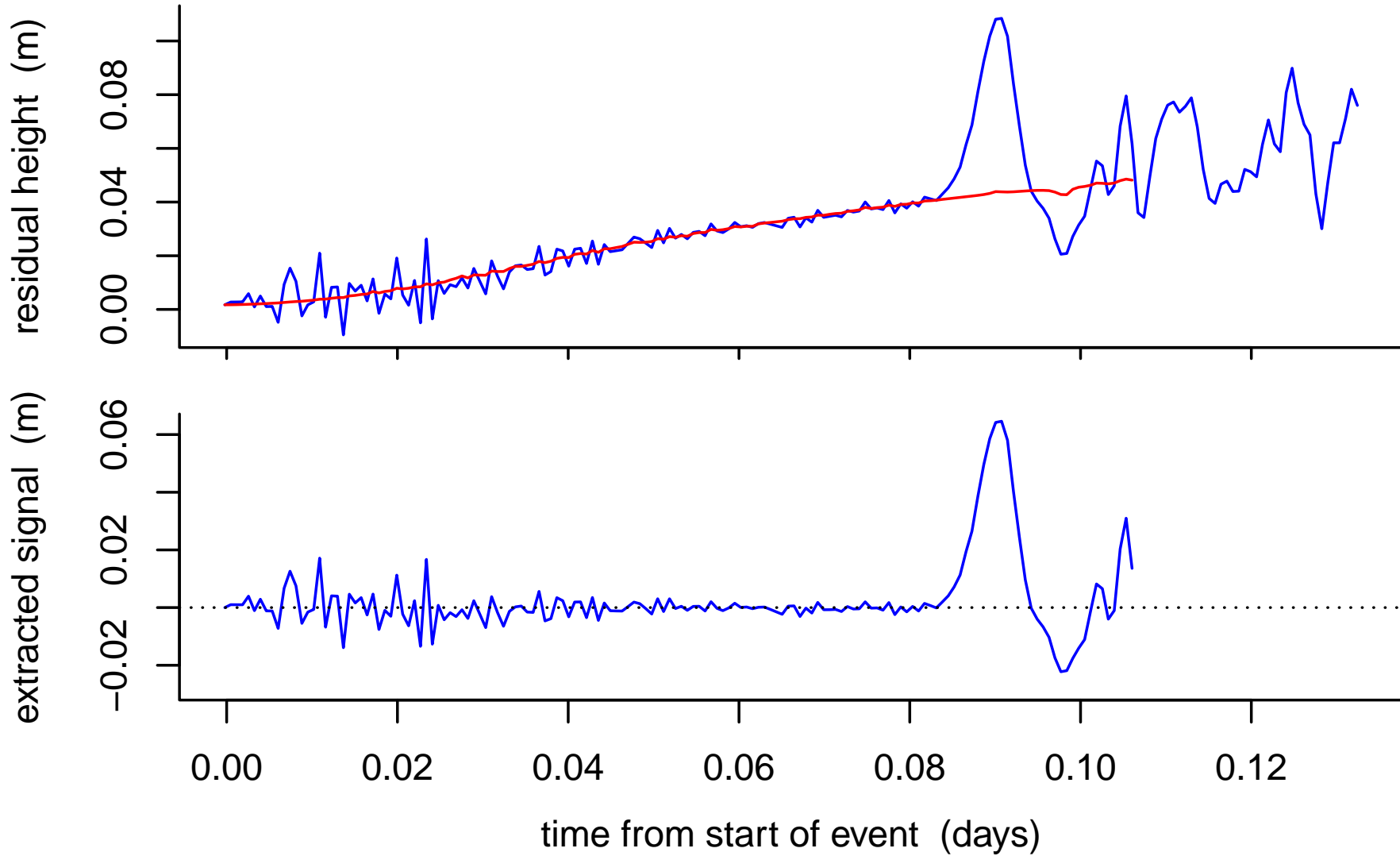
# Kalman Filtering/Smoothing (Local Level Model)



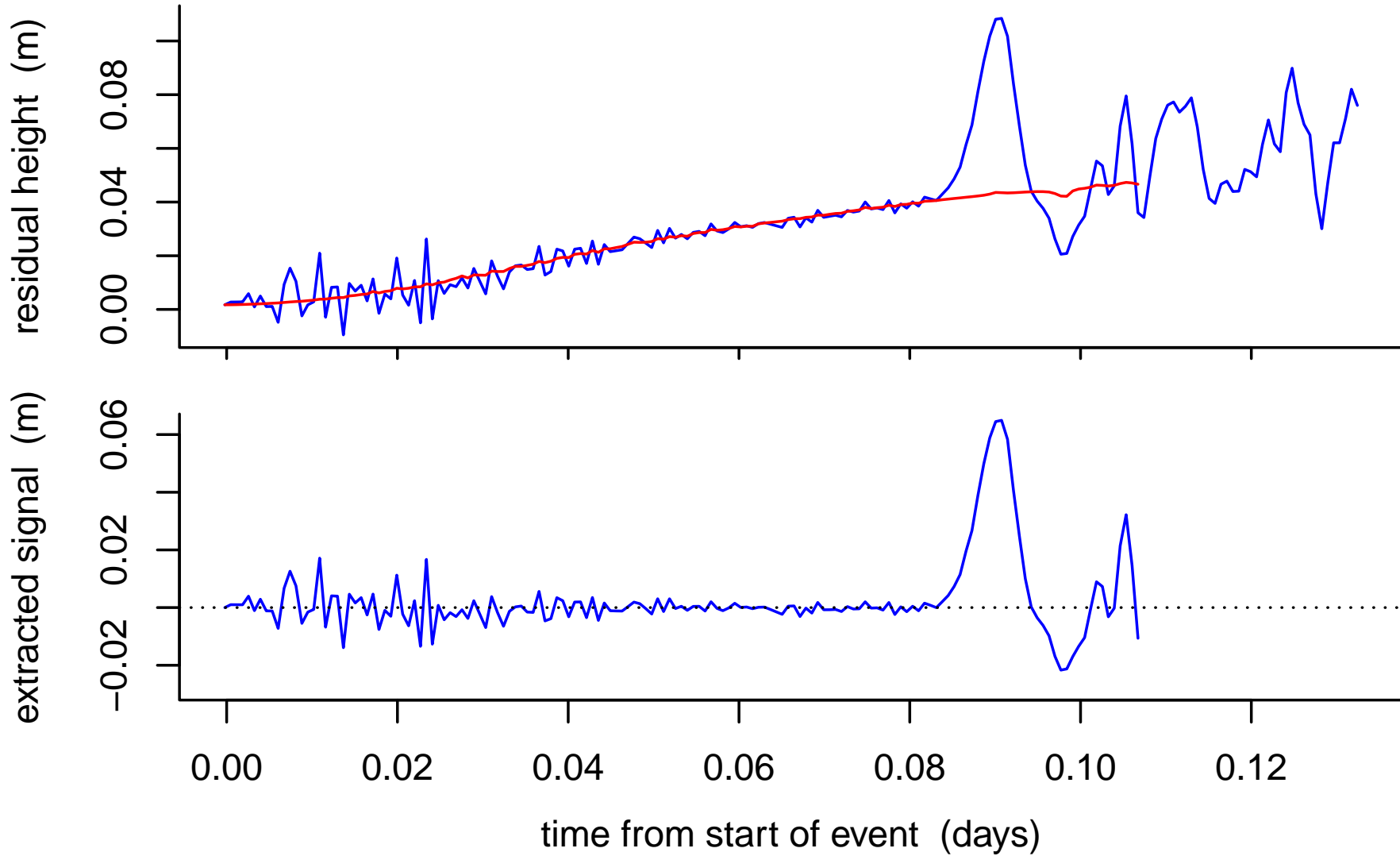
# Kalman Filtering/Smoothing (Local Level Model)



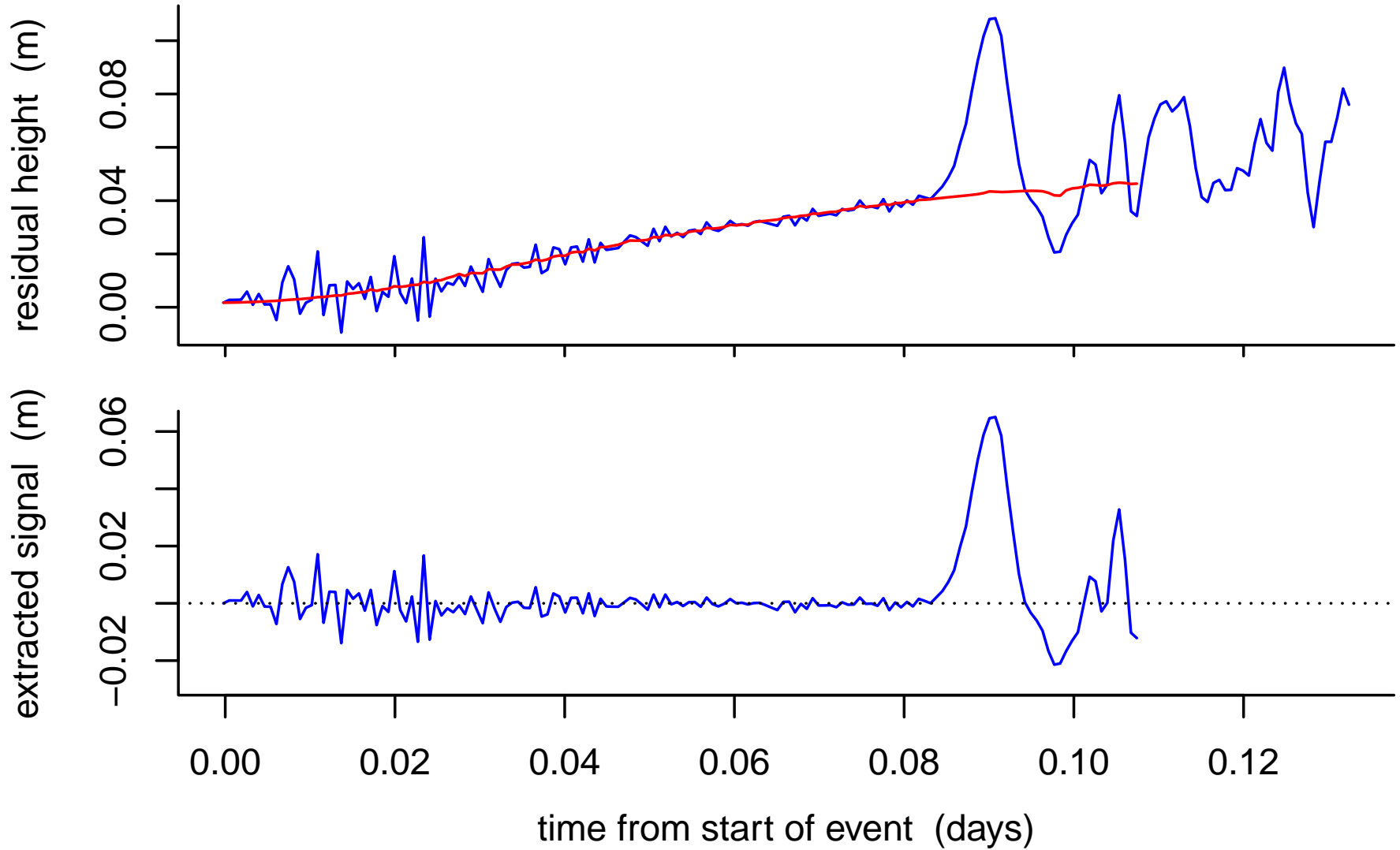
# Kalman Filtering/Smoothing (Local Level Model)



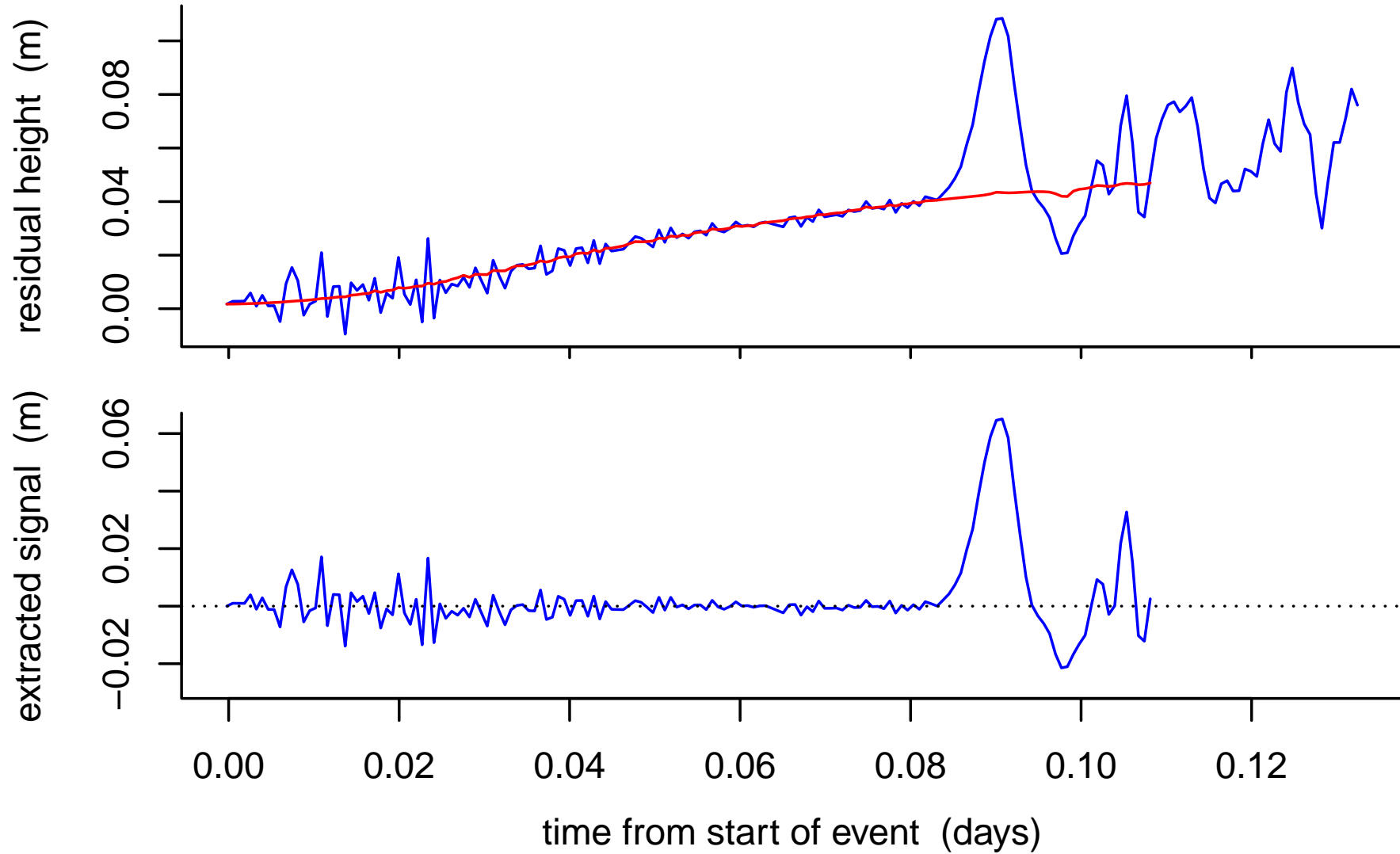
# Kalman Filtering/Smoothing (Local Level Model)



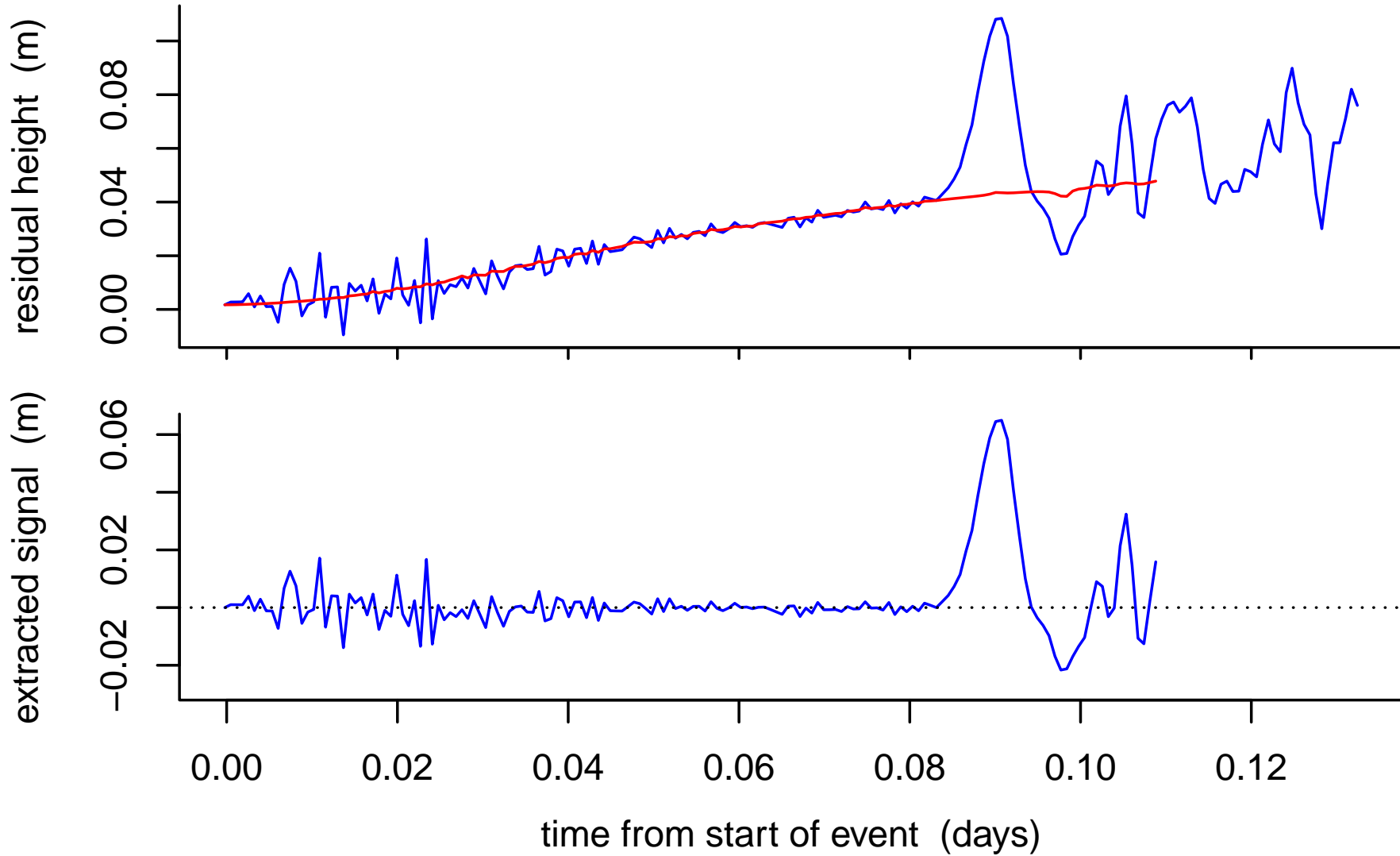
# Kalman Filtering/Smoothing (Local Level Model)



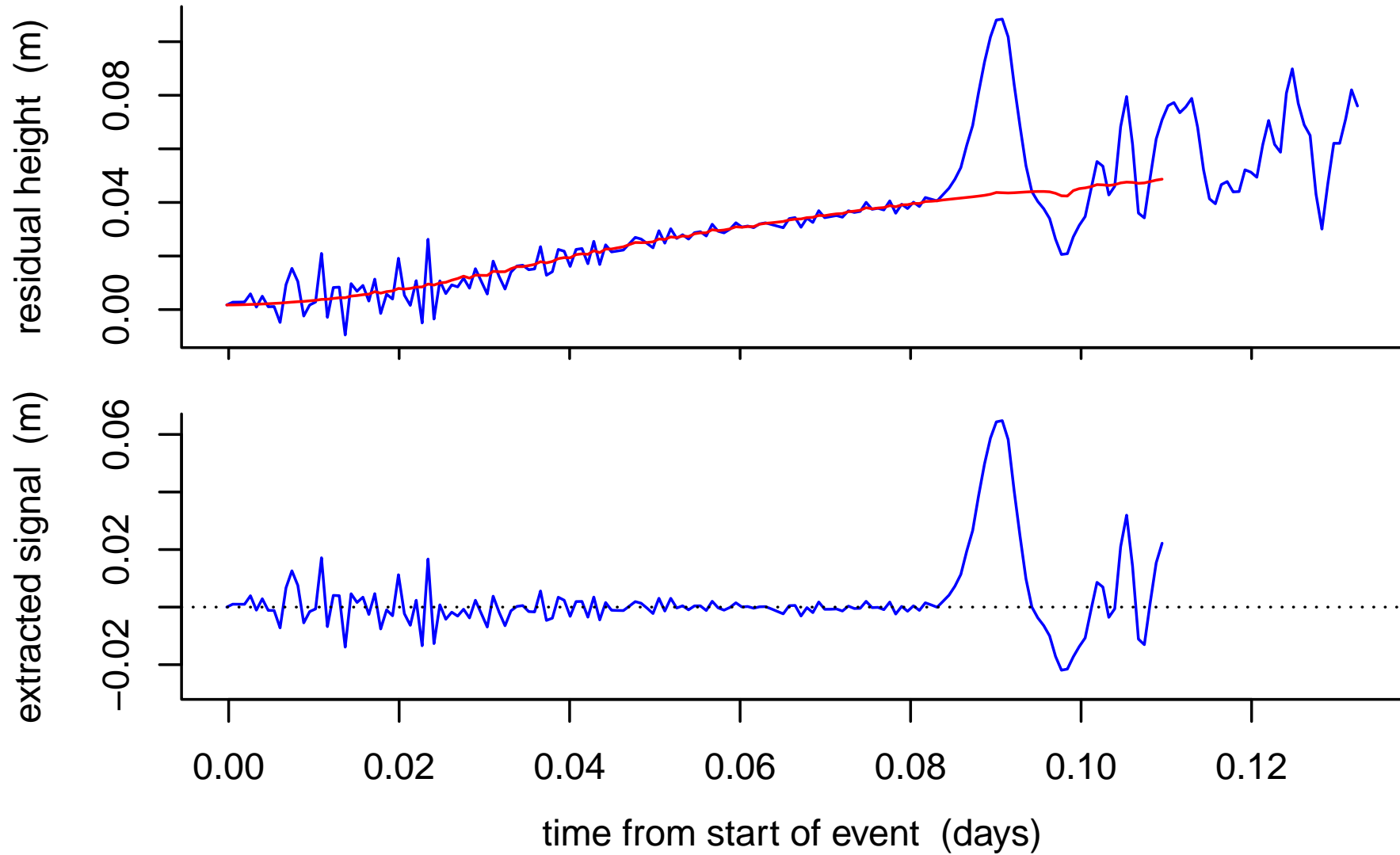
# Kalman Filtering/Smoothing (Local Level Model)



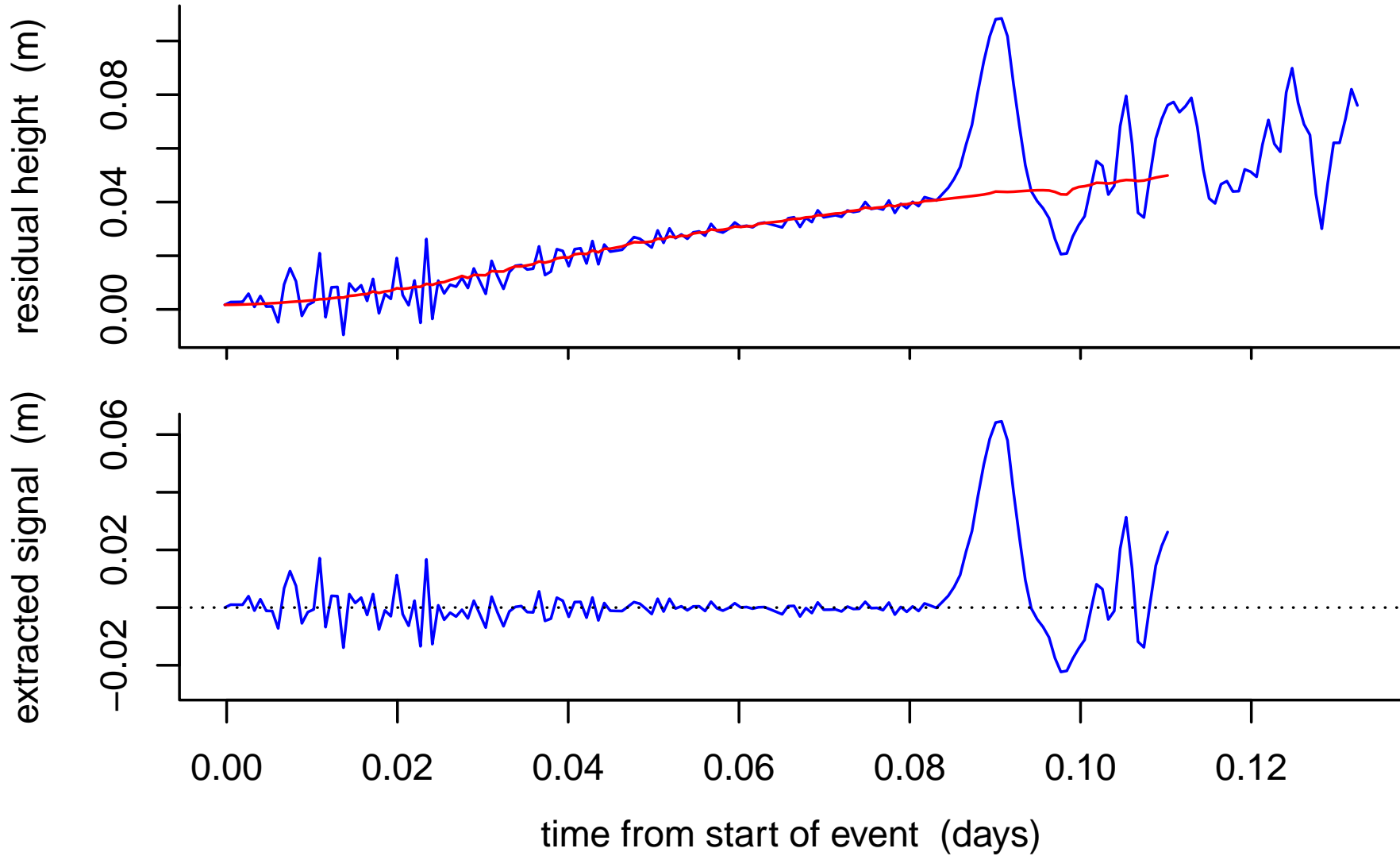
# Kalman Filtering/Smoothing (Local Level Model)



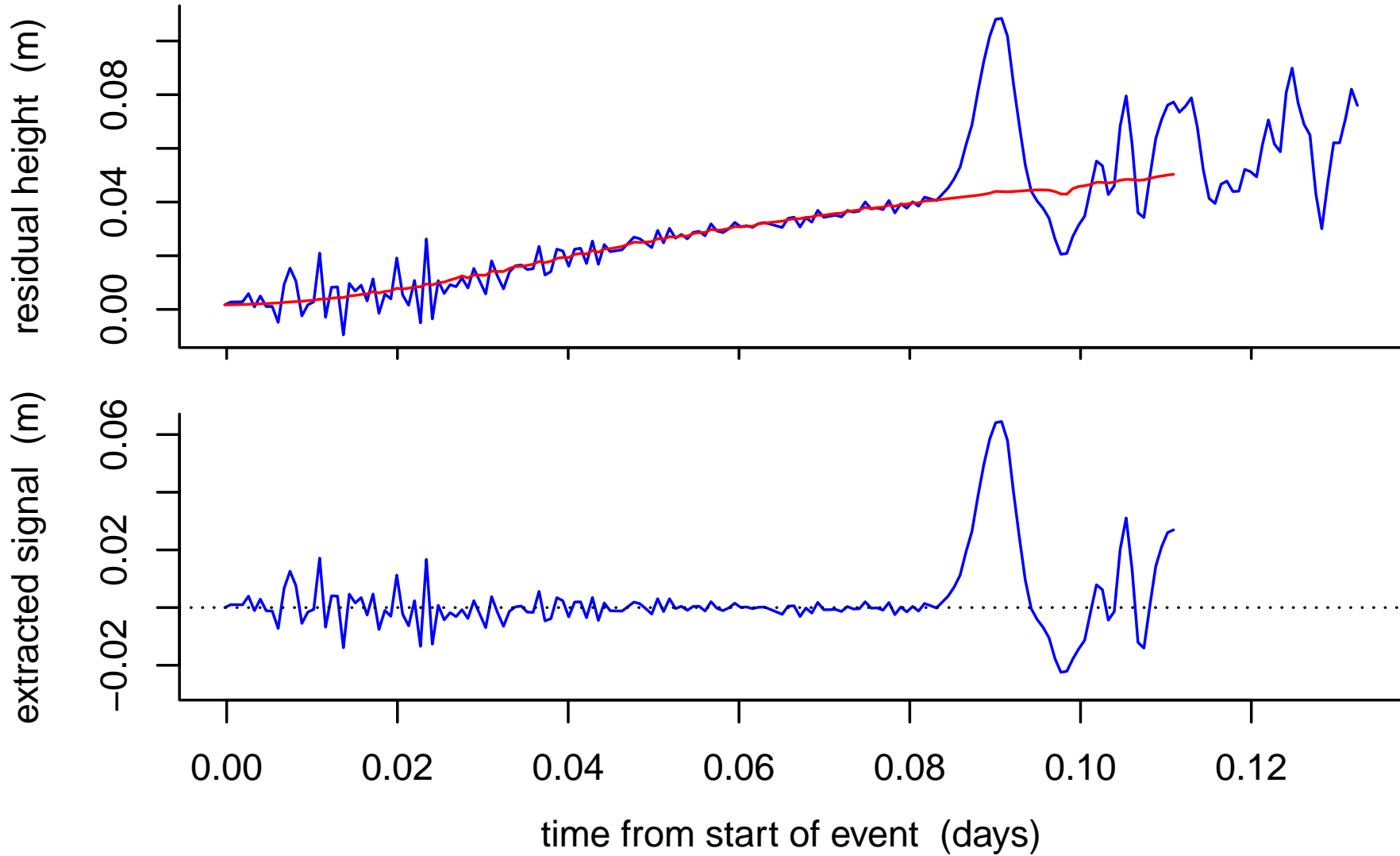
# Kalman Filtering/Smoothing (Local Level Model)



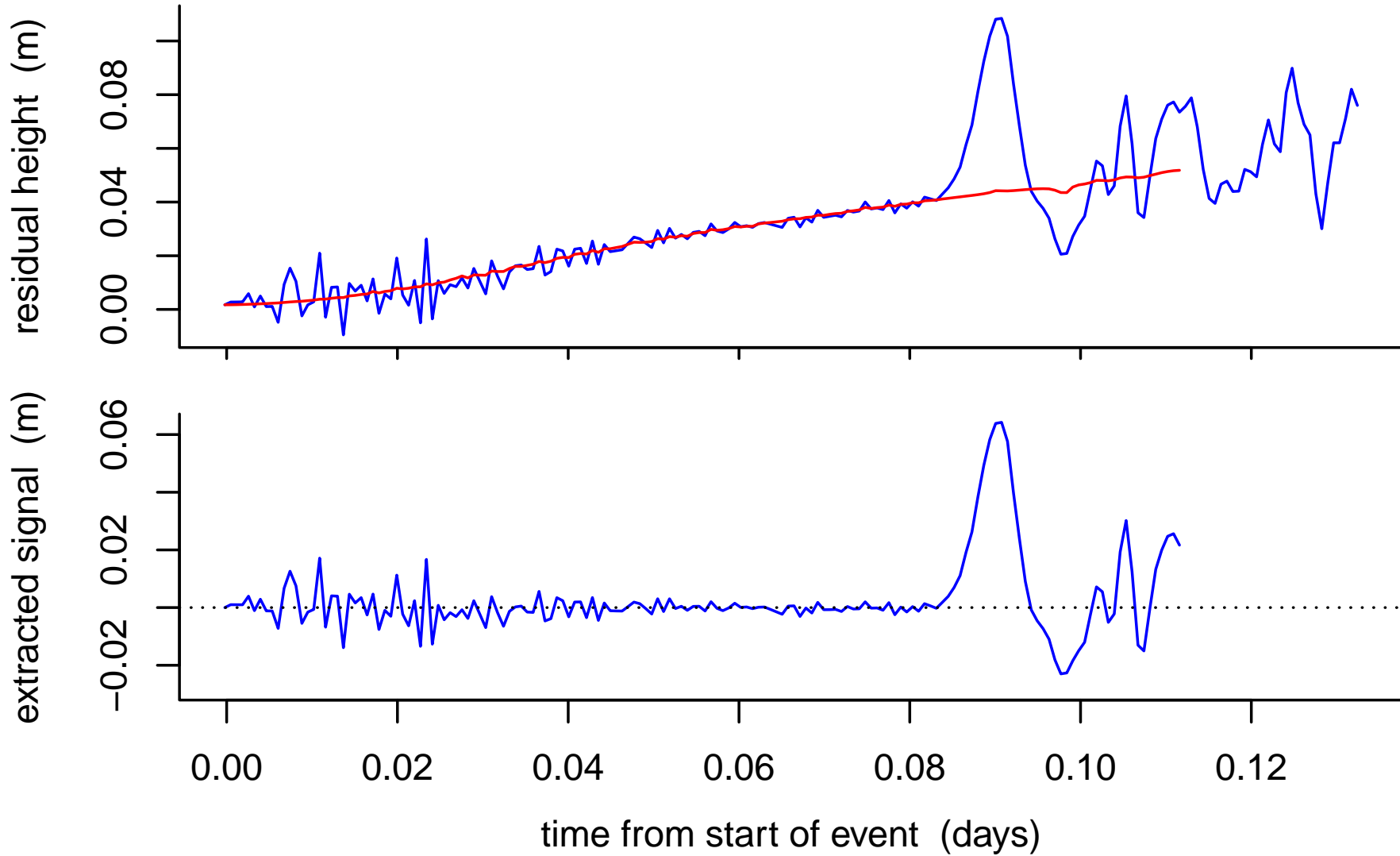
# Kalman Filtering/Smoothing (Local Level Model)



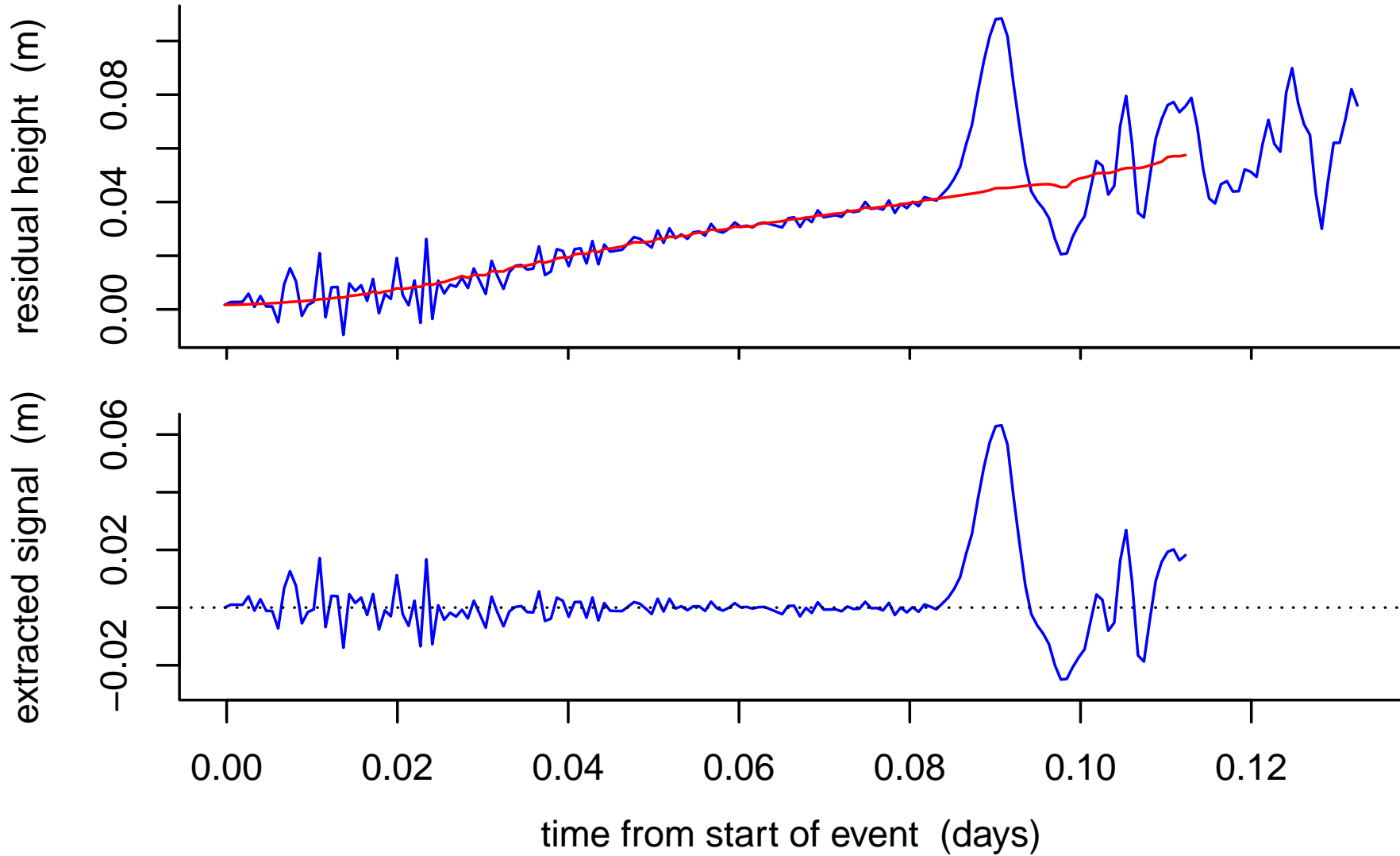
# Kalman Filtering/Smoothing (Local Level Model)



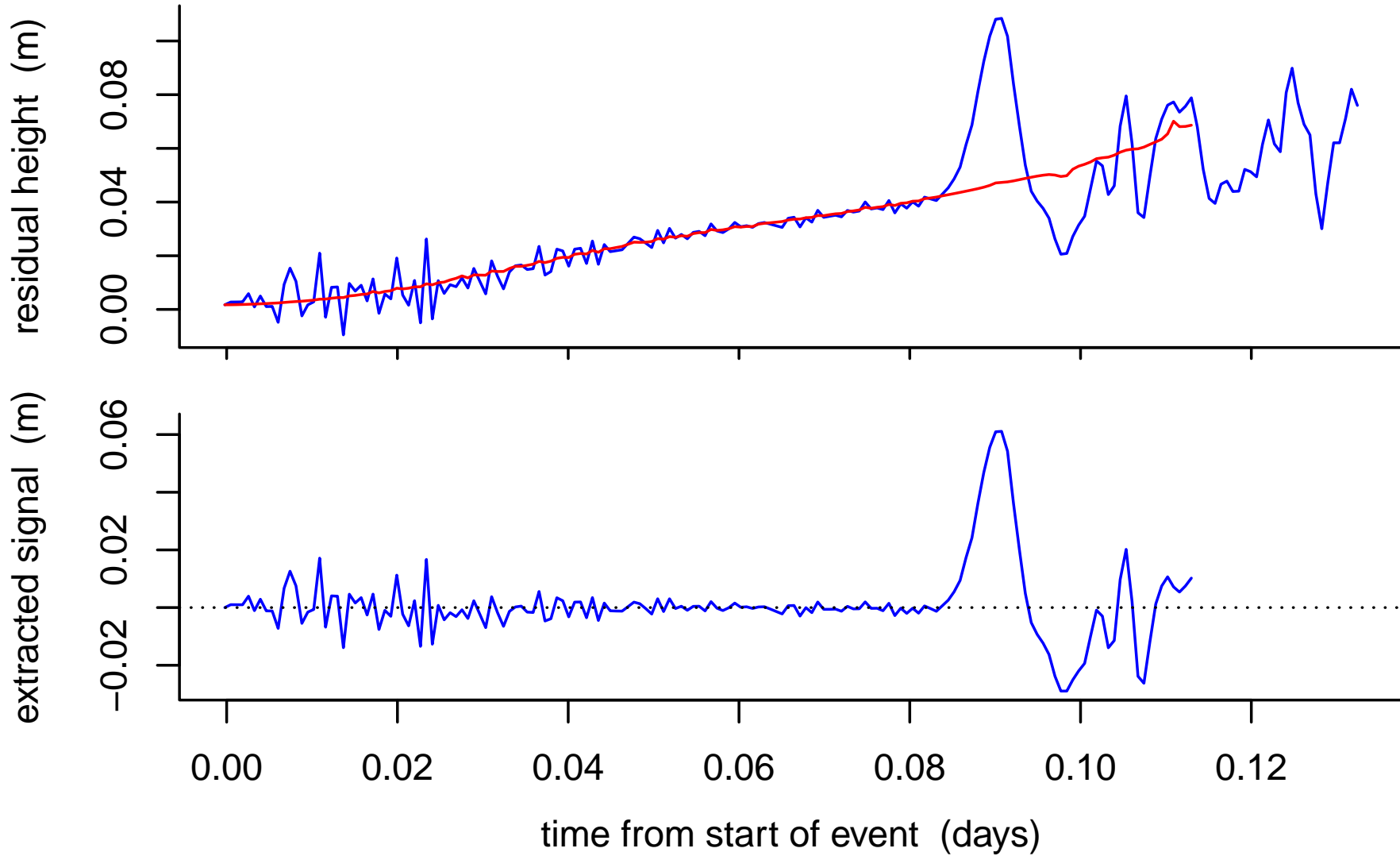
# Kalman Filtering/Smoothing (Local Level Model)



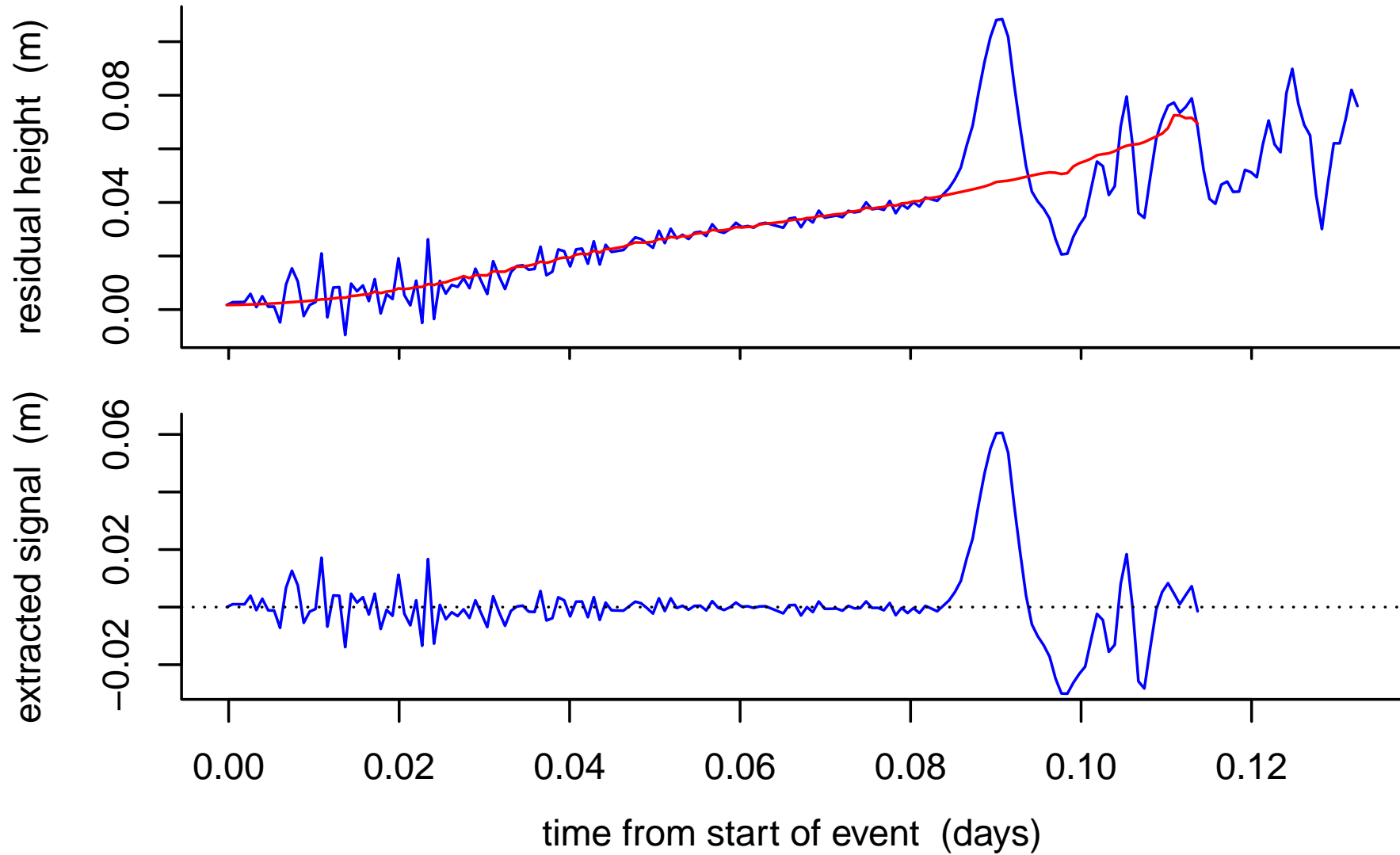
# Kalman Filtering/Smoothing (Local Level Model)



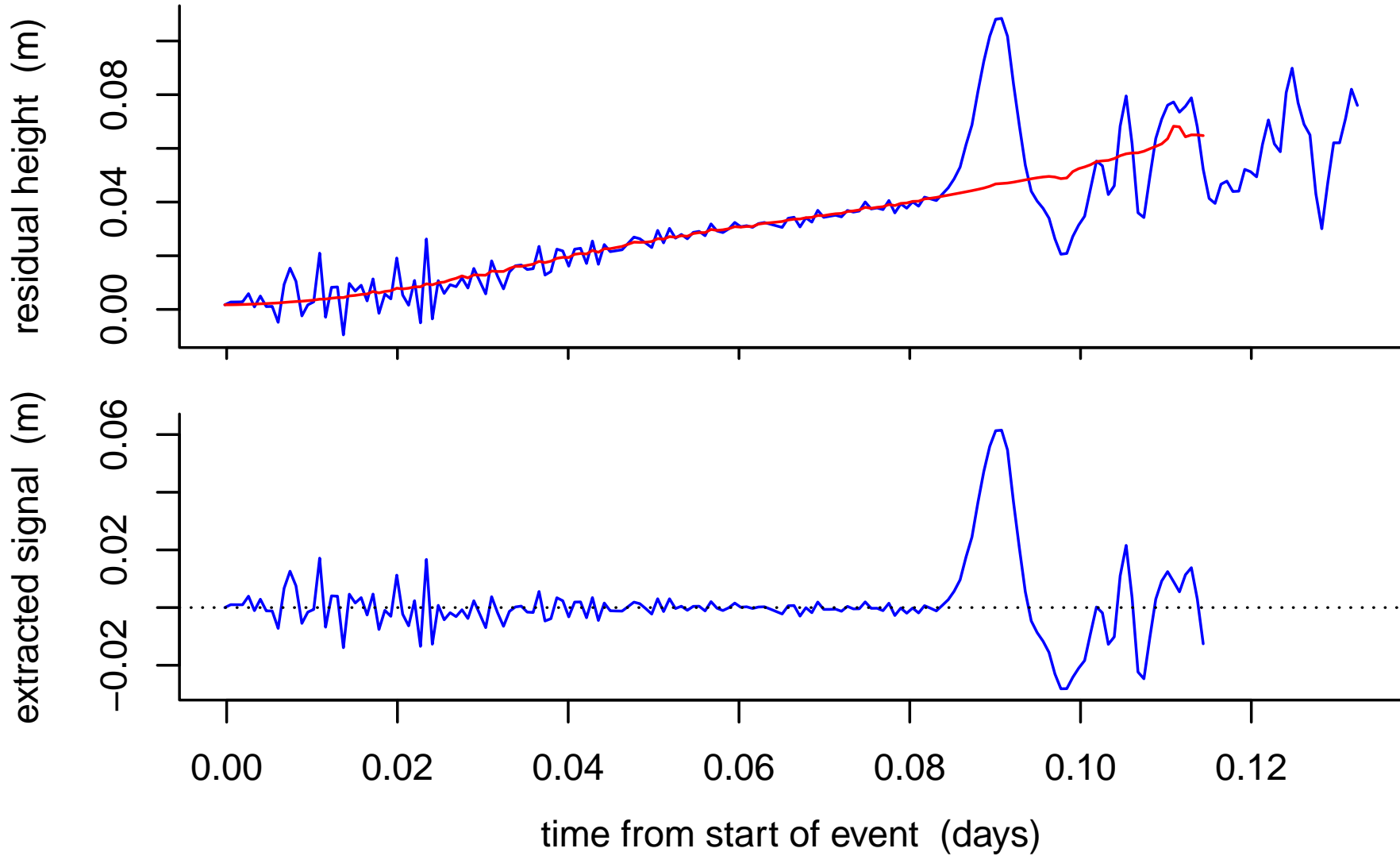
# Kalman Filtering/Smoothing (Local Level Model)



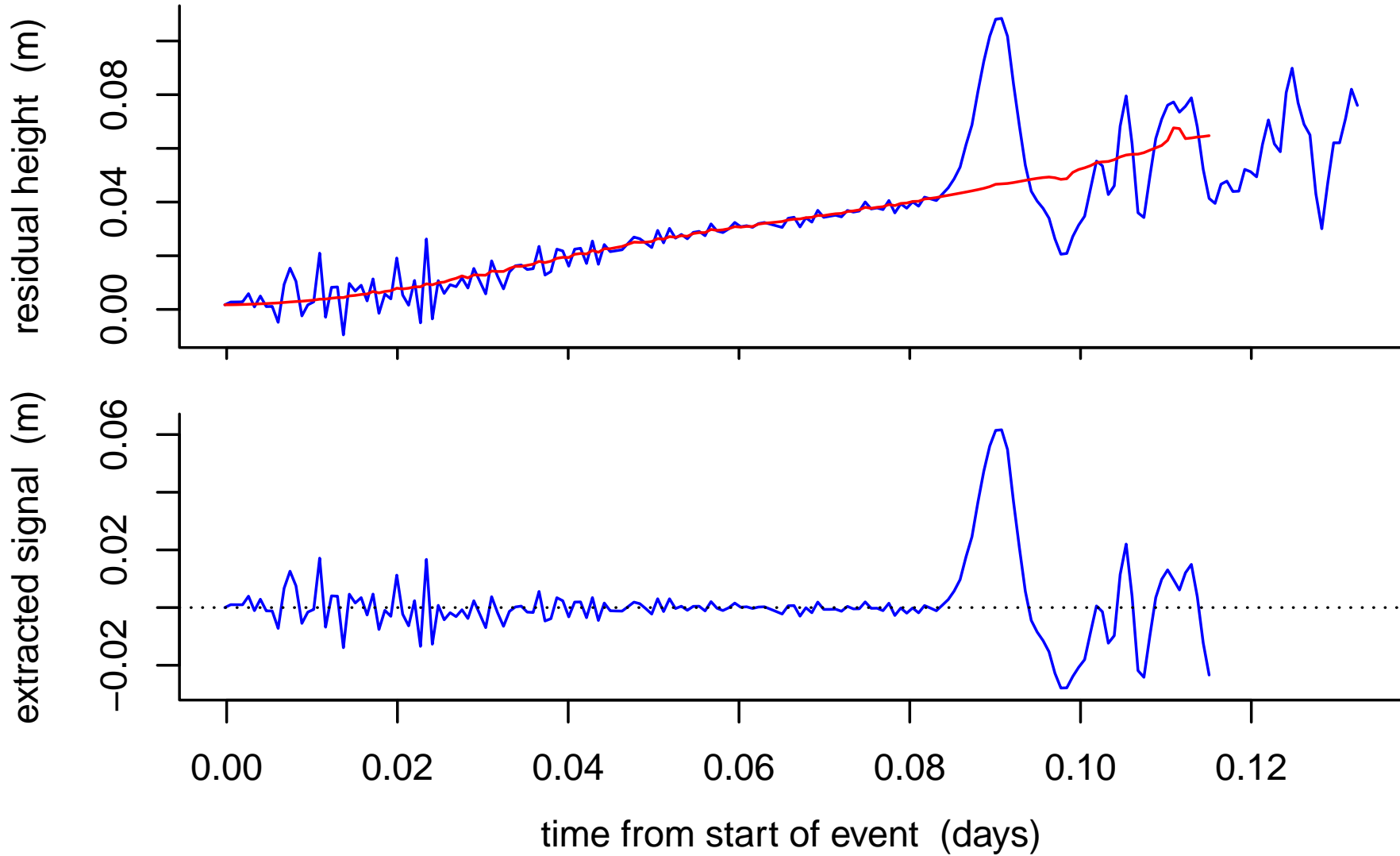
# Kalman Filtering/Smoothing (Local Level Model)



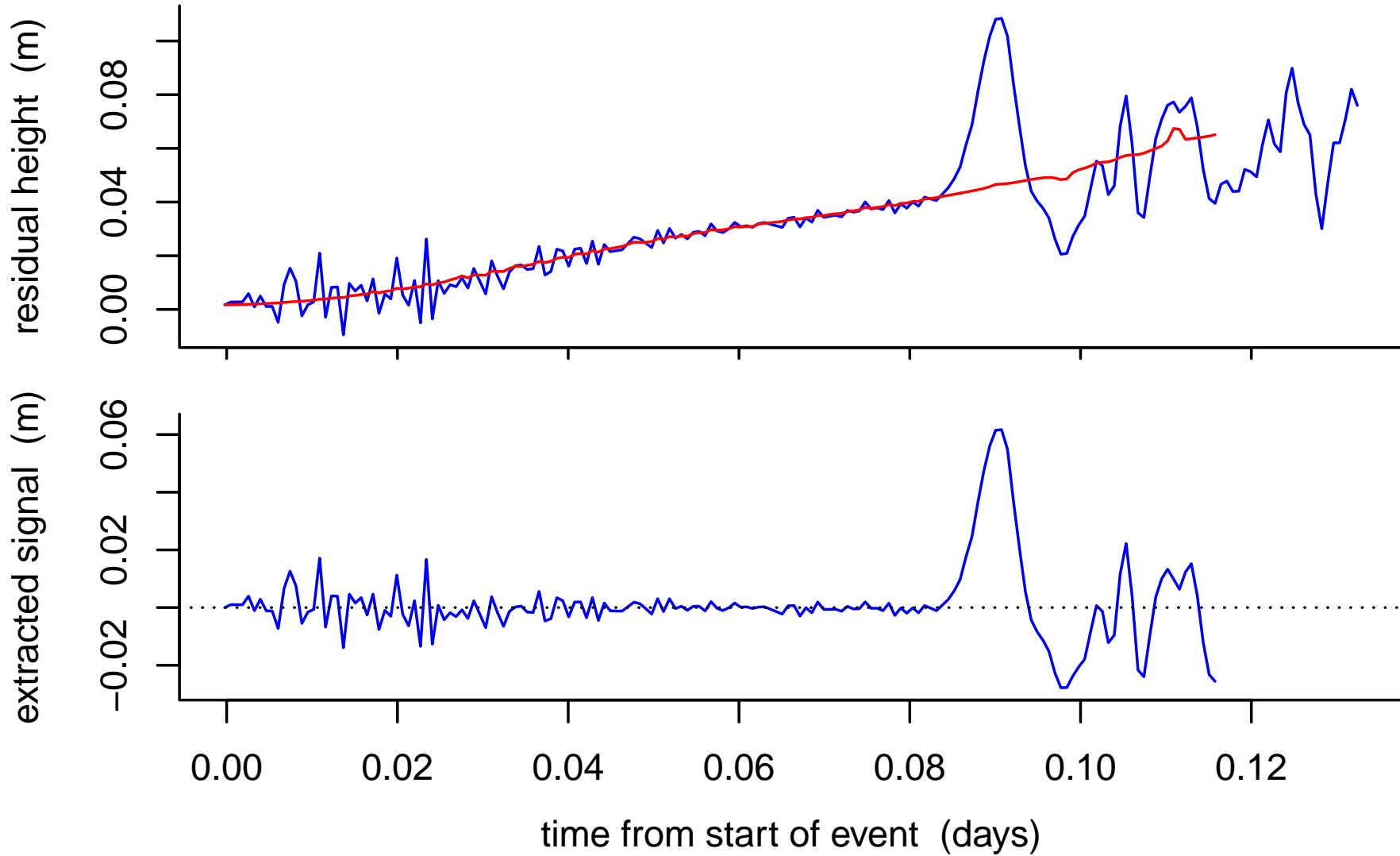
# Kalman Filtering/Smoothing (Local Level Model)



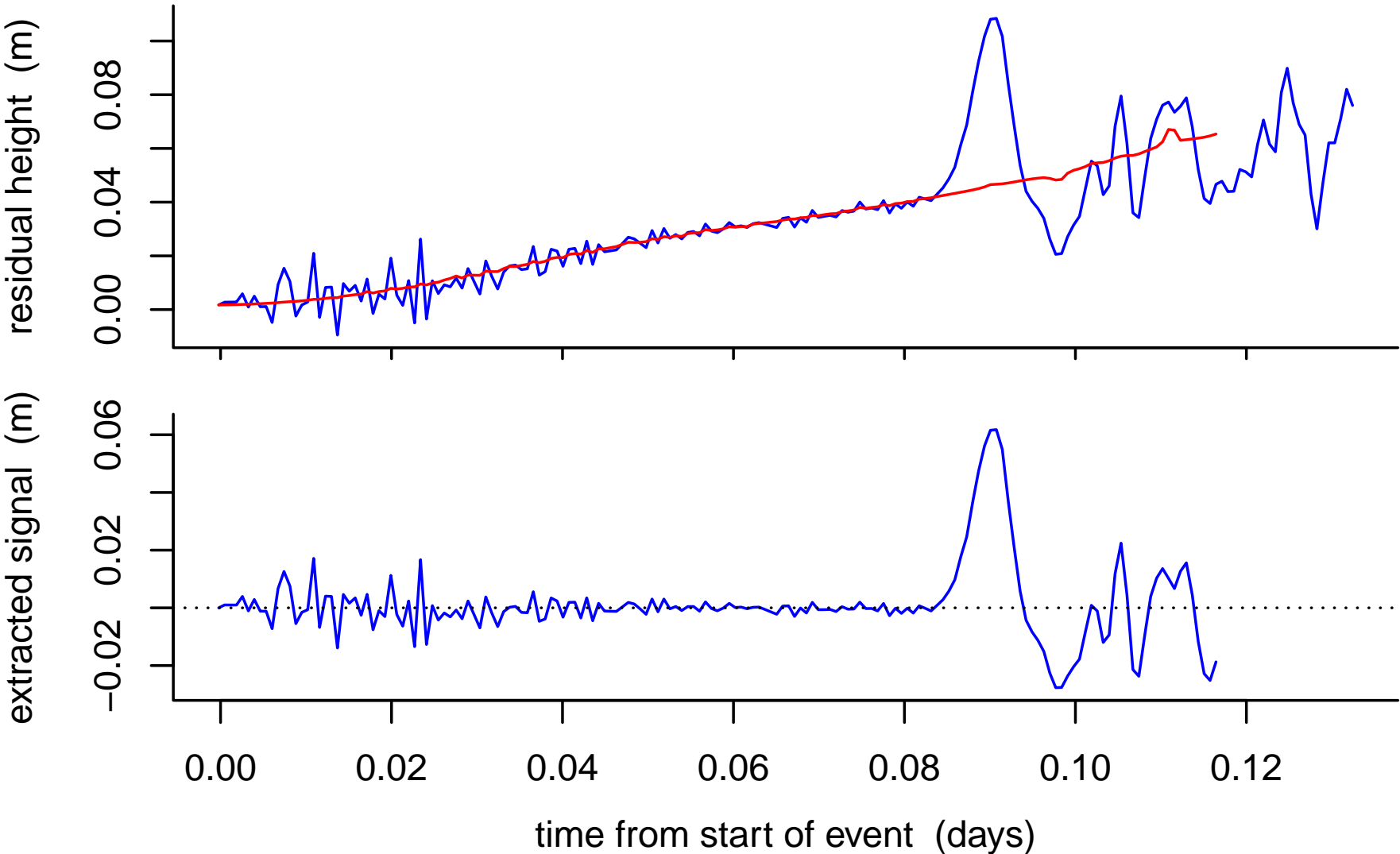
# Kalman Filtering/Smoothing (Local Level Model)



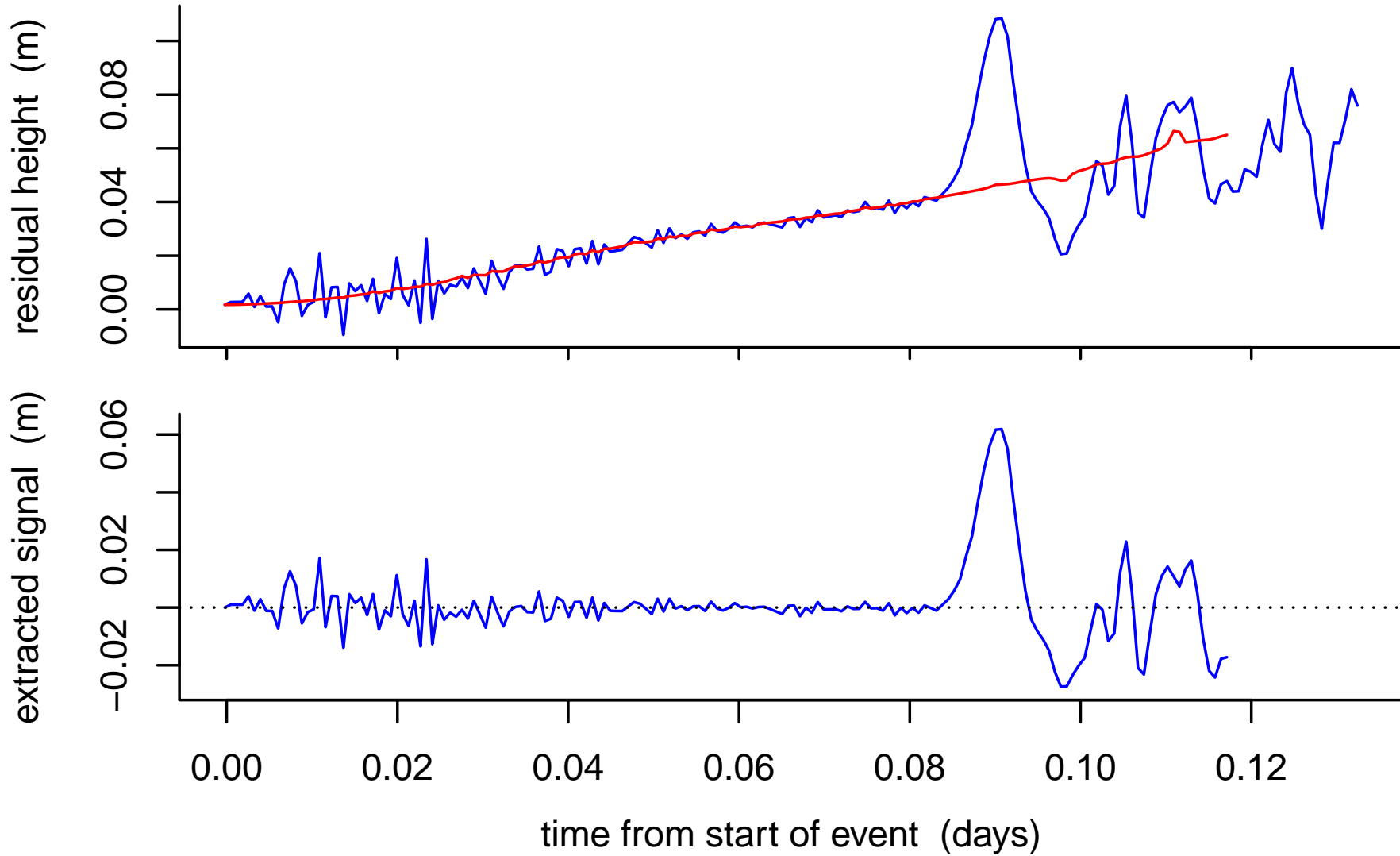
# Kalman Filtering/Smoothing (Local Level Model)



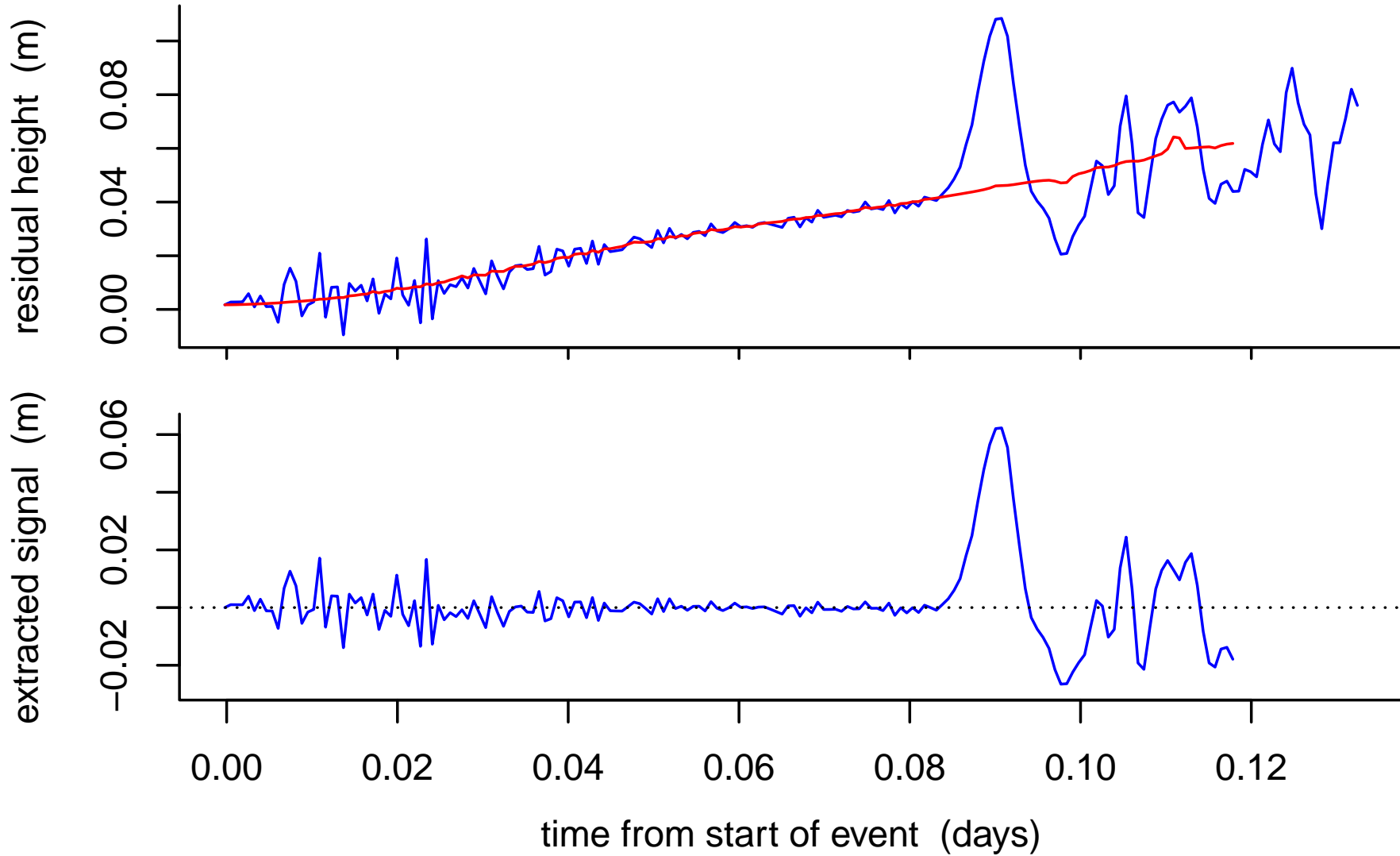
# Kalman Filtering/Smoothing (Local Level Model)



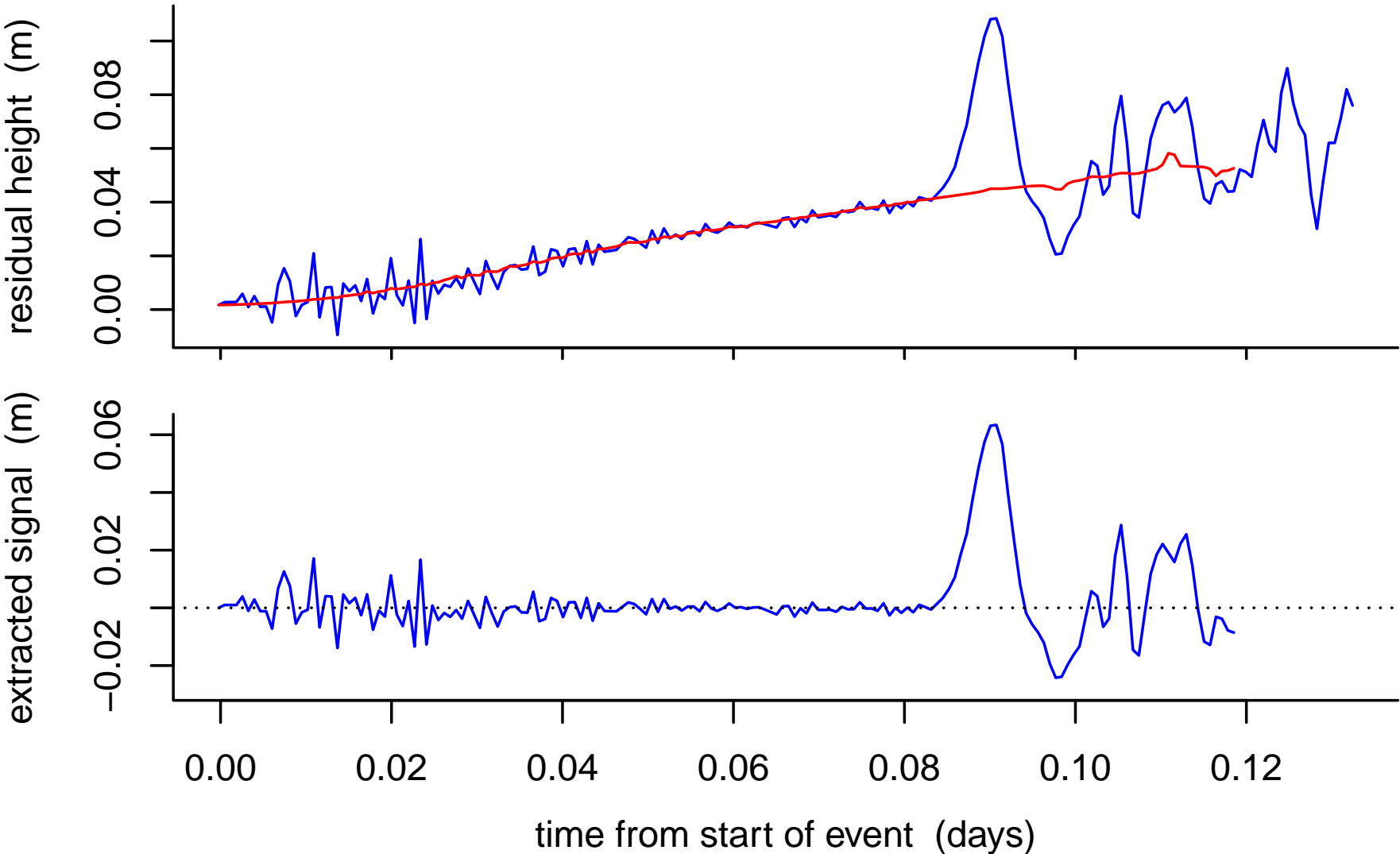
# Kalman Filtering/Smoothing (Local Level Model)



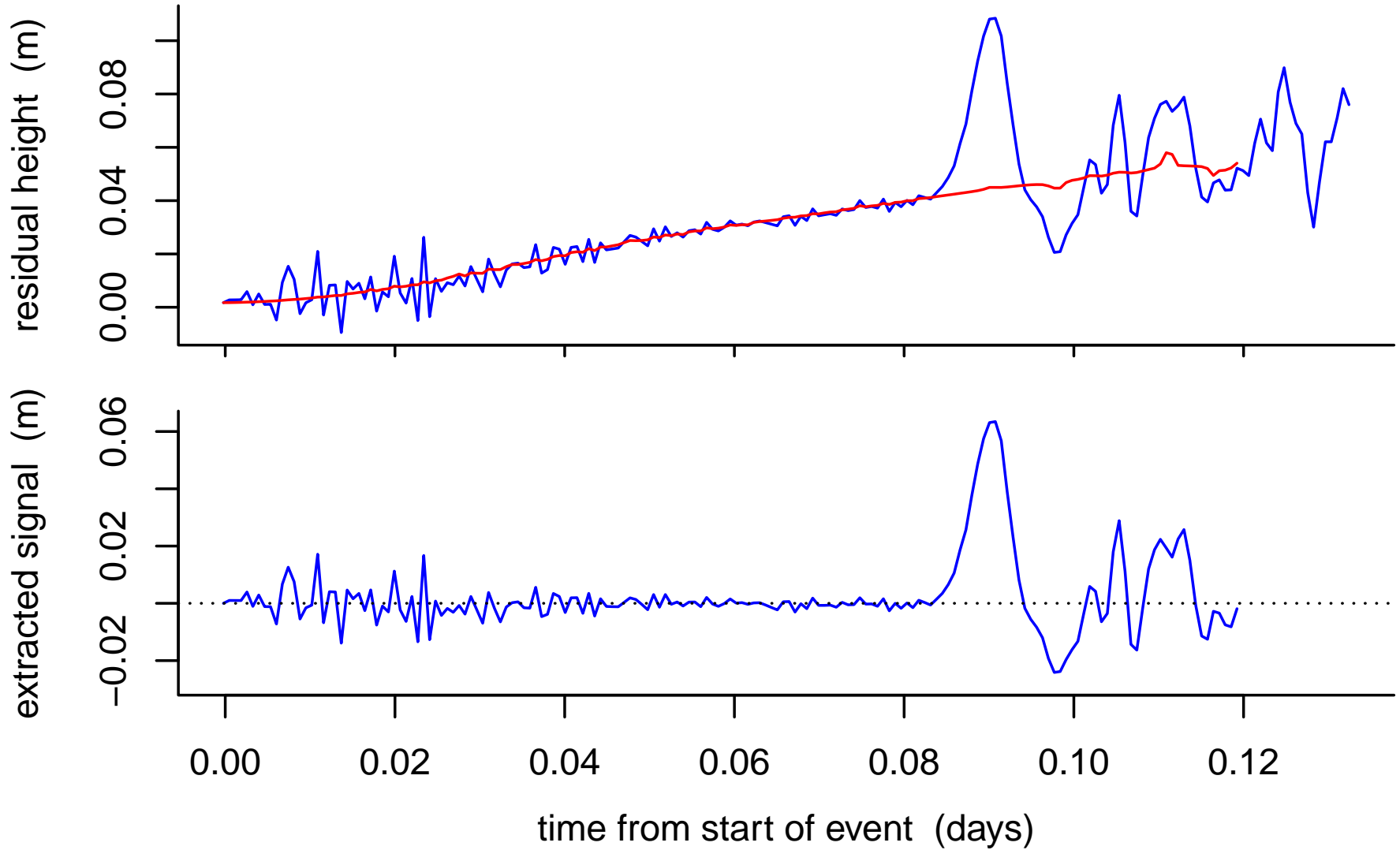
# Kalman Filtering/Smoothing (Local Level Model)



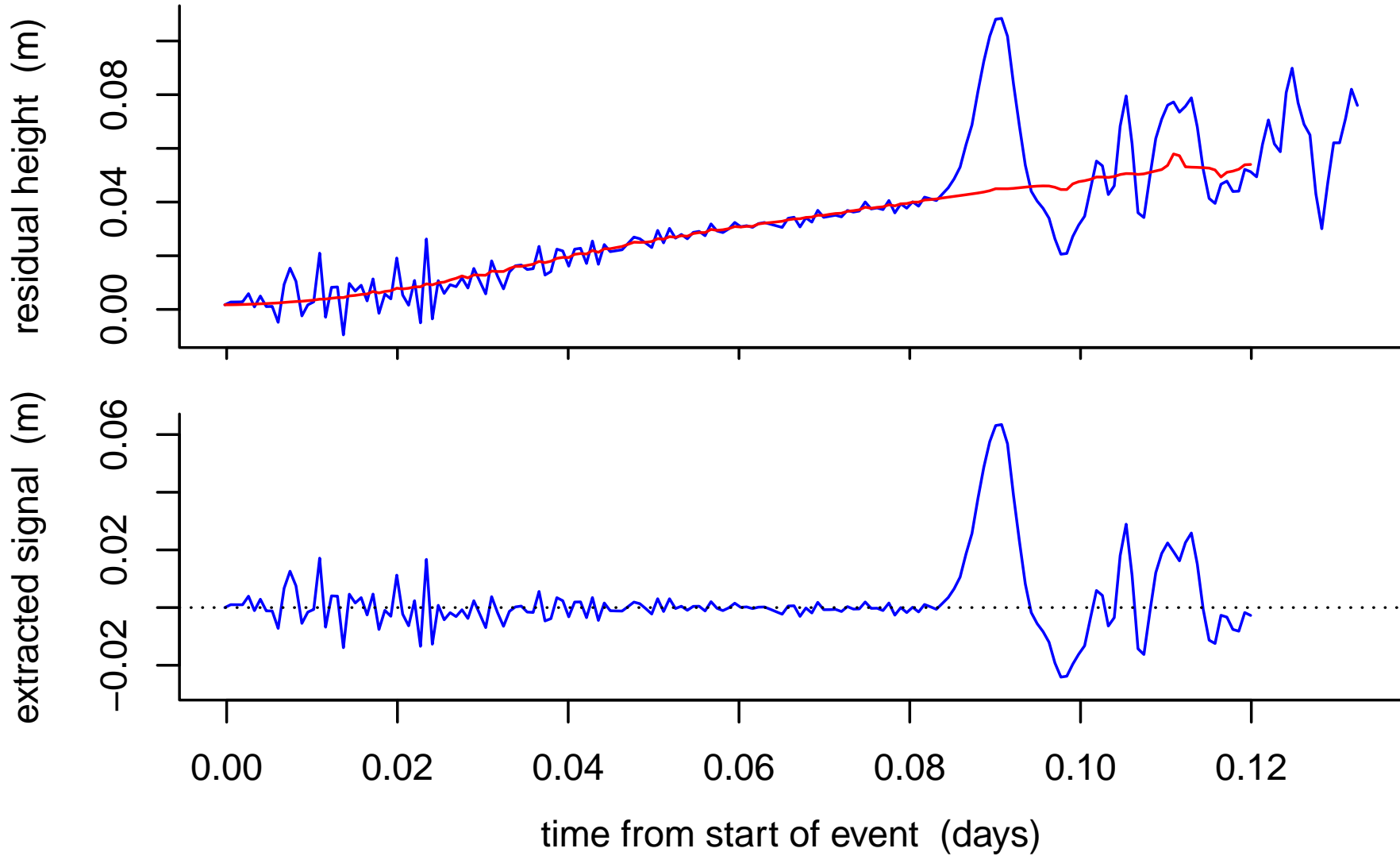
# Kalman Filtering/Smoothing (Local Level Model)



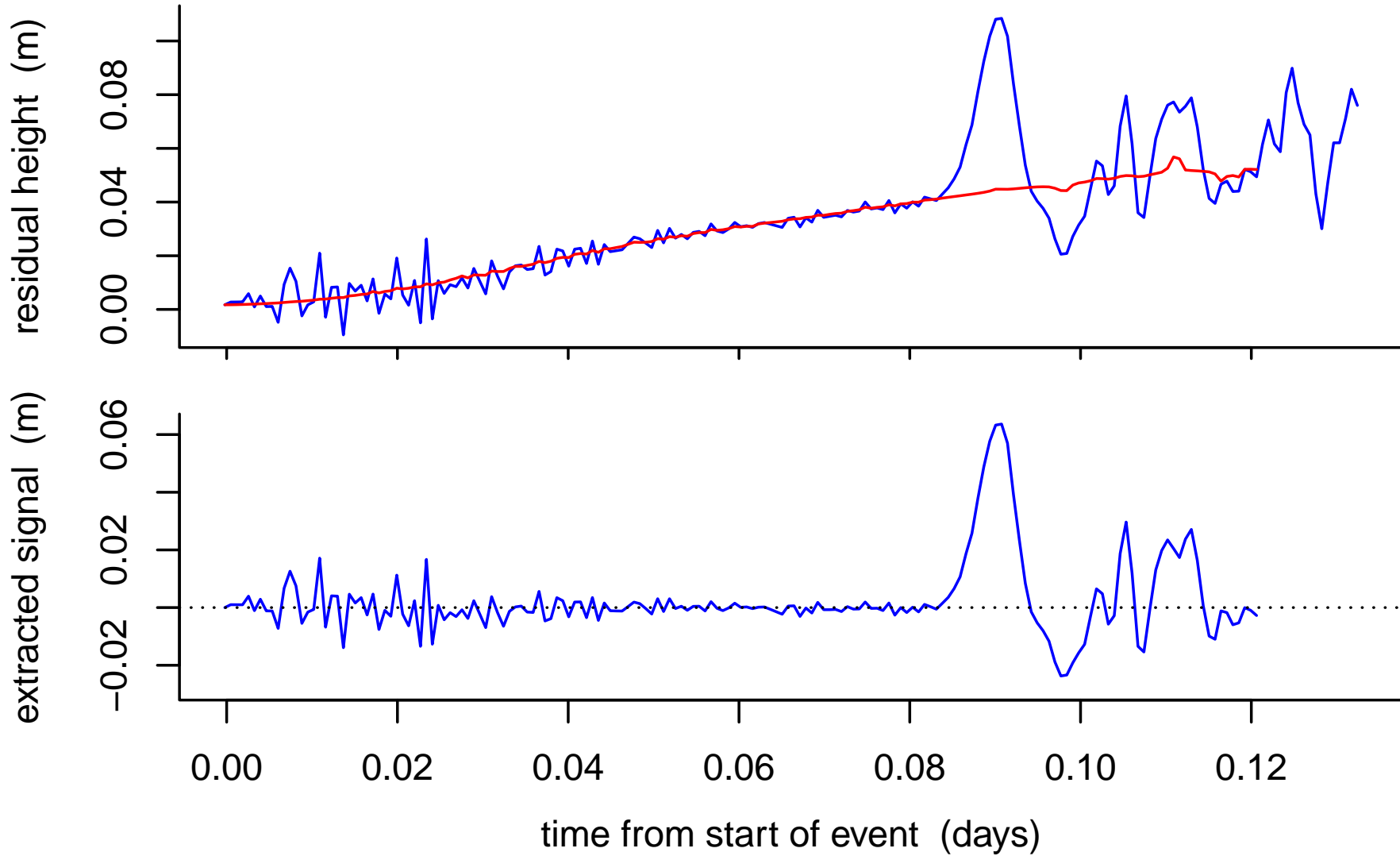
# Kalman Filtering/Smoothing (Local Level Model)



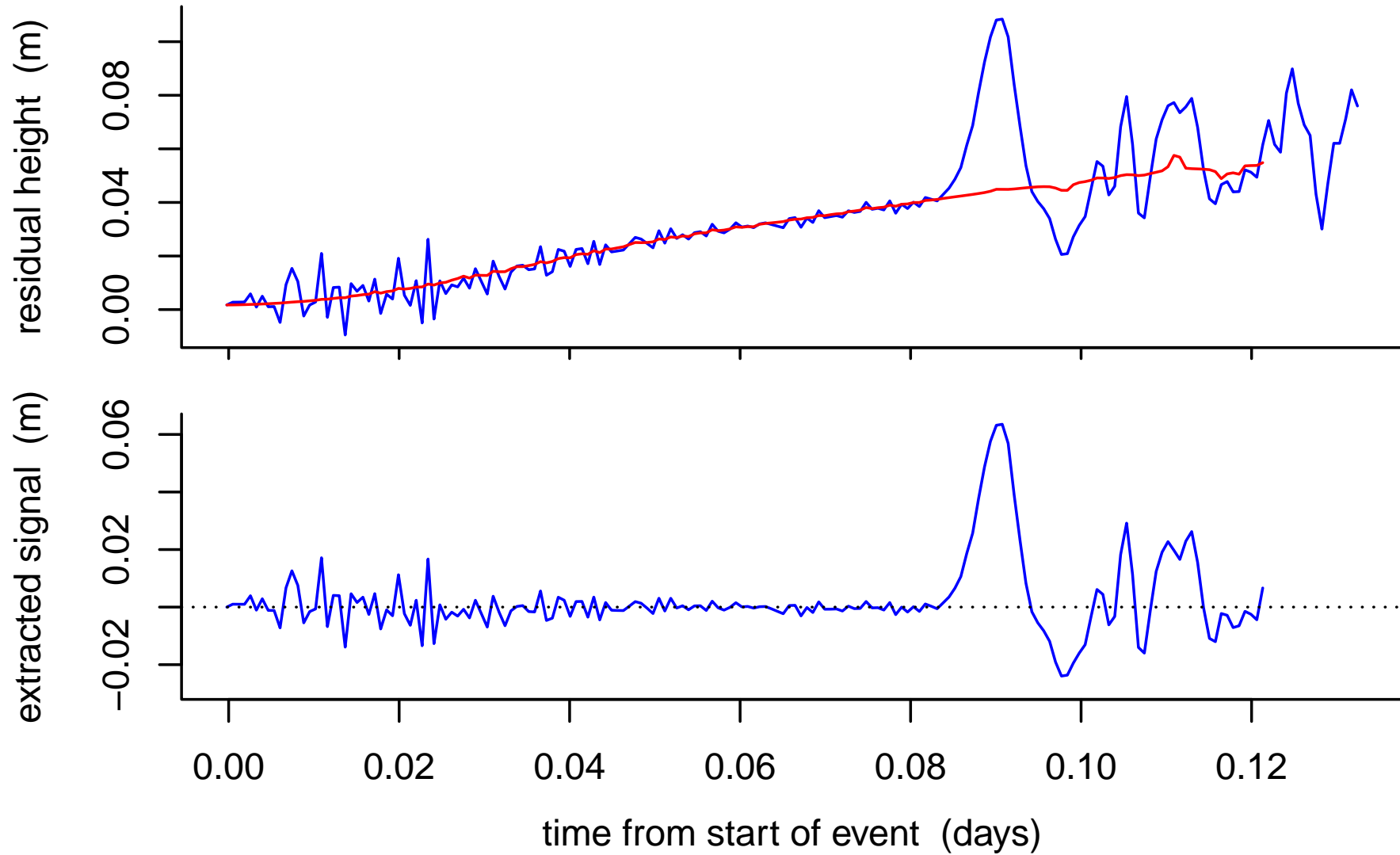
# Kalman Filtering/Smoothing (Local Level Model)



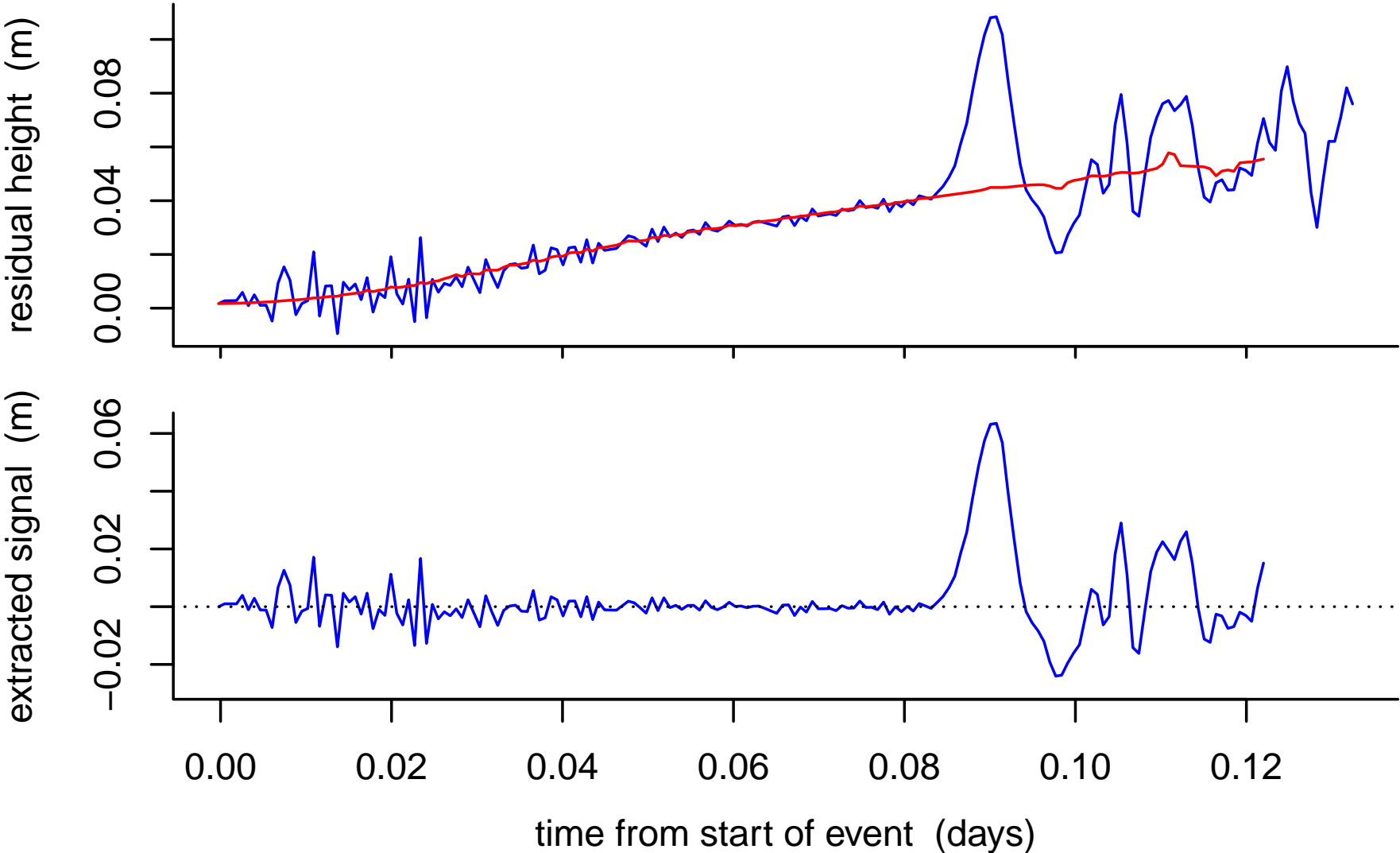
# Kalman Filtering/Smoothing (Local Level Model)



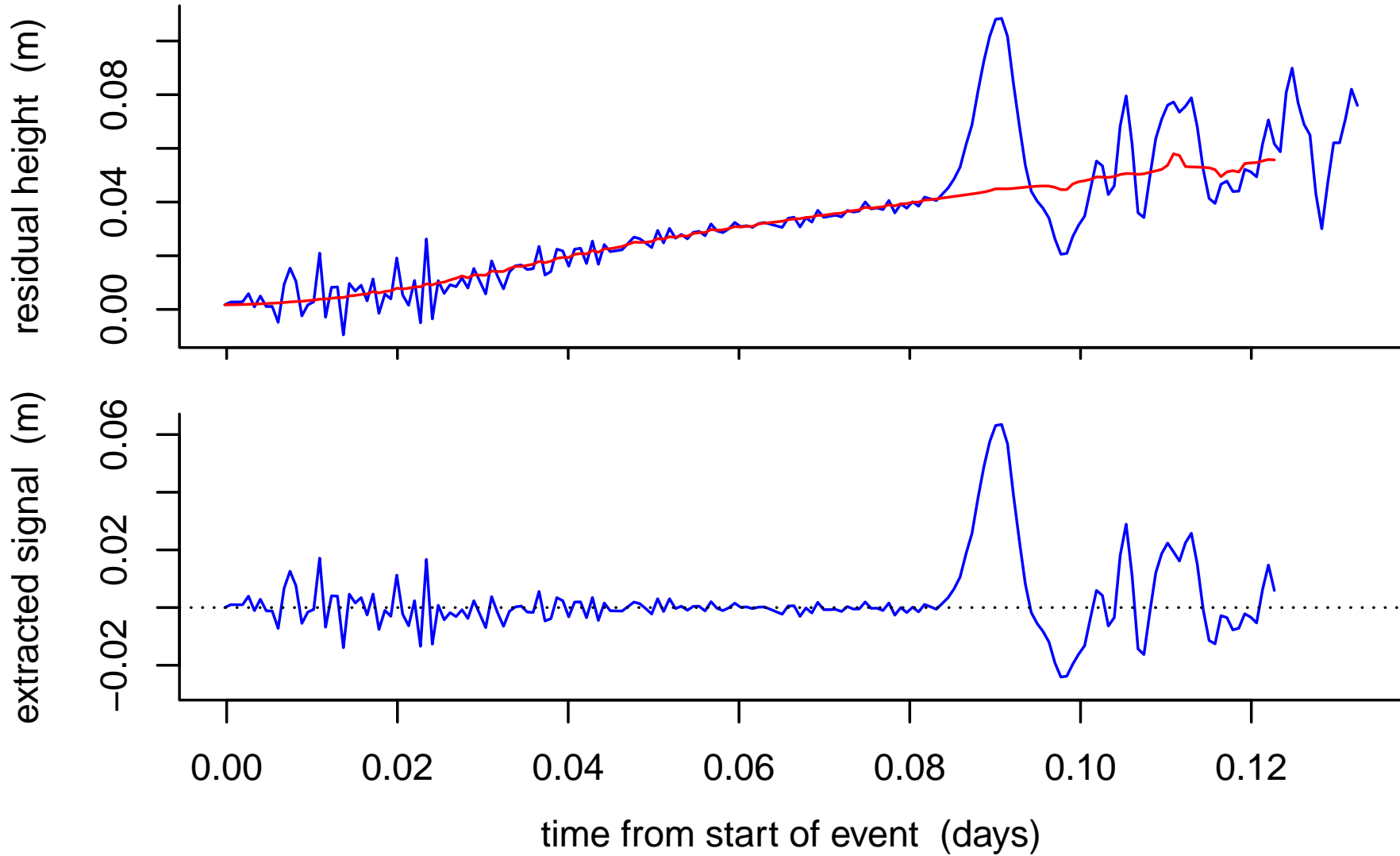
# Kalman Filtering/Smoothing (Local Level Model)



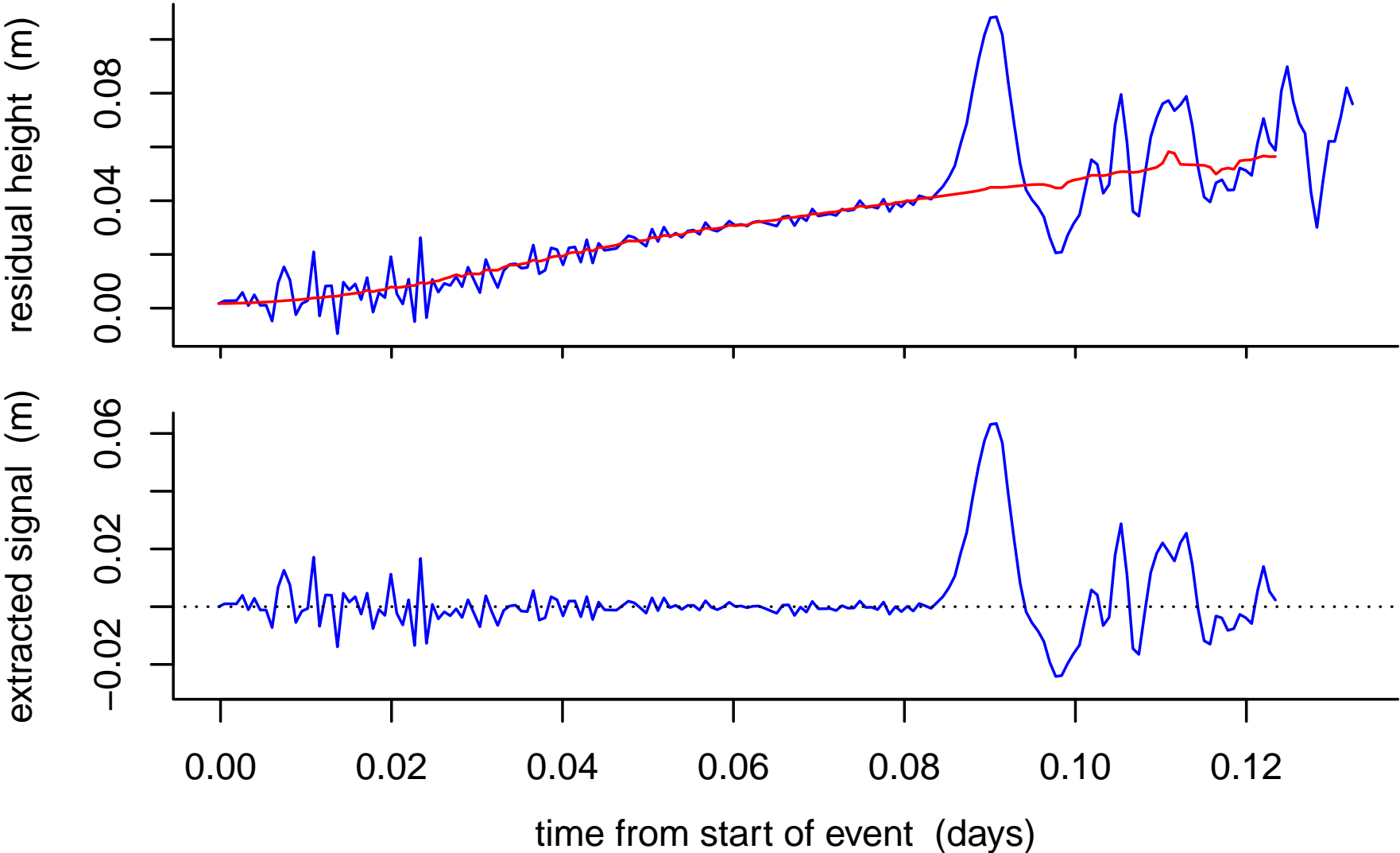
# Kalman Filtering/Smoothing (Local Level Model)



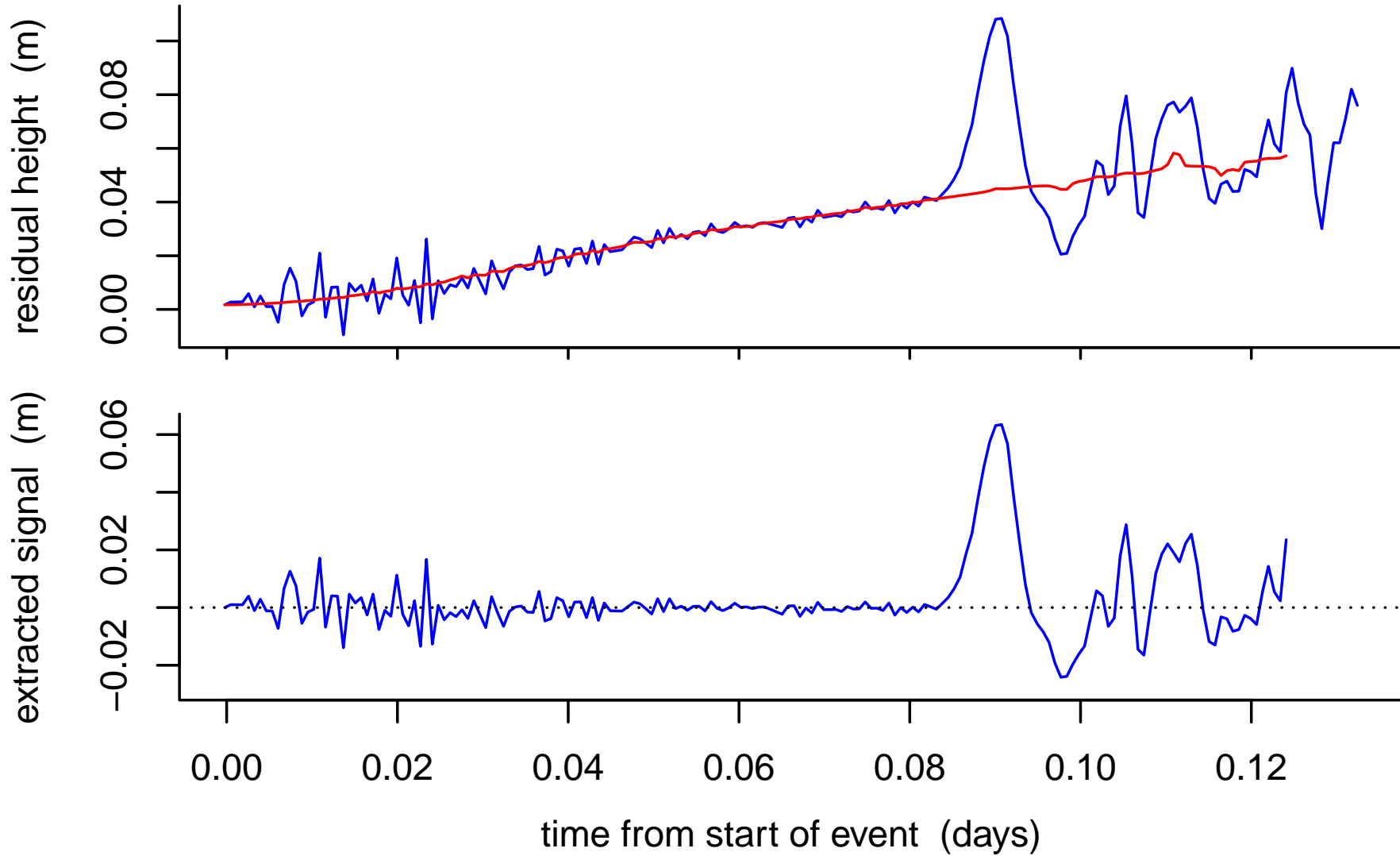
# Kalman Filtering/Smoothing (Local Level Model)



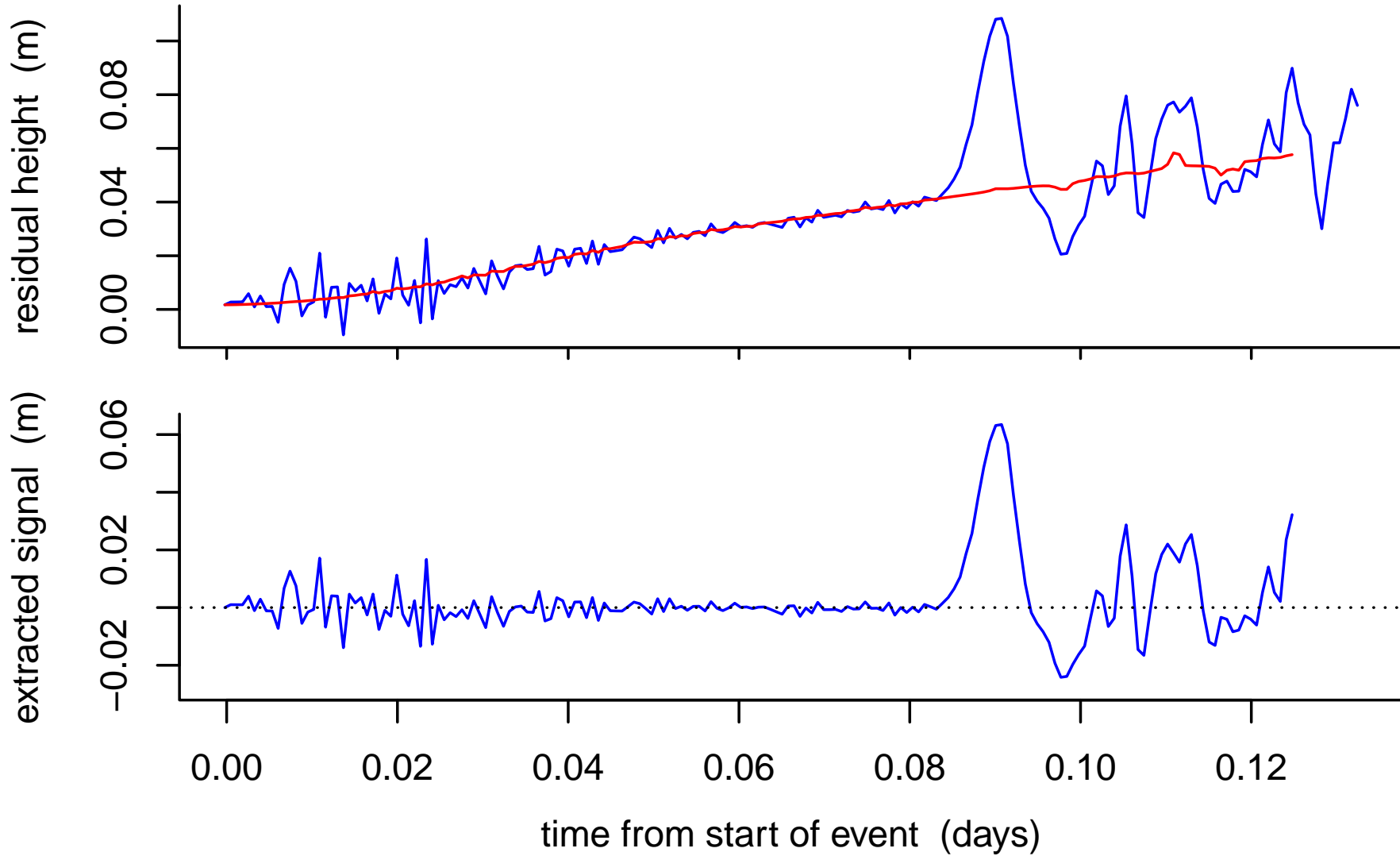
# Kalman Filtering/Smoothing (Local Level Model)



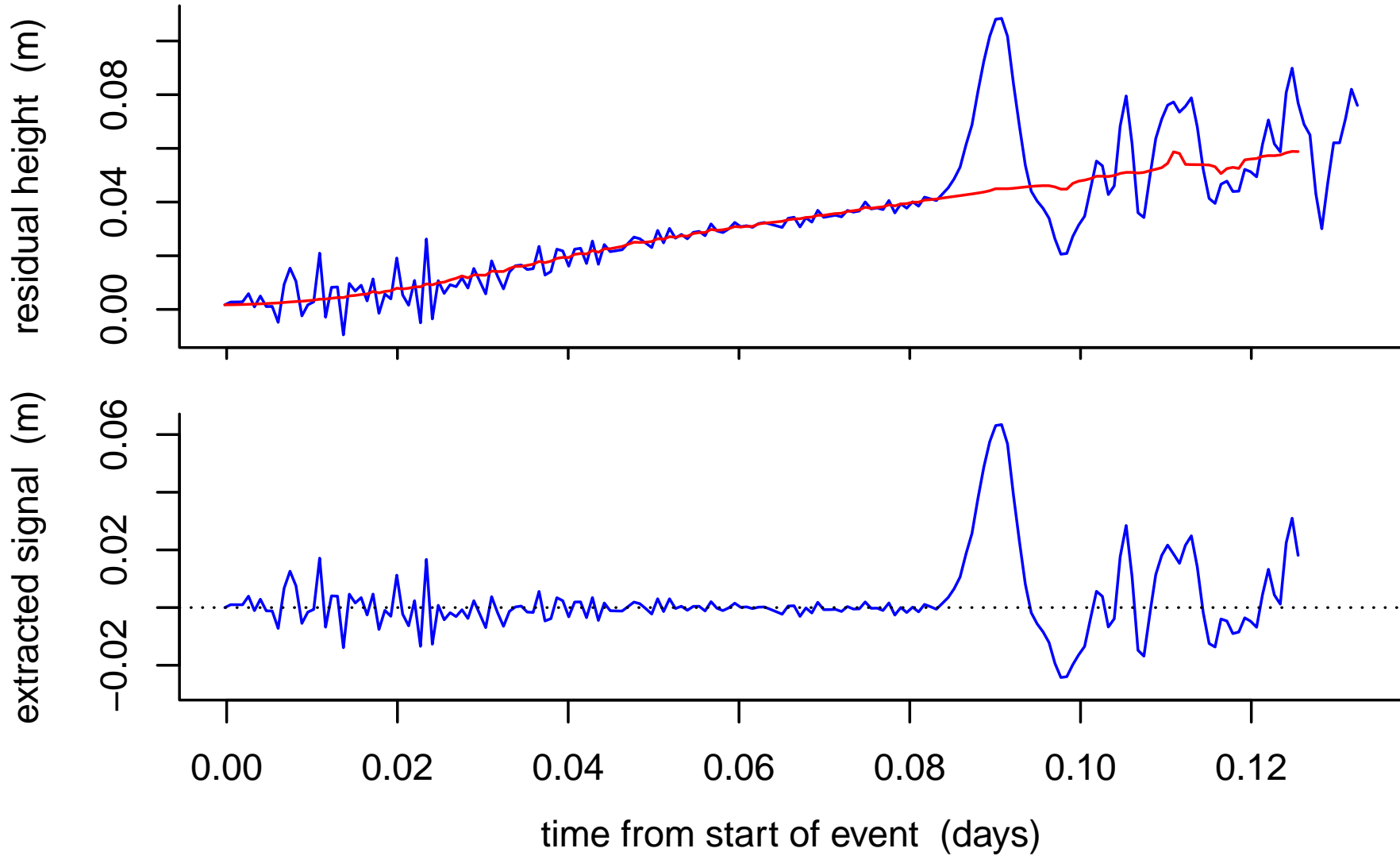
# Kalman Filtering/Smoothing (Local Level Model)



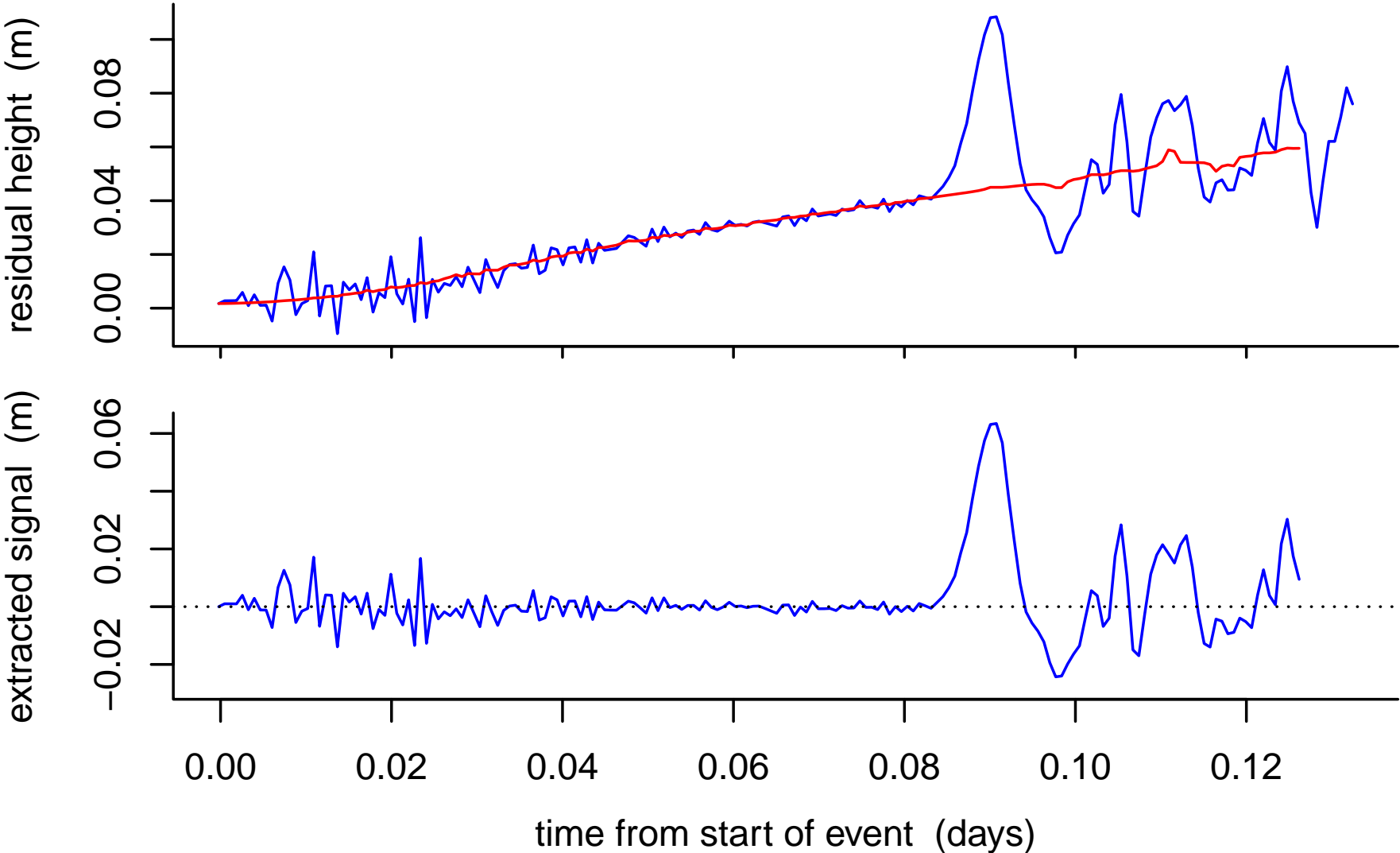
# Kalman Filtering/Smoothing (Local Level Model)



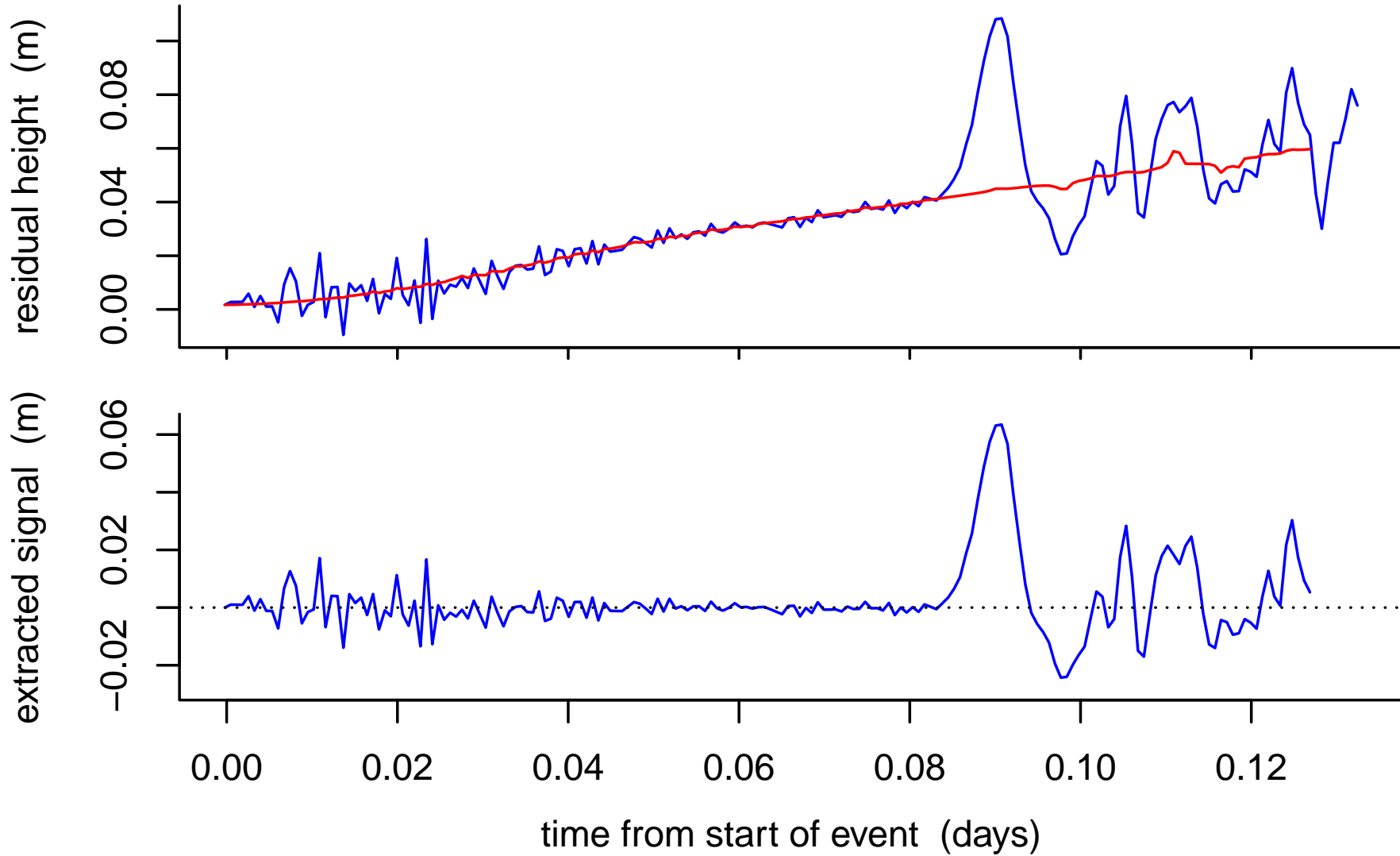
# Kalman Filtering/Smoothing (Local Level Model)



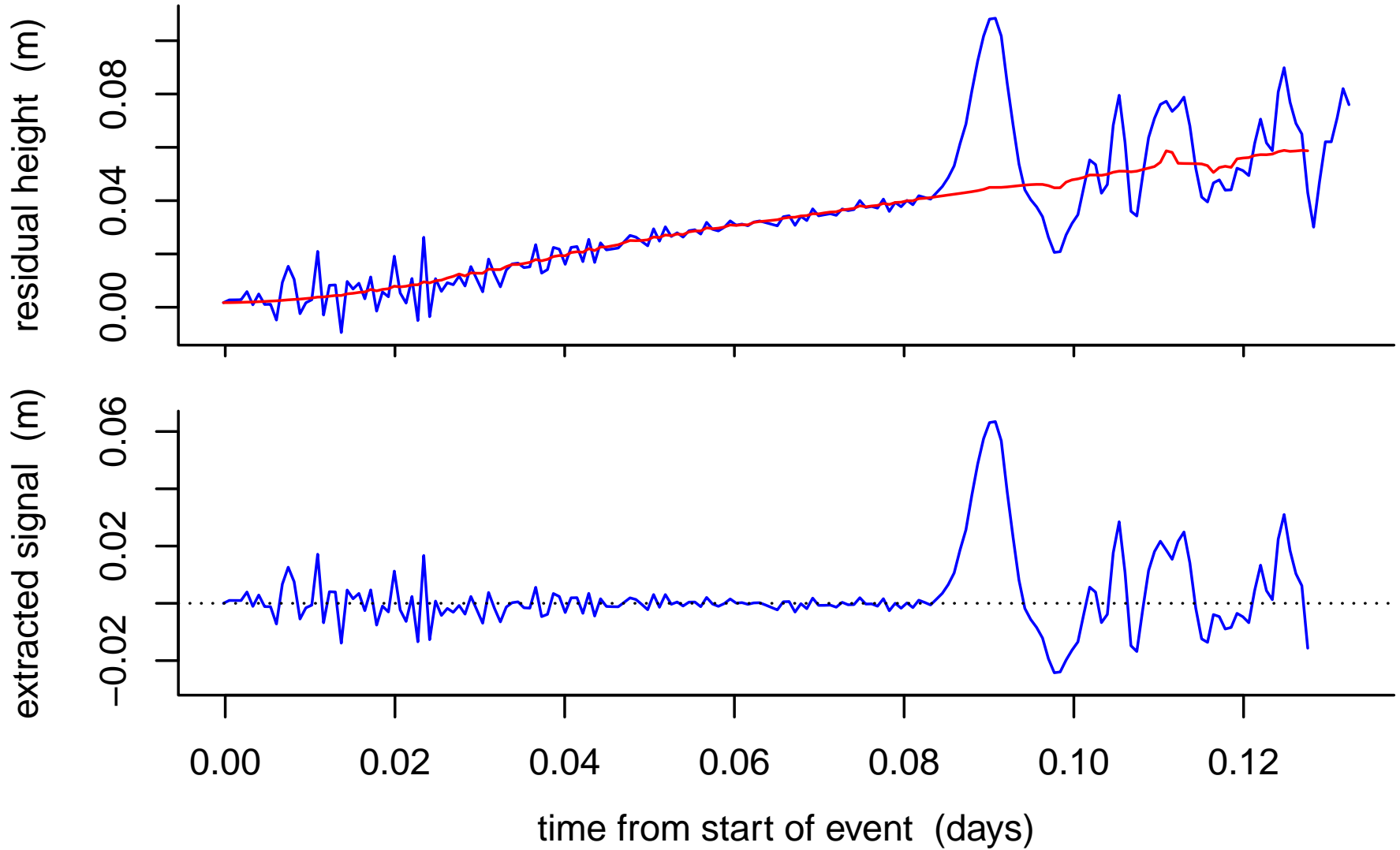
# Kalman Filtering/Smoothing (Local Level Model)



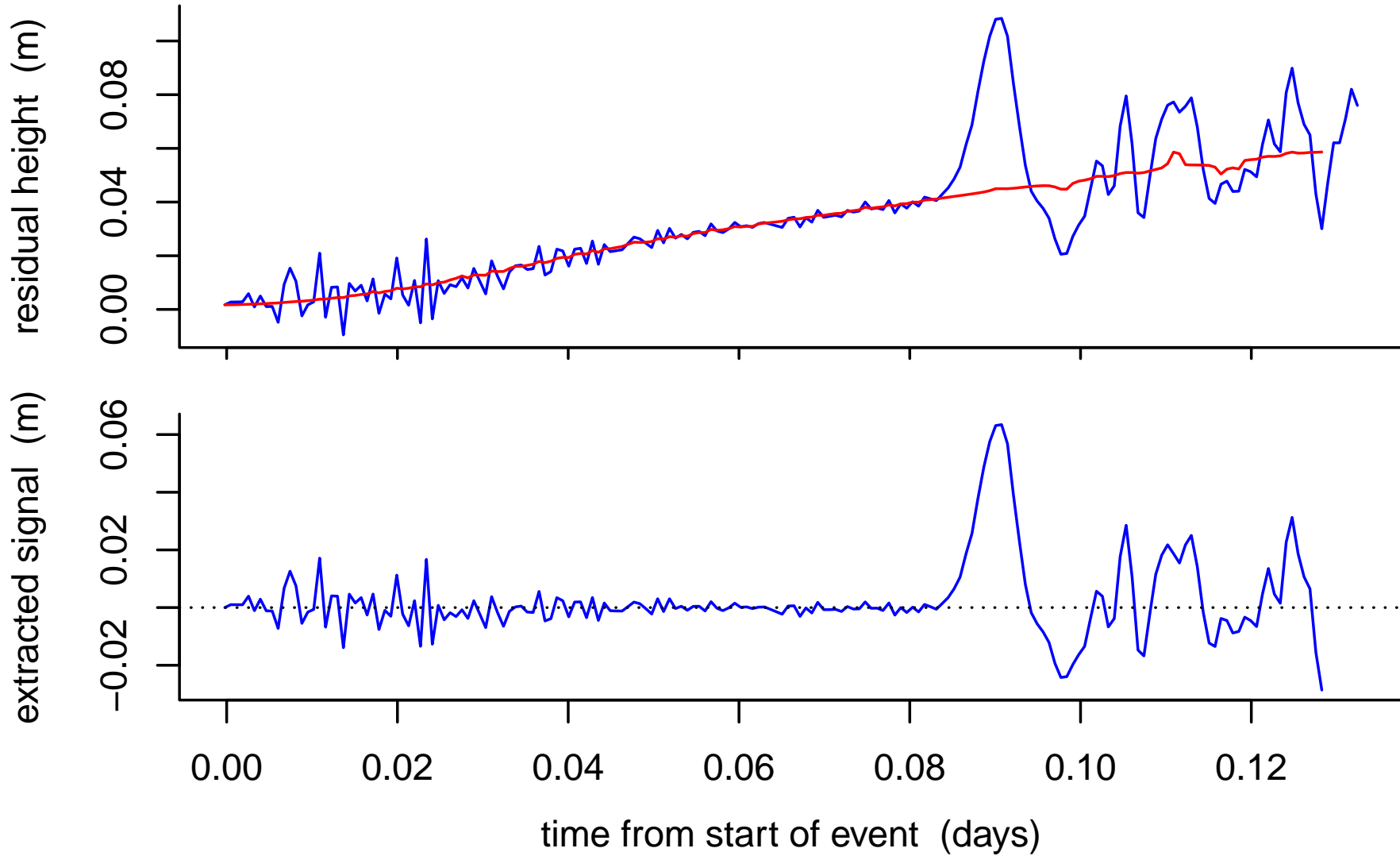
# Kalman Filtering/Smoothing (Local Level Model)



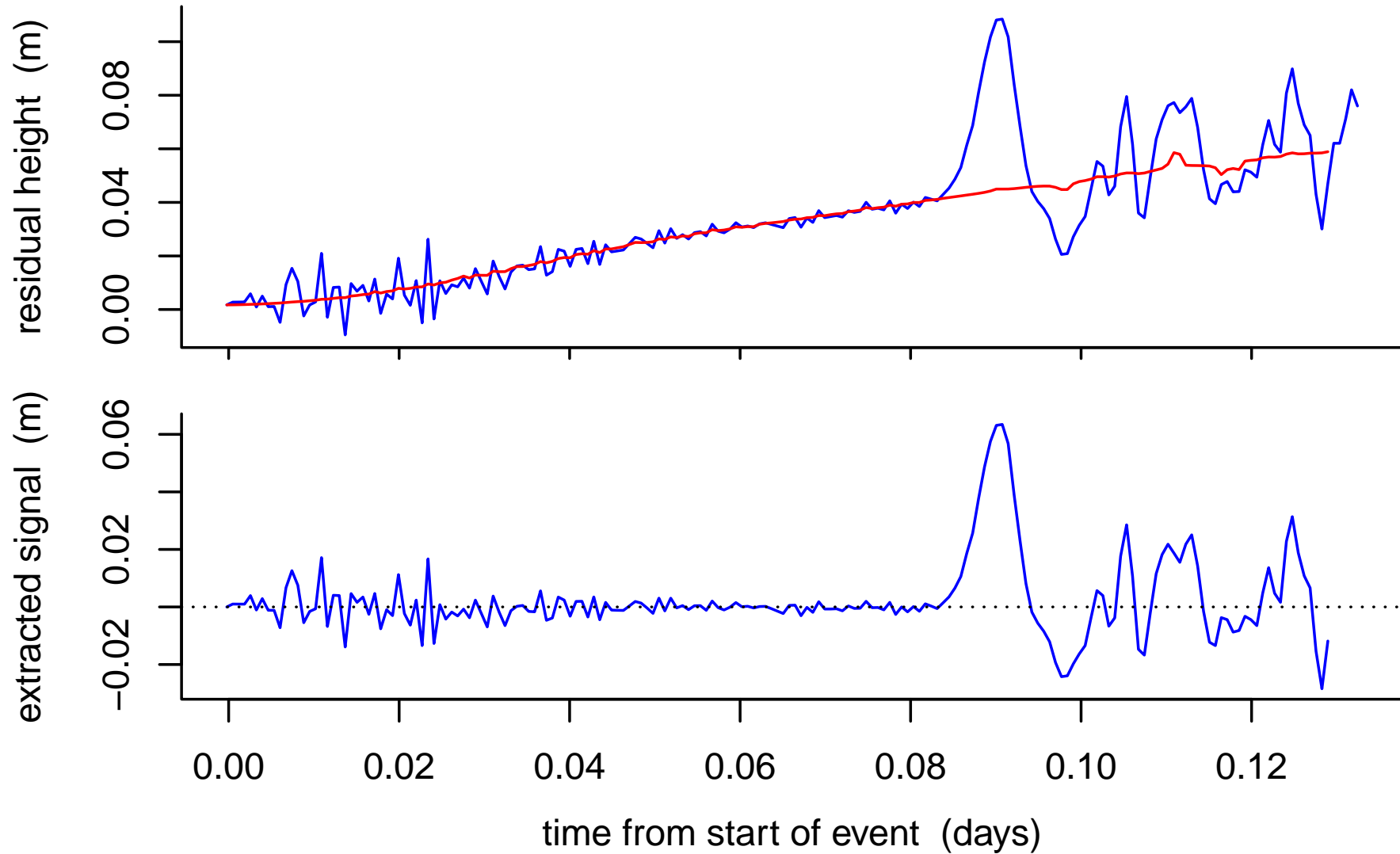
# Kalman Filtering/Smoothing (Local Level Model)



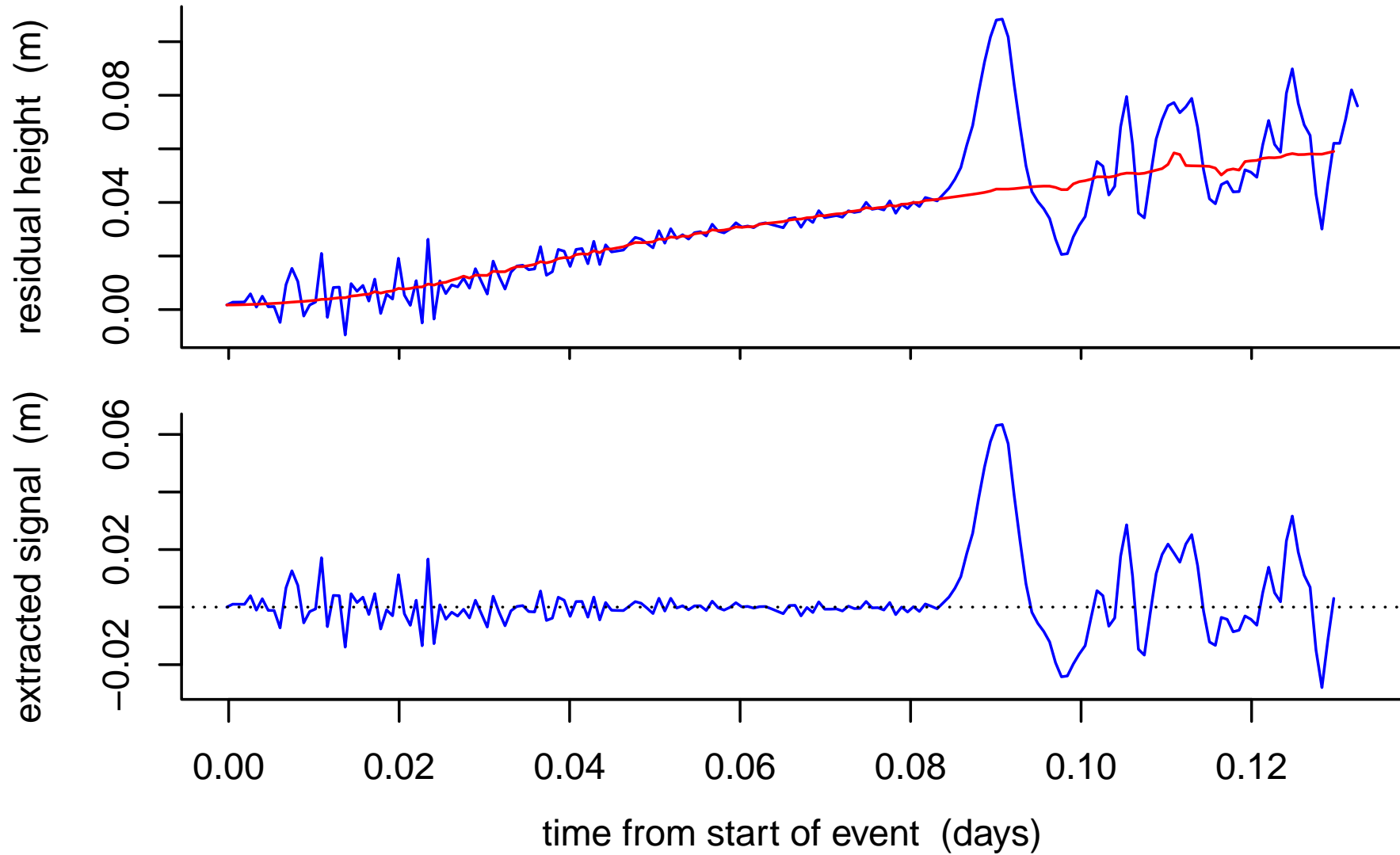
# Kalman Filtering/Smoothing (Local Level Model)



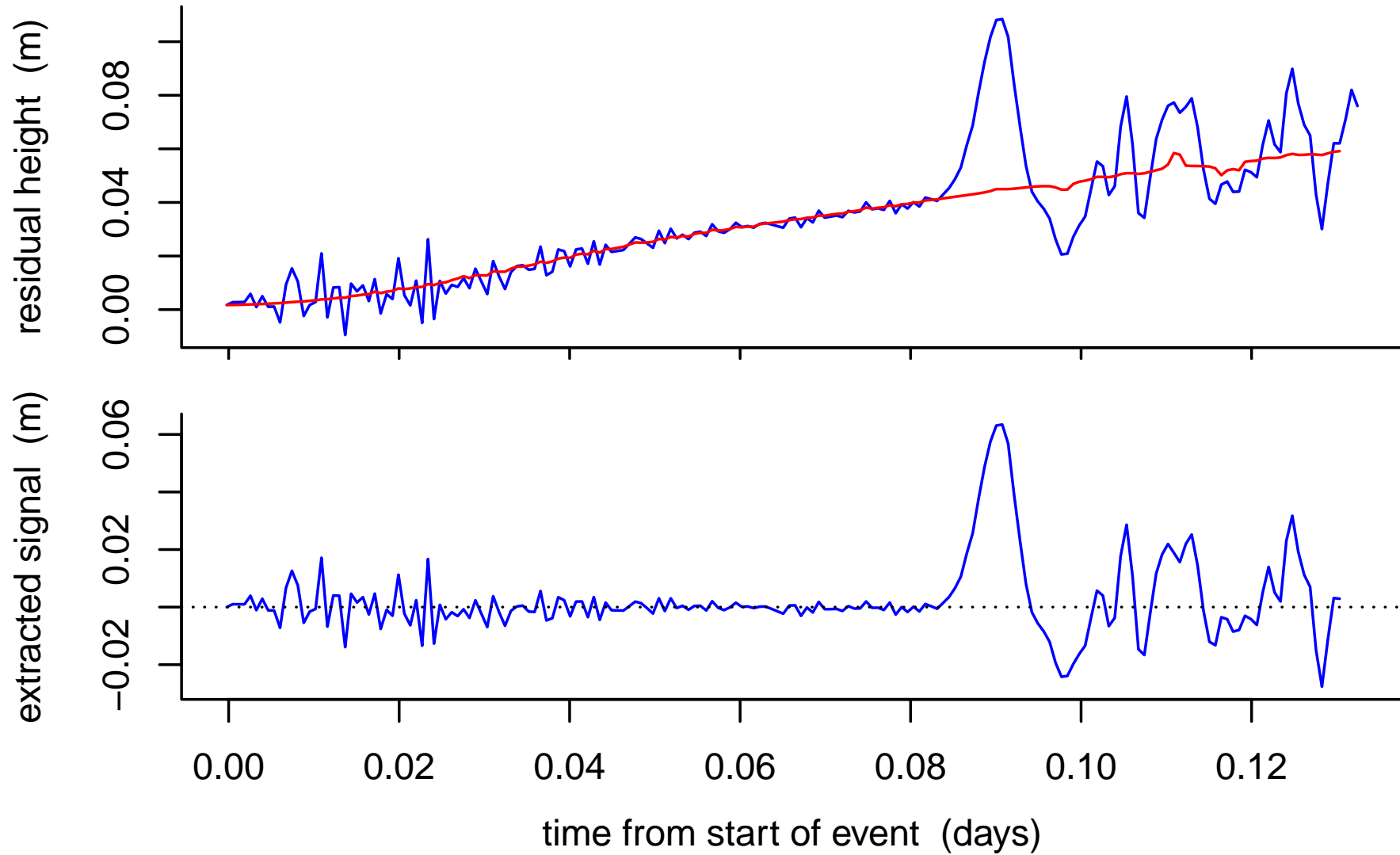
# Kalman Filtering/Smoothing (Local Level Model)



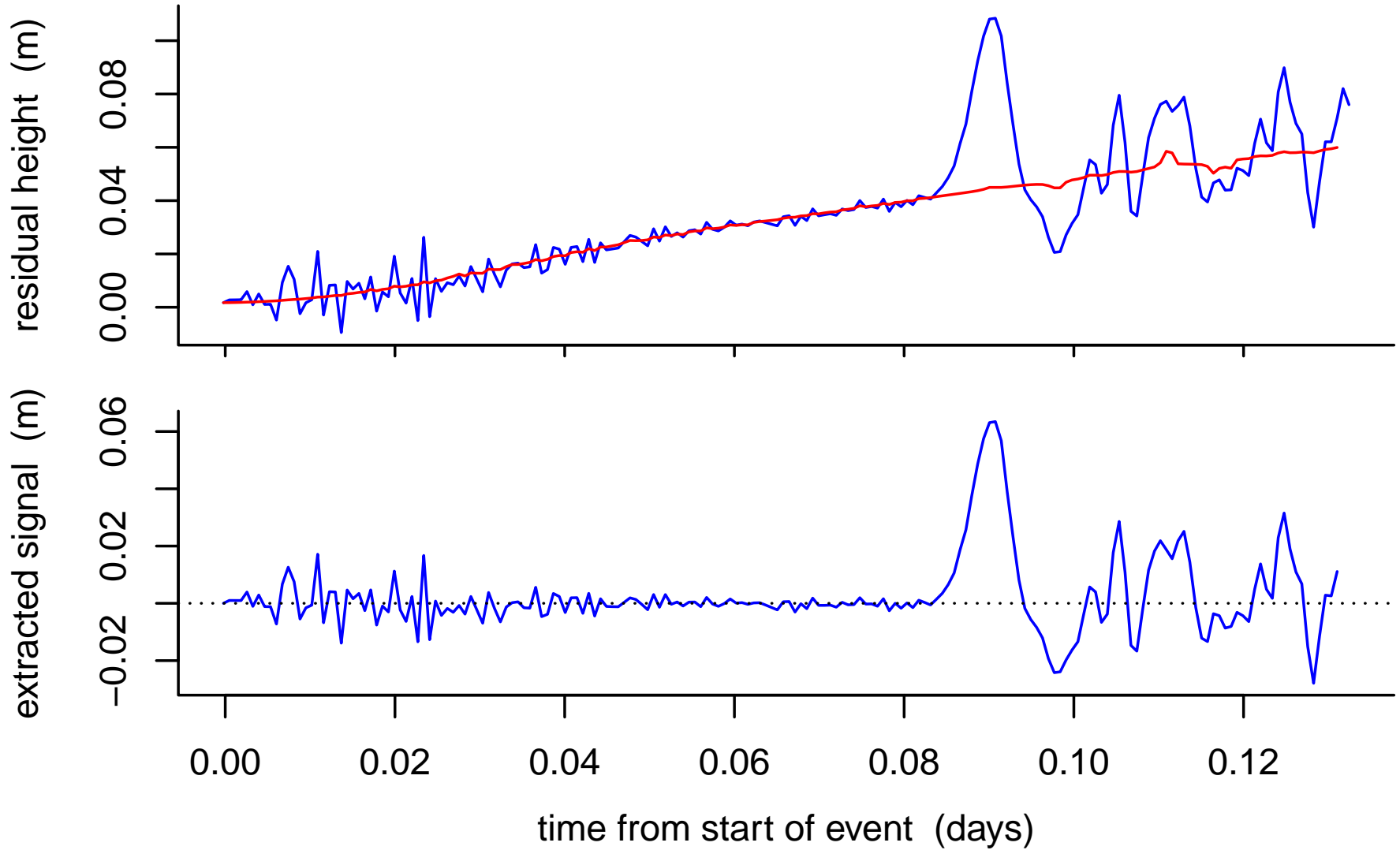
# Kalman Filtering/Smoothing (Local Level Model)



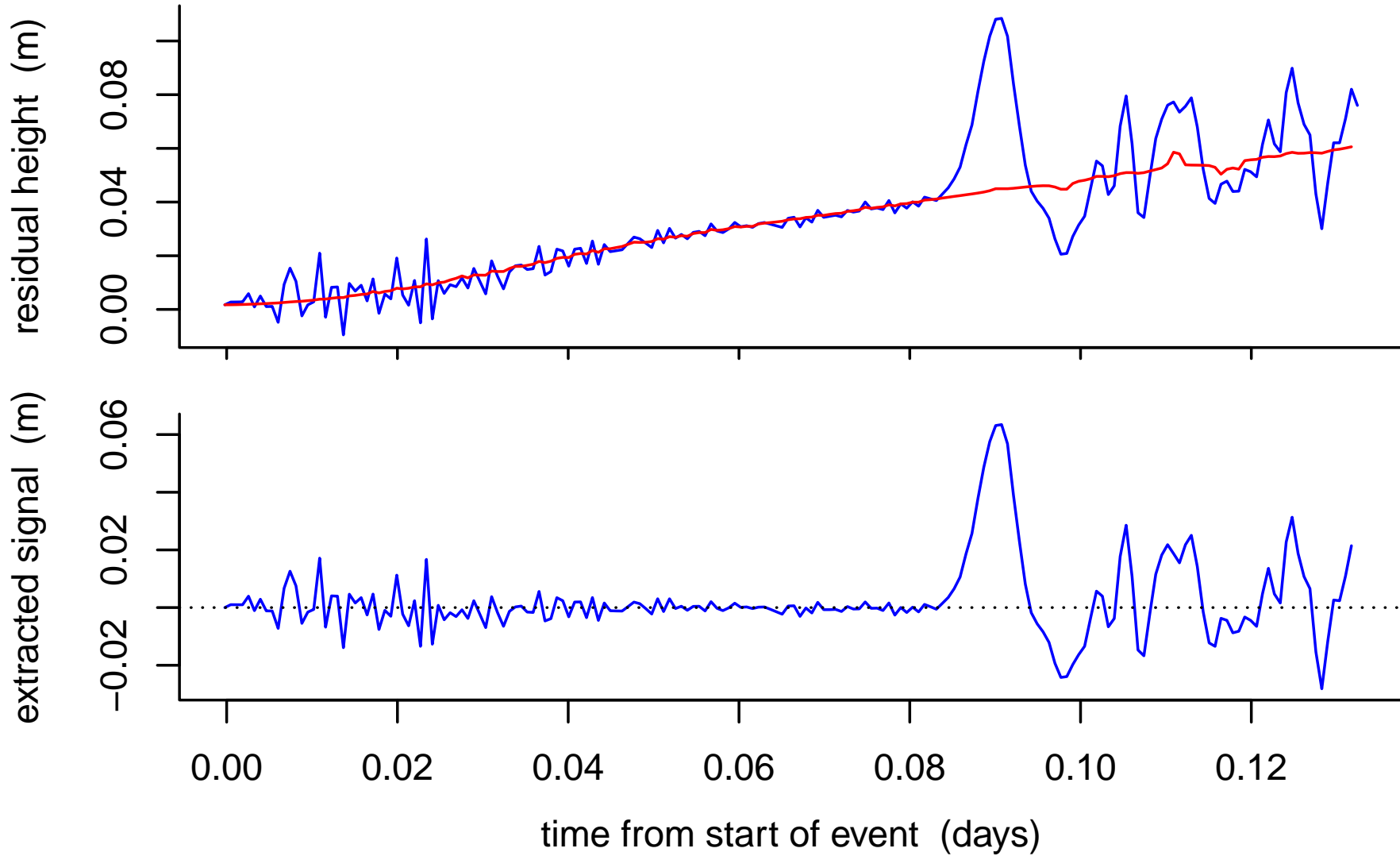
# Kalman Filtering/Smoothing (Local Level Model)



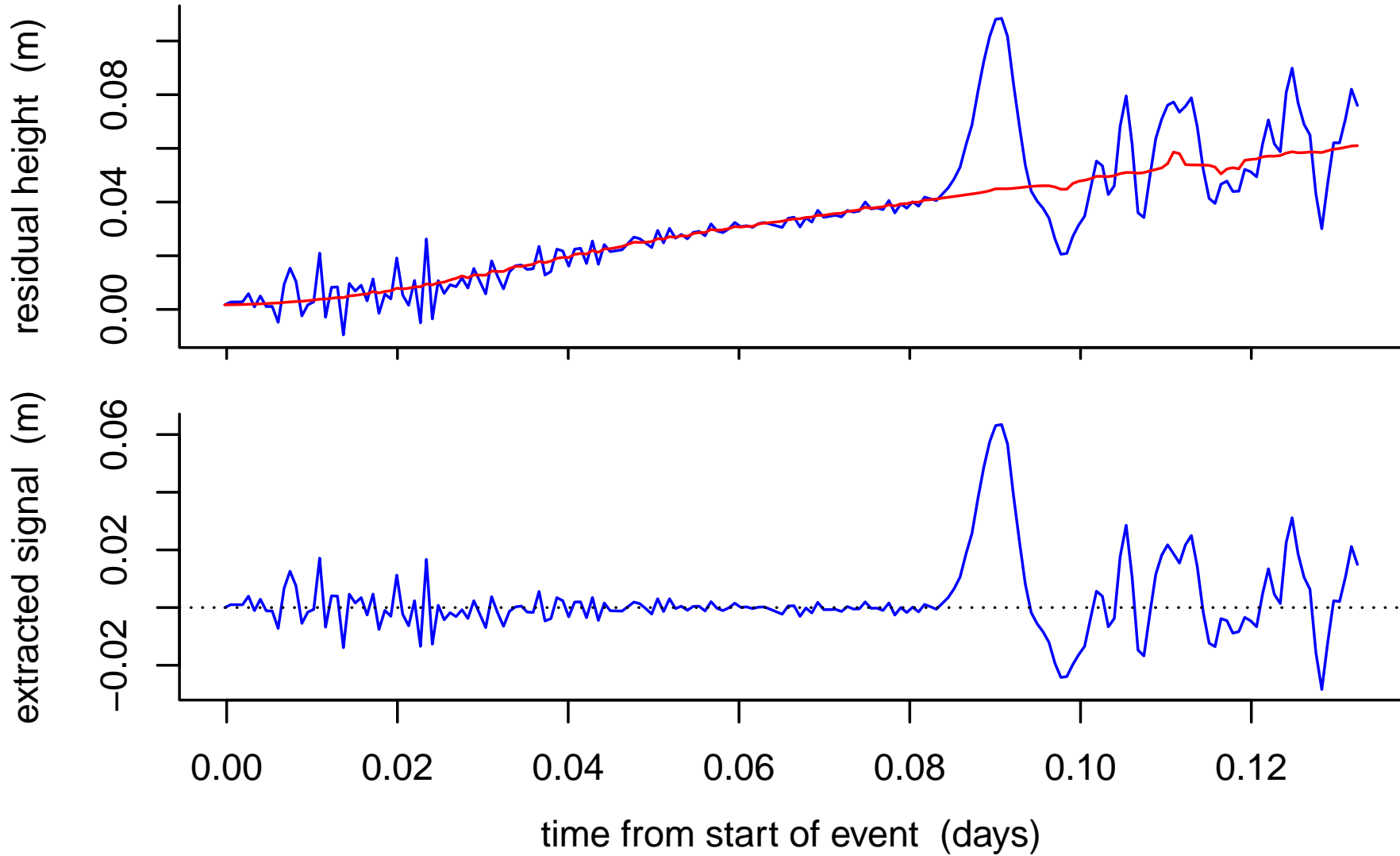
# Kalman Filtering/Smoothing (Local Level Model)



# Kalman Filtering/Smoothing (Local Level Model)



# Kalman Filtering/Smoothing (Local Level Model)



## References

- G. T. Diderrich (1985), ‘The Kalman Filter from the Perspective of Goldberger–Theil Estimators,’ *The American Statistician*, **39**, pp. 193–198
- J. Durbin and S. J. Koopman (2012), *Time Series Analysis by State Space Methods* (Second Edition), Oxford, UK: Oxford University Press
- R. L. Eubank (2006), *A Kalman Filter Primer*, Boca Raton: Chapman & Hall/CRC
- R. E. Kalman (1960), ‘A New Approach to Linear Filtering and Prediction Problems,’ *Journal of Basic Engineering*, **82**, Part D, pp. 35–45
- R. J. Meinhold and N. D. Singpurwalla (1983), ‘Understanding the Kalman Filter,’ *The American Statistician*, **37**, pp. 123–127
- D. B. Percival, D. W. Denbo, M. C. Eblé, E. Gica, P. Y. Huang, H. O. Mofjeld, M. C. Spillane, V. V. Titov and E. I. Tolkova (2015), ‘Detiding DART™ Buoy Data for Operational Tsunami Forecasting,’ *Pure and Applied Geophysics*, DOI 10.1007/s00024-014-0962-0
- G. Petris, S. Petrone and P. Campagnoli (2009), *Dynamic Linear Models with R*, Dordrecht: Springer
- F. Tusell (2011), ‘Kalman Filtering in R,’ *Journal of Statistical Software*, **39**, Issue 2 <http://www.jstatsoft.org/v39/i02>