Data Assimilation into a Primitive-Equation Model with a Parallel Ensemble Kalman Filter

CHRISTIAN L. KEPPENNE

NASA Seasonal to Interannual Prediction Project, Goddard Space Flight Center, Greenbelt, Maryland, and General Sciences Corporation, Beltsville, Maryland

(Manuscript received 18 August 1990, in final form 15 July 1999)

ABSTRACT

Data assimilation experiments are performed using an ensemble Kalman filter (EnKF) implemented for a two-layer spectral shallow water model at triangular truncation T100 representing an abstract planet covered by a strongly stratified fluid. Advantage is taken of the inherent parallelism in the EnKF by running each ensemble member on a different processor of a parallel computer. The Kalman filter update step is parallelized by letting each processor handle the observations from a limited region.

The algorithm is applied to the assimilation of synthetic altimetry data in the context of an imperfect model and known representation-error statistics. The effect of finite ensemble size on the residual errors is investigated and the error estimates obtained with the EnKF are compared to the actual errors.

1. Introduction and motivation

The Kalman filter (Kalman and Bucy 1961) is the statistically optimal sequential-estimation procedure for linear dynamical systems. In a Kalman filter, observations are fed to a numerical model with weights that minimize error variance. The information content of observations is advected from data-rich areas to data-poor areas with the help of an optimally estimated error-covariance matrix. The latter is propagated in time together with the model flow. This provides a major advantage over data asimilation schemes such as optimal interpolation in which the error-covariance distribution is assumed known a priori. However, time stepping the model-error covariance matrix requires two matrix multiplications of order *n*, where *n* is the number of model prognostic variables. This cost is prohibitive for even a simple numerical model with no more than $O(10^4)$ variables. Therefore, reduced-phase-space Kalman filters (e.g., Cane et al. 1996) have been proposed. The ensemble Kalman filter (EnKF), introduced by Evensen (1994), is another alternative to the traditional Kalman filter in which the error covariances are estimated at a cost directly proportional to that of the forward model.

The traditional Kalman filter integrates an approximate equation (exact only in the case of linear systems) for the error-covariance matrix where all third- and higher-order statistical moments are neglected. This closure approximation has been shown not to be sufficient under all circumstances for strongly nonlinear dynamics (e.g., Evensen 1992; Miller et al. 1994). The EnKF avoids this problem by integrating an ensemble of model trajectories from which error-covariance estimates (and thus the gain matrix) can be calculated. This can be understood as an ad hoc method to integrate the Fokker–Planck or Kolmogorov–Chapman equation, which describes completely the evolution of error statistics by using sample trajectories rather than a closure approximation or linearization.

The EnKF has been implemented by Evensen (1994), Evensen and van Leeuwen (1996), and Houtekamer and Mitchell (1998) for quasigeostrophic models with relatively few state-vector variables. This paper is meant as a first step from these implementations toward applications involving high-resolution general circulation models (GCMs) with several-million state variables. It is concerned with the assimilation of synthetic altimetry data into a two-layer spectral shallow water model at triangular truncation T100, with about 120 000 spectral coefficients and 270 000 gridpoint variables. The data come from an integration of the same model at the same resolution but with a different layer configuration. The model imperfections are taken into account by introducing a stochastic forcing term with the same mean and variance as the known process noise. For linear systems, the EnKF is equivalent to the traditional Kalman filter in the limit of infinitely large ensembles. The effect of ensemble size on the filter's performance is examined in this paper.

Corresponding author address: Dr. Christian Keppenne, Oceans and Ice Branch, Code 971, NASA Goddard Space Flight Center, Greenbelt, MD 20771.
E-mail: clk@janus.gsfc.nasa.gov

A description of the parallel EnKF algorithm and model is given in section 2. The data assimilation experiments are discussed in section 3, where the errors predicted by the EnKF in the case of relatively few ensemble members are compared to the true errors. Section 4 contains the conclusions.

2. Methodology

a. The ensemble Kalman filter

The EnKF is discussed in detail in Evensen and van Leeuwen (1996) and in Houtekamer and Mitchell (1998). The presentation that follows focuses on the details of its implementation that differ from the work of Evensen and van Leeuwen.

To best understand how the algorithm works, it is convenient to rewrite the traditional update equation of the Kalman filter (e.g., Bierman 1977),

$$\mathbf{x}^a = \mathbf{x}^f + \mathbf{PH}^{\mathrm{T}}(\mathbf{HPH}^{\mathrm{T}} + \mathbf{R})^{-1}(\mathbf{z} - \mathbf{Hx}), \qquad (1)$$

as (Evensen and van Leeuwen 1996)

$$\mathbf{x}^a = \mathbf{x}^f + \mathbf{B}^{\mathrm{T}}\mathbf{b}, \qquad \mathbf{B} = \mathbf{HP}^{\mathrm{T}}, \qquad (2)$$

where $\mathbf{x}^f$ and $\mathbf{x}^a$ correspond to the forecast and analysis, and the superscript $(\cdot)^{\mathrm{T}}$ stands for matrix transposition. Using n and n_{obs} to denote state-vector size and the number of observations processed during one assimilation cycle, $\mathbf{x}$ ($n \times 1$) is the state vector of a given ensemble member, $\mathbf{P}$ ($n \times n$) is the model-error covariance matrix, $\mathbf{R}$ ($n_{\mathrm{obs}} \times n_{\mathrm{obs}}$) is the error-covariance matrix for the observations, and $\mathbf{H}$ ($n_{\mathrm{obs}} \times n$) is the measurement matrix that relates the data linearly to the model state. The rows of matrix $\mathbf{B}$ ($n_{\mathrm{obs}} \times n$) are the representer vectors and $\mathbf{HPH}^{\mathrm{T}}$ is known as the representer matrix. Vector $\mathbf{b}$ ($n_{\mathrm{obs}} \times 1$) contains the amplitudes of the representers and $\mathbf{z}$ ($n_{\mathrm{obs}} \times 1$) contains the observations. Following Burgers et al. (1998), the data are treated as random variables and $\mathbf{z}$ is replaced by $\mathbf{y} = \mathbf{z} + \boldsymbol{\zeta}$, where $\boldsymbol{\zeta}$ is normally distributed and has the same mean (0) and variance as the observational errors.

To calculate b, the following linear system is solved for each ensemble member in turn:

$$(\mathbf{HPH}^{\mathrm{T}} + \mathbf{R})\mathbf{b} = \mathbf{y} - \mathbf{Hx}^f. \qquad (3)$$

In the case of ocean or atmospheric GCMs, $n \gg n_{\mathrm{obs}}$ and $\mathbf{P}$ is too large to be contained in computer memory. However, $\mathbf{P}$ is never explicitly formed. Instead, we start by computing the $n_{\mathrm{obs}} \times m$ matrix $\mathbf{Q} = \mathbf{H}(\mathbf{X} - \overline{\mathbf{x}})$ where the columns of $\mathbf{X}$ ($n \times m$) each contain the state of one ensemble member and $\overline{\mathbf{x}}$ ($n \times 1$) is the ensemble mean, which is subtracted from every column of $\mathbf{X}$ in forming $\mathbf{Q}$. The calculation of $\mathbf{H}$ is unnecessary in most cases. Rather, $\mathbf{Q}$ can be calculated by taking the difference between every state vector and the mean state vector after interpolation to the location of each observation. A matrix multiplication is thereby avoided.

Once $\mathbf{Q}$ has been computed, the representer matrix is calculated with the only matrix multiplication in the algorithm as $\mathbf{HPH}^{\mathrm{T}} = \mathbf{QQ}^{\mathrm{T}}/(m - 1)$ and the linear equation (3) is solved m times, each time with the $\mathbf{x}^f$ corresponding to a different ensemble member in the right-hand side. Precautions must be taken when $m < n_{\mathrm{obs}}$ and $\mathbf{W} = \mathbf{HPH}^{\mathrm{T}} + \mathbf{R}$ is rank deficient. We have used two different solvers for (3), usually with very similar results. The first and fastest solver is a form of incomplete Cholesky decomposition in which rows of the system matrix, $\mathbf{W}$, on which the algorithm encounters a negative diagonal element are essentially zeroed out. However, this solver gives improper results when $\mathbf{W}$ is too ill-conditioned. The second and more robust solver (LAPACK's SGELSS) relies on singular value decomposition to throw away the singular vectors of $\mathbf{W}$ that correspond to near-zero singular values. Between 0% and 5% of the variance of the elements of $\mathbf{W}$ has typically been discarded when using this solver. Our criterion is to retain as many singular vectors as we can while at the same time making sure that

$$\|\mathbf{H}(\mathbf{x}^a - \mathbf{x}^f)\| \leq \|\mathbf{y} - \mathbf{Hx}^f\|, \qquad (4)$$

where $\|\mathbf{v}\|$ is the L_2 norm, or length, of vector $\mathbf{v}$. After $\mathbf{b}$ has been found one calculates, for each ensemble member in turn, the $m \times 1$ vector, $\mathbf{w} = \mathbf{Q}^{\mathrm{T}}\mathbf{b}$, and then computes the analysis increment as

$$\mathbf{x}^a - \mathbf{x}^f = \mathbf{B}^{\mathrm{T}}\mathbf{b} = \frac{(\mathbf{X} - \overline{\mathbf{x}})\mathbf{w}}{m - 1}, \qquad (5)$$

where $\overline{\mathbf{x}}$ is subtracted from every column of $\mathbf{X}$. The increment thus combines the columns of $\mathbf{X} - \overline{\mathbf{x}}$ linearly with weights $\mathbf{u}/(m - 1)$.

b. Parallelization approach

The EnKF's error-covariance-forecasting step can be parallelized in the most straightforward manner by running each ensemble member on a separate processor of a parallel computer, provided there is enough memory for this operation (Fig. 1a). This approach is most suitable when dealing with a serial model designed, for instance, to run on a traditional vector supercomputer. However, the analysis involves the entire state vector of every ensemble member. Thus, without some sort of domain decomposition for the analysis, one runs into difficulties on a typical parallel computer with distributed-memory architecture, where no processor has enough memory to contain the entire ensemble. Therefore, if each model copy resides on a different processor, the ensemble matrix, $\mathbf{X}$, which is distributed one column per processor during the forecast step, must be transposed on the processor lattice so that each processor ends up with a piece of the state vector of every ensemble member (Fig. 1b). The analysis can then proceed regionally on each processor. A second transposition must take place to return to the configuration of Fig. 1a, where the regional analysis increments computed on every processor are assembled into global increments.

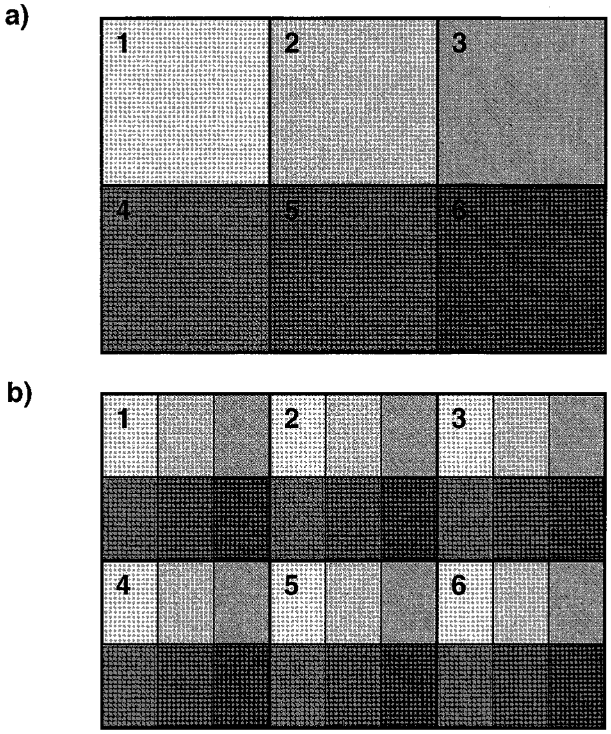

FIG. 1. (a) Domain decomposition used to propagate the ensemble. Each shade of gray represents the state vector of a different ensemble member. Each numbered square represents a processor. The ensemble members (six in this example) are distributed one per processor. (b) Domain decomposition used for the analysis. The memory of each processor contains the same state-vector elements from every ensemble member.

Another argument for doing the analysis on a regional basis is given by Houtekamer and Mitchell (1998). They find that, unless very large ensembles are used, it is best to do the analysis regionally and introduce a cutoff radius beyond which covariances are ignored. Their Fig. 5 shows little improvement in performance for ensemble sizes of about 100 when the cutoff radius increases beyond 20 degrees. The domain decomposition used here for the analysis essentially achieves the same result.

Since interprocessor communications are generally expensive, it is preferable to avoid transposing **X** twice by using a domain decomposition such as that of Figure 1b to (a) run the model and (b) perform the analysis. In this case, each processor must evolve the state-vector elements of every ensemble member that correspond to the area of Fig. 1b for which that processor does the analysis. However, this requires using a prealably distributed model and thus could only be applied to most GCMs presently in use after a major model parallelization effort.

Although the model used in this study does not have the complexity of a typical GCM, it relies on the spectral transform method (Orzsag 1970) to switch between the spectral and gridpoint representations of the state vector. Fast Fourier transforms (FFTs) and Legendre transforms (LTs) are used to this end. Thus, to efficiently parallelize this kind of spectral model, parallel FFTs and LTs are required. Although the former are now available on most parallel architectures, the latter are not. The domain decomposition of Fig. 1a is thus used to evolve the ensemble and that of Fig. 1b for the analysis. A CRAY T3E with 128 MB (32 million 4-byte words) of memory per processor is used. This is amply sufficient to run each model copy on a separate processor.

The algorithm is written in C++ and relies on the Message Passing Interface (MPI; Message Passing Interface Forum 1994) for the communications between processors. Prior to the analysis, each processor gathers from every other processor the state-vector elements it requires with a call to MPI_Gather(). Each processor then processes with (1)–(3) the observations that are local to it in the decomposition of Fig. 1b resulting in a set of **w** coefficient vectors, one for each ensemble member. Each processor then sends to every other processor the **w** that corresponds to the ensemble member that resides in the memory of that processor in the decomposition of Fig. 1a, using a call to MPI_Scatter(). Following this, each processor combines the coefficient vectors it has received from the other processors. In doing so, the influence from nearby processors is weighted more heavily than that from processors farther away:

$$\mathbf{w}_i = \frac{\sum_j \gamma_{ij}\mathbf{w}_{ij}}{\sum_j \gamma_{ij}}, \tag{6}$$

$$\gamma_{ij} = \exp\left\{-\frac{1}{d}[(\theta_i - \theta_j)^2 + (\lambda_i - \lambda_j)^2]^{1/2}\right\}. \tag{7}$$

In (6), $\mathbf{w}_{ij}$ is the coefficient vector calculated by processor j for the ensemble member that resides in the memory of processor i, and γ_{ij} is the corresponding weight that is inversely proportional to the distance between (θ_i, λ_i) and (θ_j, λ_j), the central points of the regions of processors i and j in the decomposition of Fig. 1b. The parameter d is the side of each region in radians and w_i is the final coefficient vector on processor i. The analysis increment for the ith ensemble member is then obtained using (5) with **w** replaced by $\mathbf{w}_i$.

c. *The model*

The model is adapted from the one used by Keppenne et al. (2000) to investigate orographicaly forced intraseasonal atmospheric oscillations. It represents an abstract planet covered by a stratified fluid interacting with a fluid bottom topography representative of a deep isentropic layer in steady-state motion. In this sense, the system bears some resemblance with the atmospheres of Jovian planets.

A similar model is derived and discussed in detail by Haltiner and Williams (1980, pp. 54–59). It is described by the following dimensionless system of prognostic equations:

$$\frac{\partial}{\partial t}\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\psi_j = \boldsymbol{\nabla}\cdot[\mathbf{u}_j(\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\psi_j + f)] - \tau\delta_{j2}U_2\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\psi_j + (-1)^{n+1}\gamma(\boldsymbol{\nabla}\cdot\boldsymbol{\nabla})^n\psi_j, \qquad (8)$$

$$\frac{\partial}{\partial t}\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\chi_j = \mathbf{k}\cdot\boldsymbol{\nabla}\times[\mathbf{u}_j(\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\psi_j + f)] - \tau\delta_{j2}U_2\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\chi_j + (-1)^{n+1}\gamma(\boldsymbol{\nabla}\cdot\boldsymbol{\nabla})^n\chi_j - \boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\left[\frac{\mathbf{u}_j\cdot\mathbf{u}_j}{2} + \phi_2 + \alpha_j\phi_1 + \beta_j\phi_b\right], \qquad (9)$$

$$\frac{\partial}{\partial t}\phi_j = -\boldsymbol{\nabla}\cdot(\mathbf{u}_j\phi_j) - \Phi_j\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\chi_j - \boldsymbol{\epsilon}\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}(\phi_j - \phi_j^*). \qquad (10)$$

In (8)–(10) the subscript j takes the values 1 and 2 for the upper and lower layers, respectively. Equations (8) and (9) are the vorticity-divergence form of the momentum equation and (10) is the mass-conservation equation. For each value of j, $\psi(\lambda, \theta, t)$, $\chi(\lambda, \theta, t)$, and $\phi(\lambda, \theta, t)$ are the corresponding layer's streamfunction, velocity potential and the deviation of the geopotential height from its global average, Φ_j. Here, λ is longitude, θ latitude and t time. Time is rendered dimensionless by measuring it in units equal to the planetary rotation period, $2\pi/\Omega$, while the dimensional length unit is the planetary radius. The horizontal fluid velocity in each layer is $\mathbf{u}_j(\lambda, \theta, t)$. The Coriolis parameter is $f = 2\Omega \sin\theta$, τ is a drag coefficient, α is the ratio of each layer's density to that of the lower layer ($\alpha_1 = \rho_1/\rho_2$, $\alpha_2 = 1$), and β is the ratio of layer density to that of the bottom topography ($\beta_1 = \rho_1/\rho_b$, $\beta_2 = \rho_2/\rho_b$). The latter is 0 in the case of a solid topography and in]0–1[when the bottom topography is fluid. The parameter, γ, is a diffusion coefficient, U the layer's root-mean-square (rms) zonal kinetic energy per unit mass, ϕ_b is topographic height, $\boldsymbol{\nabla}\cdot$ is the horizontal divergence operator, $\boldsymbol{\nabla}\times$ is the horizontal curl operator, and $\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}$ the horizontal Laplacian operator. The forcing field is idealized and depends only on latitude. Hence, the relaxation term, $\boldsymbol{\epsilon}\boldsymbol{\nabla}\cdot\boldsymbol{\nabla}(\phi_j - \phi_j^*)$, is applied exclusively to the zonally symmetric component of each layer's geopotential field.

The system (8)–(10) is expanded in terms of spherical harmonics (e.g., Hochstadt 1971) at triangular truncation T100, using an associated Gaussian grid with 150 latitudes and 300 meridians. As mentioned in the preceding section, the transform method (Orzsag 1970; Bourke 1972) is used to iterate back and forth between the spectral and gridpoint representations of model variables. The spectral representation of (8)–(10) is integrated with a semi-implicit leapfrog scheme with time step equal to 0.05 in dimensionless units, that is, 1/20th of the planetary rotation period. An Asselin (1972) filter with coefficient 0.025 is applied to damp the spectrum of internal and external gravity waves. The first time step after each analysis is a modified Euler backward step (Kurihara 1965).

As alluded to in the introduction, the version of the model used to generate the observations (hereafter control model) differs from the version into which the data are assimilated (hereafter analysis model). In the control model, $\alpha_1 = 0.9$, $\beta_1 = 0.81$, and $\beta_2 = 0.9$. In the analysis model, $\alpha_1 = 0.5$, $\beta_1 = 0.25$, and $\beta_2 = 0.5$. The analysis model's layer configuration is consequently more stable (baroclinically) than that of the control model. Figures 2 and 3 illustrate the differences between the two models.

Figure 2 shows time series of spatial-mean angular momentum (AM: $u \cos\theta$) in the upper and lower layers of the control (panels a and b) and analysis (panels c and d) models. It illustrates that the control model's variability on the short timescales that are of concern here is significantly higher than that of the analysis model. Indeed, the ratio of the standard deviation of the signal of Fig. 1c to that of Fig. 1a is 0.67. The corresponding ratio for the signals of Figs. 1d and 1b is 0.64.

Figure 3 shows maximum entropy method (Burg 1967) AM power spectra in both layers of both models. Fifty poles in the complex plane are used to compute the spectra. The AM spectra for both layers of the control model are dominated by one peak near 85 days and two smaller peaks near 35 and 60 days. For the analysis model, a peak near 55 days dominates the AM power spectra of both layers. Although the spectral features become less pronounced when less than 20 poles are used and new spurious peaks appear with more than 100 poles, these spectra are quite robust with respect to variations in the number of poles between these two limits. In each panel of Fig. 3, the dashed and solid lines correspond to the spectra of the first and seconds halves of a 1000-day AM time series and illustrate that the spectra are stationary. Thus, the differences between Figs. 3a and 3b for the control model and Figs. 3c and 3d for the analysis model are not artifacts caused by the time series being too short. They are a manifestation of what is commonly referred to as system noise: errors and approximations made in formulating the model or errors in the specification of the forcing fields. How this system noise is accounted for in the data assimilation experiments is discussed in the next section.

d. The data assimilation experiments

The observations are gridded total-height [$\phi_1(\lambda, \theta, t) + \phi_2(\lambda, \theta, t)$] fields sampled daily at every 16th grid point during the last 30 days of a 1000-day integration of the control model, for a total of 2775 measurements for each assimilation cycle. At T100 resolution, the Gaussian latitudes are approximately evenly spaced. Therefore, the data locations resemble a typical finite-difference grid with even meridional and longitudinal spacings. To simulate the uncertainty of meteorological

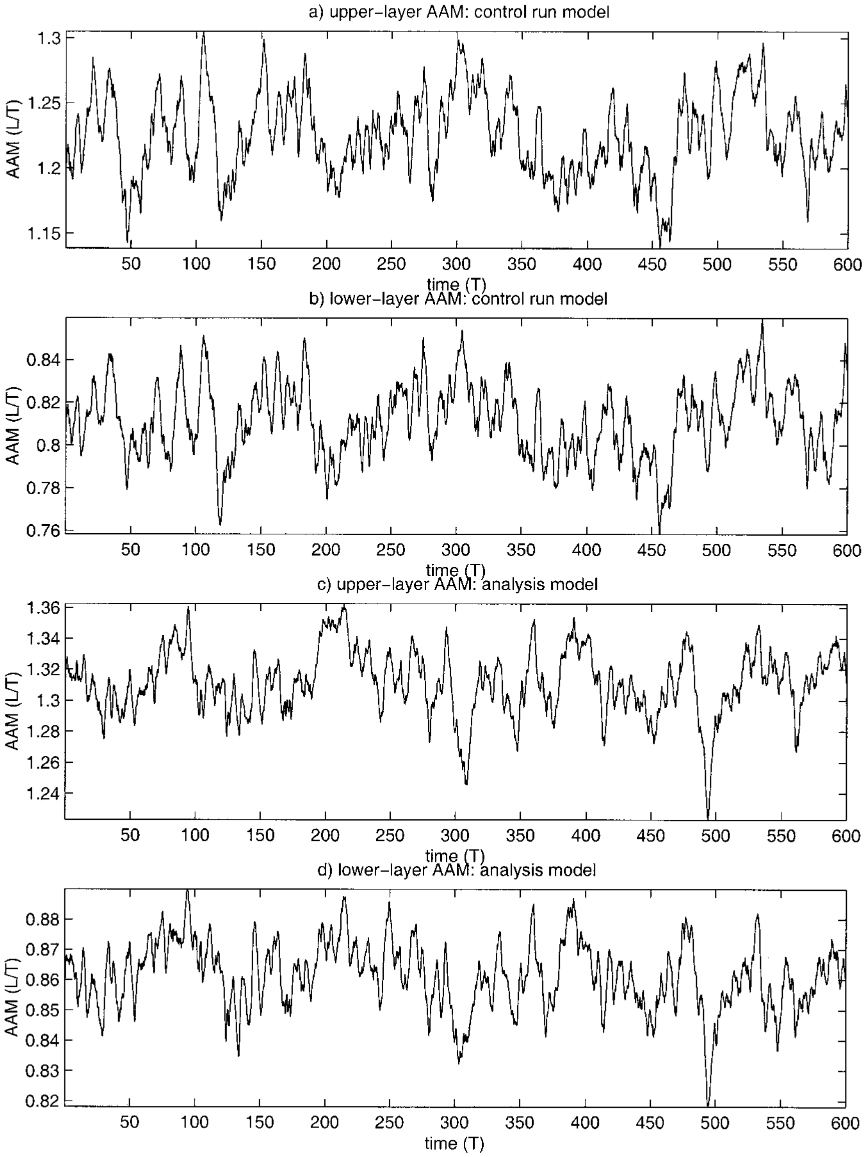

FIG. 2. Time series of horizontal-mean angular momentum in the upper and lower layers of the (a), (b) control and (c), (d) analysis models. Dimensionless units are used.

and oceanographic observations, white noise with a standard deviation equal to the horizontal-mean standard deviation of the control model's total height field is added to the observations.

The initial ensemble configuration before any data are assimilated is such that the ensemble mean is close to the analysis model's climatology. To arrive at this configuration, the ensemble is integrated for a sufficiently long time, with a stochastic forcing term representing the process noise added to the right-hand sides of (8)–(10), for the state vectors of all ensemble members to become uncorrelated.

The term representing the process noise is obtained as follows. Let the control and analysis models be, respectively, represented by

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}^c[\mathbf{x}(t)] \quad \text{and} \tag{11}$$

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}^a[\mathbf{x}(t)] = \mathbf{f}^c[\mathbf{x}(t)] - \boldsymbol{\Theta}(t), \tag{12}$$

where $\boldsymbol{\Theta}(t)$ is the process noise. Since the truth (control) is known, the first- and second-order moments of $\boldsymbol{\Theta}$ can be obtained by computing a large number, l, of realizations of $\mathbf{f}^c(\mathbf{x}_k)$ and $\mathbf{f}^a(\mathbf{x}_k)$, for given $\mathbf{x}_k$, and calculating the mean, $\overline{\boldsymbol{\Theta}}$, and variance, $\boldsymbol{\sigma}^2_{\boldsymbol{\Theta}}$, of the set $\{\mathbf{f}^c(\mathbf{x}_k) - \mathbf{f}^a(\mathbf{x}_k),\ k = 1, \ldots, l\}$. White noise with mean $\overline{\boldsymbol{\Theta}}$ and variance $\boldsymbol{\sigma}^2_{\boldsymbol{\Theta}}$ is then included in the calculation of the time derivatives for every ensemble member. To ensure that the resulting forcing fields are horizontally contin-

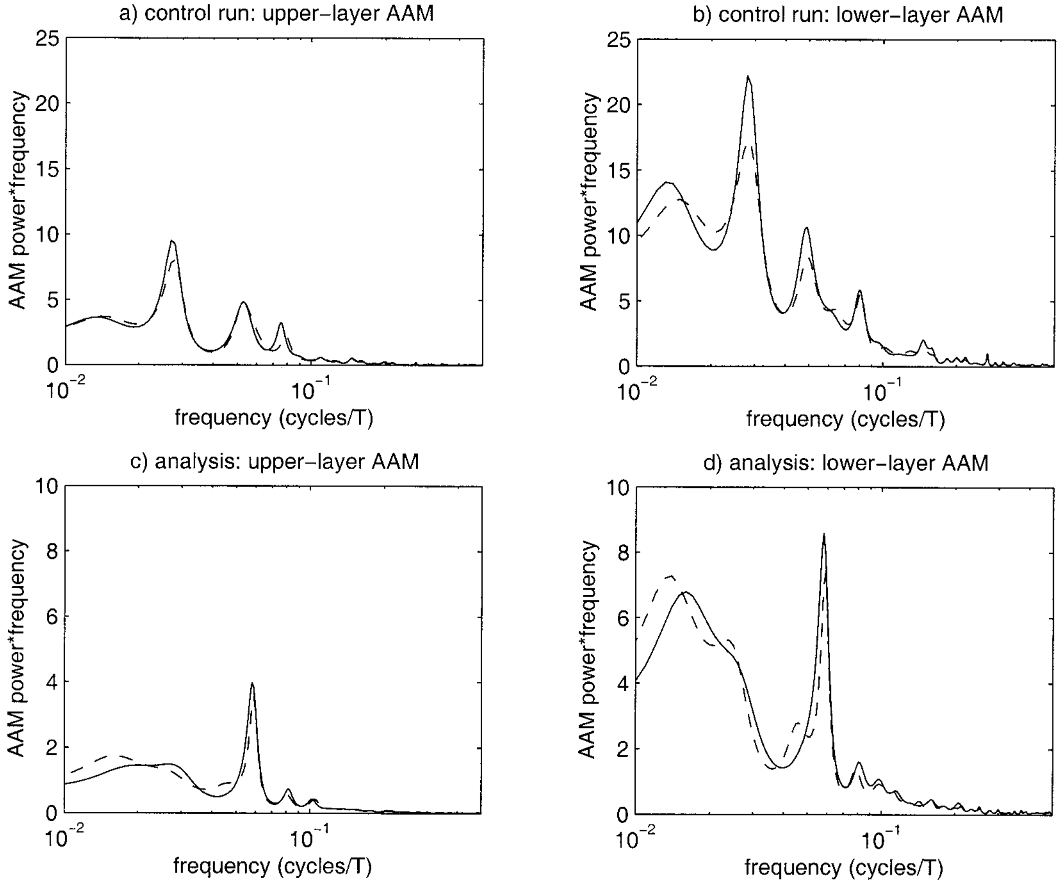

FIG. 3. Maximum entropy power spectra of horizontal-mean angular momentum in the upper and lower layers of the (a), (b) control and (c), (d) analysis models. In each panel, the solid and dashed lines correspond to the first and second halves of a 1000-day time series. The frequency axes are labeled in units of cycles per day.

uous, the process noise term is applied to the spectral, spherical harmonic representation of (8)–(10), rather than in gridpoint space.

Figure 4 shows how the standard deviation of $\boldsymbol{\Theta}$, $\boldsymbol{\sigma}_{\boldsymbol{\Theta}}$, compares to that of $\mathbf{f}^c$, $\boldsymbol{\sigma}_{\mathbf{f}^c}$. The solid line in each panel shows the meridional distribution of $\boldsymbol{\sigma}_{\mathbf{f}^c}$ for a given model field. The dashed line corresponds to $\boldsymbol{\sigma}_{\boldsymbol{\Theta}}$ for the same field. The standard deviations shown are rms deviations averaged over time and and longitude.

Each analysis cycle proceeds as follows. The ensemble is integrated for one day from its configuration, $\mathbf{X}^a(t_{i-1})$, after the previous analysis at time t_{i-1}, to yield a new ensemble configuration, $\mathbf{X}^f(t_i)$, and the observations are processed as discussed in the preceeding sections to arrive at $\mathbf{X}_a(t_i)$. The mean ensemble state vector at this point in time, $\overline{\mathbf{x}}^a(t_i)$, estimates the true state at time t_i. The rms spread between ensemble members etimates the rms error. The new configuration, $\mathbf{X}^a(t_i)$, is the starting point for the next analysis cycle.

The ensemble size is varied between 20 and 140, in increments of 20, and the forecast and analysis errors are compared to the errors from a climatological forecast made with the climatology of the control model. Since the control and analysis models have differing dynamics (Figs. 2–4), the control model's climatology is a more rigorous benchmark to evaluate the performance of the algorithm against than the mean of a no-assimilation ensemble forecast with the analysis model.

With 2775 observations per assimilation cycle, the algorithm becomes unnecessarily expensive with larger ensembles than are considered here, as the local $\mathbf{W}$ matrix on each processor becomes so small that most of the time is spent communicating between processors rather than calculating analysis increments. Fortunately, about 100 ensemble members are enough for the analysis errors to converge, as will be seen in the next section.

3. Results

Figure 5 corresponds to the experiment with 100 ensemble members. It shows the total-height observations 30 days into the experiment, that is, at the time of the last analysis (Fig. 5a). Also shown is the true total-height field from the control model at 30 days (Fig. 5b) and the true height of the layer interface (Fig. 5c). The

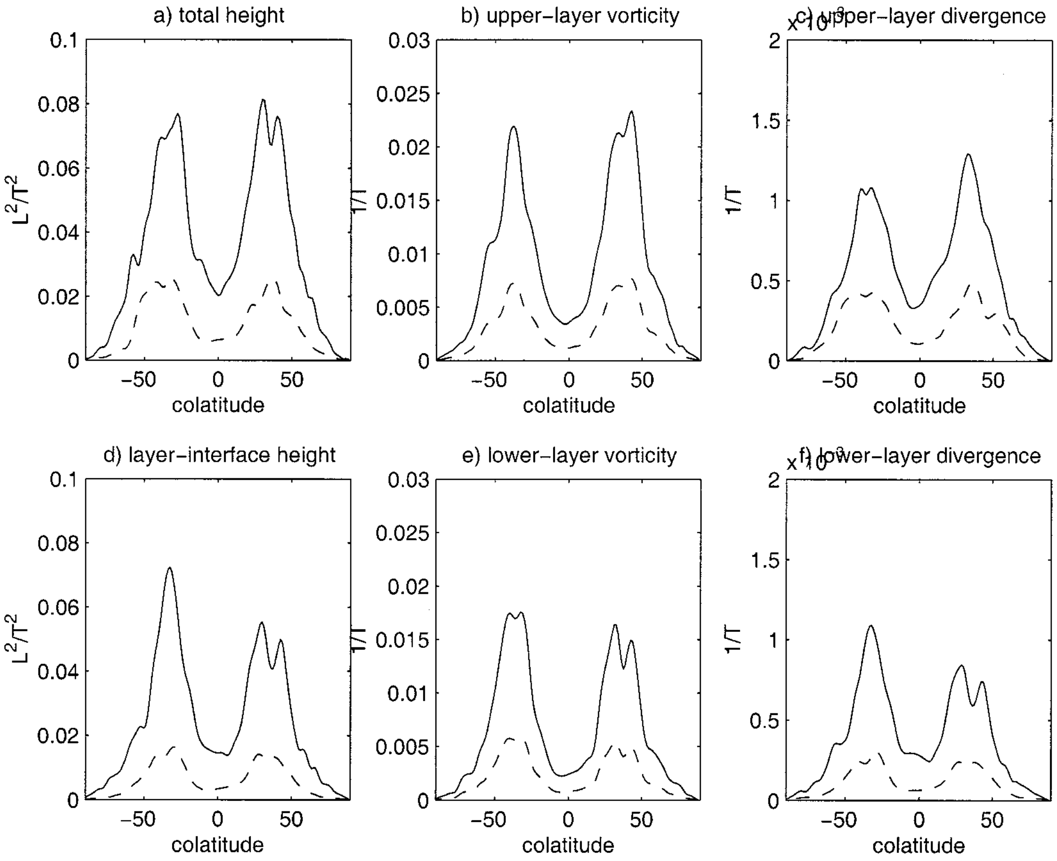

FIG. 4. Meridional distribution of σ_{f_c} (solid) and σ_Θ (dashed) for (a) the total height, (b) the upper-layer vorticity, (c) the upper-layer divergence, (d) the layer-interface height, (e) the lower-layer vorticity, and (f) the lower-layer divergence. Dimensionless units are used.

total-height and layer-interface-height forecasts are shown in Figs. 5d and 5e, respectively. The corresponding analyses are shown in Figs. 5f and 5g. Careful examination of Figs. 5b, 5d, and 5f confirms that although the total-height forecast already reproduces many features of the true total height, the analysis improves further on the forecast estimate. Indeed, the correlation coefficient between the true total height and the total-height forecast is 0.81. The correlation between the true total height and the total-height analysis is 0.97. The correlations between the true layer-interface height of Fig. 1c and the corresponding forecast in Fig. 5e is 0.76. Between the true layer-interface height and the corresponding analysis shown in Fig. 5g, the correlation is 0.89. Naturally, the correspondence between the control and analysis is not as high for the vorticity and divergence fields (not shown), since only total-height observations are being assimilated.

The evolution of rms errors over the course of the experiment with 100 ensemble members is shown in Fig. 6. In each panel, the solid line corresponds to the true rms forecast error for a given field. The dotted line shows the EnKF estimate of the forecast error. The dashed line is the true analysis error and the dashed–dotted line is the analysis-error estimate. The true rms error, σ_t, is the rms difference between the ensemble mean and the truth from the control model, that is, $\sigma_t^2 = \sum_{k=1}^{n_g} (\overline{\xi}_k - \xi_k^t)^2)/(n_g - 1)$, where $\overline{\boldsymbol{\xi}}$ is the ensemble mean for field $\boldsymbol{\xi}$, $\boldsymbol{\xi}^t$ is the corresponding field from the true state, and the sum runs over all n_g grid points. The estimated rms errror, σ_e, is the horizontal average of the rms spread between ensemble members, $\sigma_e = \sum_{k=1}^{n_g} [\sum_{l=1}^{m} (\xi_{k,l} - \overline{\xi}_{k,l})^2/(m-1)]^{1/2}/n_g$, where subscripts k and l refer to the kth grid point and lth ensemble member.

The true rms forecast errors for the total height are about 50% of the corresponding climatological-forecast errors during the entire experiment (Fig. 6a). The estimated forecast errors are very close to the true forecast errors. The true total-height analysis errors are about 20% of the climatological-forecast errors for the 30 days of the experiment. The EnKF overestimates them somewhat: the estimated analysis errors are about 30% of the climatological-forecast errors. The interface-height errors behave like the total-height errors but the true analysis errors for this field stay at about 40% of the climatological-forecast reference errors (Fig. 6d). They are slightly underestimated by the EnKF. However, the estimated interface-height forecast errors are very close

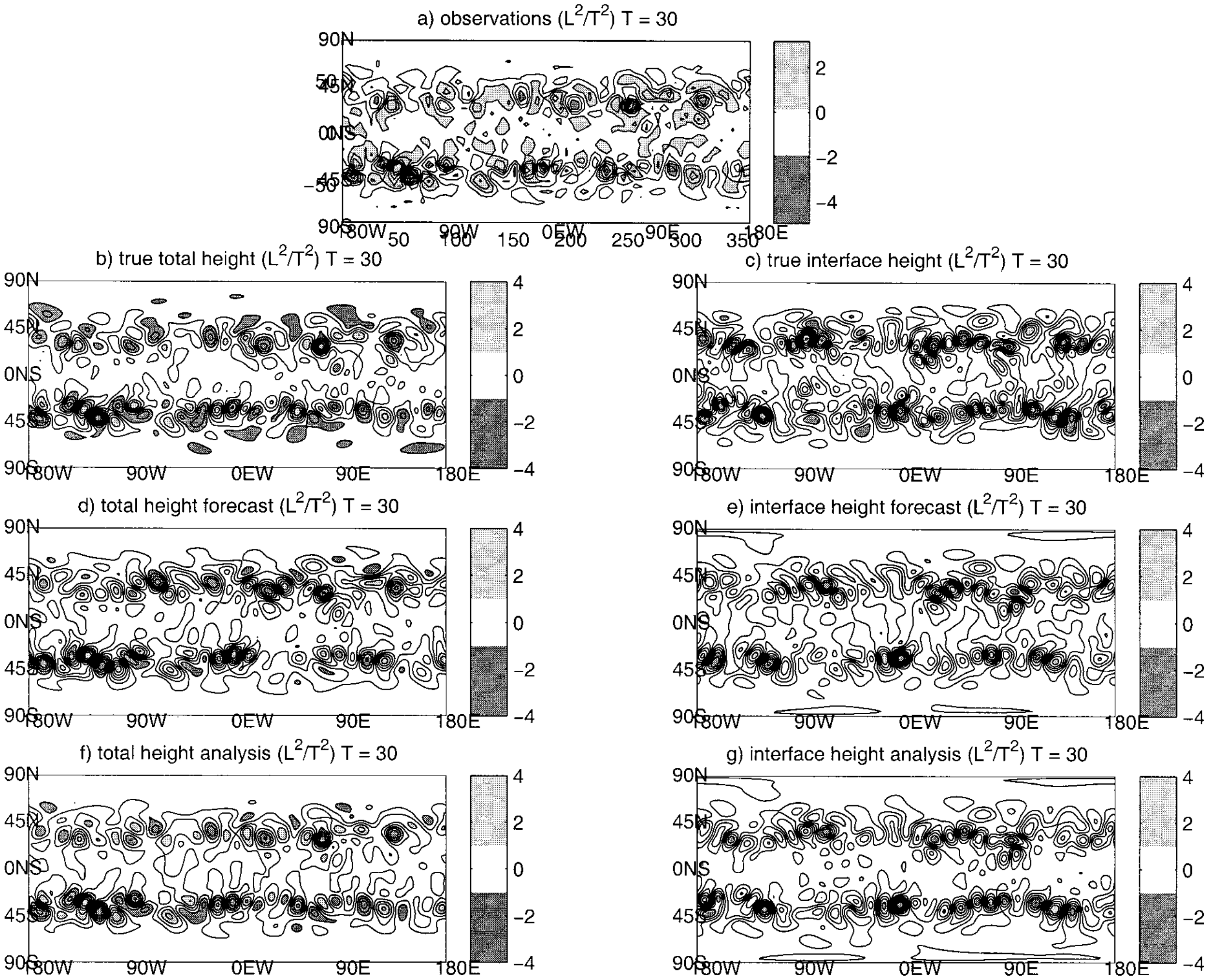
a) observations (L^2/T^2) T = 30
90N
45N
0NS
-45
90S
180W
90W
0EW
90E
180E
50
100
150
200
250
300
350
2
0
−2
−4
b) true total height (L^2/T^2) T = 30
c) true interface height (L^2/T^2) T = 30
d) total height forecast (L^2/T^2) T = 30
e) interface height forecast (L^2/T^2) T = 30
f) total height analysis (L^2/T^2) T = 30
g) interface height analysis (L^2/T^2) T = 30
90N
45N
0NS
45S
90S
180W
90W
0EW
90E
180E
4
2
0
−2
−4

FIG. 5. (a) Gridded total-height observations at day 30 of the experiments. (b) True total-height field from the control model. (c) True layer-interface-height field. (d) Total-height forecast with 100 ensemble members. (e) Layer-interface-height forecast. (f) Total-height analysis. (g) Layer-interface-height analysis. Dimensionless units are used in each panel.

to the true forecast errors. The forecast errors for the upper- and lower-layer vorticity are also well estimated by the EnKF (Figs. 6b and 6e). The analysis errors for the upper-layer vorticity are slightly underestimated. The underestimation is more severe for the lower-layer-vorticity analysis errors. The error estimates are very poor for the upper- and lower-layer divergence fields. There is no skill at all for these fields and the true forecast and analysis errors are comparable to the climatological-forecast errors (Figs. 6c and 6f). The EnKF grossly understimates them.

The effect of ensemble size on the rms errors is shown in Fig. 7. The errors shown here are time averages over the length of the experiments. The true forecast errors for the total height and layer-interface height reach a near-asymptotic state with about 100 ensemble members (Figs. 7a and 7d). The corresponding true analysis errors do not decrease much when the ensemble size is increased beyond 100. The forecast-error estimates for these two fields are very good (within a few percent) for the larger ensemble sizes. The analysis-error estimates are not as accurate. With 100 ensemble members, the ratio of estimated-to-true analysis error is 1.12 for the total height and 0.92 for the layer-interface height. The forecast errors for the upper- and lower-layer vorticity also reach a nearly asymptotic state but the analysis errors do not (Figs. 7b and 7e). The forecast-error estimates are also within a few percent for large ensembles. The analysis-error estimates are better for the upper-layer vorticity than for the lower-layer vorticity. With 100 ensemble members, the upper-layer-vorticity analysis-error estimate is 91% of the true analysis error. For the lower-layer vorticity it is 74% of the true analysis error. As already seen in Fig. 6, the true errors for the divergence fields are no better then the corresponding climatological-forecast errors, regardless of how many ensemble members there are (Figs. 7c and 7f). In fact, the error estimates for the divergence become

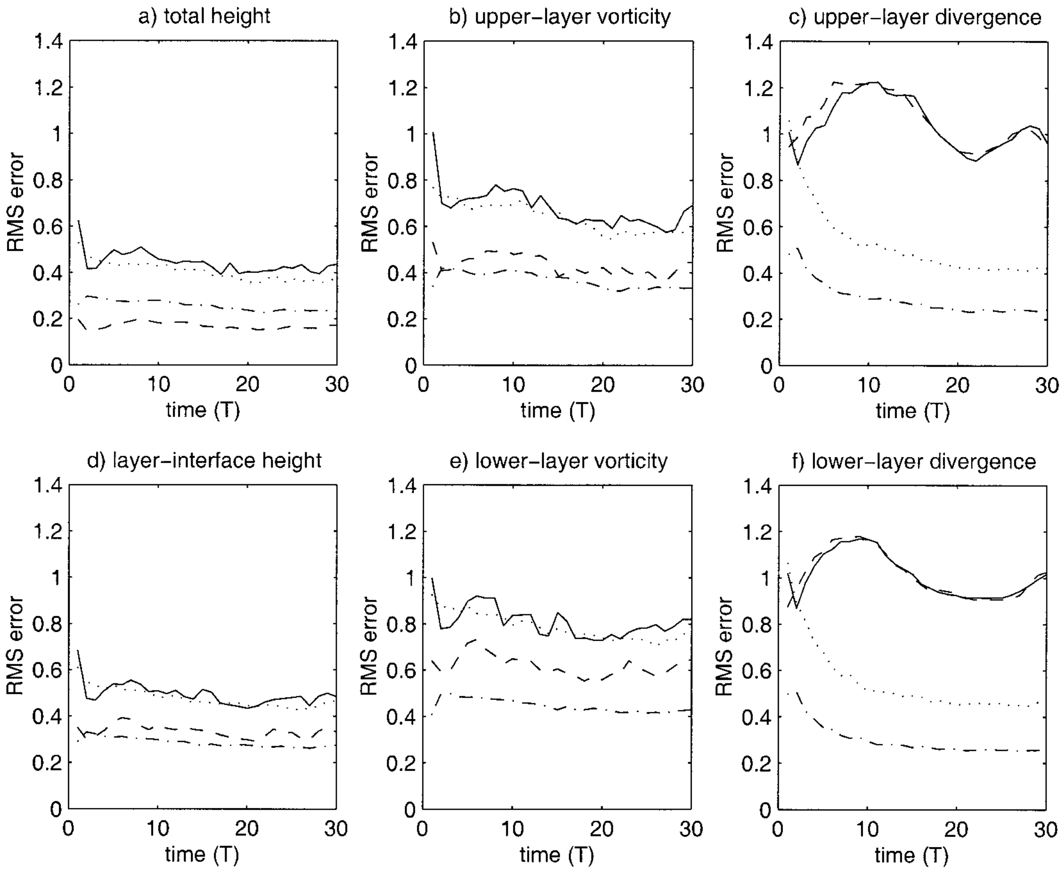

FIG. 6. Temporal evolution of rms errors for (a) the total height, (b) the upper-layer vorticity, (c) the upper-layer divergence, (d) the layer-interface height, (e) the lower-layer vorticity, and (f) the lower-layer divergence, in the experiment with 100 ensemble members. The errors are expressed in units of the error of a climatological forecast made with the climatology of the control model. The solid line in each panel corresponds to the true forecast error, the dotted line to the forecast-error estimate, the dashed line to the true analysis error, and the dashed–dotted line to the analysis-error estimate.

worse when the ensemble size increases. This is especially true for the forecast-error estimates. This behavior of the divergence-error estimates is likely due to the fact that the rms spread between ensemble members for the divergence fields is proportional to the rms spread for the other fields. Therefore, as the total-height and vorticity error estimates decrease with increasing ensemble sizes, so do the estimated divergence errors.

4. Conclusions and future directions

This work demonstrates that the EnKF can be efficiently implemented on a massively parallel computer with a minimal level of interprocessor communications. In this study, a serial model is used and each ensemble member is time stepped on a separate processor. To permit the analysis to also occur in parallel, a costly transposition of the ensemble matrix takes place so that each processor can process the observations that are local to the region for which it is responsible. This transposition can be avoided if the model dealt with is designed to run on a parallel computer architecture. This suggests that the EnKF could become a method of choice to assimilate observations into next-generation ocean and atmospheric GCMs running on distributed-memory computer architectures. The EnKF has the further advantage of providing time-dependent error-covariance estimates without the need for backward integrations of an adjoint model, as in the representer method (Bennett 1992).

The results of this study are encouraging. They indicate that moderate-size ensembles can be used with success to assimilate noisy observations into baroclinic primitive equation models in the context of an imperfect model. Although only total-height measurements are processed, their assimilation has a positive impact on the total and layer-interface heights and on the upper- and lower-layer-vorticity estimates. Th EnKF also does a reasonable job in estimating the error amplitude for these four fields. The assimilation of total height has no detectable impact on the divergence-field estimates. This is not surprising since quasigeostrophic theory tells

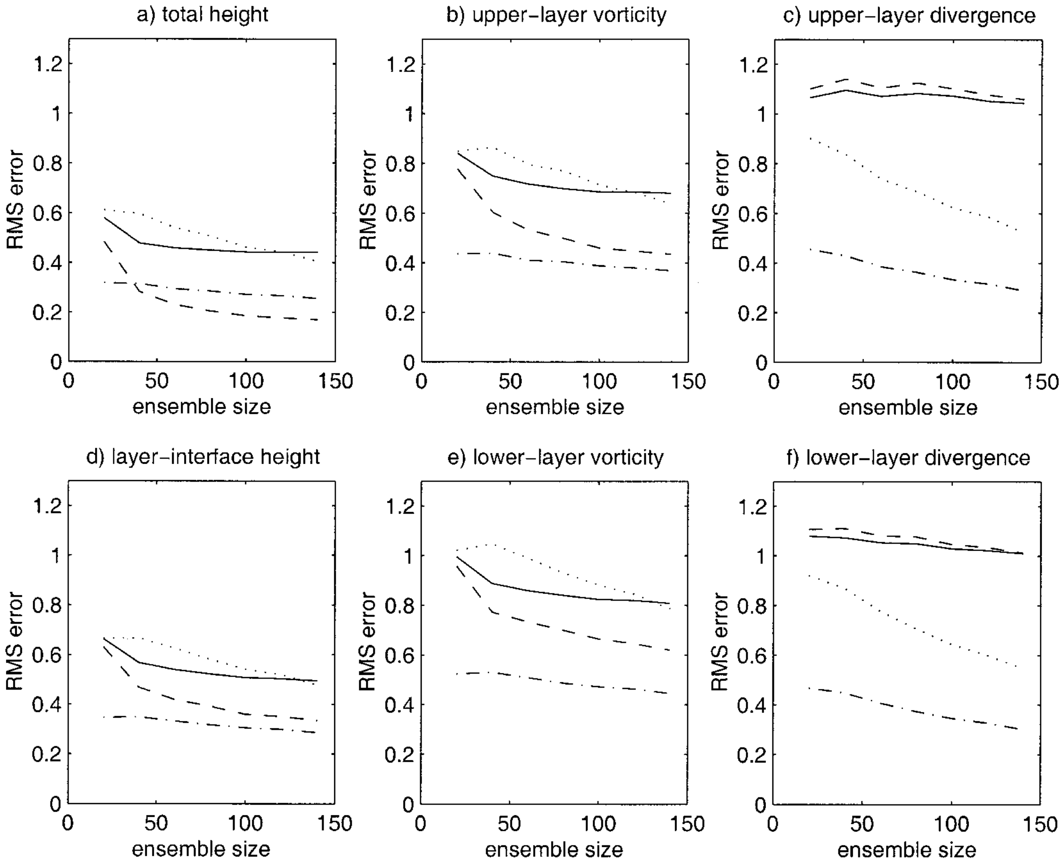

FIG. 7. Effect of ensemble size on time-mean rms errors for (a) the total height, (b) the upper-layer vorticity, (c) the upper-layer divergence, (d) the layer-interface height, (e) the lower-layer vorticity, and (f) the lower-layer divergence. The errors are expressed in units of the error of a climatological forecast made with the climatology of the control model. The solid line in each panel corresponds to the true forecast error, the dotted line to the forecast-error estimate, the dashed line to the true analysis error, and the dashed–dotted line to the analysis-error estimate.

us to expect zero covariances between the total-height and divergence fields. However, the error estimates for the upper- and lower-layer divergence are also very poor. This shortcoming is not due to the relatively small ensemble sizes considered here, since the error estimates become worse when more ensemble members are used. It indicates that the analysis scheme could still be improved. Perhaps, an approach similar to the one used in Houtekamer and Mitchell (1998), where the ensemble is divided into two parts and the covariances from each part are used to update the other part, could lead to better divergence-error estimates.

One could also argue that the process-noise statistics, which are known exactly in this study since the observations are produced by integrating the control model, are quite difficult to estimate in the real world. Yet, the present study represents a step in the right direction from identical-twin studies using a perfect model. We are currently working on an implementation of the EnKF for a realistic ocean GCM in which the process noise is modeled by using different atmospheric forcing fields to integrate each ensemble member (Keppenne and Rienecker 1999).

Acknowledgments. The author thanks Geir Evensen and Michele Rienecker for stimulating conversations. The computations took place on a CRAY T3E-600 at NASA' Goddard Space Flight Center. This project is supported by the Global Modeling and Analysis Program of NASA's Earth Science Enterprise under RTOP 622-24-47.

REFERENCES

Asselin, R., 1972: Frequency filter for time integations. *Mon. Wea. Rev.,* **100,** 487–490.

Bennett, A., 1992: *Inverse Methods in Physical Oceanography.* Cambridge University Press, 346 pp.

Bierman, G., 1977: *Factorization Methods for Discrete Sequential Estimation.* Academic Press, 241 pp.

Bourke, W., 1972: An efficient, one-level, primitive-equation spectral model. *Mon. Wea. Rev.,* **100,** 683–689.

Burg, J., 1967: Maximum entropy spectral analysis. *Proc. 37th Meeting of the Society of Exploration Geophysicists,* Oklahoma City, OK, Society of Exploration Geophysicists. Reprinted 1978, in *Modern Spectral Analysis,* D. G. Childers, Ed., IEEE Press, 334 pp.

Burgers, G., P. van Leeuwen, and G. Evensen, 1998: Analysis scheme in the ensemble Kalman filter. *Mon. Wea. Rev.,* **126,** 1719–1724.

Cane, M., A. Kaplan, R. Miller, B. Tang, E. Hackert, and A. Busalacchi, 1996: Mapping tropical Pacific sea level: Data assimilation via a reduced state space Kalman filter. *J. Geophys. Res.,* **101** (C), 22 599–22 617.

Evensen, G., 1992: Using the extended Kalman filter with a multilayer quasi-geostrophic ocean model. *J. Geophys. Res.,* **97,** 17 905–17 924.

——, 1994: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics *J. Geophys. Res.,* **99,** 10 143–10 162.

——, and P. van Leeuwen, 1996: Assimilation of GEOSAT altimeter data for the Agulhas Current using the ensemble Kalman filter with a quasigeostrophic model. *Mon. Wea. Rev.,* **124,** 85–96.

Haltiner, G., and R. Williams, 1980: *Numerical Prediction and Dynamic Meteorology.* John Wiley and Sons, 477 pp.

Hochstadt, H., 1971: *The Functions of Mathematical Physics.* John Wiley and Sons, 322 pp.

Houtekamer, P., and H. Mitchell, 1998: Data assimilation using an ensemble Kalman filter technique. *Mon. Wea. Rev.,* **126,** 796–811.

Kalman, R., and R. Bucy, 1961: New results in linear filtering and prediction theory. *Trans. ASME, J. Basic Eng.,* **82D,** 35–45.

Keppenne, C., and M. Rienecker, 1999: Assimilation of temperature into an ocean general circulation model using a massively parallel ensemble Kalman filter. *Proc. Third World Meteorological Organization Symp. on the Assimilation of Observations in Meteorology and Oceanography,* Quebec City, PQ, Canada, WMO, in press.

——, S. Marcus, M. Kimoto, and M. Ghil, 2000: Intraseasonal variability in a two-layer model and observations. *J. Atmos. Sci.,* **57,** 1010–1028.

Kurihara, A., 1965: On the use of implicit and iterative methods for the time integration of the wave equation. *Mon. Wea. Rev.,* **93,** 33–46.

Message Passing Interface Forum, 1994: A message-passing interface standard. CS-94-230, Computer Science Department Tech. Rep. University of Tenessee, Knoxville, TN, 228 pp. [Available online at http://www.cs.utk.edu./~library/1994.html]

Miller, R., M. Ghil, and F. Gauthiez, 1994: Advanced data assimilation in strongly nonlinear systems. *J. Atmos. Sci.,* **51,** 1037–1056.

Orzsag, S., 1970: Transform method for the calculation of vector-coupled sums: Application to the spectral form of the vorticity equation. *J. Atmos. Sci.,* **27,** 890–895.