Automated weighing by sequential inference in dynamic environments

A. D. Martin and T. C. A. Molteno

Department of Physics

University of Otago

Dunedin, 9016, New Zealand

Email: amartin@elec.ac.nz

Abstract—We demonstrate sequential mass inference of a suspended bag of milk powder from simulated measurements of the vertical force component at the pivot while the bag is being filled. We compare the predictions of various sequential inference methods both with and without a physics model to capture the system dynamics. We find that non-augmented and augmented-state unscented Kalman filters (UKFs) in conjunction with a physics model of a pendulum of varying mass and length provide rapid and accurate predictions of the milk powder mass as a function of time. The UKFs outperform the other method tested - a particle filter. Moreover, inference methods which incorporate a physics model outperform equivalent algorithms which do not.

I. Introduction

Sequential inference has been used for measurement and control in many contexts including vehicle navigation [1], target tracking [2] and chemical process plant control [3]. Sequential inference algorithms take a time-series of noisy measurements of a system, and produce increasingly accurate estimates of the system parameters or state variables in the form of a posterior distribution. We adapt sequential inference algorithms for use in automated weighing systems, where the system under consideration exhibits dynamics according to physical laws. Such methods provide dynamically updated estimates of mass, along with estimates of its uncertainty, which will be useful for the control of automated weighing systems.

We use the example of a milk powder bagging system, where the bag is suspended from a point at its top from which it may swing under the influence of gravity. The bag is gradually filled with milk powder while measurements of the vertical force component are made at discrete times. We test sequential inference algorithms by simulating such measurements, taking into account both process and measurement noise, and run the algorithms on the simulated data-set. We analyse the predictions for accuracy, precision and speed of inference, and draw conclusions about the most suitable algorithm. We stress the benefits of considering the underlying physics when designing sensors and control systems.

II. Methods

A. Physics model

We model the bag-filling system as a pendulum with mass increasing at rate $\dot{m}$ (as the bag fills), and with effective length
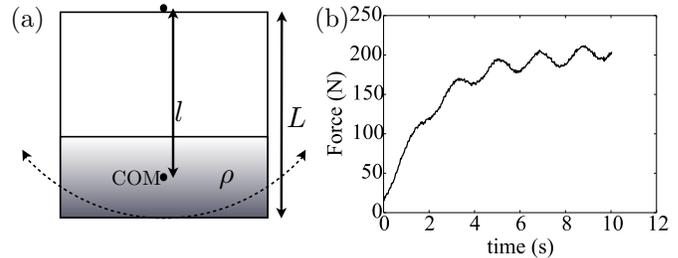

Fig. 1. (a) Schematic of the system. The box is a model of the bag of length L, which pivots at the indicated point at its top. The cross-sectional area is A (not indicated on the figure). The milk powder is represented by the shaded region, and has density ρ, and its centre of mass (COM) is indicated. The system behaves as a pendulum of length l (from pivot to COM). (b) Simulated data for the observed vertical component of force at the pivot during bag filling.

$l = L - x_{\text{com}}$, where $x_{\text{com}} = m/(2\rho A)$, ρ is the density of the milk powder and A is the cross-sectional area of the bag. The rate $\dot{m}$ is assumed to vary randomly (see below). Figure 1 (a) shows a schematic of the system. The equations of motion for the pendulum are simply:

$$\dot{\theta} = \omega, \tag{1}$$

$$\dot{\omega} = -g \sin\theta/l, \tag{2}$$

where θ is the angle of the pendulum, and the effective pendulum length l is not constant but decreases as the milk powder's centre of mass rises. The system is observed via the vertical component of the force on the measuring device from which the bag is attached, which is given by:

$$F = m \cos\theta \left(l\omega^2 + g \cos\theta \right). \tag{3}$$

We simulate the system at discrete times t_n to generate some example data. We propagate the dynamical variables between each time according to Eqs. (1) and (2), and at each timestep evolve the flow of mass into the bag according to:

$$\log\left(\dot{m}_n/1\text{kg s}^{-1} \right) = \log\left(\dot{m}_{n-1}/1\text{kg s}^{-1} \right) + d_n, \tag{4}$$

where $d_n \sim N(0, \Sigma_m)$ is a process noise term, which models the tendency of the filling-rate to slow down and speed up, whilst remaining strictly positive. Then each force measurement is simulated as:

$$F_n = F(\theta_n, m_n, l_n) + v_n, \tag{5}$$

where the measurement noise $v_n \sim N(0, \Sigma_F)$.

B. Inference methods

To compare different algorithms' performance in estimating the mass of milk powder in a bag during filling we use three different sequential inference algorithms, each either in conjunction with the physics model described in Sec. II-A, or without such a model. Each inference model is fed the same set of measurements $\{\mathbf{y}_n\}$ (in this case equal to the simulated forces F_n), and is given an initial state estimate in the form of a prior distribution. When using the physics model, the algorithms estimate a state vector containing the components θ, ω, $\log(m/1\text{kg})$, $\log(\dot{m}/1\text{kg s}^{-1})$, $\log(L/1\text{m})$ and $\log(\rho A/1\text{kg m}^{-1})$. Otherwise, the algorithms estimate $\log(m/1\text{kg})$ and $\log(\dot{m}/1\text{kg s}^{-1})$ only. Note that log variables of some quantities are estimated to ensure strictly positive estimates of those quantities.

Each algorithm assumes the state vector propagates between timesteps using the forward map $\mathbf{f} : \mathbf{x}_n \mapsto \mathbf{x}_{n+1}$. When the physics model is used, this map integrates the equations of motion [Eqs. (1) and (2)], and evolves the mass as dictated by $\dot{m}$. Otherwise, only the mass is evolved. Each algorithm also maps a state to a measurement estimate by an observation map $\mathbf{g} : \mathbf{x} \mapsto \hat{\mathbf{y}}$, which is given by Eq. (3) when the physics model is used, and $F = mg$ otherwise.

We briefly describe the functioning of each algorithm below, and provide detailed description in Figs. 2-4.

1) Kalman filters: Firstly, we use two sequential-inference algorithms based on the Kalman Filter, namely the augmented-state Unscented Kalman Filter (UKF), developed in Ref. [1], [4], and the non-augmented UKF, described in Ref. [5]. These algorithms are extensions of the Kalman Filter designed to perform well with nonlinear forward maps and/or nonlinear observation maps, on the principle that it is easier to approximate the probability distribution than to approximate (linearise) the nonlinear functions. The ability to perform well using nonlinear maps is vital for the current problem, since the equations of motion of the physics model are nonlinear. Even if the physics model is not used, the use of log variables to ensure positive estimates of the mass and mass-flow rate requires the forward map for these variables also to be nonlinear.

Both algorithms approximate the state distribution by specially chosen 'sigma points', which capture at least the first two moments of the distribution. The sigma points are run through the forward map and the transformed mean and covariance are used in order to perform a usual Kalman Filter update [1].

The augmented-state UKF differs from the non-augmented UKF in 3 ways: in the augmented-state UKF the sigma-point states are augmented with parameters representing the process noise and measurement noise; the non-augmented state UKF samples the sigma-points twice, while the augmented-state UKF only once; also, in the non-augmented state UKF the process and measurement uncertainties are included in the calculation through the 'augmented maps' $\mathbf{f}_a$ and $\mathbf{g}_a$ (see Fig.

Initialise Set the prior mean μ_0 and covariance $\mathbf{K}_0$ with a well-motivated estimate, along with parameters α, β and γ which determine the distribution of sigma-points [4]. Provide the estimated process and measurement covariances: Σ_d and Σ_v.

for $n = 1...n_t - 1$ **do**

1) Calculate sigma points and weights $\{\mathbf{x}^{(j)}, W^{(j)}\}$:
$$\mathbf{x}^{(0)} = \mu_{n-1},$$
$$W_m^{(0)} = \frac{1}{N_x + \lambda}, \quad W_c^{(0)} = W_m^{(0)} + (1 - \alpha^2 + \beta),$$
$$\mathbf{x}^{(i)} = \mu_{n-1} + \sqrt{(N_x + \lambda)\mathbf{K}_{n-1}},$$
$$\mathbf{x}^{(i+N_x)} = \mu_{n-1} - \sqrt{(N_x + \lambda)\mathbf{K}_{n-1}},$$
$$W_m^{(i)} = W_c^{(i)} = W_m^{(i+N_x)} = W_c^{(i+N_x)} = \frac{1}{2(N_x + \lambda)},$$
for $i = 1...N_x$, where $\lambda = \alpha^2(N_x + \kappa)$, and N_x is the state dimension.

2) Transform sigma points using the forward map:
$$\hat{\mathbf{x}}_n^{(i)} = \mathbf{f}(\mathbf{x}_n^{(i)}).$$

3) Calculate the predicted mean and covariance:
$$\hat{\mu}_n = \sum_{i=0}^{p} W_m^{(i)} \hat{\mathbf{x}}_n^{(i)},$$
$$\hat{\mathbf{K}}_n = \Sigma_d + \sum_{i=0}^{p} W_c^{(i)} (\hat{\mathbf{x}}_n^{(i)} - \hat{\mu}_n)(\hat{\mathbf{x}}_n^{(i)} - \hat{\mu}_n)^T.$$

4) Recalculate the sigma points using the predicted mean and covariance:
$$\tilde{\mathbf{x}}^{(0)} = \hat{\mu}_n,$$
$$\tilde{\mathbf{x}}^{(i)} = \hat{\mu}_n + \sqrt{(N_x + \lambda)\hat{\mathbf{K}}_n},$$
$$\tilde{\mathbf{x}}^{(i+N_x)} = \hat{\mu}_n - \sqrt{(N_x + \lambda)\hat{\mathbf{K}}_n}.$$

5) Apply the observation model to each new sigma point :
$$\hat{\mathbf{y}}_n^{(i)} = \mathbf{g}(\tilde{\mathbf{x}}_n^{(i)}).$$

6) Calculate the predicted observation:
$$\hat{\mathbf{y}}_n = \sum_{i=0}^{2N_x} W_m^{(i)} \hat{\mathbf{y}}_n^{(i)}.$$

7) Calculate the innovation covariance:
$$\hat{\mathbf{S}}_n = \Sigma_v + \sum_{i=0}^{2N_x} W_c^{(i)} (\hat{\mathbf{y}}_n^{(i)} - \hat{\mathbf{y}}_n)(\hat{\mathbf{y}}_n^{(i)} - \hat{\mathbf{y}}_n)^T.$$

8) Calculate the cross covariance:
$$\mathbf{K}_n^{xy} = \sum_{i=0}^{2N_x} W_c^{(i)} (\hat{\mathbf{x}}_n^{(i)} - \hat{\mu}_n)(\hat{\mathbf{x}}_n^{(i)} - \hat{\mu}_n)^T.$$

9) Perform a usual Kalman Filter update :
$$\mu_n = \hat{\mu}_n + \mathbf{W}_n v_n,$$
$$\mathbf{K}_n = \hat{\mathbf{K}}_n - \mathbf{W}_n \hat{\mathbf{S}}_n \mathbf{W}_n^T,$$
where $v_n = \mathbf{y}_n - \hat{\mathbf{y}}_n$ and $\mathbf{W}_n = \mathbf{K}_n^{xy} \hat{\mathbf{S}}_n^{-1}$.

end

Fig. 2.   Non-augmented-state UKF [5]

Initialise Set the prior mean μ_0 and covariance $\mathbf{K}_0$ with a well-motivated estimate, along with parameters α, β and γ which determine the distribution of sigma-points [4]. Provide the estimated process and measurement covariances: Σ_d and Σ_v.

for $n = 1 ... n_t - 1$ **do**

  1) Augment the state mean μ_{n-1} and covariance $\mathbf{K}_{n-1}$ with the the process noise and measurement noise means (zeros) and their respective covariances, Σ_d, Σ_v:
$$\mu_n^a = \begin{pmatrix} \mu_{n-1} \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \quad \mathbf{K}_{a,n} = \mathrm{diag}\,(\mathbf{K}_{n-1}, \Sigma_d, \Sigma_v).$$

  2) Calculate sigma points and weights $\{\mathbf{x}^{(j)}, W^{(j)}\}$:
$$\mathbf{x}^{(0)} = \mu_n^a,$$
$$W_m^{(0)} = \frac{1}{N_x + \lambda}, \; W_c^{(0)} = W_m^{(0)} + \left(1 - \alpha^2 + \beta\right),$$
$$\mathbf{x}^{(i)} = \mu_n^a + \sqrt{(N_x + \lambda)\,\mathbf{K}_{a,n}},$$
$$\mathbf{x}^{(i+N_x)} = \mu_n^a - \sqrt{(N_x + \lambda)\,\mathbf{K}_{a,n}},$$
$$W_m^{(i)} = W_c^{(i)} = W_m^{(i+N_x)} = W_c^{(i+N_x)} = \frac{1}{2\,(N_x + \lambda)},$$
for $i = 1 ... N_x$, where $\lambda = \alpha^2\,(N_x + \kappa)$, and N_x is the dimension of the augmented state vector.

  3) Transform sigma points using the forward map:
$$\hat{\mathbf{x}}_{a,n}^{(i)} = \mathbf{f}_a\left(\mathbf{x}_{a,n}^{(i)}\right), \text{ where } \mathbf{f}_a(\mathbf{x}_a) = \begin{pmatrix} \mathbf{f}(\mathbf{x}) + \mathbf{x}_d \\ \mathbf{x}_d \\ \mathbf{x}_v \end{pmatrix}, \text{ and}$$
$\mathbf{x}$, $\mathbf{x}_d$ and $\mathbf{x}_v$ are the state, process-noise and measurement-noise parts of $\mathbf{x}_a$.

  4) Calculate the predicted mean and covariance:
$$\hat{\mu}_{a,n} = \sum_{i=0}^{2N_x} W^{(i)}\hat{\mathbf{x}}_{a,n}^{(i)},$$
$$\hat{\mathbf{K}}_{a,n} = \sum_{i=0}^{2N_x} \left(\hat{\mathbf{x}}_{a,n}^{(i)} - \hat{\mu}_{a,n}\right)\left(\hat{\mathbf{x}}_{a,n}^{(i)} - \hat{\mu}_{a,n}\right)^T.$$

  5) Apply the observation model to each transformed sigma point :
$$\hat{\mathbf{y}}_n^{(i)} = \mathbf{g}_a\left(\hat{\mathbf{x}}_{a,n}^{(i)}\right), \text{ where } \mathbf{g}_a(\mathbf{x}_a) = \mathbf{g}(\mathbf{x}) + \mathbf{x}_v.$$

  6) Calculate the predicted observation:
$$\hat{\mathbf{y}}_n = \sum_{i=0}^{2N_x} W_m^{(i)}\hat{\mathbf{y}}_n^{(i)}.$$

  7) Calculate the innovation covariance:
$$\hat{\mathbf{S}}_n = \sum_{i=0}^{2N_x} W_c^{(i)}\left(\hat{\mathbf{y}}_n^{(i)} - \hat{\mathbf{y}}_n\right)\left(\hat{\mathbf{y}}_n^{(i)} - \hat{\mathbf{y}}_n\right)^T.$$

  8) Calculate the cross covariance (for the non-augmented state):
$$\mathbf{K}_n^{xy} = \sum_{i=0}^{2N_x} W^{(i)}\left(\hat{\mathbf{x}}_n^{(i)} - \hat{\mu}_n\right)\left(\hat{\mathbf{x}}_n^{(i)} - \hat{\mu}_n\right)^T.$$

  9) Perform a usual Kalman Filter update :
$$\mu_n = \hat{\mu}_n + \mathbf{W}_n\nu_n,$$
$$\mathbf{K}_n = \hat{\mathbf{K}}_n - \mathbf{W}_n\hat{\mathbf{S}}_n\mathbf{W}_n^T,$$
where $\nu_n = \mathbf{y}_n - \hat{\mathbf{y}}_n$ and $\mathbf{W}_n = \mathbf{K}_n^{xy}\hat{\mathbf{S}}_n^{-1}$.

end

Fig. 3.   Augmented-state UKF [1], [4]

Initialise Sample n_p 'particles' from well-motivated prior:
$$\mathbf{x}_0^{(i)} \sim P(\mathbf{x}).$$
Select initial weights $W_0^{(i)} = 1/n_p$.
Provide the estimated process and measurement covariances: Σ_d and Σ_v.

for $n = 1 ... n_t - 1$ **do**

  1) Get next state sample:
Propagate 'particles' through forward map, adding process noise sampled from relevant distribution (in this case, gaussian)
$$\mathbf{x}_n^{(i)} \sim N\left(\mathbf{f}\left(\mathbf{x}_{n-1}^{(i)}\right), \Sigma_d\right).$$

  2) Get log measurement probability:
$$\log p\left(\mathbf{y}_n | \mathbf{x}_n^{(i)}\right) = -\frac{1}{2}\nu_i^T\Sigma_v^{-1}\nu_i, \text{ where } \nu_i = \mathbf{y}_n - \mathbf{g}\left(\mathbf{x}_n^{(i)}\right)$$
and set weights
$$\hat{W}_n^{(i)} = \exp\left(\log W_{n-1}^{(i)} + \log p\left(\mathbf{y}_n | \mathbf{x}_n^{(i)}\right)\right).$$
Normalise weights: $W_n^{(i)} = \dfrac{\hat{W}_n^{(i)}}{\sum_j \hat{W}_n^{(j)}}$.

  3) Get effective number of particles:
$$n_{\mathrm{eff}} = \frac{1}{\sum_j \left(\hat{W}_n^{(i)}\right)^2}.$$

  if $n_{eff} < n_{thr}$ **then**
Draw n_p particles $\left\{x_n^{(j)}\right\}$ from the current particle set $\left\{x_n^{(i)}\right\}$ with probabilities proportional to $\left\{W_n^{(i)}\right\}$.
Reset weights $W_n^{(j)} = 1/n_p$.
  end

end

Fig. 4.   Particle filter

3) rather than added to the predicted covariance and innovation covariance as in the non-augmented UKF (see Fig. 2).

The performance of these two algorithms is compared in Ref. [6], which found that for the problems considered there the augmented-state UKF generally performs better.

2) Particle filters: As well as the two varieties of UKF, we also test the performance of a particle filter. We use a sequential importance resampling algorithm, first introduced in Ref. [7]. This algorithm has the advantage that as well as permitting a nonlinear forward map and observation function, it permits the prior distribution for the state vector to have any form, as well as the estimated noise distributions (although we do not need to exploit this flexibility for the current problem). The algorithm approximates a distribution of states by a sample of weighted states ('particles'). It evolves each 'particle' through the forward map before performing a Bayesian update of the sample. Resampling prevents weights becoming concentrated in a small number of particles.

III. Results

We ran the algorithms on measurements simulated using the following parameters/initial conditions (partly motivated by Ref. [8]): $\theta_0 = 0.2$ rad, $\omega_0 = 0.2$ rad/s, $L = 3.5$ m, $m_0 = 1.7$
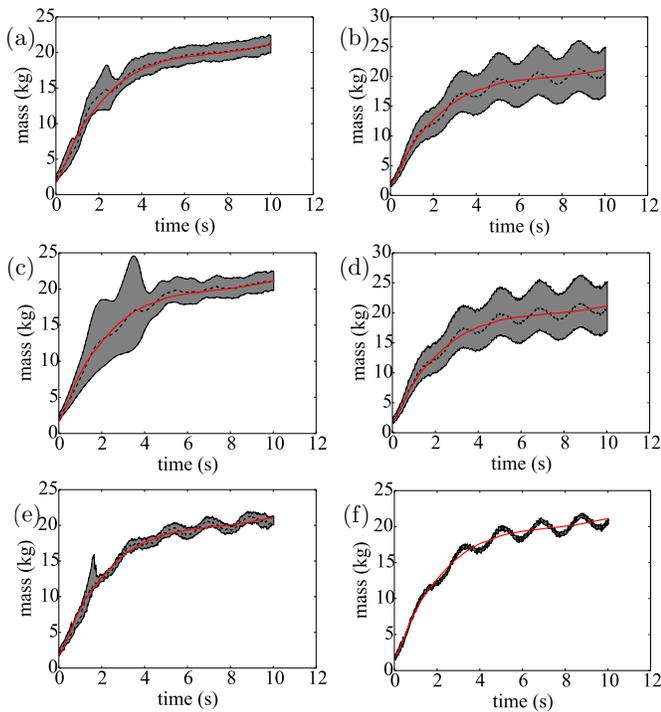
Fig. 5. Sequential mass estimates (shaded regions) and true mass values [light (red) lines] as a function of time. Subplots (a) and (b) show results for the non-augmented-state UKF, (c) and (d) for the augmented-state UKF, and (e) and (f) for the particle filter. Results shown in (a), (c) and (e) are computed using the physics model and those in (b), (d) and (f) without. For the UKF results, grey regions represent the central 90% prediction regions for the mass, dotted lines represent the region boundaries and mean mass prediction; for the particle filter, grey regions represent masses between the 5th and 95th percentile of the particle sample (indicated by dotted lines), the 50th mass percentile is also given by a dotted line.
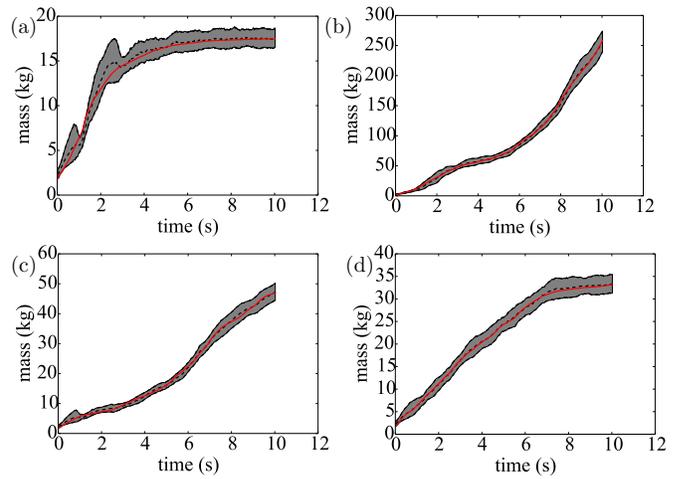

Fig. 6. Sequential mass estimates (shaded regions) from for non-augmented-state UKF and true mass values [light (red) lines] as a function of time. Grey regions represent the central 90% prediction regions for the mass, dotted lines represent the region boundaries and mean mass prediction. Plots (a)-(d) show results obtained for simulated data with different realisations of the process and measurement noise.

kg, $\dot{m}_0 = 5.5$ kg/s, $\rho A = 161.25$ kg/m. We used $\Sigma_m = 0.1$ to generate process noise in the mass flow [via Eq. (4)] at samples every 0.025s during 10s of filling time. Both UKF algorithms used a prior distribution for the state space with mean values equivalent to: $\theta_0 = 0.21$ rad, $\omega_0 = 0.15$ rad/s, $L = 2.5$ m, $m_0 = 2.2$ kg, $\dot{m}_0 = 5.36$ kg/s, $\rho A = 177.38$ kg/m and variances: $\sigma_\theta^2 = 0.2$ rad^2, $\sigma_\omega^2 = 0.2$ rad^2, $\sigma_{\log L/1m}^2 = 0.2$, $\sigma_{\log m/1kg}^2 = 0.02$, $\sigma_{\log \dot{m}/1kg\ s^{-1}}^2 = 0.02$, and $\sigma_{\log \rho A/1kg\ m^{-1}}^2 = 0.2$. This prior overlapped the 'true' state, but was not centred thereon. The particle filter sampled 1000 particles from a gaussian prior of the same mean and covariance as used by the UKF algorithms. For the estimated process noise covariance, Σ_d, we used Σ_m for the $\log\left(\dot{m}/1kg\ s^{-1}\right)$ variance component, and small but non-zero values for the other diagonal components. These non-zero values for the variables without expected process noise are necessary for the correct functioning of all the algorithms, and are usually justified in real applications by invoking the need to account for the difference between the (necessarily simplistic) model used and the laws governing the true system dynamics. In the simulated measurements, the noise covariance $\Sigma_F = 1.5$ N^2, and the estimated noise covariance supplied to the inference algorithms $\Sigma_\nu = 2.5$ N^2

We ran all algorithms on the same set of simulated mea-surements illustrated in Fig. 1(b). The sequential estimates of mass are expressed as 90% prediction regions, and are shown in Fig. 5. The most successful methods were the UKFs and particle-filter using the physics model [Figs. 5(a), (c) and (e)]. These methods produced suitably narrow regions which contained the 'true' mass value. The UKFs which did not use a physics model provided wide 90% prediction intervals which contained the true mass value [Figs. 5(b) and (d)]; the particle filter which did not use a physics model produced a narrow 90% prediction interval which often did not contain the true mass value [Fig. 5(f)]. The particle filter without a model clearly attributed the oscillations in the measured vertical force component to oscillations in the mass flow, even though the magnitude of these oscillations was larger than would be expected from the stated process noise. We tested the non-augmented UKF for different realisations of the process and measurement noise, and it was found to produce consistently good estimates [see Fig. 6].

As time $t \to \infty$, the widths of the prediction intervals are limited by the effect of the finite process and measurement noise. By using idealised equipment, where the process and measurement noise is very small, it should be possible to obtain very accurate estimates of the mass. We demonstrate this by simulating a force dataset using negligible process noise in $\dot{m}$. We ran all inference algorithms on this dataset using estimates of the process covariance σ_d with very small diagonal elements of order 10^{-8} in each appropriate unit (and zero-valued off-diagonal elements). All inference methods using the physics model produced estimates with very narrow prediction region [Fig. 7(a), (c) and (e)]; however, the particle filter's 90% prediction region did not contain the true mass value. For these simulations with no process noise [Figs. 7 (b), (d) and (f)], the methods without a physics model produced poor estimates, either with wide prediction regions (in the case
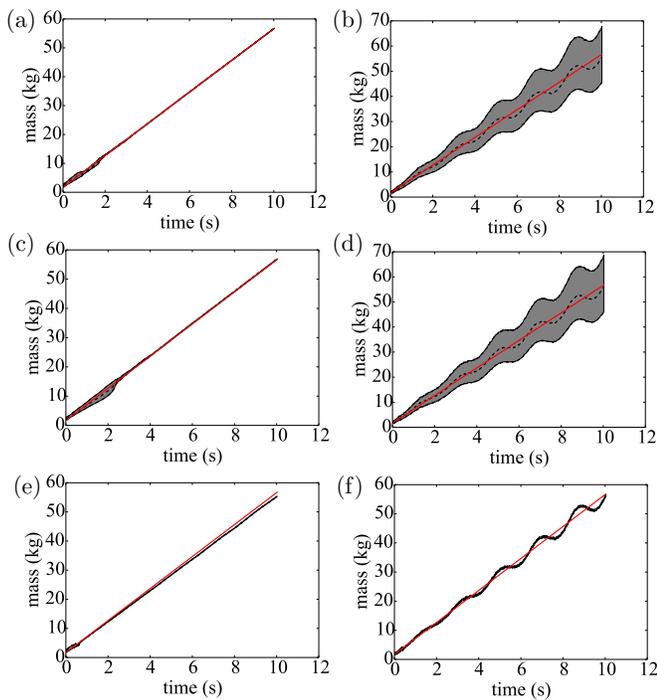
277

Fig. 7. As Fig. 5, but all with all inferences performed on simulated data with negligible process noise in $\dot{m}$, using negligible estimated process noise.

of the UKFs), or with narrow regions not always containing the true mass value (in the case of the particle filter).

IV. CONCLUSIONS

We have shown the benefit of including a physics model in sequential inference algorithms used in control-systems for automated weighing - in particular for weighing a suspended bag of powdered milk during the filling process. When used in conjunction with a UKF or particle filter algorithm, these methods are expected to produce narrow 90% prediction regions containing the true mass value, which would not be possible without the consideration of the dynamics generated by the physics model. Reference [6] found that the augmented-state UKF performed better than the non-augmented UKF in the problems considered there; however, for the problem considered in this paper, both UKF algorithms produce very similar output in similar running times. UKF algorithms are expected to be more useful than particle filters, since their running time is comparable to the filling times used in this paper. The particle filter's running time is several times longer than these filling times, so in practice, particle filters would require slower filling rates and longer times between force measurements in order to inform a synchronous control system. Improved predictive performance of the particle filter in the cases where the prediction intervals fail to contain the true mass value would be expected if a larger number of particles were sampled. However, this would slow the running time even further.

Consideration of systems with process noise in the density ρ and cross-sectional area A would provide straightforward and

important extensions of this work. The density of milk would be expected to fluctuate as it settles during packing, and the cross-sectional area A would also be expected to vary with height within the bag.

REFERENCES

[1] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, Mar 2004.
[2] P. Costa, "Adaptive model architecture and extended kalman-bucy filters," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 30, no. 2, pp. 525–533, Apr 1994.
[3] G. Prasad, G. Irwin, E. Swidenbank, and B. Hogg, "Plant-wide predictive control for a thermal power plant based on a physical plant model," *Control Theory and Applications, IEE Proceedings -*, vol. 147, no. 5, pp. 523–537, Sep 2000.
[4] E. Wan and R. van der Merwe, "The unscented kalman filter for nonlinear estimation," in *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, 2000, pp. 153–158.
[5] R. van der Merwe and E. A. Wan, "The square-root unscented kalman filter for state and parameter-estimation," in *in International Conference on Acoustics, Speech, and Signal Processing*, 2001, pp. 3461–3464.
[6] Y. Wu, D. Hu, M. Wu, and X. Hu, "Unscented kalman filtering for additive noise case: augmented vs. non-augmented," in *American Control Conference, 2005. Proceedings of the 2005*, June 2005, pp. 4051–4055 vol. 6.
[7] N. Gordon, D. Salmond, and A. Smith, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *Radar and Signal Processing, IEE Proceedings F*, vol. 140, no. 2, pp. 107–113, Apr 1993.
[8] J. J. Tuohy, "Some physical properties of milk powders," *Irish Journal of Food Science and Technology*, vol. 13, no. 2, pp. pp. 141–152, 1989. [Online]. Available: http://www.jstor.org/stable/25619580