State-Space Inference for Non-Linear Latent Force Models with Application to Satellite Orbit Prediction

Jouni Hartikainen\(^1\)
(joint work with Mari Seppänen\(^2\) and Simo Särkkä\(^1\))

\(^1\)Department of Biomedical Engineering and Computational Science (BECS)
Aalto University, Espoo, Finland

\(^2\)Department of Mathematics,
Tampere University of Technology, Tampere, Finland

June, 2012
Contents

1. Latent Force Models
2. SDE View of Latent Force Models
3. Filtering and Smoothing of SDEs
4. Simulated Experiments
5. GPS Satellite Orbit Prediction
6. Conclusion
Contents

1. Latent Force Models
2. SDE View of Latent Force Models
3. Filtering and Smoothing of SDEs
4. Simulated Experiments
5. GPS Satellite Orbit Prediction
6. Conclusion
1 Latent Force Models

2 SDE View of Latent Force Models

3 Filtering and Smoothing of SDEs

4 Simulated Experiments

5 GPS Satellite Orbit Prediction

6 Conclusion
Latent Force Models (LFMs)

- LFMs combine **mechanistic principles with data-driven components**.
- In general, can be written in form

\[ dx(t) = f(x(t), u(t), t) \, dt, \]  

where \( x(t) \) is a process with some mechanistic a priori information.

- The latent forces \( u(t) = \{u_r(t)\}_{r=1}^R \) have **Gaussian process (GP)** priors

\[ u_r(t) \sim \mathcal{GP}(m(t), k_{ur}(t, t')), \quad r = 1, \ldots, R. \]  

- In several key applications \( f(\cdot) \) is **non-linear**.
Latent Force Models (LFMs)

- LFMs combine mechanistic principles with data-driven components.
- In general, can be written in form

\[ dx(t) = f(x(t), u(t), t) \, dt, \]

where \( x(t) \) is a process with some mechanistic a priori information.

- The latent forces \( u(t) = \{u_r(t)\}_{r=1}^R \) have Gaussian process (GP) priors

\[ u_r(t) \sim \mathcal{GP}(m(t), k_{ur}(t, t')), \quad r = 1, \ldots, R. \]

- In several key applications \( f(\cdot) \) is non-linear.
Latent Force Models (LFMs)

- LFM combines mechanistic principles with data-driven components.
- In general, can be written in form

\[ dx(t) = f(x(t), u(t), t) \, dt, \]  

(1)

where \( x(t) \) is a process with some mechanistic a priori information.

- The latent forces \( u(t) = \{ u_r(t) \}_{r=1}^R \) have Gaussian process (GP) priors

\[ u_r(t) \sim \mathcal{GP}(m(t), k_{ur}(t, t')), \quad r = 1, \ldots, R. \]  

(2)

- In several key applications \( f(\cdot) \) is non-linear.
Latent Force Models (LFMs)

- LFMś combine mechanistic principles with data-driven components.
- In general, can be written in form

\[ dx(t) = f(x(t), u(t), t) \, dt, \] (1)

where \( x(t) \) is a process with some mechanistic a priori information.

- The latent forces \( u(t) = \{u_r(t)\}_{r=1}^R \) have Gaussian process (GP)
  priors

\[ u_r(t) \sim GP(m(t), k_{ur}(t, t')), \quad r = 1, \ldots, R. \] (2)

- In several key applications \( f(\cdot) \) is non-linear.
Linear LFMss with certain covariance functions: standard GP regression techniques.

Non-linear LFMss need approximations.

Previous approaches:
- Laplace approximation (Lawrence et al. 2007)
- Markov chain Monte Carlo (Titsias et al., 2009)

Disadvantages:
- Need the solution of \( x(t) \) explicitly as a function of \( u(t) \) to (numerically) calculate the needed likelihood \( p(Y|u(\cdot), \theta) \).
- Cubic scaling of computations with respect number of time points.

In this work, we propose a framework, which alleviates these problems.
Linear LFMs with **certain** covariance functions: **standard** GP regression techniques.

Non-linear LFMs need approximations.

Previous approaches:
- Laplace approximation (Lawrence et al. 2007)
- Markov chain Monte Carlo (Titsias et al., 2009)

Disadvantages:
- Need the solution of $x(t)$ explicitly as a function of $u(t)$ to (numerically) calculate the needed likelihood $p(Y|u(\cdot), \theta)$.
- Cubic scaling of computations with respect number of time points.

In this work, we propose a framework, which alleviates these problems.
Linear LFMs with **certain** covariance functions: **standard GP regression techniques**.

Non-linear LFMs need approximations.

Previous approaches:
- Laplace approximation (Lawrence et al. 2007)
- Markov chain Monte Carlo (Titsias et al., 2009)

Disadvantages:
- Need the solution of \( x(t) \) explicitly as a function of \( u(t) \) to (numerically) calculate the needed likelihood \( p(\mathcal{Y}|u(\cdot), \theta) \).
- Cubic scaling of computations with respect number of time points.

In this work, we propose a framework, which alleviates these problems.
Linear LFMs with certain covariance functions: standard GP regression techniques.

Non-linear LFMs need approximations.

Previous approaches:
- Laplace approximation (Lawrence et al. 2007)
- Markov chain Monte Carlo (Titsias et al., 2009)

Disadvantages:
- Need the solution of $x(t)$ explicitly as a function of $u(t)$ to numerically calculate the needed likelihood $p(Y|u(·), \theta)$.
- Cubic scaling of computations with respect number of time points.

In this work, we propose a framework, which alleviates these problems.
Linear LFMs with certain covariance functions: standard GP regression techniques.

Non-linear LFMs need approximations.

Previous approaches:
- Laplace approximation (Lawrence et al. 2007)
- Markov chain Monte Carlo (Titsias et al., 2009)

Disadvantages:
- Need the solution of $x(t)$ explicitly as a function of $u(t)$ to (numerically) calculate the needed likelihood $p(Y|u(\cdot), \theta)$.
- Cubic scaling of computations with respect number of time points.

In this work, we propose a framework, which alleviates these problems.
Inference with Latent Force Models

- Linear LFMs with **certain** covariance functions: **standard GP regression techniques**.
- Non-linear LFMs need approximations.
- Previous approaches:
  - Laplace approximation (Lawrence et al. 2007)
  - Markov chain Monte Carlo (Titsias et al., 2009)
- Disadvantages:
  - Need the solution of $x(t)$ explicitly as a function of $u(t)$ to (numerically) calculate the needed likelihood $p(Y|u(\cdot), \theta)$.
  - **Cubic** scaling of computations with respect number of time points.
- In this work, we propose a framework, which alleviates these problems.
• Linear LFMs with certain covariance functions: standard GP regression techniques.
• Non-linear LFMs need approximations.
• Previous approaches:
  • Laplace approximation (Lawrence et al. 2007)
  • Markov chain Monte Carlo (Titsias et al., 2009)
• Disadvantages:
  • Need the solution of $x(t)$ explicitly as a function of $u(t)$ to (numerically) calculate the needed likelihood $p(Y|u(\cdot), \theta)$.
  • Cubic scaling of computations with respect number of time points.
• In this work, we propose a framework, which alleviates these problems.
Inference with Latent Force Models

- Linear LFMs with certain covariance functions: standard GP regression techniques.
- Non-linear LFMs need approximations.
- Previous approaches:
  - Laplace approximation (Lawrence et al. 2007)
  - Markov chain Monte Carlo (Titsias et al., 2009)
- Disadvantages:
  - Need the solution of \( x(t) \) explicitly as a function of \( u(t) \) to (numerically) calculate the needed likelihood \( p(\mathcal{Y}|u(\cdot), \theta) \).
  - Cubic scaling of computations with respect number of time points.
- In this work, we propose a framework, which alleviates these problems.
Linear LFMs with certain covariance functions: standard GP regression techniques.

Non-linear LFMs need approximations.

Previous approaches:
- Laplace approximation (Lawrence et al. 2007)
- Markov chain Monte Carlo (Titsias et al., 2009)

Disadvantages:
- Need the solution of $x(t)$ explicitly as a function of $u(t)$ to (numerically) calculate the needed likelihood $p(Y|u(\cdot), \theta)$.
- Cubic scaling of computations with respect number of time points.

In this work, we propose a framework, which alleviates these problems.
Contents
1 Latent Force Models
2 SDE View of Latent Force Models
3 Filtering and Smoothing of SDEs
4 Simulated Experiments
5 GPS Satellite Orbit Prediction
6 Conclusion
GPs with certain covariance functions (including the Matérn class) can be represented as solutions to linear time-invariant (LTI) stochastic differential equations (SDEs).

Using this, the GP prior on $u(t)$ can be formed as LTI SDE of form

$$dz(t) = F \cdot z(t) \, dt + L \cdot d\beta(t)$$

where $z(t) = (u(t) \, \frac{du(t)}{dt} \ldots \frac{d^{p-1}u_{r(t)}}{dt^{p-1}}) \, T$ and

$$F = \begin{pmatrix} 0 & 1 & \cdots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ -a^0 & \cdots & 0 & 1 \\ \cdots & \cdots & -a^{p-2} & -a^{p-1} \end{pmatrix}, \quad L = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ q \end{pmatrix}.$$
GPs with certain covariance functions (including the Matérn class) can be represented as solutions to linear time-invariant (LTI) stochastic differential equations (SDEs).

Using this, the GP prior on $u(t)$ can be formed as LTI SDE of form

$$dz(t) = F \, z(t) \, dt + L \, d\beta(t)$$

where $z(t) = (u(t) \frac{du(t)}{dt} \ldots \frac{d^{p-1}u_r(t)}{dt^{p-1}})^T$ and

$$F = \begin{pmatrix} 0 & 1 & & \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -a^0 & \ldots & -a^{p-2} & -a^{p-1} \end{pmatrix}, \quad L = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ q \end{pmatrix}.$$
By combining this view with the dynamic model of \( x(t) \), the overall latent force model can be formulated as a

\[
dx(t) = f(x(t), u(t), t) \, dt,
\]
\[
dz_r(t) = F_r z_r(t) \, dt + L_r \, d\beta_r(t), \quad r = 1, \ldots, R
\]

The measurements at time instants \( t_1, \ldots, t_T \) can be modeled as

\[
y_k = h_k(x(t_k)) + r_k, \quad k = 1, \ldots, T.
\]

where \( h(\cdot) \) is the measurement model function and \( r_k \sim N(0, R_k) \) is the measurement noise.
By combining this view with the dynamic model of $x(t)$, the overall latent force model can be formulated as a

$$
dx(t) = f(x(t), u(t), t) \, dt,$$

$$
dz_r(t) = F_r \, z_r(t) \, dt + L_r \, d\beta_r(t), \; r = 1, \ldots, R
$$

The measurements at time instants $t_1, \ldots, t_T$ can be modeled as

$$
y_k = h_k(x(t_k)) + r_k, \; k = 1, \ldots, T. \quad (4)
$$

where $h(\cdot)$ is the measurement model function and $r_k \sim N(0, R_k)$ is the measurement noise.
Actually, we are interested in inferring the state of continuous-discrete system of the form

\[ d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t) \, dt + \mathbf{L}(\mathbf{x}(t), t) \, d\beta(t) \]
\[ \mathbf{y}_k = \mathbf{h}_k(\mathbf{x}(t_k)) + \mathbf{r}_k, \quad k = 1, \ldots, T. \] (5)

In particular, we aim to infer the filtering and smoothing distributions at a time instant \( t \):

\[ p(\mathbf{x}(t)|\mathcal{Y}_{1:k}), \quad t \in [t_k, t_{k+1}) \] (6)

and

\[ p(\mathbf{x}(t)|\mathcal{Y}_{1:T}), \quad t \in [t_0, t_T], \] (7)

where \( \mathcal{Y}_M = \{\mathbf{y}_1, \ldots, \mathbf{y}_M\} \).
Actually, we are interested in inferring the state of continuous-discrete
system of the form

\[ d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t), t) \, dt + \mathbf{L}(\mathbf{x}(t), t) \, d\beta(t) \]

\[ \mathbf{y}_k = \mathbf{h}_k(\mathbf{x}(t_k)) + \mathbf{r}_k, \quad k = 1, \ldots, T. \]

In particular, we aim to infer the filtering and smoothing distributions
at a time instant \( t \):

\[ p(\mathbf{x}(t)|\mathcal{Y}_{1:k}), \quad t \in [t_k, t_{k+1}) \]

and

\[ p(\mathbf{x}(t)|\mathcal{Y}_{1:T}), \quad t \in [t_0, t_T], \]

where \( \mathcal{Y}_M = \{\mathbf{y}_1, \ldots, \mathbf{y}_M\} \).
Classic Bayesian filtering theory provides an efficient recursive solution – but requires approximations as well.

Here we form a Gaussian approximation

$$p(x(t) | \mathcal{Y}_k) \approx N(x(t) | m(t), P(t)).$$

Approximation methods from non-linear Kalman filtering.

Another way would be to use a particle filter, but it can require very high amount of particles to be precise enough.
Classic Bayesian filtering theory provides an efficient recursive solution – but requires approximations as well.

Here we form a Gaussian approximation

\[ p(x(t) | \mathcal{Y}_k) \approx N(x(t) | m(t), P(t)). \]

Approximation methods from non-linear Kalman filtering.

Another way would be to use a particle filter, but it can require very high amount of particles to be precise enough.
• Classic Bayesian filtering theory provides an efficient recursive solution – but requires approximations as well.

• Here we form a Gaussian approximation

\[ p(x(t) | \mathcal{Y}_k) \approx N(x(t) | m(t), P(t)) . \]

• Approximation methods from non-linear Kalman filtering.

• Another way would be to use a particle filter, but it can require very high amount of particles to be precise enough.
Classic Bayesian filtering theory provides an efficient recursive solution – but requires approximations as well.

Here we form a Gaussian approximation

\[
p(x(t) | Y_k) \approx N(x(t) | m(t), P(t)).
\]

Approximation methods from non-linear Kalman filtering.

Another way would be to use a particle filter, but it can require very high amount of particles to be precise enough.
Gaussian Continuous-Discrete Filter

- For each measurement $y_k$ for $k = 1, \ldots, T$ do:
  1. **Prediction step:** Integrate the following ODEs from time $t_{k-1}$ to $t_k$:
     
     $$ \frac{dm(t)}{dt} = \int f(x, t) N(x | m(t), P(t)) \, dx $$
     $$ \frac{dP(t)}{dt} = \int f(x, t) (x - m(t))^T N(x | m(t), P(t)) \, dx $$
     $$ + \int (x - m(t)) f(x, t)^T N(x | m(t), P(t)) \, dx $$
     $$ + \int L(x, t) Q L^T(x, t) N(x | m(t), P(t)) \, dx. $$

  2. **Update step:** Use Bayes’ rule, which is the (non-linear) Kalman filter update step and gives a Gaussian result:
     $$ p(x(t_k) | Y_k) \approx N(x(t_k) | m(t_k), P(t_k)). $$
Gaussian Continuous-Discrete Filter

For each measurement $y_k$ for $k = 1, \ldots, T$ do:

1. **Prediction step:** Integrate the following ODEs from time $t_{k-1}$ to $t_k$:

\[
\frac{dm(t)}{dt} = \int f(x, t) N(x | m(t), P(t)) \, dx \\
\frac{dP(t)}{dt} = \int f(x, t) (x - m(t))^T N(x | m(t), P(t)) \, dx \\
+ \int (x - m(t)) f(x, t)^T N(x | m(t), P(t)) \, dx \\
+ \int L(x, t) Q L^T(x, t) N(x | m(t), P(t)) \, dx.
\]

2. **Update step:** Use Bayes’ rule, which is the (non-linear) Kalman filter update step and gives a Gaussian result:

\[
p(x(t_k) | Y_k) \approx N(x(t_k) | m(t_k), P(t_k)).
\]
Gaussian Continuous-Discrete Filter

For each measurement \( y_k \) for \( k = 1, \ldots, T \) do:

1. **Prediction step:** Integrate the following ODEs from time \( t_{k-1} \) to \( t_k \):

\[
\frac{d\mathbf{m}(t)}{dt} = \int f(x, t) N(x | \mathbf{m}(t), P(t)) \, dx
\]
\[
\frac{dP(t)}{dt} = \int f(x, t) (x - \mathbf{m}(t))^T N(x | \mathbf{m}(t), P(t)) \, dx
\]
\[
+ \int (x - \mathbf{m}(t)) f(x, t)^T N(x | \mathbf{m}(t), P(t)) \, dx
\]
\[
+ \int L(x, t) Q L^T (x, t) N(x | \mathbf{m}(t), P(t)) \, dx.
\]

2. **Update step:** Use Bayes’ rule, which is the (non-linear) Kalman filter update step and gives a Gaussian result:

\[
p(\mathbf{x}(t_k) | Y_k) \approx N(\mathbf{x}(t_k) | \mathbf{m}(t_k), P(t_k)).
\]
The smoothing distributions \( p(x(t)|y_{1:T}) \approx \mathcal{N}(x(t)|m^s(t), P^s(t)) \) can be obtained by a simple backward recursion.

The filter also provides conditional likelihood approximations

\[
p(y_k | Y_{k-1}, \theta) \approx \mathcal{N}(y_k | \mu_k, S_k).
\]

Usable in parameter estimation (MAP, MCMC etc.) via the chain-rule

\[
p(Y_T | \theta) = \prod_{k=1}^{T} p(y_k | Y_{k-1}, \theta) \approx \prod_{k=1}^{T} \mathcal{N}(y_k | \mu_k, S_k).
\]
The smoothing distributions $p(x(t)|y_{1:T}) \approx N(x(t)|m^s(t), P^s(t))$ can be obtained by a simple backward recursion.

The filter also provides *conditional likelihood* approximations

$$p(y_k | Y_{k-1}, \theta) \approx N(y_k | \mu_k, S_k).$$

Usable in *parameter estimation* (MAP, MCMC etc.) via the chain-rule

$$p(Y_T | \theta) = \prod_{k=1}^{T} p(y_k | Y_{k-1}, \theta) \approx \prod_{k=1}^{T} N(y_k | \mu_k, S_k).$$
The smoothing distributions \( p(x(t) | y_{1:T}) \approx N(x(t) | m^s(t), P^s(t)) \) can be obtained by a simple backward recursion.

The filter also provides **conditional likelihood** approximations

\[
p(y_k | \mathcal{Y}_{k-1}, \theta) \approx N(y_k | \mu_k, S_k).
\]

Usable in **parameter estimation** (MAP, MCMC etc.) via the chain-rule

\[
p(\mathcal{Y}_T | \theta) = \prod_{k=1}^{T} p(y_k | \mathcal{Y}_{k-1}, \theta) \approx \prod_{k=1}^{T} N(y_k | \mu_k, S_k).
\]
Contents

1. Latent Force Models
2. SDE View of Latent Force Models
3. Filtering and Smoothing of SDEs
4. Simulated Experiments
5. GPS Satellite Orbit Prediction
6. Conclusion
Comparison to state-of-the-art (Laplace and MCMC).

Transcription factor model (Lawrence et al. 2007)

\[
\frac{dx_j(t)}{dt} = B_j + \sum_{r=1}^{R} S_{j,r} g_j(u_r(t)) - D_j x_j(t), \quad j = 1, \ldots, N
\]

with \( N = 3, \, R = 1 \) and \( T = 13 \).

Randomized parameters \( \theta = \{B_j, S_j, D_j, A_j\}_{j=1}^{N} \) over 100 simulations.

Tested functions: \( g(u(t)) = \frac{e^{u(t)}}{\gamma + e^{u(t)}} \) (saturation), \( g(u(t)) = \frac{1}{\gamma + e^{u(t)}} \) (repression) and \( g(u(t)) = e^{u(t)} \) (exponential).

Criterions: root-mean-square error (RMSE) over finer time grid and number of divergences (DIV).
Comparison to state-of-the-art (Laplace and MCMC).

Transcription factor model (Lawrence et al. 2007)

\[
\frac{dx_j(t)}{dt} = B_j + \sum_{r=1}^{R} S_{j,r} g_j(u_r(t)) - D_j x_j(t), \ j = 1, \ldots, N
\]

with \( N = 3 \), \( R = 1 \) and \( T = 13 \).

Randomized parameters \( \theta = \{B_j, S_j, D_j, A_j\}_{j=1}^{N} \) over 100 simulations.

Tested functions: \( g(u(t)) = \frac{e^{u(t)}}{\gamma + e^{u(t)}} \) (saturation), \( g(u(t)) = \frac{1}{\gamma + e^{u(t)}} \) (repression) and \( g(u(t)) = e^{u(t)} \) (exponential).

Criterions: root-mean-square error (RMSE) over finer time grid and number of divergences (DIV).
Comparison to state-of-the-art (Laplace and MCMC).

Transcription factor model (Lawrence et al. 2007)

\[
\frac{dx_j(t)}{dt} = B_j + \sum_{r=1}^{R} S_{j,r} g_j(u_r(t)) - D_j x_j(t), \quad j = 1, \ldots, N
\]

with \( N = 3, R = 1 \) and \( T = 13 \).

Randomized parameters \( \theta = \{B_j, S_j, D_j, A_j\}_{j=1}^{N} \) over 100 simulations.

Tested functions: \( g(u(t)) = \frac{e^{u(t)}}{\gamma + e^{u(t)}} \) (saturation), \( g(u(t)) = \frac{1}{\gamma + e^{u(t)}} \) (repression) and \( g(u(t)) = e^{u(t)} \) (exponential).

Criterions: root-mean-square error (RMSE) over finer time grid and number of divergences (DIV).
Estimation Transcription Factors 1/2

- Comparison to state-of-the-art (Laplace and MCMC).
- Transcription factor model (Lawrence et al. 2007)

\[
\frac{dx_j(t)}{dt} = B_j + \sum_{r=1}^{R} S_{j,r} g_j(u_r(t)) - D_j x_j(t), \quad j = 1, \ldots, N
\]

with \(N = 3, R = 1\) and \(T = 13\).

- Randomized parameters \(\theta = \{B_j, S_j, D_j, A_j\}_{j=1}^{N}\) over 100 simulations.

- Tested functions: \(g(u(t)) = \frac{e^{u(t)}}{\gamma + e^{u(t)}}\) (saturation), \(g(u(t)) = \frac{1}{\gamma + e^{u(t)}}\) (repression) and \(g(u(t)) = e^{u(t)}\) (exponential).

- Criterions: root-mean-square error (RMSE) over finer time grid and number of divergences (DIV).
Comparison to state-of-the-art (Laplace and MCMC).

Transcription factor model (Lawrence et al. 2007)

\[
\frac{dx_j(t)}{dt} = B_j + \sum_{r=1}^{R} S_{j,r} g_j(u_r(t)) - D_j x_j(t), \quad j = 1, \ldots, N
\]

with \( N = 3, \quad R = 1 \) and \( T = 13 \).

Randomized parameters \( \theta = \{B_j, S_j, D_j, A_j\}_{j=1}^{N} \) over 100 simulations.

Tested functions: \( g(u(t)) = \frac{e^{u(t)}}{\gamma + e^{u(t)}} \) (saturation), \( g(u(t)) = \frac{1}{\gamma + e^{u(t)}} \) (repression) and \( g(u(t)) = e^{u(t)} \) (exponential).

Criterions: root-mean-square error (RMSE) over finer time grid and number of divergences (DIV).
<table>
<thead>
<tr>
<th></th>
<th>LA</th>
<th>ESLS</th>
<th>GFS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Saturation</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\gamma = 0.1$</td>
<td>1.758 (0)</td>
<td>0.737 (0)</td>
<td><strong>0.720 (0)</strong></td>
</tr>
<tr>
<td>$\gamma = 0.5$</td>
<td>0.737 (0)</td>
<td>0.484 (0)</td>
<td><strong>0.483 (0)</strong></td>
</tr>
<tr>
<td>$\gamma = 1$</td>
<td>0.489 (0)</td>
<td><strong>0.483 (0)</strong></td>
<td>0.484 (0)</td>
</tr>
<tr>
<td><strong>Repression</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\gamma = 0.1$</td>
<td>1.933 (40)</td>
<td><strong>0.327 (0)</strong></td>
<td>0.374 (0)</td>
</tr>
<tr>
<td>$\gamma = 0.5$</td>
<td>1.267 (1)</td>
<td><strong>0.363 (0)</strong></td>
<td>0.367 (0)</td>
</tr>
<tr>
<td>$\gamma = 1$</td>
<td>0.483 (0)</td>
<td>0.476 (0)</td>
<td><strong>0.474 (0)</strong></td>
</tr>
<tr>
<td><strong>Exponential</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.300 (55)</td>
<td><strong>0.290 (0)</strong></td>
<td>0.358 (9)</td>
</tr>
</tbody>
</table>
Contents

1. Latent Force Models
2. SDE View of Latent Force Models
3. Filtering and Smoothing of SDEs
4. Simulated Experiments
5. GPS Satellite Orbit Prediction
6. Conclusion
Orbit prediction is needed in a number of real-time applications.

- For example, can improve Time To First Fix (TTFF) when network is not available (e.g. Assisted GPS, A-GPS).
- Previous prediction methods not accurate enough for longer time periods (over a week or so).
- Here we use the LFM approach to improve the predictions of a deterministic physical model.
- Computational efficiency is important – predictions are typically performed on embedded systems (stand-alone GPS devices or mobile handsets)
 Orbit prediction is needed in a number of real-time applications.

For example, can improve Time To First Fix (TTFF) when network is not available (e.g. Assisted GPS, A-GPS).

Previous prediction methods not accurate enough for longer time periods (over a week or so).

Here we use the LFM approach to improve the predictions of a deterministic physical model.

Computational efficiency is important – predictions are typically performed on embedded systems (stand-alone GPS devices or mobile handsets).
Orbit prediction is needed in a number of real-time applications.

For example, can improve Time To First Fix (TTFF) when network is not available (e.g. Assisted GPS, A-GPS).

Previous prediction methods not accurate enough for longer time periods (over a week or so).

Here we use the LFM approach to improve the predictions of a deterministic physical model.

Computational efficiency is important – predictions are typically performed on embedded systems (stand-alone GPS devices or mobile handsets)
Orbit prediction is needed in a number of real-time applications.

For example, can improve Time To First Fix (TTFF) when network is not available (e.g. Assisted GPS, A-GPS).

Previous prediction methods not accurate enough for longer time periods (over a week or so).

Here we use the LFM approach to improve the predictions of a deterministic physical model.

Computational efficiency is important – predictions are typically performed on embedded systems (stand-alone GPS devices or mobile handsets)
GPS Satellite Orbit Prediction

- Orbit prediction is needed in a number of real-time applications.
- For example, can improve Time To First Fix (TTFF) when network is not available (e.g. Assisted GPS, A-GPS).
- Previous prediction methods not accurate enough for longer time periods (over a week or so).
- Here we use the LFM approach to improve the predictions of a deterministic physical model.
- Computational efficiency is important – predictions are typically performed on embedded systems (stand-alone GPS devices or mobile handsets).
Motion Model

- The equation of motion for the satellite can be written as

\[
\frac{d}{dt} \begin{bmatrix} r \\ v \end{bmatrix} = \begin{bmatrix} v \\ a(r, t) + u(r, v, t) \end{bmatrix}.
\]

- The deterministic model for acceleration is

\[
a(r, t) = a_g + a_{\text{moon}} + a_{\text{sun}} + a_{\text{srp}}.
\]

- Most modelling errors reside in the solar radiation pressure \( a_{\text{srp}} \).

- In this work we model the unknown forces \( u(r, v, t) \) non-parametrically.
The equation of motion for the satellite can be written as

\[
\frac{d}{dt} \begin{bmatrix} r \\ v \end{bmatrix} = \begin{bmatrix} v \\ a(r, t) + u(r, v, t) \end{bmatrix}.
\]

The deterministic model for acceleration is

\[
a(r, t) = a_g + a_{moon} + a_{sun} + a_{srp}.
\]

Most modelling errors reside in the solar radiation pressure \( a_{srp} \).

In this work we model the unknown forces \( u(r, v, t) \) non-parametrically.
The equation of motion for the satellite can be written as

\[
\frac{d}{dt} \begin{bmatrix} r \\ v \end{bmatrix} = \begin{bmatrix} v \\ a(r, t) + u(r, v, t) \end{bmatrix}.
\]

The deterministic model for acceleration is

\[
a(r, t) = a_g + a_{\text{moon}} + a_{\text{sun}} + a_{\text{srp}}.
\]

Most modelling errors reside in the solar radiation pressure \( a_{\text{srp}} \).

In this work we model the unknown forces \( u(r, v, t) \) non-parametrically.
The equation of motion for the satellite can be written as
\[
\frac{d}{dt} \begin{bmatrix} r \\ v \end{bmatrix} = \begin{bmatrix} v \\ a(r, t) + u(r, v, t) \end{bmatrix}.
\]

The deterministic model for acceleration is
\[
a(r, t) = a_g + a_{\text{moon}} + a_{\text{sun}} + a_{\text{srp}}.
\]

Most modelling errors reside in the solar radiation pressure \( a_{\text{srp}} \).

In this work we model the unknown forces \( u(r, v, t) \) non-parametrically.
First step of modelling: assume smoothness priors (of Matérn form) for each component of $\mathbf{u}(r, v, t)$ (in RTN coordinates).

Forces clearly exhibit quasi-periodic behaviour with a period of little less than one day.
Quasi-Periodic Model for Latent Forces

- We modelled the quasi-periodicities with a superposition of resonators

\[
\frac{d^2 c_n(t)}{dt^2} = -(2\pi nf)^2 c_n(t) + w_n(t), \quad n = 1, \ldots, N.
\]

- Can be written in LTI SDE form.

(a) \(c_1(t)\)  
(b) \(c_2(t)\)  
(c) \(c_3(t)\)  
(d) \(\sum_{i=1}^{3} c_i(t)\)
Comparison of deterministic and latent force models on an online prediction scenario with 30 days of data.

The observations (with sampling period of 15 minutes) on only the gray shaded intervals were used.

(a) Deterministic model pos. error  
(b) Latent force model pos. error
Contents

1. Latent Force Models
2. SDE View of Latent Force Models
3. Filtering and Smoothing of SDEs
4. Simulated Experiments
5. GPS Satellite Orbit Prediction
6. Conclusion
We have reformulated non-linear LFMs as partially observed SDEs.

Enables the use of sequential algorithms for state and parameter inference in linear time.

Does not need explicit solutions of the underlying dynamic system.

The proposed Gaussian approximation is computationally efficient but might be inadequate in some cases.

Works well in many practical applications.