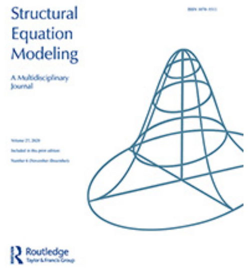

Structural Equation Modeling: A Multidisciplinary Journal

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/hsem20

A Computationally More Efficient Bayesian Approach for Estimating Continuous-Time Models

Martin Hecht & Steffen Zitzmann

STRUCTURAL EQUATION MODELING: A MULTIDISCIPLINARY JOURNAL
2020, VOL. 27, NO. 6, 829–840
https://doi.org/10.1080/10705511.2020.1719107

Routledge
Taylor & Francis Group

OPEN ACCESS Check for updates

A Computationally More Efficient Bayesian Approach for Estimating Continuous-Time Models

Martin Hecht[1] and Steffen Zitzmann[2]

[1]Humboldt-Universität zu Berlin; [2]University of Kiel

ABSTRACT

Continuous-time modeling is gaining in popularity as more and more intensive longitudinal data need to be analyzed. Current Bayesian software implementations of continuous-time models suffer from rather high, inadequate run times. Therefore, we apply a model reformulation approach to reduce run time. In a simulation study, we investigate the estimation quality and run time gain. We then illustrate our optimized Bayesian continuous-time model estimation and compare it to established continuous-time modeling software using an empirical example. Parameter estimates and inference statistics were very comparable, while run times were very different. Our approach reduces the run times for Bayesian estimations of continuous-time models from hours to minutes.

KEYWORDS

Continuous-time modeling; Bayesian analysis; run time optimization; intensive longitudinal data; structural equation modeling

Continuous-time modeling has finally found its way into psychological and behavioral research – one indication of which is the recently published edited volume by Van Montfort, Oud, and Voelkle (2018). The increasing popularity of intensive longitudinal data from experience sampling (ES), ecological momentary assessment (EMA), and ambulatory assessment (AA), has fueled the growing interest in continuous-time models because they are inherently well suited to handling unequally spaced measurement occasions and individualized assessment designs. Furthermore, continuous-time models make it possible to compare the results of studies with different time intervals between measurement occasions.

Several software solutions for estimating continuous-time models have been introduced, such as the R package ctsem, which includes both a frequentist (Driver, Oud, & Voelkle, 2017) and a Bayesian estimation module (Driver & Voelkle, 2018). Whereas frequentist estimation in ctsem seems appropriately fast, model run times in the Bayesian ctsem are rather high. For instance, Hecht and Voelkle (2019) report that “[r]un time and RAM usage of the frequentist estimation were barely noticeable, whereas the Bayesian estimation needed approximately 2.64GB RAM and 2 days and 20 hours run time” for a continuous-time model using real data. Hence, run time is definitely an obstacle to more widespread use of existing Bayesian estimation software for continuous-time models. One solution might be to simply switch to the faster frequentist estimation approach; however, users might wish to benefit from the advantages of Bayesian estimation, which include the ability to include previous knowledge, the potential to estimate otherwise intractable models (e.g., van de Schoot, Winter, Ryan, Zondervan-Zwijnenburg, & Depaoli, 2017), and the stabilization of parameter estimates (e.g., Zitzmann, 2018).

Many approaches for run time optimization exist and could potentially be implemented in Bayesian model estimation. One promising approach comes from Hecht, Gische, Vogel, and Zitzmann (2019), who suggest reformulating the model in terms of the model implied covariance matrix and model implied mean vector (which can be easily derived using a structural equation modeling [SEM] framework), and using these for MCMC sampling. By applying this method, the information to be modeled is reduced from possibly millions of data points to one sample scatter matrix (with a Wishart distribution) and one sample mean vector (with a multivariate normal distribution). One of the few drawbacks of this approach is that individual person parameters (random effects) are no longer part of the model and thus cannot be estimated. However, this is also the case in the frequentist likelihood approaches for estimating structural equation models many users are accustomed to. Moreover, as in SEM, individual score methods (e.g., Hardt, Hecht, Oud, & Voelkle, 2019; Hardt, Hecht, & Voelkle, 2019) can be used to estimate person parameters if there is substantive interest in them, for instance, for individual diagnostic purposes. Another issue is that the distributional forms of the sample scatter matrix and the sample mean vector need to be known and Bayesian samplers for these distributions need to be provided. Fortunately, for the most common two-level data structure, the sample scatter matrix has a Wishart distribution, which is included in most general multi-purpose Bayesian software. Continuous-time models involve this two-level data structure (repeatedly measured values nested within persons); hence, we can straightforwardly implement the model reformulation approach for Bayesian estimation of continuous-time models.

CONTACT Martin Hecht martin.hecht@hu-berlin.de Department of Psychology, Humboldt-Universität zu Berlin, Unter den Linden 6, 10099 Berlin, Germany.

Supplemental data for this article can be accessed here.

Purpose and scope

In the present work, we apply the approach proposed by Hecht, Gische, et al. (2019) to reduce the run time of a Bayesian univariate continuous-time model for unequally spaced assessment designs. We chose to use the Bayesian software JAGS for its flexibility and stability. The article is organized into the following sections. First, we present the univariate continuous-time model and describe the necessary model reformulations. Second, we report results from a simulation study in which we compared a classic Bayesian model implementation with the reformulated Bayesian model implementation with respect to run time and estimation quality. Third, we present a real data example for which we calculated estimates using our Bayesian JAGS implementations as well as the established R package ctsem. Finally, we conclude with a discussion of our work. Annotated R code, JAGS/BUGS code, ctsem code, and an example data file with which to run all presented analyses are provided in the Online Supplemental Material.

Optimization of the univariate continuous-time model

Unequal-interval non-individualized longitudinal designs involve responses of $j = 1, \ldots, J$ persons at several points in time, t_p, with $p = 1, \ldots, P$ being a running index denoting the discrete measurement occasion and P being the number of measurement occasions (see, e.g., Hecht, Hardt, Driver, & Voelkle, 2019, for details and illustrations). Time intervals Δ_{p-1} between measurement occasions are given by $\Delta_{p-1} = t_p - t_{p-1}$ for all $p \geq 2$, and y_{jp} is the manifest response of person j at measurement occasion p.

The continuous-time model is given by (adapted from Hecht, Hardt, et al., 2019; Oud & Delsing, 2010):

$$\text{for } p \geq 2,\ y_{jp} = a^*_{\Delta_{p-1}} y_{j(p-1)} + b^*_{j\Delta_{p-1}} + \omega_{j(p-1)}, \tag{1}$$

$$\text{with } \omega_{j(p-1)} \sim \mathcal{N}\left(0, q^*_{\Delta_{p-1}}\right), \tag{2}$$

$$q^*_{\Delta_{p-1}} = (2a)^{-1}\left[\exp(2a\Delta_{p-1}) - 1\right]q, \tag{3}$$

$$a^*_{\Delta_{p-1}} = \exp(a\Delta_{p-1}), \tag{4}$$

$$b^*_{j\Delta_{p-1}} = h^*_{\Delta_{p-1}} b_j, \tag{5}$$

$$\text{with } b_j \sim \mathcal{N}\left(b, \sigma^2_{\text{b}}\right) \text{ and} \tag{6}$$

$$h^*_{\Delta_{p-1}} = a^{-1}\left(a^*_{\Delta_{p-1}} - 1\right), \tag{7}$$

$$\text{and for } p = 1,\ y_{j1} \sim \mathcal{N}\left(\mu^*_{j\infty} + \mu_{\text{dev}}, \sigma^2_{\text{fw}}\right), \tag{8}$$

$$\text{with } \mu^*_{j\infty} = -a^{-1} b_j, \tag{9}$$

where a is the continuous-time auto-effect; q the continuous-time diffusion variance; b_j the person-specific continuous-time intercepts; $a^*_{\Delta_{p-1}}$,[1] $q^*_{\Delta_{p-1}}$, and $b^*_{j\Delta_{p-1}}$ their discrete-time counterparts; and $\omega_{j(p-1)}$ is the process error term. The person-specific continuous-time intercepts are normally distributed with mean b and variance σ^2_{b} and determine the person-specific long-range process means $\mu^*_{j\infty}$. At the first measurement occasion, the value y_{j1} for each person is assumed to be normally distributed with the person-specific process mean plus a deviation μ_{dev} as location and a within-person variance σ^2_{fw} (constant across persons). For more explanations, examples, and illustrations of this (and other) continuous-time models see Hecht, Hardt, et al. (2019), Hecht and Voelkle (2019), Driver et al. (2017), Driver and Voelkle (2018), and Voelkle, Oud, Davidov, and Schmidt (2012).

To stabilize the estimation, we express the discrete-time process intercepts in terms of the long-range process means – an approach also employed by Driver et al. (2017). Equivalent to Equations 1 and 6, this yields:

$$\text{for } p \geq 2, \\ y_{jp} = a^*_{\Delta_{p-1}} y_{j(p-1)} + \left(1 - a^*_{\Delta_{p-1}}\right)\mu^*_{j\infty} + \omega_{j(p-1)}, \tag{10}$$

$$\mu^*_{j\infty} \sim \mathcal{N}\left(\mu^*_{\infty}, \sigma^{2^*}_{\infty}\right), \tag{11}$$

$$\text{with } \sigma^{2^*}_{\infty} = a^{-2}\ \sigma^2_{\text{b}}, \tag{12}$$

The complete model is given by the autoregressive process formulation (Equation 10), the distributional assumption for the person-specific process means $\mu^*_{j\infty}$ (Equation 11), the distributional assumption for the values y_{j1} at the first measurement occasion (Equation 8), the distributional assumption for the process errors $\omega_{j(p-1)}$ (Equation 2), the function to relate discrete-time interval-dependent process error variances to the continuous-time diffusion variance q (Equation 3), and the function to relate discrete-time, interval-dependent autoregressive effects to the auto-effect a (Equation 4). This model is depicted in Figure 1 for $P = 3$ measurement occasions, with the estimated model parameters set in light-colored text on a dark background: the continuous-time auto-effect a, the continuous-time diffusion variance q, the between-person variance of the process means $\sigma^{2^*}_{\infty}$, the within-person variance at the first measurement occasion σ^2_{fw} (assumed constant across persons), the total process mean μ^*_{∞}, and the deviation of the mean at the first measurement occasion μ_{dev} from the total process mean.

To optimize the Bayesian estimation run time for this model, we use the approach described by Hecht, Gische, et al. (2019). The central idea is to use the sample scatter matrix **S** (also called the mean-corrected sums of squares and cross product [SSCP] matrix, Carroll & Green, 1997) and the sample mean vector $\bar{\mathbf{y}}$ instead of the raw response data. As we have a two-level data structure (i.e., repeatedly measured responses nested within persons), the sample scatter matrix has a Wishart distribution (e.g., Pham-Gia & Choulakian, 2014):

[1] In line with Oud and Delsing (2010) and Hecht, Hardt, et al. (2019) we use the asterisk symbol * to denote discrete-time parameters that can be calculated from continuous-time parameters. In the present article, we limited ourselves to first-order continuous-time models with auto-effects, a, in the range $(-\infty, 0)$, which implies discrete-time autoregressive effects, $a^*_{\Delta_{p-1}}$, in the range $(0, 1)$.

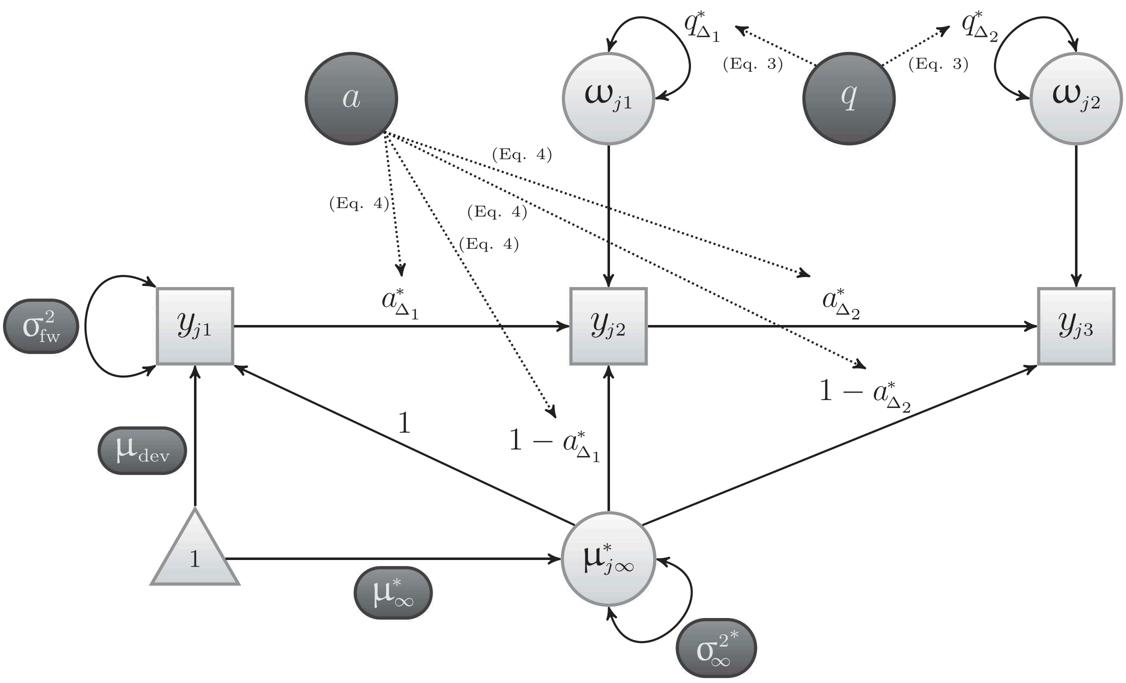

Figure 1. The univariate continuous-time model with three measurement occasions. Model parameters that are estimated are set in light text color on dark background.

$$\mathbf{S} \sim \mathcal{W}_P(\mathbf{\Sigma}, J - 1), \tag{13}$$

where the scale matrix $\mathbf{\Sigma}$ is the model implied covariance matrix and the degrees of freedom are $J - 1$. The sample mean vector is multivariate normally distributed (adapted from Flury, 1997, Chapter 4):

$$\bar{\mathbf{y}} \sim \mathcal{N}_P(\boldsymbol{\mu}, \frac{1}{J}\mathbf{\Sigma}). \tag{14}$$

where $\boldsymbol{\mu}$ is the model implied mean vector. To build the model implied covariance matrix and mean vector, any structural equation modeling framework such as LISREL (Jöreskog, Olsson, & Wallentin, 2016) or RAM (McArdle & McDonald, 1984; see also Boker, 2019) is a suitable and convenient choice. In Appendix A, we use the RAM approach to derive $\mathbf{\Sigma}$ and $\boldsymbol{\mu}$ and give the solutions in Table B1 in Appendix B.

Simulation study

In this simulation study, we show that our model reformulation approach leads to massive run time reductions. We chose the Bayesian software JAGS for its flexibility and widespread use, although the approach should be suitable for and easily implementable in the vast majority of Bayesian software programs (e.g., in the software Stan by Carpenter et al., 2017). As run time gains are of less value if they are at the expense of parameter estimation quality, we also include parameter recovery and precision statistics in our comparison.

Data generation

The data-generating model was the univariate continuous-time model described in Equations 10, 11, 8, 2, 3, and 4 and depicted in Figure 1 with true parameter values $a = -0.40$, $q = 0.40$, $\sigma^{2^*}_\infty = 0.50$, $\sigma^2_{\text{fw}} = 0.50$, $\mu^*_\infty = 0$, and $\mu_{\text{dev}} = 0$; $J = 2{,}000$ persons; and $P = 20$ measurement occasions. The full data-generating model is:

$$\mu^*_{j\infty} \sim \mathcal{N}(0, 0.50),$$

$$\text{for } p = 1,\ y_{j1} \sim \mathcal{N}\left(\mu^*_{j\infty} + 0, 0.50\right),$$

$$\text{for } p \geq 2,\ \ \Delta_{p-1} \sim \mathcal{U}_{\{0.25, 0.50, 0.75, 1, 1.25, 1.50, 1.75, 2\}},$$

$$q^*_{\Delta_{p-1}} = [2(-0.40)]^{-1}\{\exp[2(-0.40)\Delta_{p-1}] - 1\}0.40,$$

$$\omega_{j(p-1)} \sim \mathcal{N}\left(0, q^*_{\Delta_{p-1}}\right),$$

$$a^*_{\Delta_{p-1}} = \exp[(-0.40)\Delta_{p-1}],$$

$$y_{jp} = a^*_{\Delta_{p-1}} y_{j(p-1)} + \left(1 - a^*_{\Delta_{p-1}}\right)\mu^*_{j\infty} + \omega_{j(p-1)},$$

where $\mathcal{U}$ denotes a uniform distribution.

Analysis

We generated $N_{\text{repl}} = 1{,}000$ data sets, each of which was analyzed with three Bayesian implementations of the univariate continuous-time model: (1) The classic way to set up a Bayesian model is to use the model formulations that include individual parameters (Equations 10, 11, 8, 2, 3, and 4) and transfer those equations into Bayesian software syntax (*classic implementation*). We ran two versions of the classic implementation: one in which the individual parameters $\mu^*_{j\infty}$ and the individual state estimates were tracked, and one in which they were not. (2) In the second implementation, we used the reformulated model based on the model implied covariance matrix (Equation 13; *covariance-based implementation*). (3) In the third implementation, we added the estimation of means (Equation 14) to the covariance-based implementation, resulting in the *covariance- and mean-based implementation*. All models were estimated with JAGS 4.3.0 (Plummer, 2017) interfaced via the R package rjags 4–9 (Plummer, 2019) running on R version 3.6.1 (R Core Team, 2019). We chose a quite uninformative prior for each variance q, $\sigma^{2^*}_{\infty}$, and σ^2_{fw}, namely, an inverse gamma distribution with a shape and scale of 0.001. The auto-effect a had a truncated normal prior $\mathcal{N}(0, 10000)\text{T}(-\infty, -1/10^5]$ and both mean parameters, μ^*_{∞} and μ_{dev}, were assigned a more narrow normal prior distribution $\mathcal{N}(0, 9)$ in order to stabilize the estimation. Such stabilization techniques were developed for multilevel SEM (e.g., Chung, Gelman, Rabe-Hesketh, Liu, & Dorie, 2015; McNeish, 2016; Zitzmann, Lüdtke, Robitzsch, & Marsh, 2016) and have been shown to be an attractive choice, particularly when users run into issues with convergence (see Zitzmann, 2018). Starting values for the model parameters were random draws from the uniform distribution $\mathcal{U}(-3, -1/10^5)$ for a; from $\mathcal{N}(0, 1)\text{T}[1/10^5, \infty)$ (R package truncnorm, Mersmann, Trautmann, Steuer, & Bornkamp, 2018) for q, σ^{2*}_{∞}, and σ^2_{fw}; and from $\mathcal{N}(0, 1)$ for μ^*_{∞} and μ_{dev}. For all three Bayesian model implementations, the same 1,000 generated data sets and starting values were used. To monitor sampling and stop sampling when a certain stopping rule applies, we used the procedure described in Hecht, Gische, et al. (2019) with one MCMC chain, an iteration block size of 50, a burning share of 0.10, a thinning interval of 1 (no thinning), and maximum number of iterations = 200,000. Sampling was stopped when all parameters reached an effective sample size ESS $\geq$ 400 and a potential scale reduction factor PSR $\leq$ 1.001 (see Zitzmann & Hecht, 2019, for a discussion of these statistics and threshold values). ESS, PSR, and autocorrelations (AC) of the MCMC samples were computed with the R package shinystan (Gabry, 2018). After the stopping criteria were met, the mode of the converged chain served as the parameter estimate. We used the mode because it can be considered a natural extension of the ML estimator (see DeCarlo, Kim, & Johnson, 2011) and can outperform the mean and the median (e.g., Zitzmann, Lüdtke, & Robitzsch, 2015). However, we also checked whether using the mean instead of the mode made any difference in our analyses, and this was not the case. As parameter recovery and precision statistics, the bias, root mean squared error (RMSE), coverage rate, and standard error accuracy were calculated as described in Hecht, Hardt, et al. (2019, equations 30 and 32–35 in the Appendix), except that we used the mode instead of the mean as the parameter estimate. Each analysis was run on one Intel Xeon Gold 5120 (2.20 GHz) CPU of a 64-bit Linux Debian 9 "Stretch" computer.

Results

Table 1 shows the results and run times for all three implementations. All parameters were recovered very well in all implementations with practically no bias. The RMSE values were comparable across implementations. Coverage rates and standard error accuracy were very good for all parameters in all implementations. However, the implementations differed considerably with respect to run time. Whereas the classic implementation needed on average 50.1 min without tracking and 56.8 min with tracking of the individual parameters, the covariance-based implementation was roughly eight times faster ($M = 6.5$ min) and the covariance- and mean-based implementation was approximately four times faster ($M = 14.6$ min).[2] One mechanism contributing to the run time reduction can be identified by inspecting the autocorrelation of the MCMC chain. In the classic implementation, the autocorrelation was $M = 0.326$ on average, whereas it was much lower in the reformulated implementations ($M = 0.182$ and $M = 0.133$). Therefore, fewer iterations were needed to reach the ESS stopping criterion, resulting in faster estimation.

In summary, our model reformulation approach led to considerable run time reductions, while estimation quality was very high.

Empirical example

We use publicly available data from the 'Midlife in the United States (MIDUS 2): Daily Stress Project, 2004-2009' (Ryff & Almeida, 2017), a longitudinal study of health and well-being, to illustrate our Bayesian covariance- and mean-based implementation of the univariate continuous-time model. For reference purposes, we also report results obtained from Bayesian and frequentist estimation with the continuous-time modeling R package ctsem (Driver, Oud, & Voelkle, 2019). The MIDUS 2 data set contains data from 2,022 persons assessed at 8 measurement occasions; after data screening and preparation (see Hecht & Voelkle, 2019), 1,650 persons remained. For illustrative purposes, we focus on the variable "symptom severity" (B2DSYMAV, 1 = very mild, 10 = very severe). There are no missing values; thus, all persons have data for all measurement occasions (leading to a total of $8 \cdot 1{,}650 = 13{,}200$ observations).

The analyses were conducted on the same machine and with the same specifications and run parameters as described in the simulation study, except for the following modifications. The minimum required effective sample size was set to 1,000. In line with Hecht and Voelkle (2019), and because freely estimating the parameters at the first measurement occasion is not possible in the used Bayesian version of ctsem, we constrained the parameters for the first measurement occasion to the stationary process parameters. In our implementation, this is achieved by the constraints $\mu_{\text{dev}} = 0$ and $\sigma^2_{\text{fw}} = [-1/(2a)]q$. The location of the normal prior for μ^*_{∞} was set to the total mean of the sample data: $\mu^*_{\infty} \sim \mathcal{N}(2.4, 9)$. The raw MCMC chain (with 4,444 iterations) returned by ctStanFit() (Bayesian ctsem) was post-processed exactly like the chains from our JAGS implementations with 10% burnin and the mode as the parameter estimate.

Results and MCMC statistics are provided in Tables 2 and 3. The MCMC chains for all parameters in all Bayesian analyses converged (PSR very close to 1), and parameter precision was

[2] Some few analyses with extreme run times occurred. Inspecting these, we noticed that the starting value for at least one of the variances was very close to 0. Therefore, many more iterations were needed to reach convergence.

Table 1. Parameter quality and run time for three Bayesian implementations of the univariate continuous-time model (simulated data).

		Classic	Cov.-based	Cov./mean-based
Statistic	Parameter	Value	Value	Value
Bias	a	−0.000151	−0.000111	−0.000036
	q	−0.000023	−0.000018	−0.000028
	$\sigma^{2^*}_{\infty}$	−0.000251	−0.000119	−0.000047
	σ^2_{fw}	−0.000471	−0.000680	−0.000467
	μ^*_{∞}	0.000109	–	0.000181
	μ_{dev}	−0.000806	–	−0.000736
RMSE	a	0.0073	0.0073	0.0074
	q	0.0036	0.0036	0.0036
	$\sigma^{2^*}_{\infty}$	0.0188	0.0188	0.0190
	σ^2_{fw}	0.0187	0.0184	0.0187
	μ^*_{∞}	0.0177	–	0.0175
	μ_{dev}	0.0180	–	0.0181
Coverage rate 95%	a	0.948	0.948	0.949
	q	0.950	0.948	0.950
	$\sigma^{2^*}_{\infty}$	0.956	0.957	0.961
	σ^2_{fw}	0.957	0.960	0.957
	μ^*_{∞}	0.947	–	0.950
	μ_{dev}	0.942	–	0.934
SE accuracy	a	0.985	0.980	0.976
	q	1.015	1.008	1.001
	$\sigma^{2^*}_{\infty}$	1.021	1.019	1.011
	σ^2_{fw}	1.018	1.034	1.020
	μ^*_{∞}	0.996	–	1.007
	μ_{dev}	0.982	–	0.976
Run time (minutes)	M	50.1 / 56.8[1]	6.5	14.6
	min	13.2 / 16.4[1]	1.0	1.2
	$Q_{0.95}$	108.9 / 114.1[1]	11.7	16.2
	max	217.4 / 403.6[1]	330.7	3770.7
PSR	M	1.0000	0.9989	0.9990
	min	0.9991	0.9976	0.9976
	max	1.0010	1.0010	1.0010
ESS	M	1046	777	1018
	min	400	400	400
	max	5238	6409	119308
AC	M	0.326	0.182	0.133
	min	0.102	−0.111	−0.121
	max	0.607	0.718	0.760
Number of iterations	M	2380	1335	1637
	min	1200	700	700
	max	19950	42650	144700

Cov. = covariance, $Q_{0.95}$ = 0.95 quantile, N_{repl} = 1,000, PSR $\leq$ 1.001 and ESS $\geq$ 400 for all parameters, J = 2,000 persons, P = 20 measurement occasions, one chain ran on one Intel Xeon Gold 5120 (2.20 GHz) CPU of a 64-bit Linux Debian 9 "Stretch" computer. RMSE = root mean squared error; SE = standard error; PSR = potential scale reduction factor; ESS = effective sample size; AC = autocorrelation. [1] The first value is the run time without tracking, the second value with tracking of individual parameters.

Table 2. Results of the univariate continuous-time model for the variable "symptom severity" from the MIDUS 2 study.

	Classic			Covariance-/mean-based			Bayesian ctsem			Frequentist ctsem		
Parameter	Est.	LL95	UL95	Est.	LL95	UL95	Est.	LL95	UL95	Est.	LL95	UL95
a	−1.61	−1.74	−1.50	−1.62	−1.74	−1.50	−1.62	−1.74	−1.50	−1.61	−1.72	−1.49
q	6.22	5.91	6.70	6.30	5.90	6.69	6.26	5.90	6.69	6.26	5.87	6.65
$\sigma^{2^*}_{\infty}$	1.38	1.27	1.51	1.38	1.27	1.51	1.39	1.27	1.52	1.38	1.26	1.50
μ^*_{∞}	2.55	2.48	2.61	2.54	2.48	2.61	2.55	2.48	2.61	2.55	2.48	2.61
Run time (minutes)	51.0 / 78.7[1]			3.9			851.2			0.05		

J = 1,650 persons, P = 8 measurement occasions. LL95 and UL95 are the lower and upper limit of the 95% credible interval in the Bayesian models and of the 95% confidence interval in the frequentist model. The parameterizations of the software implementations differ, therefore, parameters were transformed for comparison. [1] The first value is the run time without tracking, the second value with tracking of individual parameters.

Table 3. Bayesian MCMC estimation statistics for the univariate continuous-time model (MIDUS 2 data).

	Classic			Covariance-/mean-based			Bayesian ctsem		
Parameter	PSR	ESS	AC	PSR	ESS	AC	PSR	ESS	AC
a	1.0008	1003	0.886	1.0000	1001	0.831	1.0000	1812	0.183
q	1.0009	1043	0.857	1.0001	1004	0.831	1.0000	2382	0.094
$\sigma^{2^*}_{\infty}$	1.0000	6212	0.387	0.9999	9982	0.005	1.0001	1177	0.299
μ^*_{∞}	1.0000	10269	0.208	0.9999	10080	−0.020	0.9998	1853	0.265

J = 1,650 persons, P = 8 measurement occasions. PSR = potential scale reduction factor; ESS = effective sample size; AC = autocorrelation. The parameterizations of the software implementations differ, therefore, parameters were transformed for comparison.

good (ESS $\geq$ 1,000). The estimates were very similar across all software implementations. The Bayesian credible interval widths and the frequentist confidence interval widths were very similar as well. We again observed pronounced differences regarding run time. The frequentist ctsem model achieved the fastest estimation (0.05 min). Second place goes to our covariance-/mean-based implementation, with a 3.9 min run time. The classic implementation ran for 51.0 min without tracking, and 78.7 min with tracking individual parameters. The longest run time by far, 851.2 min (roughly 14 h), was observed for Bayesian ctsem.

In summary, we have shown that our classic and reformulated Bayesian implementations produce similar results as the currently most prominent continuous-time modeling software ctsem, but are by far computationally more efficient than Bayesian ctsem.

Discussion

Run time for Bayesian models can be unsatisfactorily high, especially for more advanced and complex models such as continuous-time models. We employed the run time optimization approach described in Hecht, Gische, et al. (2019), which is based on a reformulation of the model in terms of covariances and means. This massively reduces the amount of information that needs to be modeled, leading to a massive reduction in run time. In an empirical example, we compared our approach to the leading continuous-time modeling software ctsem and found no differences in estimates, but large differences with respect to run time. Our approach reduces the run time of Bayesian estimations of continuous-time models from hours (possibly days) to minutes.

Several issues and limitations of our work need to be taken into consideration: (1) The model we employed was one of the simpler models from the class of continuous-time models, namely a univariate continuous-time model with between-person differences in process means. Moreover, the design used had unequal spacing between measurement occasions but was not individualized across persons. Future run time optimization research should address more complex continuous-time models, such as those involving multiple processes, measurement models, predictors, trends, and between-person variance in other parameters, and should extend the approach to unequal-interval individualized designs. (2) Our simulation study involved only one simulation design factor, namely, the Bayesian model implementations. All other factors (e.g., the number of persons and the number of measurement occasions) that might have an effect on run time were kept constant. Nevertheless, we believe that our approach should generalize to a large number of other conditions and should be especially advantageous for large sample sizes, because increasing of the sample size (by including more persons) is most likely neutral with respect to run time in our approach, whereas run time in classic Bayesian approaches increases with increasing sample size. We illustrate the plausibility of this effect in our example in the Online Supplemental Material, which involves data from $J = 10,000$ persons and $P = 3$ measurement occasions. For this example, the classic implementation runs over 2 h, whereas the reformulated implementations estimate the model parameters in less than 1 min. Furthermore, we ran our simulation again, this time with $J = 200$ instead of J = 2,000 persons. Even with many fewer persons, the reformulated models still yield a run time gain. The covariance-based model was roughly three times faster and the covariance- and mean-based model almost twice as fast as the classic implementation (see Table C1 in Appendix C). (3) To check whether deviations from the assumed normal distributions matter, we ran the simulation yet again with data generated from skewed normal distributions using the function rsnorm() from the R package fGarch (Würtz et al., 2019). We used a skewness of 2, as this value is one of the "realistic conditions that are worthwhile to explore" (Reinartz, Echambadi, & Chin, 2002, p. 231 and Table 1). The results of this simulation are presented in Table D1 in Appendix D. Bias is just a tiny bit larger (maximum absolute difference = 0.000528) than in the simulation study without skewness misspecifications. Coverage rates and standard error accuracies are marginally worse, but still quite good. Thus, our presented implementations are robust to violations of the normality assumption in the form of skewed distributions. However, in future research, it would be interesting to test whether the proposed implementations are robust to other forms of non-normality (e.g., kurtosis $\neq$ 0) and other model misspecifications. (4) We compared our approach to established continuous-time modeling software. Due to the elevated run time of the Bayesian ctsem estimation and limited computational resources, we could not include Bayesian ctsem in our simulation study; instead, we illustrated similarities and differences between the two software (JAGS vs. Bayesian ctsem) in a real data analysis and tried to ensure comparability to the greatest extent possible (e.g., by using the same post-processing of MCMC chains and assuring approximately equivalent effective sample sizes and convergence statistics). (5) The advantage of Bayesian ctsem over our covariance-/mean-based implementation is, of course, that individual state values are estimated and made available. This might be worth the additional run time. However, if researchers are not substantively interested in these parameters, there is no need to wait days for their estimation. Moreover, if individual parameters are of interest, our classic Bayesian implementation (with a tracking of these parameters) could be used, which is still by far computationally more efficient than Bayesian ctsem. Another approach would be to use individual score methods in a second step after estimating the model parameters. For autoregressive panel models, Hardt, Hecht, and Voelkle (2019) recommend either the Bartlett method, the regression method, or the Kalman filter. Run times for these methods do not seem to be an issue, as run times are often not even reported, and Estabrook and Neale (2013) speak of "trivial computation time" for obtaining Bartlett scores (p. 18). We additionally ran the Kalman filter for all replications in our simulation study using the R package FKF (Luethi, Erb, & Otziger, 2018). Run times for estimating 40,000 state values were $M = 0.34$ seconds on average (min = 0.26, max = 0.71). Thus, person score estimation as a second subsequent step requires virtually no additional computational time. (6) Frequentist estimation might be faster than Bayesian estimation (as shown in our empirical example). However, Bayesian estimation has several advantages (e.g., inclusion of previous knowledge, estimation of otherwise intractable models, stabilization of parameter estimates, a different (presumably more intuitive) philosophy for making inferences) that might be attractive to users. (7) We used JAGS because it is a popular and stable multipurpose Bayesian software. The proposed approach should easily work with other Bayesian software programs as well. However, the run time gain depends on the efficiency and speed of the implemented samplers and thus might differ across programs. (8) The presented implementations in their current form only work with complete data. One straightforward option for dealing with missing data without modifying our implementations is multiple imputation (e.g., Rubin, 1987). Alternatively, FIML-like procedures for handling missing data could be programmed into the Bayesian model. However, this would be a subject for future research. (9) Running multiple MCMC chains in parallel on multiple cores is another option to reduce run time even more and can be combined with all of the presented implementations. For example, Hecht and Voelkle (2019) report that a parallelization with 32 chains on 32 cores reduced the run time to one fourth of its previous value.

In conclusion, we have shown that Hecht, Gische, et al.'s (2019) model reformulation approach can be used for computationally more efficient Bayesian estimation of continuous-time models and have made Bayesian continuous-time modeling feasible for large sample analysis.

Acknowledgments

We thank Charles Driver and Manuel Voelkle for the many insightful discussions about continuous-time modeling over the past years. Many thanks also to Keri Hartman for carefully proofreading our manuscript.

Funding

We acknowledge support by the Open Access Publication Fund of Humboldt-Universität zu Berlin.

ORCID

Martin Hecht http://orcid.org/0000-0002-5168-4911

References

Boker, S. M. (2019). The reticular action model: A remarkably lasting achievement. In E. Ferrer, S. M. Boker, & K. J. Grimm (Eds.), *Longitudinal multivariate psychology* (pp. 126–142). New York, NY: Routledge.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., … Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of Statistical Software*, *76*, 1–32. doi:10.18637/jss.v076.i01

Carroll, J. D., & Green, P. E. (1997). Vector and matrix operations for multivariate analysis. In Eds. J. D. Carroll & P. E. Green, *Mathematical tools for applied multivariate analysis* (pp. 26–76). doi:10.1016/B978-012160954-2/50003-2

Chung, Y., Gelman, A., Rabe-Hesketh, S., Liu, J., & Dorie, V. (2015). Weakly informative prior for point estimation of covariance matrices in hierarchical models. *Journal of Educational and Behavioral Statistics*, *40*, 136–157. doi:10.3102/1076998615570945

DeCarlo, L. T., Kim, Y., & Johnson, M. S. (2011). A hierarchical rater model for constructed responses, with a signal detection rater model. *Journal of Educational Measurement*, *48*, 333–356. doi:10.1111/j.1745-3984.2011.00143.x

Driver, C. C., Oud, J. H. L., & Voelkle, M. C. (2017). Continuous time structural equation modelling with R package ctsem. *Journal of Statistical Software*, *77*, 1–35. doi:10.18637/jss.v077.i05

Driver, C. C., Oud, J. H. L., & Voelkle, M. C. (2019). ctsem: Continuous time structural equation modelling (Version 2.9.6) [Computer software]. Retrieved from http://CRAN.R-project.org/package=ctsem

Driver, C. C., & Voelkle, M. C. (2018). Hierarchical Bayesian continuous time dynamic modeling. *Psychological Methods*, *23*, 774–799. doi:10.1037/met0000168

Estabrook, R., & Neale, M. (2013). A comparison of factor score estimation methods in the presence of missing data: Reliability and an application to nicotine dependence. *Multivariate Behavioral Research*, *48*, 1–27. doi:10.1080/00273171.2012.730072

Flury, B. (1997). *A first course in multivariate statistics*. New York, NY: Springer.

Gabry, J. (2018). shinystan: Interactive visual and numerical diagnostics and posterior analysis for Bayesian models (Version 2.5.0) [Computer software]. Retrieved from http://cran.r-project.org/package=shinystan

Hardt, K., Hecht, M., Oud, J. H. L., & Voelkle, M. C. (2019). Where have the persons gone? – An illustration of individual score methods in autoregressive panel models. *Structural Equation Modeling*, *26*, 310–323. doi:10.1080/10705511.2018.1517355

Hardt, K., Hecht, M., & Voelkle, M. C. (2019). Robustness of individual score methods against model misspecification in autoregressive panel models. *Structural Equation Modeling*. Advance online publication. doi: 10.1080/10705511.2019.1642755

Hecht, M., Gische, C., Vogel, D., & Zitzmann, S. (2019). Integrating out nuisance parameters for computationally more efficient Bayesian estimation – An illustration and tutorial. *Structural Equation Modeling*. Advance online publication. doi:10.1080/10705511.2019.1647432

Hecht, M., Hardt, K., Driver, C. C., & Voelkle, M. C. (2019). Bayesian continuous-time Rasch models. *Psychological Methods*, *24*, 516–537. doi:10.1037/met0000205

Hecht, M., & Voelkle, M. C. (2019). Continuous-time modeling in prevention research: An illustration. *International Journal of Behavioral Development*. Advance online publication. doi:10.1177/0165025419885026

Jöreskog, K. G., Olsson, U. H., & Wallentin, F. Y. (2016). *Multivariate analysis with LISREL*. New York, NY: Springer International Publishing.

Luethi, D., Erb, P., & Otziger, S. (2018). FKF: Fast Kalman filter (Version 0.1.5) [Computer software]. Retrieved from http://CRAN.R-project.org/package=FKF

McArdle, J. J., & McDonald, R. P. (1984). Some algebraic properties of the reticular action model for moment structures. *British Journal of Mathematical and Statistical Psychology*, *37*, 234–251. doi:10.1111/j.2044-8317.1984.tb00802.x

McNeish, D. M. (2016). Using data-dependent priors to mitigate small sample bias in latent growth models: A discussion and illustration using M*plus*. *Journal of Educational and Behavioral Statistics*, *41*, 27–56. doi:10.3102/1076998615621299

Mersmann, O., Trautmann, H., Steuer, D., & Bornkamp, B. (2018). truncnorm: Truncated normal distribution (Version 1.0-8) [Computer software]. Retrieved from http://cran.r-project.org/package=truncnorm

Oud, J. H. L., & Delsing, M. J. M. H. (2010). Continuous time modeling of panel data by means of SEM. In K. van Montfort, J. H. L. Oud, & A. Satorra (Eds.), *Longitudinal research with latent variables* (pp. 201–244). Berlin, Germany: Springer.

Pham-Gia, T., & Choulakian, V. (2014). Distribution of the sample correlation matrix and applications. *Open Journal of Statistics*, *4*, 330–344. doi:10.4236/ojs.2014.45033

Plummer, M. (2017). JAGS (Version 4.3.0) [Computer software]. Retrieved from https://sourceforge.net/projects/mcmc-jags/files

Plummer, M. (2019). rjags: Bayesian graphical models using MCMC (Version 4-9) [Computer software]. Retrieved from http://cran.r-project.org/package=rjags

R Core Team. (2019). R: A language and environment for statistical computing (Version 3.6.1) [Computer Software]. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from http://www.R-project.org

Reinartz, W. J., Echambadi, R., & Chin, W. W. (2002). Generating nonnormal data for simulation of structural equation models using mattson's method. *Multivariate Behavioral Research*, *37*, 227–244. doi:10.1207/S15327906MBR3702_03

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. New York, NY: Wiley.

Ryff, C. D., & Almeida, D. M. (2017). *Midlife in the United States (MIDUS 2): Daily stress project, 2004–2009* [data file, documentation, and code book]. Ann Arbor, MI: Inter-university Consortium for Political and Social Research. doi:10.3886/ICPSR26841.v2

Van Montfort, K., Oud, J. H. L., & Voelkle, M. C. (Eds.). (2018). *Continuous time modeling in the behavioral and related sciences*. Cham, Switzerland: Springer.

Voelkle, M. C., Oud, J. H. L., Davidov, E., & Schmidt, P. (2012). An SEM approach to continuous time modeling of panel data: Relating authoritarianism and anomia. *Psychological Methods*, *17*, 176–192. doi:10.1037/a0027543

van de Schoot, R., Winter, S. D., Ryan, O., Zondervan-Zwijnenburg, M., & Depaoli, S. (2017). A systematic review of Bayesian articles in psychology: The last 25 years. *Psychological Methods*, *22*, 217–239. doi:10.1037/met0000100

Würtz, D., Setz, T., Chalabi, Y., Boudt, C., Chausse, P., & Miklovac, M. (2019). *fGarch: Rmetrics – autoregressive conditional heteroskedastic modelling (version 3042.83.1) [Computer software]*. Retrieved from http://CRAN.R-project.org/package=fGarch

Zitzmann, S. (2018). A computationally more efficient and more accurate stepwise approach for correcting for sampling error and measurement error. *Multivariate Behavioral Research*, *53*, 612–632. doi:10.1080/00273171.2018.1469086

Zitzmann, S., & Hecht, M. (2019). Going beyond convergence in Bayesian estimation: Why precision matters too and how to assess it. *Structural Equation Modeling*, *26*, 646–661. doi:10.1080/10705511.2018.1545232

Zitzmann, S., Lüdtke, O., & Robitzsch, A. (2015). A Bayesian approach to more stable estimates of group-level effects in contextual studies. *Multivariate Behavioral Research*, *50*, 688–705. doi:10.1080/00273171.2015.1090899

Zitzmann, S., Lüdtke, O., Robitzsch, A., & Marsh, H. W. (2016). A Bayesian approach for estimating multilevel latent contextual models. *Structural Equation Modeling*, *23*, 661–679. doi:10.1080/10705511.2016.1207179

Appendix A RAM model formulation

Here we specify the univariate continuous-time model (depicted in Figure 1 and described in Equations 10, 11, 8, 2, 3, and 4) within the RAM structural equation modeling framework (McArdle & McDonald, 1984; see also Boker, 2019) and derive the model implied covariance matrix and mean vector. Empty cell entries are zero. The manifest variables y_{jp} are directly mapped onto latent variables θ_{jp} (i.e., we have no measurement model). Building upon Equation 10, the **A** matrix is:

$$\underset{3P\times 3P}{\mathbf{A}} = \begin{array}{c|cccccccccccccccccc}
 & \theta_{j1} & \theta_{j2} & \theta_{j3} & \cdots & \theta_{j(P-1)} & \theta_{jP} & \omega_{j1} & \omega_{j2} & \cdots & \omega_{j(P-2)} & \omega_{j(P-1)} & \mu^*_{j\infty} & y_{j1} & y_{j2} & y_{j3} & \cdots & y_{j(P-1)} & y_{jP} \\
\hline
\theta_{j1} & & & & & & & & & & & & 1 & & & & & & \\
\theta_{j2} & a^*_{\Delta_1} & & & & & & 1 & & & & & 1-a^*_{\Delta_1} & & & & & & \\
\theta_{j3} & & a^*_{\Delta_2} & & & & & & 1 & & & & 1-a^*_{\Delta 2} & & & & & & \\
\vdots & & & \ddots & & & & & & \ddots & & & \vdots & & & & & & \\
\theta_{j(P-1)} & & & & \ddots & & & & & & 1 & & 1-a^*_{\Delta P-2} & & & & & & \\
\theta_{jP} & & & & & a^*_{\Delta_{P-1}} & & & & & & 1 & 1-a^*_{\Delta P-1} & & & & & & \\
\omega_{j1} & & & & & & & & & & & & & & & & & & \\
\omega_{j2} & & & & & & & & & & & & & & & & & & \\
\vdots & & & & & & & & & & & & & & & & & & \\
\omega_{j(P-2)} & & & & & & & & & & & & & & & & & & \\
\omega_{j(P-1)} & & & & & & & & & & & & & & & & & & \\
\mu^*_{j\infty} & & & & & & & & & & & & & & & & & & \\
y_{j1} & 1 & & & & & & & & & & & & & & & & & \\
y_{j2} & & 1 & & & & & & & & & & & & & & & & \\
y_{j3} & & & 1 & & & & & & & & & & & & & & & \\
\vdots & & & & \ddots & & & & & & & & & & & & & & \\
y_{j(P-1)} & & & & & 1 & & & & & & & & & & & & & \\
y_{jP} & & & & & & 1 & & & & & & & & & & & &
\end{array},$$

building upon Equations 2, 8, and 11, the **S** matrix is:

$$\underset{3P\times 3P}{\mathbf{S}} = \begin{array}{c|cccccccccccccccccc|}
 & \theta_{j1} & \theta_{j2} & \theta_{j3} & \cdots & \theta_{j(P-1)} & \theta_{jP} & \omega_{j1} & \omega_{j2} & \cdots & \omega_{j(P-2)} & \omega_{j(P-1)} & \mu^*_{j\infty} & y_{j1} & y_{j2} & y_{j3} & \cdots & y_{j(P-1)} & y_{jP} \\
\theta_{j1} & \sigma^2_{\text{fw}} & & & & & & & & & & & & & & & & & \\
\theta_{j2} & & & & & & & & & & & & & & & & & & \\
\theta_{j3} & & & & & & & & & & & & & & & & & & \\
\vdots & & & & & & & & & & & & & & & & & & \\
\theta_{j(P-1)} & & & & & & & & & & & & & & & & & & \\
\theta_{jP} & & & & & & & & & & & & & & & & & & \\
\omega_{j1} & & & & & & & q^*_{\Delta_1} & & & & & & & & & & & \\
\omega_{j2} & & & & & & & & q^*_{\Delta_2} & & & & & & & & & & \\
\vdots & & & & & & & & & \ddots & & & & & & & & & \\
\omega_{j(P-2)} & & & & & & & & & & q^*_{\Delta P-2} & & & & & & & & \\
\omega_{j(P-1)} & & & & & & & & & & & q^*_{\Delta P-1} & & & & & & & \\
\mu^*_{j\infty} & & & & & & & & & & & & \sigma^{2*}_{\infty} & & & & & & \\
y_{j1} & & & & & & & & & & & & & & & & & & \\
y_{j2} & & & & & & & & & & & & & & & & & & \\
y_{j3} & & & & & & & & & & & & & & & & & & \\
\vdots & & & & & & & & & & & & & & & & & & \\
y_{j(P-1)} & & & & & & & & & & & & & & & & & & \\
y_{jP} & & & & & & & & & & & & & & & & & &
\end{array},$$

the **F** matrix is:

$$\underset{P\times 3P}{\mathbf{F}} = \begin{array}{c|cccccccccccccccccc|}
 & \theta_{j1} & \theta_{j2} & \theta_{j3} & \cdots & \theta_{j(P-1)} & \theta_{jP} & \omega_{j1} & \omega_{j2} & \cdots & \omega_{j(P-2)} & \omega_{j(P-1)} & \mu^*_{j\infty} & y_{j1} & y_{j2} & y_{j3} & \cdots & y_{j(P-1)} & y_{jP} \\
y_{j1} & & & & & & & & & & & & & 1 & & & & & \\
y_{j2} & & & & & & & & & & & & & & 1 & & & & \\
y_{j3} & & & & & & & & & & & & & & & 1 & & & \\
\vdots & & & & & & & & & & & & & & & & \ddots & & \\
y_{j(P-1)} & & & & & & & & & & & & & & & & & 1 & \\
y_{jP} & & & & & & & & & & & & & & & & & & 1
\end{array},$$

and, building upon Equations 11 and 8 the **M** column vector is:

$$\underset{3P\times 1}{\mathbf{M}} = \begin{array}{cccccccccccccccccccc}
 & \theta_{j1} & \theta_{j2} & \theta_{j3} & \cdots & \theta_{j(P-1)} & \theta_{jP} & \omega_{j1} & \omega_{j2} & \cdots & \omega_{j(P-2)} & \omega_{j(P-1)} & \mu^*_{j\infty} & y_{j1} & y_{j2} & y_{j3} & \cdots & y_{j(P-1)} & y_{jP} & \\
(& \mu_{\text{dev}} & & & & & & & & & & & \mu^*_{\infty} & & & & & & &)'.
\end{array}$$

With these terms, the model implied covariance matrix $\boldsymbol{\Sigma}$ and the model implied mean vector $\boldsymbol{\mu}$ can then be calculated as

$$\underset{P\times P}{\boldsymbol{\Sigma}} = \mathbf{F}(\mathbf{I}_{3P} - \mathbf{A})^{-1}\mathbf{S}(\mathbf{I}_{3P} - \mathbf{A})^{-1'}\mathbf{F}',$$

$$\underset{P\times 1}{\boldsymbol{\mu}} = \mathbf{F}(\mathbf{I}_{3P} - \mathbf{A})^{-1}\mathbf{M},$$

where $\mathbf{I}_{3P}$ is an identity matrix of size $3P$. The solutions can be found in Table B1 in Appendix B.

Appendix B Model Implied Covariance Matrix and Mean Vector

Table B1. Model implied variances, covariances, and means for the univariate continuous-time model.

r	c	value
Variances		
1	1	$\sigma_{fw}^2 + \sigma_{\infty}^{2^*}$
2	2	$q_{\Delta_1}^* + \sigma_{fw}^2 \, a_{\Delta_1}^{*2} + \sigma_{\infty}^{2^*}$
3	3	$q_{\Delta_2}^* + q_{\Delta_1}^* a_{\Delta_2}^{*2} + \sigma_{fw}^2 a_{\Delta_2}^{*2} a_{\Delta_1}^{*2} + \sigma_{\infty}^{2^*}$
4	4	$q_{\Delta_3}^* + q_{\Delta_2}^* a_{\Delta_3}^{*2} + q_{\Delta_1}^* a_{\Delta_3}^{*2} a_{\Delta_2}^{*2} + \sigma_{fw}^2 a_{\Delta_3}^{*2} a_{\Delta_2}^{*2} a_{\Delta_1}^{*2} + \sigma_{\infty}^{2^*}$
5	5	$q_{\Delta_4}^* + q_{\Delta_3}^* a_{\Delta_4}^{*2} + q_{\Delta_2}^* a_{\Delta_4}^{*2} a_{\Delta_3}^{*2} + q_{\Delta_1}^* a_{\Delta_4}^{*2} a_{\Delta_3}^{*2} a_{\Delta_2}^{*2} + \sigma_{fw}^2 a_{\Delta_4}^{*2} a_{\Delta_3}^{*2} a_{\Delta_2}^{*2} a_{\Delta_1}^{*2} + \sigma_{\infty}^{2^*}$
⋮	⋮	
P	*P*	$q_{\Delta_{P-1}}^* + q_{\Delta_{P-2}}^* a_{\Delta_{P-1}}^{*2} + q_{\Delta_{P-3}}^* a_{\Delta_{P-1}}^{*2} a_{\Delta_{P-2}}^{*2} + \ldots + \sigma_{fw}^2 a_{\Delta_{P-1}}^{*2} a_{\Delta_{P-2}}^{*2} \cdots a_{\Delta_1}^{*2} + \sigma_{\infty}^{2^*}$
Covariances		
2	1	$a_{\Delta_1}^* \sigma_{fw}^2 + \sigma_{\infty}^{2^*}$
3	1	$a_{\Delta_2}^* a_{\Delta_1}^* \sigma_{fw}^2 + \sigma_{\infty}^{2^*}$
4	1	$a_{\Delta_3}^* a_{\Delta_2}^* a_{\Delta_1}^* \sigma_{fw}^2 + \sigma_{\infty}^{2^*}$
5	1	$a_{\Delta_4}^* a_{\Delta_3}^* a_{\Delta_2}^* a_{\Delta_1}^* \sigma_{fw}^2 + \sigma_{\infty}^{2^*}$
⋮	⋮	
P	1	$a_{\Delta_{P-1}}^* \ldots a_{\Delta_4}^* a_{\Delta_3}^* a_{\Delta_2}^* a_{\Delta_1}^* \sigma_{fw}^2 + \sigma_{\infty}^{2^*}$
3	2	$a_{\Delta_2}^* \left(q_{\Delta_1}^* + \sigma_{fw}^2 a_{\Delta_1}^{*2}\right) + \sigma_{\infty}^{2^*}$
4	2	$a_{\Delta_3}^* a_{\Delta_2}^* \left(q_{\Delta_1}^* + \sigma_{fw}^2 a_{\Delta_1}^{*2}\right) + \sigma_{\infty}^{2^*}$
5	2	$a_{\Delta_4}^* a_{\Delta_3}^* a_{\Delta_2}^* \left(q_{\Delta_1}^* + \sigma_{fw}^2 a_{\Delta_1}^{*2}\right) + \sigma_{\infty}^{2^*}$
⋮	⋮	
P	2	$a_{\Delta_{P-1}}^* \ldots a_{\Delta_4}^* a_{\Delta_3}^* a_{\Delta_2}^* \left(q_{\Delta_1}^* + \sigma_{fw}^2 a_{\Delta_1}^{*2}\right) + \sigma_{\infty}^{2^*}$
4	3	$a_{\Delta_3}^* \left(q_{\Delta_2}^* + q_{\Delta_1}^* a_{\Delta_2}^{*2} + \sigma_{fw}^2 a_{\Delta_2}^{*2} a_{\Delta_1}^{*2}\right) + \sigma_{\infty}^{2^*}$
5	3	$a_{\Delta_4}^* a_{\Delta_3}^* \left(q_{\Delta_2}^* + q_{\Delta_1}^* a_{\Delta_2}^{*2} + \sigma_{fw}^2 a_{\Delta_2}^{*2} a_{\Delta_1}^{*2}\right) + \sigma_{\infty}^{2^*}$
⋮	⋮	
P	3	$a_{\Delta_{P-1}}^* \ldots a_{\Delta_4}^* a_{\Delta_3}^* \left(q_{\Delta_2}^* + q_{\Delta_1}^* a_{\Delta_2}^{*2} + \sigma_{fw}^2 a_{\Delta_2}^{*2} a_{\Delta_1}^{*2}\right) + \sigma_{\infty}^{2^*}$
5	4	$a_{\Delta_4}^* \left(q_{\Delta_3}^* + q_{\Delta_2}^* a_{\Delta_3}^{*2} + q_{\Delta_1}^* a_{\Delta_3}^{*2} a_{\Delta_2}^{*2} + \sigma_{fw}^2 a_{\Delta_3}^{*2} a_{\Delta_2}^{*2} a_{\Delta_1}^{*2}\right) + \sigma_{\infty}^{2^*}$
⋮	⋮	
P	4	$a_{\Delta_{P-1}}^* \ldots a_{\Delta_4}^* \left(q_{\Delta_3}^* + q_{\Delta_2}^* a_{\Delta_3}^{*2} + q_{\Delta_1}^* a_{\Delta_3}^{*2} a_{\Delta_2}^{*2} + \sigma_{fw}^2 a_{\Delta_3}^{*2} a_{\Delta_2}^{*2} a_{\Delta_1}^{*2}\right) + \sigma_{\infty}^{2^*}$
P	*P*-1	$a_{\Delta_{P-1}}^* \left(q_{\Delta_{P-2}}^* + q_{\Delta_{P-3}}^* a_{\Delta_{P-2}}^{*2} + q_{\Delta_{P-4}}^* a_{\Delta_{P-2}}^{*2} a_{\Delta_{P-3}}^{*2} + \ldots + \sigma_{fw}^2 a_{\Delta_{P-2}}^{*2} a_{\Delta_{P-3}}^{*2} \ldots a_{\Delta_1}^{*2}\right) + \sigma_{\infty}^{2^*}$
Means		
1		$\mu_{dev} + \mu_{\infty}^*$
2		$a_{\Delta_1}^* \mu_{dev} + \mu_{\infty}^*$
3		$a_{\Delta_2}^* a_{\Delta_1}^* \mu_{dev} + \mu_{\infty}^*$
4		$a_{\Delta_3}^* a_{\Delta_2}^* a_{\Delta_1}^* \mu_{dev} + \mu_{\infty}^*$
5		$a_{\Delta_4}^* a_{\Delta_3}^* a_{\Delta_2}^* a_{\Delta_1}^* \mu_{dev} + \mu_{\infty}^*$
⋮		
P		$a_{\Delta_{P-1}}^* \ldots a_{\Delta_4}^* a_{\Delta_3}^* a_{\Delta_2}^* a_{\Delta_1}^* \mu_{dev} + \mu_{\infty}^*$

r = row, c = column of covariance matrix and mean vector; *P* = number of measurement occasions.

Appendix C Results from simulations with reduced number of persons

Table C1. Parameter quality and run time for three Bayesian implementations of the univariate continuous-time model (simulated data with reduced number of persons, J = 200).

		Classic	Cov.-based	Cov./mean-based
Statistic	Parameter	Value	Value	Value
Bias	a	−0.001126	−0.001394	0.000881
	q	0.000429	0.000414	0.000379
	$\sigma^{2^*}_{\infty}$	−0.007554	−0.007579	−0.007829
	σ^2_{fw}	−0.003437	−0.003176	−0.003690
	μ^*_{∞}	0.001115	–	0.000905
	μ_{dev}	−0.000494	–	−0.000361
RMSE	a	0.0235	0.0233	0.0235
	q	0.0117	0.0118	0.0118
	$\sigma^{2^*}_{\infty}$	0.0618	0.0616	0.0615
	σ^2_{fw}	0.0601	0.0610	0.0610
	μ^*_{∞}	0.0578	–	0.0579
	μ_{dev}	0.0554	–	0.0557
Coverage rate 95%	a	0.930	0.936	0.932
	q	0.951	0.951	0.950
	$\sigma^{2^*}_{\infty}$	0.954	0.956	0.951
	σ^2_{fw}	0.950	0.947	0.947
	μ^*_{∞}	0.936	–	0.941
	μ_{dev}	0.953	–	0.949
SE accuracy	a	0.967	0.978	0.971
	q	0.986	0.978	0.978
	$\sigma^{2^*}_{\infty}$	1.002	1.006	1.010
	σ^2_{fw}	1.027	1.010	1.012
	μ^*_{∞}	0.967	–	0.966
	μ_{dev}	1.014	–	1.006
Run time (minutes)	M	10.1	3.3	5.6
	min	3.2	1.2	2.0
	$Q_{0.95}$	19.5	5.0	8.3
	max	73.4	192.5	589.3
PSR	M	1.0000	0.9997	0.9997
	min	0.9991	0.9984	0.9983
	max	1.0010	1.0010	1.0010
ESS	M	1052	753	959
	min	400	400	400
	max	7879	6610	38393
AC	M	0.329	0.202	0.149
	min	0.086	−0.097	−0.090
	max	0.685	0.580	0.552
Number of iterations	M	2406	1335	1493
	min	1250	700	650
	max	14250	50750	45550

Cov. = covariance, $Q_{0.95}$ = 0.95 quantile, N_{repl} = 1,000, PSR $\leq$ 1.001 and ESS $\geq$ 400 for all parameters, J = 200 persons, P = 20 measurement occasions, one chain ran on one Intel Xeon Gold 5120 (2.20 GHz) CPU of a 64-bit Linux Debian 9 “Stretch” computer. RMSE = root mean squared error; SE = standard error; PSR = potential scale reduction factor; ESS = effective sample size; AC = autocorrelation.

Appendix D Results from simulations with skewed normal distributions

Table D1. Parameter quality and run time for three Bayesian implementations of the univariate continuous-time model (simulated data with skewed normal distributions).

		Classic	Cov.-based	Cov./mean-based
Statistic	Parameter	Value	Value	Value
Bias	a	−0.000243	−0.000228	−0.000175
	q	−0.000039	−0.000044	−0.000051
	σ^{2*}_{∞}	0.000363	0.000455	0.000427
	σ^{2}_{fw}	−0.000999	−0.000974	−0.000737
	μ^{*}_{∞}	−0.000130	–	−0.000237
	μ_{dev}	−0.001254	–	−0.001159
RMSE	a	0.0072	0.0072	0.0072
	q	0.0039	0.0040	0.0040
	σ^{2*}_{∞}	0.0213	0.0211	0.0214
	σ^{2}_{fw}	0.0199	0.0199	0.0200
	μ^{*}_{∞}	0.0182	–	0.0183
	μ_{dev}	0.0184	–	0.0181
Coverage rate 95%	a	0.949	0.946	0.952
	q	0.931	0.928	0.932
	σ^{2*}_{∞}	0.921	0.928	0.922
	σ^{2}_{fw}	0.939	0.937	0.938
	μ^{*}_{∞}	0.949	–	0.945
	μ_{dev}	0.939	–	0.940
SE accuracy	a	0.994	0.988	0.985
	q	0.930	0.915	0.913
	σ^{2*}_{∞}	0.899	0.905	0.896
	σ^{2}_{fw}	0.954	0.951	0.951
	μ^{*}_{∞}	0.968	–	0.967
	μ_{dev}	0.963	–	0.973
Run time (minutes)	M	94.0	5.7	8.4
	min	29.1	1.4	2.0
	$Q_{0.95}$	185.9	10.6	16.0
	max	584.1	362.6	916.5
PSR	M	1.0000	0.9996	0.9997
	min	0.9992	0.9984	0.9984
	max	1.0010	1.0010	1.0010
ESS	M	1053	767	950
	min	400	400	400
	max	6570	5322	42467
AC	M	0.325	0.181	0.133
	min	0.086	−0.128	−0.094
	max	0.669	0.744	0.685
Number of iterations	M	2380	1265	1406
	min	1250	700	700
	max	11700	42450	51250

Cov. = covariance, $Q_{0.95}$ = 0.95 quantile, N_{repl} = 1,000, PSR $\leq$ 1.001 and ESS $\geq$ 400 for all parameters, J = 2,000 persons, P = 20 measurement occasions, one chain ran on one Intel Xeon Gold 5120 (2.20 GHz) CPU of a 64-bit Linux Debian 9 "Stretch" computer. RMSE = root mean squared error; SE = standard error; PSR = potential scale reduction factor; ESS = effective sample size; AC = autocorrelation. In the data generation, the skewness of the normal distributions was set to 2.