Supplementary of "Multivariate Reparameterized Inverse Gaussian Processes with Common Effects for Degradation-based Reliability Prediction"

Liangliang Zhuang[a], Ancha Xu[a,b,*], Guanqi Fang[a,b], Yincai Tang[c]

[a] *School of Statistics and Mathematics, Zhejiang Gongshang University, Hangzhou 310018, China*
[b] *Collaborative Innovation Center of Statistical Data Engineering, Technology & Application Zhejiang Gongshang University, Hangzhou, China*
[c] *The KLATASDS-MOE, School of Statistics, East China Normal University, Shanghai 200241, China*

The organization of the supplementary document is as follows: Section S1 provides proofs of propositions; Section S2 details the technical aspects of the integral approximation; Section S3 elaborates on the technicalities of the EM algorithm, including the computation of conditional expectations during the E-step and the derivation of first-order partial derivatives of the Q-function. Sections S4 and S5 present additional results from simulation and case studies, respectively. Additionally, the R code for implementing the GL method and EM algorithm is available online. For detailed descriptions, see Section S6.

S1 Proof of propositions

S1.1 Proof of Proposition 1

For the degradation of the k-th PC, we have

$$\mathrm{E}\left[Y_k(t)\right] = \frac{\Lambda_0(t)+\Lambda_k(t)}{\gamma}, \quad \text{and} \quad \mathrm{Var}\left[Y_k(t)\right] = \frac{\Lambda_0(t)+\Lambda_k(t)}{\gamma^3}. \tag{S1}$$

Before proceeding to derivation $\mathrm{Cov}\left[Y_{k_1}\left(t_1\right), Y_{k_2}\left(t_2\right)\right]$, we first calculate the following results. Let $Z(t)$ be a $r\mathcal{IG}$ process, with parameters $\Lambda_0(t)$, and γ. When $t_1 \leq t_2$,

$$\mathrm{E}\left[Z\left(t_1\right) Z\left(t_2\right)\right] = \mathrm{E}\left\{Z\left(t_1\right)\left[Z\left(t_2\right) - Z\left(t_1\right) + Z\left(t_1\right)\right]\right\}$$

*Corresponding author: xuancha@mail.zjgsu.edu.cn

$$
\begin{aligned}
&= \mathrm{E}\left[Z^2\left(t_1\right)\right] + \mathrm{E}\left[Z\left(t_2\right) - Z\left(t_1\right)\right] \times \mathrm{E}\left[Z\left(t_1\right)\right] \\
&= \frac{\Lambda_0^2(t_1)}{\gamma^2} + \frac{\Lambda_0(t_1)}{\gamma^3} + \frac{\Lambda_0(t_2) - \Lambda_0(t_1)}{\gamma} \times \frac{\Lambda_0(t_1)}{\gamma} \\
&= \frac{\Lambda_0(t_1)\left[1 + \gamma\Lambda_0(t_2)\right]}{\gamma^3},
\end{aligned} \tag{S2}
$$

when $t_1 > t_2$, $\mathrm{E}\left[Z\left(t_1\right) Z\left(t_2\right)\right] = \left\{\Lambda_0(t_2)\left[1 + \gamma\Lambda_0(t_1)\right]\right\}/\gamma^3$. Thus,

$$
\mathrm{E}\left[Z\left(t_1\right) Z\left(t_2\right)\right] = \frac{\min(\Lambda_0(t_1), \Lambda_0(t_2))\left[1 + \gamma \max(\Lambda_0(t_1), \Lambda_0(t_2))\right]}{\gamma^3}.
$$

For the degradation of two different PCs $Y_{k_1}(t_1)$ and $Y_{k_2}(t_2)$ with $k_1 \neq k_2$, and $t_1 \neq t_2$ we have

$$
\begin{aligned}
&\mathrm{Cov}\left[Y_{k_1}\left(t_1\right), Y_{k_2}\left(t_2\right)\right] \\
&= \mathrm{E}\left[Y_{k_1}\left(t_1\right) Y_{k_2}\left(t_2\right)\right] - \mathrm{E}\left[Y_{k_1}\left(t_1\right)\right] \mathrm{E}\left[Y_{k_2}\left(t_2\right)\right] \\
&= \mathrm{E}\left[\left(Z\left(t_1\right) + X_{k_1}\left(t_1\right)\right)\left(Z\left(t_2\right) + X_{k_2}\left(t_2\right)\right)\right] - \frac{\left[\Lambda_0\left(t_1\right) + \Lambda_{k_1}\left(t_1\right)\right]\left[\Lambda_0\left(t_2\right) + \Lambda_{k_2}\left(t_2\right)\right]}{\gamma^2} \\
&= \mathrm{E}\left[Z\left(t_1\right) Z\left(t_2\right)\right] + \mathrm{E}\left[Z\left(t_1\right) X_{k_2}\left(t_2\right)\right] + \mathrm{E}\left[X_{k_1}\left(t_1\right) Z\left(t_2\right)\right] + \mathrm{E}\left[X_{k_1}\left(t_1\right) X_{k_2}\left(t_2\right)\right] \\
&\quad - \frac{\left[\Lambda_0\left(t_1\right) + \Lambda_{k_1}\left(t_1\right)\right]\left[\Lambda_0\left(t_2\right) + \Lambda_{k_2}\left(t_2\right)\right]}{\gamma^2} \\
&= \frac{\min(\Lambda_0(t_1), \Lambda_0(t_2))\left[1 + \gamma \max(\Lambda_0(t_1), \Lambda_0(t_2))\right]}{\gamma^3} + \frac{\Lambda_{k_2}(t_2)\Lambda_0(t_1)}{\gamma^2} + \frac{\Lambda_{k_1}(t_1)\Lambda_0(t_2)}{\gamma^2} \\
&\quad + \frac{\Lambda_{k_2}(t_2)\Lambda_{k_1}(t_1)}{\gamma^2} - \frac{\left[\Lambda_0\left(t_1\right) + \Lambda_{k_1}\left(t_1\right)\right]\left[\Lambda_0\left(t_2\right) + \Lambda_{k_2}\left(t_2\right)\right]}{\gamma^2} \\
&= \frac{\min(\Lambda_0(t_1), \Lambda_0(t_2))}{\gamma^3} + \frac{\min(\Lambda_0(t_1), \Lambda_0(t_2)) \times \max(\Lambda_0(t_1), \Lambda_0(t_2))}{\gamma^2} - \frac{\Lambda_0(t_1)\Lambda_0(t_2)}{\gamma^2} \\
&= \frac{\min(\Lambda_0(t_1), \Lambda_0(t_2))}{\gamma^3}
\end{aligned} \tag{S3}
$$

Subsequently, the Pearson's correlation coefficient between $Y_{k_1}(t)$ and $Y_{k_2}(t)$ is

$$
\rho\left[Y_{k_1}(t), Y_{k_2}(t)\right] = \frac{\mathrm{cov}\left[Y_{k_1}(t), Y_{k_2}(t)\right]}{\sqrt{\mathrm{var}\left[Y_{k_1}(t)\right] \mathrm{var}\left[Y_{k_2}(t)\right]}} = \frac{\Lambda_0(t)}{\sqrt{\left(\Lambda_0(t) + \Lambda_{k_1}(t)\right)\left(\Lambda_0(t) + \Lambda_{k_2}(t)\right)}}. \tag{S4}
$$

S1.2 Proof of Proposition 3

Let the failure time of the k-th PC be represented as follows $T_{\mathcal{D}_k} = \inf\left\{t : Y_k(t) \geq \mathcal{D}_k\right\}$. Then, the system failure time can be defined by $T_{\boldsymbol{\mathcal{D}}} = \inf\left\{t : Y_1(t) \geq \mathcal{D}_1 \text{ or } \cdots \text{ or } Y_K(t) \geq \mathcal{D}_K\right\}$,

where $\boldsymbol{\mathcal{D}} = (\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_K)'$ is a vector storing all PC failure thresholds. The CDF of system failure time $T_{\boldsymbol{\mathcal{D}}}$ can be derived as:

$$\begin{aligned} F_{T_{\boldsymbol{\mathcal{D}}}}(t \mid \boldsymbol{\Lambda}(t), \gamma, \boldsymbol{\mathcal{D}}) &= \int_0^{\tilde{y}} F_{T_{\boldsymbol{\mathcal{D}}}}(t \mid Z(t)) f(z) \mathrm{d}z, \\ &= \int_0^{\tilde{y}} \left[1 - P\left(Y_1(t) < \mathcal{D}_1, \ldots, Y_K(t) < \mathcal{D}_K \mid Z(t)\right)\right] f(z) \mathrm{d}z \\ &= \int_0^{\tilde{y}} \left[1 - P\left(X_1(t) < \mathcal{D}_1 - z, \ldots, X_K(t) < \mathcal{D}_K - z \mid Z(t)\right)\right] f(z) \mathrm{d}z \\ &= \int_0^{\tilde{y}} \left[1 - \prod_{k=1}^{K} \left(F_{rIG}(\mathcal{D}_k - z; \Lambda_k(t), \gamma)\right)\right] f(z) \mathrm{d}z. \end{aligned} \tag{S5}$$

where $\boldsymbol{\Lambda}(t) = (\Lambda_0(t), \ldots, \Lambda_K(t))'$. $\tilde{y}$ is defined as $\tilde{y} = \min\{y_1, \ldots, y_K\}$, where $y_1, \ldots, y_K$ are the observed dagradation values. Given the complexity of directly integrating the CDF of system failure time, approximation methods can be employed to simplify the integral. Next, we introduce the technical details of the integral approximation method.

S2 Technical details of the integral approximation method

The GL quadrature is a numerical integration method particularly suitable for integrating functions over the interval $[-1, 1]$. Compared with Monte Carlo integration, GL quadrature could provide an accurate approximation for the integral with a much lower computational budget (Swarztrauber, 2003). It relies on the selection of roots and weights based on Legendre polynomials to provide high accuracy in approximating integrals (Babolian et al., 2005). For the integral $\int_1^{-1} h(x)dx$, the GL quadrature approximation is given by

$$\int_{-1}^{1} h(x)dx \approx \sum_{q=1}^{l} w_q h\left(x_q\right). \tag{S6}$$

Here, x_q are the roots of the Legendre polynomial, and w_q are the corresponding weights. For a given order l, there are l roots and weights, which can be obtained through numerical methods or by consulting precomputed tables. Generally, the selection of these nodes and weights aims to achieve full accuracy for polynomials of degree $2l - 1$ (Golub and Welsch, 1969).

Next, we approximate (S5) using the GL integration method. To ensure that the integral is over the interval $[-1, 1]$, we introduce a transformed variable $u = 2z/\tilde{y} - 1$. Subsequently, we perform the GL quadrature approximation on the transformed integral, yielding the following result:

$$\begin{aligned} & F_{T_{\boldsymbol{\mathcal{D}}}}(t \mid \boldsymbol{\Lambda}(t), \gamma, \boldsymbol{\mathcal{D}}) \\ & = \frac{\tilde{y}}{2} \int_{-1}^{1} \left[1 - \prod_{k=1}^{K} \left(F_{rIG}(\mathcal{D}_k - \frac{\tilde{y}(u+1)}{2}; \Lambda_k(t), \gamma) \right) \right] f(\frac{\tilde{y}(u+1)}{2}; \Lambda_0(t), \gamma) \mathrm{d}u, \\ & \approx \frac{\tilde{y}}{2} \sum_{q=1}^{l} w_q \left[1 - \prod_{k=1}^{K} \left(F_{rIG}(\mathcal{D}_k - \frac{\tilde{y}(u_q+1)}{2}; \Lambda_k(t), \gamma) \right) \right] f(\frac{\tilde{y}(u_q+1)}{2}; \Lambda_0(t), \gamma), \end{aligned} \tag{S7}$$

where l is a given order, u_q is the root of the Legendre polynomial $P_n(u_q)$, which is formulated as

$$P_l(u_q) = \frac{1}{2^l l!} \frac{d^l}{du_q^l} (u_q^2 - 1)^l,$$

and $w_q = 2/\left\{ \left(1 - u_q^2\right) \left[P_l'(u_q)\right]^2 \right\}$ is the corresponding weight. We can easily obtain the weights and nodes for the GL method using the function *legendre.quadrature.rules()* provided in the R package *gaussquad* (Novomestky, 2022). The choice of l significantly impacts the approximation's accuracy and efficiency. In our paper, we choose $l = 10$. The reasons for this selection, along with comparisons to other approximation methods, are thoroughly detailed in Section 3 of the main text.

S3 Technical details of the EM algorithm

S3.1 Derivation of the conditional expectations in the E-step

We need to calculate the expectations required in the EM algorithm with respect to $p(\mathbb{Z}|\mathbb{Y}, \boldsymbol{\theta})$ in the E-step. In the following, we will suppress the dependence on $\boldsymbol{\theta}$ for simplicity. Based on Eq.(15), we have $p(\Delta \boldsymbol{Y_{i,:,j}})$. Then, it is ready to see that

$$p(\mathbb{Z}|\mathbb{Y}) = \prod_{i=1}^{n} \prod_{j=1}^{m_i} p(\Delta Z_{i,j} | \Delta \boldsymbol{Y_{i,:,j}}), \tag{S8}$$

and

$$p(\Delta Z_{i,j}|\Delta \boldsymbol{Y}_{\boldsymbol{i,:,j}}) = \frac{f_{rIG}\left(\Delta z_{i,j};\Delta\Lambda_{i,0,j},\gamma\right)\prod_{k=1}^{K} f_{rIG}\left(\Delta y_{i,k,j}-\Delta z_{i,j};\Delta\Lambda_{i,k,j},\gamma\right)}{\int_0^{\Delta\tilde{y}_{i,j}} f_{rIG}\left(\Delta z_{i,j};\Delta\Lambda_{i,0,j},\gamma\right)\prod_{k=1}^{K} f_{rIG}\left(\Delta y_{i,k,j}-\Delta z_{i,j};\Delta\Lambda_{i,k,j},\gamma\right)\mathrm{d}\Delta z_{i,j}} \tag{S9}$$

Then, we can easily obtain three conditional expectations: $\mathrm{E}\left[(\Delta Y_{i,k,j}-\Delta Z_{i,j})^{-1} \mid \Delta \boldsymbol{Y}_{i,:,j}\right]$, $\mathrm{E}\left[\ln \Delta Z_{i,j} \mid \Delta \boldsymbol{Y}_{i,:,j}\right]$, and $\mathrm{E}\left[\Delta Z_{i,j}^{-1} \mid \Delta \boldsymbol{Y}_{i,:,j}\right]$ as:

$$\begin{aligned}
\mathrm{E}\left[(\Delta Y_{i,k,j}-\Delta Z_{i,j})^{-1} \mid \Delta \boldsymbol{Y}_{i,:,j}\right] &= \int_0^{\Delta\tilde{\boldsymbol{Y}}_{i,j}} (\Delta Y_{i,k,j}-\Delta Z_{i,j})^{-1}\, p(\Delta Z_{i,j}|\Delta \boldsymbol{Y}_{\boldsymbol{i,:,j}})\mathrm{d}\Delta Z_{i,j}, \\
\mathrm{E}\left[\Delta Z_{i,j} \mid \Delta \boldsymbol{Y}_{i,:,j}\right] &= \int_0^{\Delta\tilde{\boldsymbol{Y}}_{i,j}} \Delta Z_{i,j} p(\Delta Z_{i,j}|\Delta \boldsymbol{Y}_{\boldsymbol{i,:,j}})\mathrm{d}\Delta Z_{i,j}, \\
\mathrm{E}\left[\Delta Z_{i,j}^{-1} \mid \Delta \boldsymbol{Y}_{i,:,j}\right] &= \int_0^{\Delta\tilde{\boldsymbol{Y}}_{i,j}} \Delta Z_{i,j}^{-1} p(\Delta Z_{i,j}|\Delta \boldsymbol{Y}_{\boldsymbol{i,:,j}})d\Delta Z_{i,j},
\end{aligned} \tag{S10}$$

where $\Delta\tilde{\boldsymbol{Y}}_{i,j} = \min\{\Delta Y_{i,1,j},\ldots,\Delta Y_{i,K,j}\}$. The integral involved in (S10) can also be evaluated by GL integral approximation, see Section S2.

S3.2 First order partial derivatives

Taking the first partial derivatives of Q-function in Eq.(18) with respect to $\boldsymbol{\theta}$ and equating each to zero, we obtain the following equations:

$$\frac{\partial Q(\boldsymbol{\theta})}{\partial \alpha_k} = \sum_{i=1}^{n}\sum_{j=1}^{m_i} \frac{\partial \Delta\Lambda_{i,k,j}}{\partial \alpha_k}\left\{\frac{1}{\Delta\Lambda_{i,k,j}} + \gamma - \Delta\Lambda_{i,k,j}\mathrm{E}\left[(\Delta Y_{i,k,j}-\Delta Z_{i,j})^{-1} \mid \Delta \boldsymbol{Y}_{i,:,j}\right]\right\} = 0, \tag{S11}$$

$$\frac{\partial Q(\boldsymbol{\theta})}{\partial \beta_k} = \sum_{i=1}^{n}\sum_{j=1}^{m_i} \frac{\partial \Delta\Lambda_{i,k,j}}{\partial \beta_k}\left\{\frac{1}{\Delta\Lambda_{i,k,j}} + \gamma - \Delta\Lambda_{i,k,j}\mathrm{E}\left[(\Delta Y_{i,k,j}-\Delta Z_{i,j})^{-1} \mid \Delta \boldsymbol{Y}_{i,:,j}\right]\right\} = 0, \tag{S12}$$

$$\frac{\partial Q(\boldsymbol{\theta})}{\partial \alpha_0} = \sum_{i=1}^{n}\sum_{j=1}^{m_i} \frac{\partial \Delta\Lambda_{i,0,j}}{\partial \alpha_0}\left[\frac{1}{\Delta\Lambda_{i,0,j}} + \gamma - \Delta\Lambda_{i,0,j}\mathrm{E}\left(\Delta Z_{i,j}^{-1} \mid \Delta \boldsymbol{Y}_{i,:,j}\right)\right] = 0, \tag{S13}$$

$$\frac{\partial Q(\boldsymbol{\theta})}{\partial \gamma} = \sum_{i=1}^{n}\sum_{j=1}^{m_i}\left\{\sum_{k=0}^{K}\Delta\Lambda_{i,k,j} + \gamma\left[-\sum_{k=1}^{K} Y_{i,k,j} + (K-1)E\left(\Delta Z_{i,j} \mid \Delta \boldsymbol{Y}_{i,:,j}\right)\right]\right\} = 0, \tag{S14}$$

for $k = 1,\ldots,K$. Given $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, the solution of (S14) is

$$\gamma = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m_i}\sum_{k=0}^{K}\Delta\Lambda_{i,k,j}}{\sum_{i=1}^{n}\sum_{j=1}^{m_i}\left[\sum_{k=1}^{K}\Delta Y_{i,k,j} - (K-1)\,E\left(\Delta Z_{i,j} \mid \Delta \boldsymbol{Y}_{i,:,j}\right)\right]}. \tag{S15}$$

S4 Additional simulation experiments

S4.1 Additional simulation results for $K = 3$

Figure S1 displays simulated degradation paths for three PCs with $n = 5$ across four scenarios, as specified in Table 2. Scenarios I and IV feature linear degradation trends, scenario II presents convex paths, and scenario III demonstrates concave paths for each PC. Table S1 presents the MTTF estimators across various scenarios and unit sizes. We observe that the MTTF estimates demonstrate low values of RRMSE, and the CPs consistently approximate 95%. Figure S2 shows the RRMSE ($\times 10^{-2}$) of the correlation estimates for various unit sizes and scenarios over time.

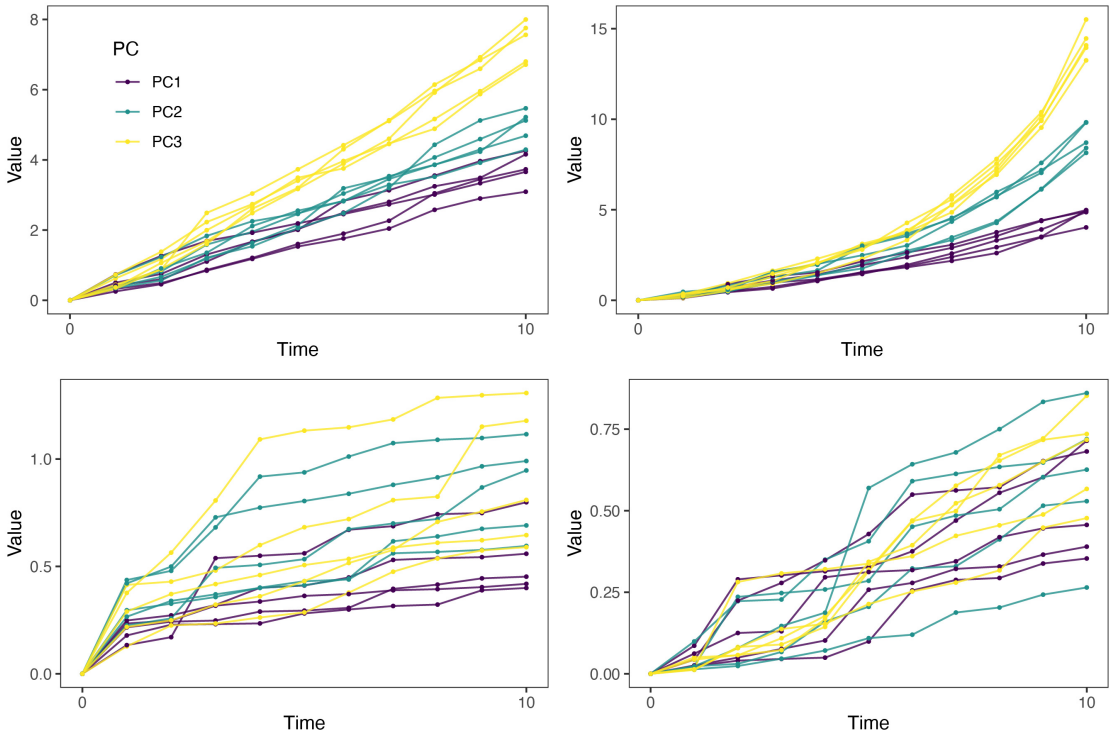

Figure S1: Simulated degradation paths for four scenarios with $n = 5$. A linear path is shown in scenarios I and IV, a convex path is shown in scenario II, and a concave path is shown in scenario III.

S4.2 Simulation results for higher dimensions

To demonstrate the effectiveness of the proposed model, we consider a higher-dimensional scenario with $K = 5$. Without loss of generality, we use Scenario I as an example and set

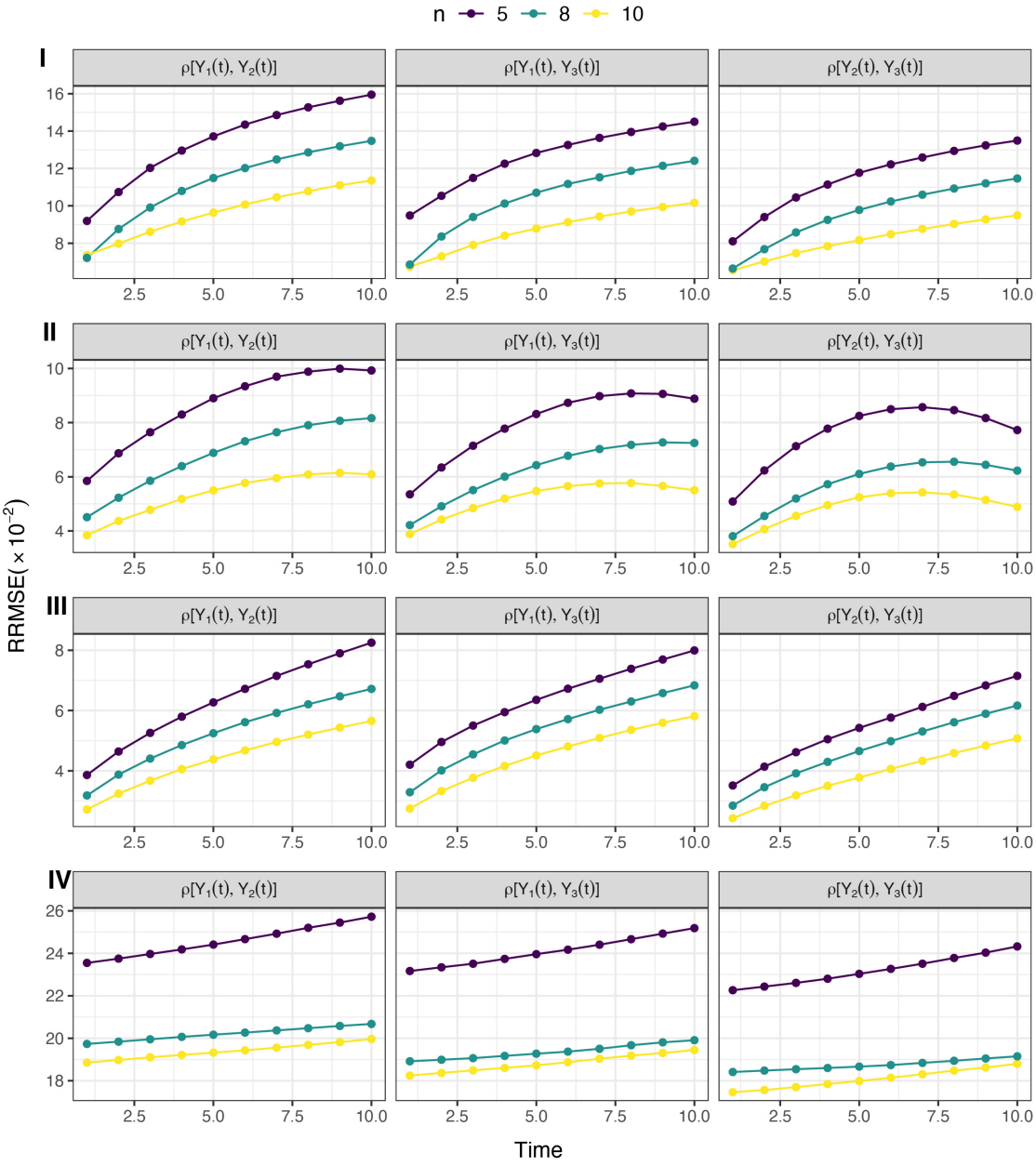

Figure S2: RRMSE ($\times 10^{-2}$) of correlation estimators across different unit sizes and scenarios.

Table S1: Results ($\times 10^{-2}$) of MTTF estimators across different scenarios and unit sizes.

	$n=5$				$n=8$				$n=10$			
	I	II	III	IV	I	II	III	IV	I	II	III	IV
RRMSE	0.32	0.14	6.47	1.43	0.23	0.12	3.60	0.97	0.20	0.07	3.39	0.82
CP	92.50	93.10	95.08	94.75	95.25	92.08	95.85	93.12	95.20	96.56	94.58	95.44

the true model parameters as $\boldsymbol{\alpha}' = (1, 0.8, 0.9, 1, 1.1, 1.2)$, $\boldsymbol{\beta}' = (1, 0.8, 1, 1.2, 1.4, 1.6)$, and $\boldsymbol{\mathcal{D}} = (3.6, 4.2, 4.8, 6, 7.2)$. The other model parameters and simulation settings remain the same as those in the case of $K = 3$.

Tables S2-S4 respectively present the model parameters, correlation coefficients, and MTTF estimates obtained through the EM algorithm, based on 500 repetitions under different n. As can be seen, the biases are small for all the parameters, and the RRMSEs decrease as the sample size n. This indicates that the model can provide a satisfactory fit with moderate n for $K = 5$. Additionally, in Table S4, we also provide MTTF estimation results for a model that assumes the PCs are independent. We utilize ML method to infer parameter estimates. It is evident from the results that under the assumption of model misidentification, the accuracy of MTTF estimates is significantly lower compared to that obtained from the proposed model.

Table S2: RRMSE ($\times 10^{-2}$) of EM estimators under Scenario I for $K = 5$ across different unit sizes.

n	α_0	α_1	α_2	α_3	α_4	α_5	β_1	β_2	β_3	β_4	β_5	γ
5	12.36	17.56	11.23	8.26	5.81	4.87	34.49	30.23	24.85	21.46	19.19	7.57
8	9.23	14.31	8.96	6.66	4.95	4.09	25.60	20.78	18.39	16.52	14.29	5.66
10	2.65	11.47	6.95	5.75	4.15	3.62	21.98	16.70	14.69	13.20	11.92	4.42

We conduct additional simulations to assess the computational efficiency and parameter

Table S3: RRMSE ($\times 10^{-2}$) of correlation estimators under Scenario I for $K = 5$ across different unit sizes.

n	$\hat{\rho}_{1,2}$	$\hat{\rho}_{1,3}$	$\hat{\rho}_{1,4}$	$\hat{\rho}_{1,5}$	$\hat{\rho}_{2,3}$	$\hat{\rho}_{2,4}$	$\hat{\rho}_{2,5}$	$\hat{\rho}_{3,4}$	$\hat{\rho}_{3,5}$	$\hat{\rho}_{4,5}$
5	11.77	11.20	10.57	10.09	10.68	10.07	9.63	9.59	9.15	8.65
8	8.66	8.30	7.91	7.45	7.89	7.53	7.10	7.24	6.81	6.49
10	4.92	4.71	4.43	4.30	4.40	4.19	4.03	4.04	3.84	3.64

Table S4: RRMSE ($\times 10^{-2}$) of MTTF estimators under Scenario I for $K = 5$ across different unit sizes.

Model	$n = 5$	$n = 8$	$n = 10$
Dep.	3.43	2.63	2.29
Indep.	6.58	6.45	6.29

estimation performance across varying numbers of PCs. The parameters β_i and α_i are defined as $\beta_i = 0.8+0.8(i-1)/(K-1)$ and $\alpha_i = 0.8+0.4(i-1)/(K-1)$, respectively, allowing β_i to linearly increase from 0.8 to 1.6 and α_i from 0.8 to 1.2. We set $n = 5$ and $m = 10$, with other settings consistent with Table 2 in the main text. Figure S3 illustrates the impact of different K values on the average RRMSE of all estimated parameters and the average runtime per iteration of the EM algorithm under Scenario I. As the dimensionality increases, there is a continuous improvement in the estimation accuracy, attributed to increased data volume. Moreover, the computational time for the EM algorithm increases linearly with K, demonstrating scalability without the typical exponential growth associated with higher dimensions. This linear parameter growth with K prevents the "curse of dimensionality."

S5 Additional results of case studies

Figure S4 and S5 give the iteration diagram of EM algorithm parameter estimation under different scenarios for PMB and fatigue crack-size data, respectively. It is evident that, after

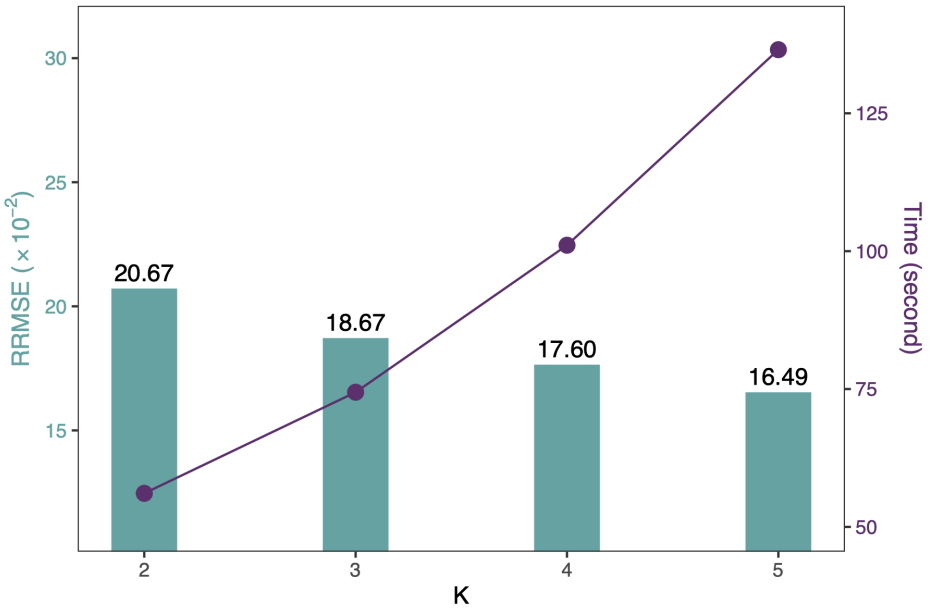

Figure S3: Average RRMSE and EM computation time for different K in Scenario I.

a substantial number of iterations, the EM algorithm provides parameter estimates for the two datasets, with each parameter achieving convergence.

S6 Online code description

The online code repository related to this paper can be found on GitHub[1]. The repository is structured into three primary directories, each designed for specific functions as outlined below:

- **case**: This directory includes:
  - "Fatigue-crack-size.xlsx" dataset, originally introduced in Meeker et al. (2022) and further processed as described in Appendix H of Fang et al. (2022).
  - "crack.R" script, which performs parameter estimation across various models and calculates their respective AIC.
  - "results" folder, where the final analytical outputs are stored.

[1]https://github.com/liangliangzhuang/multi-rIG

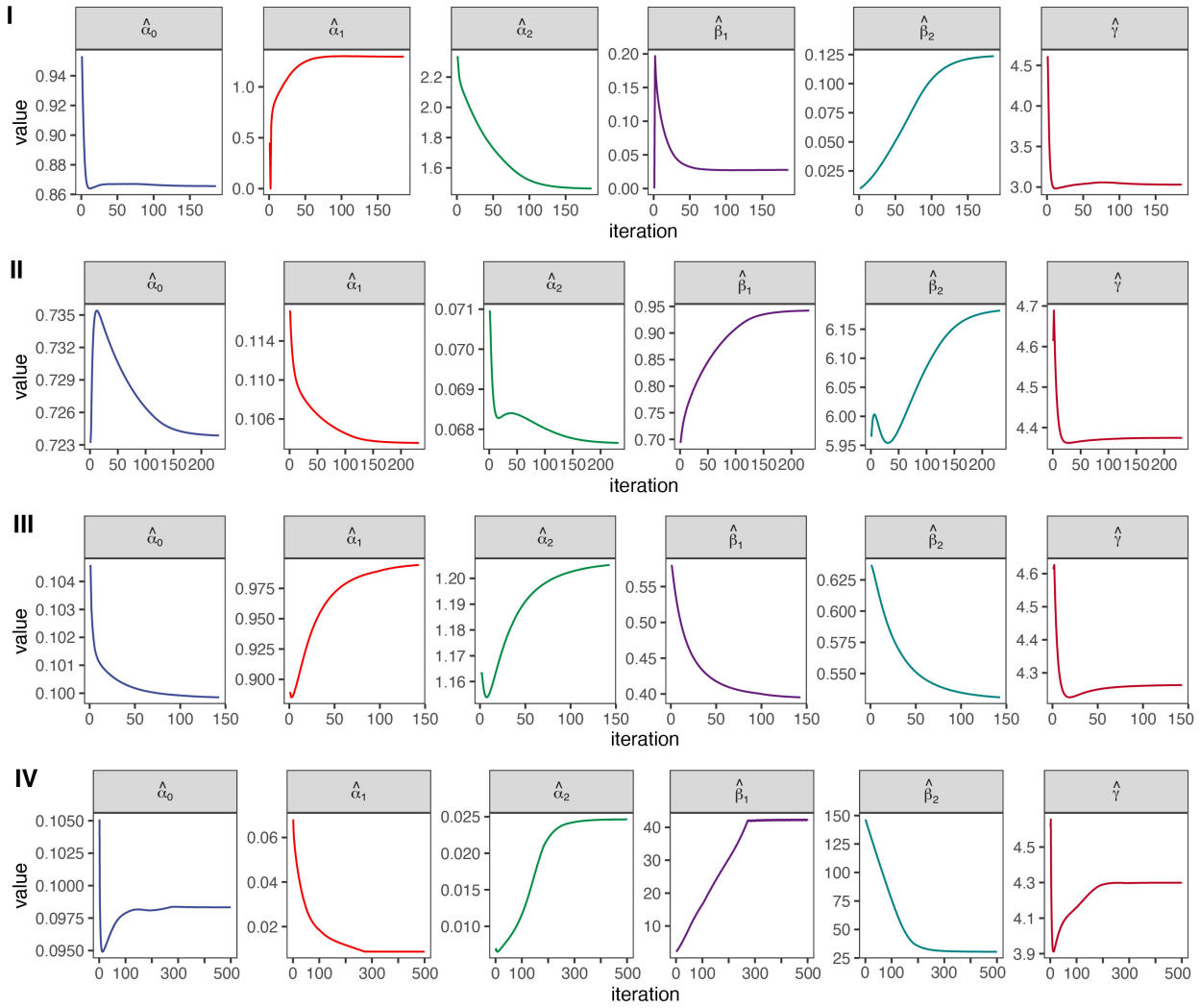

Figure S4: Iteration diagram of EM algorithm parameter estimation under different scenarios for PMB data.

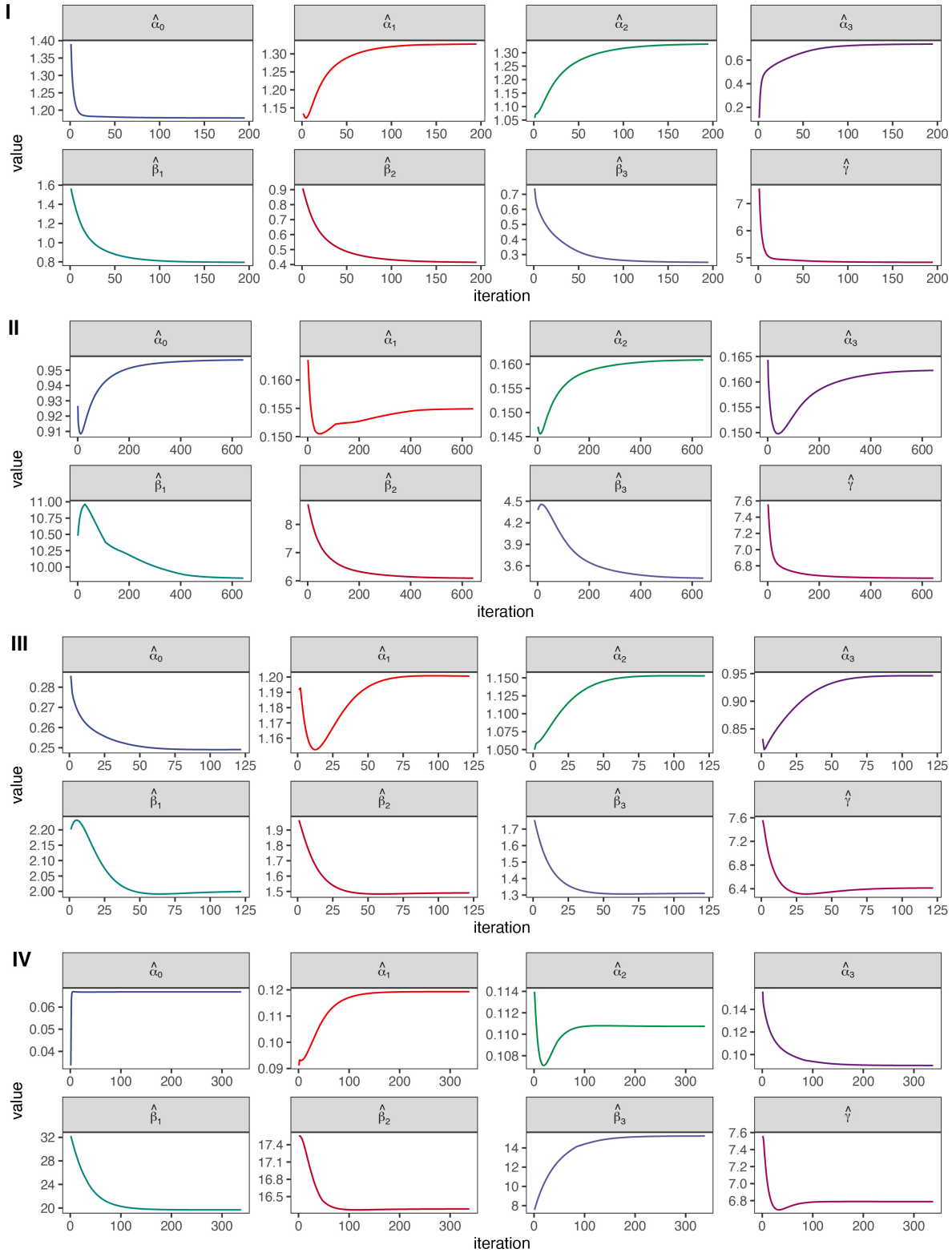

Figure S5: Iteration diagram of EM algorithm parameter estimation under different scenarios for fatigue crack-size data.

- **simulation**: "Integral_appr.R": the primary R script for replicating Figure 2 in the paper. This script evaluates the efficacy of various numerical integration techniques in approximating the CDF of failure time.
- **utility**: This directory contains a collection of essential functions for computational analysis:
  - "appr.R": Functions for numerical integration approximation methods.
  - "em.R": Functions related to the EM algorithm.
  - "fct.R": A collection of auxiliary functions regularly employed throughout the analyses.

For optimal interaction with these resources, it is recommended to open "multi-rIG.Rproj" using RStudio[2], install all necessary packages as initially specified, and proceed to execute the code sequentially, section by section.

[2]https://posit.co/download/rstudio-desktop/

References

P. N. Swarztrauber, On computing the points and weights for Gauss–Legendre quadrature, SIAM Journal on Scientific Computing 24 (2003) 945–954. doi:10.1137/S1064827500379690.

E. Babolian, M. MasjedJamei, M. Eslahchi, On numerical improvement of Gauss–Legendre quadrature rules, Applied Mathematics and Computation 160 (2005) 779–789. doi:10.1016/j.amc.2003.11.031.

G. H. Golub, J. H. Welsch, Calculation of Gauss quadrature rules, Mathematics of Computation 23 (1969) 221–230.

F. Novomestky, gaussquad: collection of functions for Gaussian quadrature, 2022. doi:10.32614/CRAN.package.gaussquad.

W. Q. Meeker, L. A. Escobar, F. G. Pascual, Statistical Methods for Reliability Data, John Wiley & Sons, 2022.

G. Fang, R. Pan, Y. Wang, Inverse Gaussian processes with correlated random effects for multivariate degradation modeling, European Journal of Operational Research 300 (2022) 1177–1193. doi:10.1016/j.ejor.2021.10.049.