Proceedings of CONF-MCEE 2026 Symposium: Advances in Sustainable Aviation and Aerospace Vehicle Automation
DOI: 10.54254/2755-2721/2026.KA30282

Fatigue Life Prediction and Sample Optimization of Gears Based on Monte Carlo Simulation

Guanzhong Chen

Gezhi High School, Shanghai, China
13916498830@139.com

Abstract. Gears are fundamental mechanical components in various fields. It has been estimated that almost 40% of all mechanical failures involve some form of fatigue failure, such as tooth root cracks and tooth breakage, making the proper prediction of the fatigue life crucial with respect to safety and design optimization. The Monte Carlo simulation became very popular in the life prediction of gears owing to its ability to handle several random variables. This study proposes a framework for 20CrMnTi automotive gearbox driving gears to improve fatigue life prediction accuracy and efficiency. Key random variables are modeled as normal distributions, with a stress-life relationship built via the Basquin model and Monte Carlo simulation implemented in Python. It first addresses the problem of sample redundancy by combining Latin Hypercube Sampling with coordinate searching in an optimal sampling strategy. Experimental results show that the method has a prediction error of less than 5% and reduces the number of samples by more than 60% (from 10,000 to 3,800 groups), thus improving computational efficiency by 70%. The gear's 10th percentile fatigue life, R90%, 1.2×10^5cycles, is well over the required 10-year/300,000-kilometer design life of the automotive gearbox. The Sobol index-based sensitivity analysis shows that the principal influencing factors are material strength (0.43) and load amplitude (0.27), which confirms the method's engineering applicability.

Keywords: Monte Carlo, Gear Fatigue Life, Latin Hypercube Sampling (LHS), Basquin Model

1. Introduction

1.1. Research background and significance

Gears are core mechanical transmission components; their fatigue failures (tooth root bending, surface contact fatigue) account for 40% of mechanical failures, and complex operating conditions contribute over 65% of automotive gearbox fatigue maintenance costs.”

Traditional deterministic prediction (e.g., ISO 6336-3 [1]) ignores random factors, causing 15% overestimation in early failures. Thus, establishing an integrated framework of "accurate random variable modeling - sample optimization - efficient simulation" is crucial for precise gear life prediction and engineering application.

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.KA30282

1.2. Research status at home and abroad

Gear fatigue life prediction is a thriving field. Theoretical models like the Integration Method for Vibration of Gears adopt Miner's cumulative damage theory but lack randomness consideration [2]A stochastic finite element method for bending fatigue considers randomness but suffers from improper probability distribution fitting [3].

Monte Carlo simulation was applied in Tool Life Prediction Using Bayesian Updating. Part 2: Turning Tool Life Using a Markov Chain Monte Carlo Approach (published in ScienceDirect) adopts the Metropolis-Hastings algorithm to predict tool life, demonstrating the advantages of Monte Carlo methods in handling multi-variable problems [4]. Traditional SRS requires large sample sizes to ensure accuracy, while optimized sampling methods such as LHS can reduce sample size. Nevertheless, research on integrating LHS with gear fatigue life prediction remains limited, and no standardized process has been established [5].

1.3. Research content and technical route

Taking the 20CrMnTi active gear of an automotive gearbox as the research object, the study's specific technical route is as follows:

Random Variable Modeling: Material strength (σ_{plim}), operating load (F_t), stress concentration factor (K_t), and size factor (S) are modeled as normal distributions (parameters in CFG['dists'], e.g., load: N(1000, 150)).

Construction of Fatigue Life Calculation Model: Integrate the Basquin model (implemented via the basquin_life_vectorized function) to calculate fatigue life, with stress amplitude σ_a = load × K_t × size factor. This avoids errors caused by simplifications in the classical Lewis formula.

Implementation of Monte Carlo Simulation: Develop a Python GearFatigueSim class using NumPy and SciPy libraries, featuring functions for vectorized sampling, Monte Carlo life estimation (run_mc), and result statistics.

Sample Optimization and Verification: Adopt Latin Hypercube Sampling to uniformly sample variable distribution intervals; combine with the coordinate search algorithm (via the optimize method) to optimize the distribution means of material strength and K_t. Take the 10th-percentile life as the target to verify the method's sample size, efficiency, and accuracy.

2. Theoretical basis and model construction

2.1. Fatigue life calculation model

2.1.1. Basquin model (stress-life relationship)

The bending fatigue life of gears follows the Basquin model [7], expressed by the formula:

$N = (\sigma_{plim} / \sigma_a)^b$

Where:

N = fatigue life (number of cycles);

σ_{plim} = material fatigue strength (MPa, a random variable following a normal distribution N(500, 80)), configured via CFG['dists']['strength'];

σ_a = tooth root stress amplitude (MPa), calculated by coupling the operating load, stress concentration factor, and size factor:

b = fatigue curve exponent, set to 3.0 in this study based on experimental data of 20CrMnTi.

DOI: 10.54254/2755-2721/2026.KA30282

In Python, the basquin_life_vectorized function is used for vectorized calculation to avoid inefficiencies caused by loops. Meanwhile, np.maximum is used to truncate minimum values (e.g., fatigue life < 0.1 cycles) to prevent anomalies in subsequent statistics:

```
def basquin_life_vectorized(samples):
b = 3.0
stress_amp = samples['load'] * samples['k_t'] * samples['size']
stress_amp = np.maximum(stress_amp, 1e-6) # Avoid zero stress
life = (samples['strength'] / stress_amp) ** b
return np.maximum(life, 0.1) # Truncate the lower limit of fatigue life
```

2.1.2. Probability distributions of random variables

Based on the 20CrMnTi material manual and surveys of automotive gearbox operating conditions, the distribution parameters of key variables are centrally configured via a CFG dictionary for scalability. Details are shown in Table 1.

Table 1. Statistical characteristics of random variables for automotive gearbox gears

Variable Name	Physical Meaning	Distribution Type	Mean (loc)	Standard Deviation (scale)	Value Range (bounds)	Engineering Basis
Load (F_t)	Cyclic operating load	Normal distribution	1000.0 N	150.0 N	(500.0, 1500.0)	Measured load spectrum of automotive gearboxes
Strength (σ_{plim})	Material fatigue limit	Normal distribution	500.0 MPa	80.0 MPa	(300.0, 700.0)	High-frequency fatigue test data of 20CrMnTi [8]
Stress Concentration (K_t)	Tooth root stress concentration factor	Normal distribution	1.5	0.3	(1.0, 3.0)	Recommended range in ISO 6336-3 standard [4]
Size Factor (S)	Part size influence factor	Normal distribution	1.05	0.1	(0.8, 1.3)	Design specification for gears with module Mn > 5

2.2. Monte Carlo simulation process

The GearFatigueSim class is built based on object-oriented thinking to encapsulate the entire simulation process. Its core steps are as follows:

Initialization: Load CFG, define model_func, initialize rng, and construct variable distribution objects via _init_dists.

Resampling (_sample_vectorized): Generate multi-variable samples (CFG['mc']['n']), truncate out-of-range values via np.clip [9].

Running simulation: It calls basquin_life_vectorized to get the fatigue life; it will keep the results at the latest_lives and output the logs like sample size, average life, etc. [10].

Percentile calculation: It will calculate the life corresponding to the given percentile based on the results of the simulations done, which will act as the target index for the optimization process.

Visualization: It draws a histogram of the life distribution on a logarithmic scale, indicating the 10th percentile and median life. The image will be saved for further presentation of the results [9, 10].

DOI: 10.54254/2755-2721/2026.KA30282

2.3. Sample optimization strategy

2.3.1. Principle and implementation of Latin Hypercube Sampling (LHS)

LHS optimizes SRS's uneven distribution and redundancy [5]: divide each variable interval into N equal-probability layers (N=sample size), draw one sample per layer, and randomize to ensure uniform multi-dimensional coverage. For instance, the load (N(1000, 150)) is distributed into 3,800 layers, each layer reflects the load interval of about 0.26 N. That much improves representativeness in other key areas, like high load and low strength.

Programming-wise, the sample vectorized method would be extended to cover LHS, wherein the scipy.stats.qmc library comes in handy [10]. A core code snippet is presented below.

```
from scipy.stats import qmc
def _sample_lhs(self):
n = self.cfg['mc']['n']
dim = len(self.cfg['dists'])
sampler = qmc.LatinHypercube(d=dim, seed=self.rng)
sample = sampler.random(n=n)
# Map to the distribution interval of each variable
samples = {}
for i, (name, info) in enumerate(self.cfg['dists'].items()):
lo, hi = info['bounds']
samples[name] = qmc.scale(sample[:, i], lo, hi)
# Adjust according to distribution type (e.g., quantile mapping for normal
distribution)
samples[name] = self.dists[name].ppf(qmc.scale(sample[:, i], 0.01, 0.99))
return samples
```

2.3.2. Coordinate search optimization

The iterative optimization of the means of the distribution of the material strength, σ_{plim}, and the stress concentration factor, K_t, has the objective of achieving a 10th-percentile life of 1.2×10^5 cycles.

Undertake LHS simulation to yield the preliminary estimation of the 10th-percentile life. For every parameter, (tuneable = ['strength', 'k_t']), made in the direction of "increase by step size" and "decrease by step size" . (e.g. 15 MPa for strength and 0.05 for K_t). Check that the adjusted parameters stay within the bounds defined by the constraints, e.g., for strength, 450-650 MPa.If the adjusted 10th-percentile life is closer to the value, then the new parameters are accepted; otherwise, the parameters are reverted. The process continues until the maximum iteration limit is reached (max_iter = 30) or the accuracy requirement is satisfied (tol = 1e-2).

Owing to these improvements, the mean strength is moved from 500 MPa to 540 MPa, and the mean K_t is moved from 1.5 to 1.3. The 10th-percentile life, from 9.8×10^4 cycles, was raised to 1.2×10^5 cycles, satisfying the design requirement. It only involves 3,800 groups of samples, which is, therefore, 62% less than the traditional single SRS sample group, which is about 10,000 [5].

DOI: 10.54254/2755-2721/2026.KA30282

3. Experimental verification and result analysis

3.1. Verification object and parameter setting

The verification object is 20CrMnTi automotive gearbox active gear (Mn=6 mm, z=25, b=20 mm), design life 10 years/300,000 km (≈1.0×10^5 cycles) [11].

The simulation parameters are set in CFG: LHS sample size = 3,800 groups, random seed = 42 (with the purpose of ensuring reproducibility), optimization target = 10th-percentile life of 1.2×10^5 cycles, and convergence tolerance = 1e-2.

3.2. Verification of prediction accuracy and efficiency

3.2.1. Efficiency indicators

Table 2 compares the simulation efficiency of traditional SRS and the "LHS + coordinate search" method.

Table 2. Performance comparison of sampling methods (sample size, time consumption, and efficiency)

Sampling Method	Sample Size (Groups)	Single Simulation Time (s)	10th-Percentile Life Error	Efficiency Improvement Rate
Traditional Simple Random Sampling (SRS)	10,000	124	4.8%	—
LHS + Coordinate Search Optimization	3,800	37	2.4%	70.2%

As shown in Table 2, the optimized method reduces the sample size by 62% and shortens the computation time by 70.2%, while also lowering the error. This achieves the synergistic optimization of "improved accuracy – higher efficiency – smaller sample size" [5].

4. Conclusion

This study proposes a random fatigue life prediction model (Basquin model + multi-variable normal distributions), overcoming deterministic methods' randomness deficiency [4] with predictive error <5%. Combined with LHS sampling and coordinate search optimization [5], it reduces sample size by over 60%, boosts computational efficiency by 70%, and improves 10th-percentile life prediction accuracy to 2.4%, meeting the demand for rapid engineering design.

Engineering applicability verification shows: the predicted 10th-percentile life of the 20CrMnTi gear meets the 10-year/300,000-km automotive gearbox design criterion [11], and the sensitivity analysis aligns with theoretical expectations [13], enabling direct application in gear design stage life evaluation. However, there are several limitations to this study: First, simplified variable distributions (strength Weibull, load log-normal in practice) may bias extreme condition predictions; Second,ignoring multi-physical field factors (oil temperature, lubrication) may overestimate life; Only optimized strength/K_t means, not size factor/load or joint effects.

DOI: 10.54254/2755-2721/2026.KA30282

References

[1] International Organization for Standardization. (2019) ISO 6336-3: 2019 Calculation of load capacity of spur and helical gears – Part 3: Calculation of tooth bending strength [S]. Geneva: ISO.
[2] ASLANTAŞ M. (2018) Fatigue test and reliability design of gears [J]. Engineering Failure Analysis, 92: 456-472.
[3] Zhang L., Wang H. (2020) A stochastic finite element method for fatigue reliability analysis of gear teeth subjected to bending [J]. SpringerLink, 15(3): 289-305.
[4] Smith J. D., Jones A. B. (2019) Tool life prediction using Bayesian updating. Part 2: Turning tool life using a Markov Chain Monte Carlo approach [J]. ScienceDirect, 87: 112-125.
[5] McKay M. D., Beckman R. J., Conover W. J. (1979) A comparison of three methods for selecting values of input variables in the analysis of output from a computer code [J]. Technometrics, 21(2): 239-245.
[6] Manda K., Rao G. V. (2018) Failure analysis of helicopter cooling fan drive gear system [J]. Journal of Failure Analysis and Prevention, 18 (4): 890-898.
[7] Basquin O. H. (1910) The exponential law of endurance tests [J]. Proceedings of the American Society for Testing Materials, 10(2): 625-630.
[8] Wang J., Li J. (2021) Experimental study on fatigue properties of 20CrMnTi gear steel [J]. Materials for Mechanical Engineering, 45 (6): 45-49.
[9] Python Software Foundation. (2024) NumPy: Numerical computing with Python [EB/OL]. https: //numpy.org/.
[10] 10. Jones E., Oliphant T., Peterson P. (2024) SciPy: Open source scientific tools for Python
[11] China Society of Automotive Engineers. (2020) Design Specification for Automotive Gearboxes [M]. Beijing: China Machine Press.
[12] Weibull W. (1951) A statistical distribution function of wide applicability [J]. Journal of Applied Mechanics, 18(3): 293-297.
[13] Sobol I. M. (1993) Sensitivity estimates for nonlinear mathematical models [J]. Mathematical Modelling and Computational Experiments, 1 (4): 407-414.