DOI: 10.2478/ata-2024-0018

Acta Technologica Agriculturae 3
Nitra, Slovaca Universitas Agriculturae Nitriae, 2024, pp. 133–141

PREDICTION OF MACHINE'S CENTRE OF GRAVITY DISLOCATION WITH MONTE CARLO SIMULATION

Jozef RÉDL[1], Grigory BOYKO[2]*, Davood KALANTARI[3]

[1]Slovak University of Agriculture in Nitra, Faculty of Engineering, Institute of Design and Engineering Technologies, Nitra, Slovakia, jozef.redl@uniag.sk
[2]Volgograd State Technical University, Chair Department of Operation and Maintenance of Automobiles, Volgograd, Russian Federation, boyko@vstu.ru
[3]Sari Agricultural Sciences and Natural Resources University, Faculty of Agricultural Engineering, Department of Mechanics and Biosystems Engineering, Sari, Iran, dkalantari2000@yahoo.com

*correspondence: boyko@vstu.ru

The estimation of machine's centre of gravity (COG) dislocation is presented in this article. Various types of agricultural tractors' parameters were analysed. Their parameters such as track width in the range of $\langle 1222\text{ mm}, 1900\text{ mm}\rangle$ and COG height in the range of $\langle 830\text{ mm}, 1340\text{ mm}\rangle$ are known from the reports of OECD tests. The static stability of the machine is defined by the critical angle of labile equilibrium or critical overturning angle in static disposition, which is determined from measurement on the tilt table. The critical overturning angle can be calculated from the known parameters of COG dislocation. The critical overturning angle for each analysed machine can be evaluated from the obtained dataset. The mean and standard deviation of parameters were used as input parameters into Monte Carlo simulation (MCS). The MCS method was used for the estimation of the critical overturning angle and height of COG for 2500 data. Simulation was performed using a Mathcad® Prime 9.0 software. The basic statistical analysis was performed using a Spyder software in the Python language. The polynomial approximation equation was set up, and then COG heights were calculated. The critical overturning angle was calculated from the real and simulated parameters. The functions of the critical overturning angle were compared, and the fit of approximation was evaluated by the R^2 parameter. The designed model was verified by the *F*-distribution test.

Keywords: mathematical modelling; random dataset; approximation; data estimation

Nowadays, in the OECD (Organization for Economic Co-operation and Development) test reports of agricultural tractors, the values of dislocation of the COG are omitted. For this reason, an experimental test must determine the dislocation of the COG height above the ground. Calculating the vertical and longitudinal coordinates of the COG could be solved by applying an ISO standard (Majdan et al., 2021). The Slovak national standard STN 27 8154, which is based on ISO 5005:1977, describes the basic principles of determining the dislocation of COG of earthmoving machines. The applied algorithm, based on the mentioned STN and ISO standards for calculating the dislocation of the COG of agricultural machines, must consider the corrections provided by error calculation (Rédl and Páleš, 2018). In many cases, experimental measurement is technically difficult or impossible for too weighty machines. The research performed by the authors about the existence of a similar presented methodology for estimating the height of the COG has not been satisfied. For this reason, they developed an alternative methodology for estimating the height of tractor's COG based on using the Monte Carlo method (MCM).

MCMs are a broad class of computational algorithms that rely on repeated random sampling to obtain a numerical result. They are often used in physical and mathematical problems when it is impossible to obtain an analytical solution, and the application of a direct algorithm is infeasible.

The MCM is mainly used in three distinct problems: numerical integration, simulation, and optimisation (Walker, 2016). The MCM is an algorithm that uses the uncertainty of problem's inputs to determine the uncertainty of outputs. The algorithm takes a sample from inputs, calculates the output, and repeats this process many times to converge to an output with uncertainty. The MCM has many past technical applications in vehicle dynamics. A review of various works has been presented, which highlights the importance of introducing the variability of the road-track/vehicle system into dynamic simulations as soon as this latter is meant to be predictive (Funfschilling and Perrin, 2019). In their work, a stochastic framework is adopted, uncertainties are propagated through the dynamic model thanks to Monte Carlo (MC) sampling, and stochastic abacus is built. Moreover, the numerical approach allows the estimation of the margin between the unloading criterion and vehicle overturning.

The combination of the research methods of vibration or vibration damage estimation under stochastic loading in the aerospace application, the combination of the finite element method and Monte Carlo simulation (MCS) and the probabilistic assessment of structures and failure predictions were presented by many authors (Páleš et al., 2014; Ptak and Czmochowski, 2023; Hamdia and Ghasemi, 2023).

A comprehensive review of MCMs, simulation, and its related subjects, which are covering 444 references in the relevant literature, was done (Song and Kawai, 2023). The authors present an extensive review of formulations and techniques along with insightful summaries of developments of existing numerical methods which are dealing with the utilisation of classic MCMs and Markov Chain MCM. They are analysing the various areas of science, for example, the areas of artificial neural networks, structural reliability, the sensitivity analysis of the probability of failure, structural reliability analysis, variance reduction techniques for structural reliability analysis and many more.

Material and methods

Parameters of machines

The test protocols of OECD (DLG, 2024) were used to gather the technical parameters of the analysed tractors. The test protocols contain many parameters determined by machine testing. All the companies providing the tests have been certified. The tractors used and their relevant parameters are shown in Table 1.

For simplification of design process of model, the parameter Δy (the shift of the y coordinate of COG) is

Table 1 Tractors' parameters

No.	Tractor type	Manufacturing year	COG height h (mm)	Shift of the y coordinate of COG Δy (mm)	Track width tw (mm)
1	CASE INTERNATIONAL 7130 4WD	1988	1074	5	1626
2	CASE INTERNATIONAL 7140 4WD	1988	1120	13	1626
3	CASE INTERNATIONAL 9110	1987	1158	0	1524
4	CASE INTERNATIONAL 9130	1987	1202	0	1524
5	CASE INTERNATIONAL 9150	1987	1334	0	1524
6	CASE INTERNATIONAL 9170	1987	1366	0	2032
7	CASE INTERNATIONAL 9180	1987	1421	0	1935
8	DEUTZ-FAHR AGROPLUS 60	1998	818	8	1400
9	DEUTZ-FAHR AGROPLUS 70	1997	890	6	1400
10	DEUTZ-FAHR DX 4.51	1989	958	12	1600
11	FIAT 140-90 DT/12	1985	1090	8	1720
12	FIAT 160-90 DT/1	1985	1090	8	1720
13	FIAT 44-28	1981	1340	12	1830
14	FIAT F100 DT/16	1992	840	3	1550
15	FIAT F130 DT/16	1992	835	3	1450
16	FORTSCHRITT ZT 323 A	1985	929	1	1766
17	KUBOTA M1-100S-DT	1993	929	4	1500
18	KUBOTA M-110-DT	1999	1009	2	1530
19	KUBOTA M-120-DT	1999	1002	1	1510
20	KUBOTA M-9580-DT	1993	929	4	1500
21	MASSEY FERGUSON 4270	1997	974	2	1525
22	MASSEY FERGUSON 4260	1998	974	2	1735
23	MASSEY FERGUSON 4255	1998	962	7	1735
24	MASSEY FERGUSON 4235	1998	945	4	1635
25	MASSEY FERGUSON 4245	1996	951	4	1635
26	NEW HOLLAND TN 65D	1999	840	10	1222
27	NEW HOLLAND TL 80	1999	830	13	1526
28	NEW HOLLAND TL 100	1999	850	12	1350
29	STEYR 9090.70	2002	936	12	1870
30	STEYR 8180	1989	1101	6	1900
31	VOLVO BM VALMET 2105	1985	1135	2	1920

Source: DLG, 2024

neglected. The effect of the offset of COG with respect to the pitch axis on the critical slope angle is minimal.

Critical slope angle

In the general position, as depicted in Fig. 1, the tractor has a stable static position on slope, and the COG is defined with h and $+\Delta y$. From known parameters, angle φ_{Lim} can be derived as follows if $+\Delta y$ is neglected:

$$\varphi_{Lim} = \text{atan}\left(\frac{tw}{2 \cdot h}\right) \quad (1)$$

During the tilting process, the tire-ground contact point C changes its position to the position C_{Lim} (Fig. 1). The angle orientation φ_{Lim} of the line $\overline{B_{Lim}C_{Lim}}$ with respect to the ground is represented by the horizontal line ToS (the tip of the slope) and $\left|\overline{B_{Lim}C_{Lim}}\right| = tw$. The solved limiting angles, which are also critical overturning angles, are listed in Table 2. The advantage of our approach is in the fact that we prevent mistakes with the exact definition of the contact point B between the tire and ground with respect to tire lateral deformation during the tilting. We assume that the mentioned factor was considered during the testing process by the OECD.

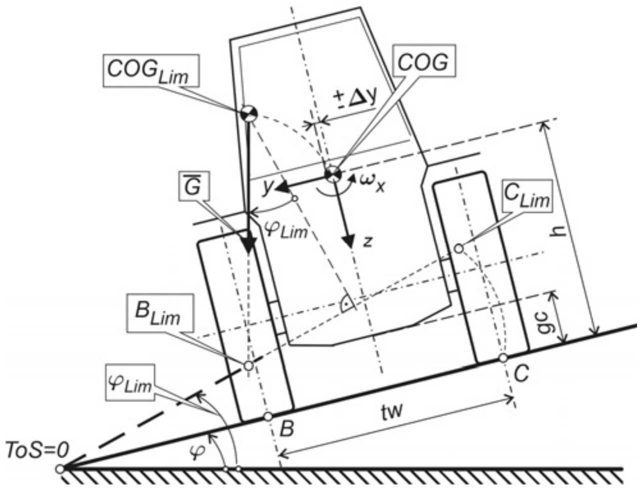

Fig. 1 Static tilt position of the tractor (on the tilt table)

Monte Carlo method

The MCM is a statistical simulation method that may be contrasted with conventional numerical discretisation methods, which are typically applied to ordinary or partial differential equations that describe some underlying physical or mathematical system. In many applications of MCM, the physical process is simulated directly, and there is no need to even write down the differential equations that describe the behaviour of the system. The only requirement is that the physical (or mathematical) system is described by a probability distribution function (PDF). Once PDFs are known, MCS can proceed by a random sampling of PDFs. Many simulations are then performed (multiple "trials" or "histories"), and the desired result is taken as an average over the number of observations (which can be a single observation or perhaps millions of observations). There are at least four crucial ingredients in a basic MC strategy (Steinhauser, 2013; Ross, 2013). These are:

1. Generating random variables;
2. PDFs, $\int_{x_{\min}}^{x_{\max}} dx\, p(x) = 1$, where $-\infty < x_{\min} < x_{\max} < +\infty$;
3. The moments of a PDF, $\langle x^n \rangle = \int_{x_{\min}}^{x_{\max}} dx\, x^n p(x)$;
4. The calculations of variances, $\text{var}\{x\} = \langle x^2 \rangle - \langle x \rangle^2$.

The number n of MCS iterations is determined by Eq. (2) (Bukaçi et al., 2016) as follows:

$$n = \left(\frac{100 \cdot z_c \cdot S_x}{\overline{x} \cdot E}\right)^2 \quad (2)$$

where: z_c – a variable depending on the confidence level for normal distribution; S_x – standard deviation; $\overline{x}$ – mean; E – percentage error bound

Results and discussion

Statistical probability tests of input data

We assumed that the probability distribution of all input values has normal (logistic) distribution. This hypothesis was tested by Wilk and Anderson-Darling test algorithms at significance level 0.05, which were performed using the Spyder software IDE (Integrated Development Environment) (Python, 2023) in the Python language with algorithms 1 and 2. Experimental values are easily imported from Excel worksheets into the Spyder and Mathcad IDEs as a matrix.

Algorithm 1 Shapiro test Python code

```
from scipy import stats
# ficritcsv – imported data from Excel
p_value = stats.shapiro(ficritcsv)
# Set a significance level (alpha) for test
alpha = 0.05
# Checking the p-value against the significance level
if p_value > alpha:
print("Sample has a normal distribution")
else:
print("Sample does not has normal distribution")
```

Output from IPython console:

ShapiroResult(statistic=0.9790266752243042, pvalue=0.7850747108459473)

Algorithm 2 Anderson-Darling test Python code

```
statistic, critical_values, significance_level = stats.
anderson(ficritcsv, dist='distr')
if statistic < critical_values[2]:
        print("Data has a 'distr' distribution (at", alpha,
                "significance level)")
else:
    print("Data does not has a 'distr' distribution (at", alpha,
                "significance level)")
```

In algorithm 2, *distr* represents the variations of tested distributions: *expon* – exponential; *normal* – normal;

logistic – logistic; *gumbel_r* – Gumbel right; *gumbel_l* – Gumbel left. The probability distribution tests proved that the input data has normal distribution. All other distribution types are rejected due to tests results. The Python code was used for its very useful form of commands and very user-friendly environment of Spyder software.

Monte Carlo simulation algorithm setup

To set up an algorithm, the basic MCS strategy mentioned above was followed. The flow chart diagram depicted in Fig. 2 describes the details for each step of algorithm. The basic technical parameters are stored in the Excel sheet. In the Mathcad solving block, the preliminary values of critical overturning angles, means, and standard deviations for angles and vehicles tracks width were solved. These values are the basic components for MCS. After solving the critical overturning angles array, the type of probability distribution of input values was determined. For this purpose, the Python programming language in the Spyder software environment was used.

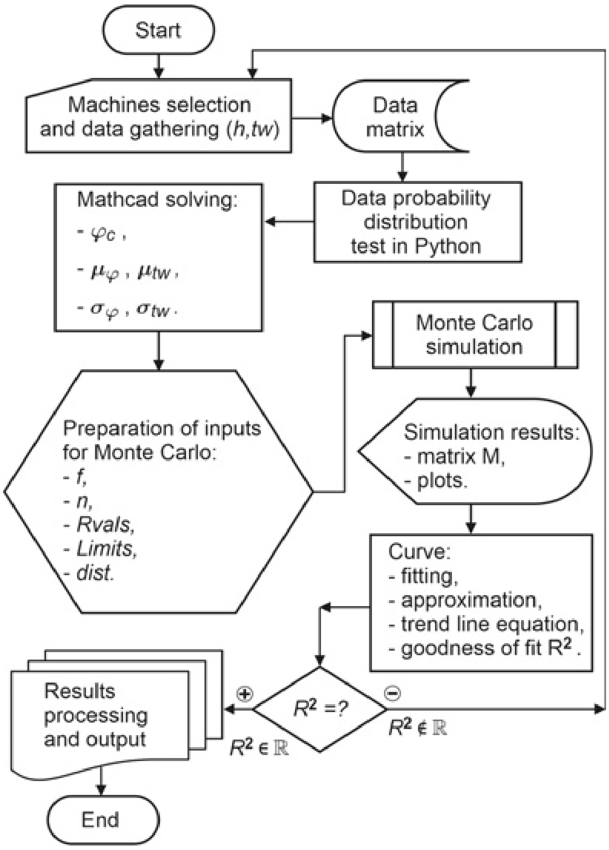

Fig. 2 Simulation flow chart

The next step was the setup of the input parameters for MCS (which are standard deviation and mean). The derivation of simulation function is based on Fig. 1. The simulation function must satisfy the real solving of Eq. (1). The results of MCS are the simulation matrix with range $n \times 3$. For MCS results utilisation, it is necessary to create a chart with simulated parameters. That belongs to the part "Curve" (Fig. 2). Desing the approximation function should by performed by polynomials or Fourier transformations approximation. The calculation of the coefficient of determination R^2 refers to the quality of the goodness of fit of designed approximation function. If R^2 has an acceptable value, then the results and outputs should be processed.

Simulation results

Firstly, from the known parameters listed in Table 1, a histogram and distribution function curve as a function variable $f(\varphi_{cri}, range)$ and $f(tw, range)$ were set up. The charts are depicted in Figs 3 and 4. All the charts are processed using the Mathcad® Prime software.

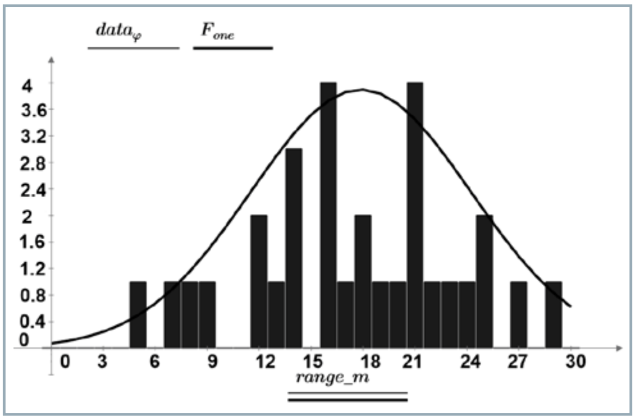

Fig. 3 Histogram and distribution function of φ_{cri}
*data*φ – the number of elements in each bin; φ_{cri}, F_{one}, F_{two} – distribution functions; *range_m* – the number of subintervals with the same length

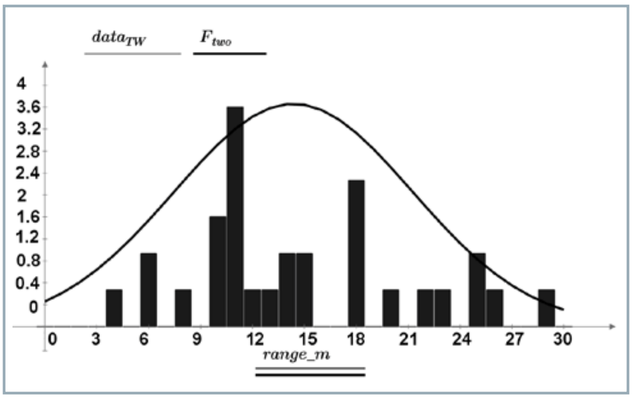

Fig. 4 Histogram and distribution function of *tw*
*data*φ – the number of elements in each bin; φ_{cri}, F_{one}, F_{two} – distribution functions; *range_m* – the number of subintervals with the same length

As proved by a test provided by Python, the input parameters in analysis have normal distribution. Normal distribution is identified in Figs 3 and 4 where the histogram charts have an additional curve of normal distribution. Setting up the simulation function (in which the term *critical* in subscript was used instead of *limit*), is as follows:

$$f(\varphi_{cr}, h) := h \cdot \tan(\varphi_{cr}) \quad (3)$$

Then, the mean and standard deviations for critical overturning angles and track width were solved as follows:

$$\mu_\varphi := mean\,(\varphi_{cr}) = 0.673,\ \mu_{tw} := mean\,(tw) = 816.613$$
$$\sigma_\varphi := Stdev\,(\varphi_{cr}) = 0.06,\ \sigma_{tw} := Stdev\,(tw) = 94.373 \quad (4)$$

The matrix *rvals* contains a string naming of each variable in the first column, their mean (μ) values in the second column, and their standard deviations (σ) in the third column:

$$rvals := \begin{bmatrix} "\varphi" & \mu_\varphi & \sigma_\varphi \\ "h" & \mu_{tw} & \sigma_{tw} \end{bmatrix} = \begin{bmatrix} "\varphi" & 0.673 & 0.06 \\ "h" & 816.613 & 94.373 \end{bmatrix} \quad (5)$$

Then, the limits matrix and distribution matrix were prepared:

$$limits := \begin{bmatrix} 0.35 & 0.785 \\ 350 & 1200 \end{bmatrix},\ disrt := \begin{bmatrix} normal \\ normal \end{bmatrix} \quad (6)$$

For setting up the limits for MCS, the Mathcad function *min*, *max* was used. For increasing the simulation interval purpose, we didn't use the solved limits, but we used the real limits based on tractors' known parameters and the expected boundaries where random variables are clipped. The distribution matrix has defined normal distribution proved above. The function of MCS in the Mathcad® Prime software has the following form:

$$M := montecarlo\,(f, n, rvals, limits, distr) \quad (7)$$

where: the variable n represents the number of samples. In our case, $n = 2500$, and solution is based on Eq. (2) with the following variables set: $z_c = 2.58$; $S_x = \sigma$; $\bar{x} = \mu$. It must be considered that the simulation function has two parameters as defined in Eq. (3). For that reason, n must be solved for the interval $\langle n\varphi, n_{tw} \rangle$ as follows:

$$n_\varphi = \left(\frac{100 \cdot z_c \cdot \sigma_\varphi}{\mu_\varphi \cdot E_\varphi} \right)^2 = 2500\,;\ E_\varphi = 0.453$$
$$n_{tw} = \left(\frac{100 \cdot z_c \cdot \sigma_{tw}}{\mu_{tw} \cdot E_{tw}} \right)^2 = 2502;\ E_{tw} = 0.585$$

$$M = \begin{bmatrix} "\varphi" & "h" & "Output" \\ 0.646 & 747.495 & 563.787 \\ 0.644 & 721.82 & 542.096 \\ 0.571 & 815.721 & 524.524 \\ 0.665 & 864.124 & 678.011 \\ 0.721 & 904.583 & 795.013 \\ 0.724 & 898.017 & 794.363 \end{bmatrix}$$

Fig. 5 MC matrix

Increasing the number of samples causes the highest precision of simulated dataset, and the error of simulation is normalised if $n \geq 1000$ (Bukaçi et al., 2016). For that reason, $n = 2500$ was chosen and it lays in interval defined above. The simulation result is the matrix of n MC samples created by calculating function f for variables randomly generated given the information found in the matrix *rvals*. A part of the results matrix is depicted in Fig. 5. The first column "φ" is the set of critical overturning angles (φ_{cr}) in radians; "*h*" is the set of COG height (h); and the "*output*" is the track width (tw), but of a half of values, all in millimetres.

The dataset is not sorted. The new matrix is sorted in the ascending order of angles and full track width. The histograms of result matrix elements are depicted in Figs 6, 7, 8. The generated values have a typical shape of normal distribution. The variables $data_\varphi$, $data_h$ are the number of elements in each bin; *range* is the number of subintervals with the same length.

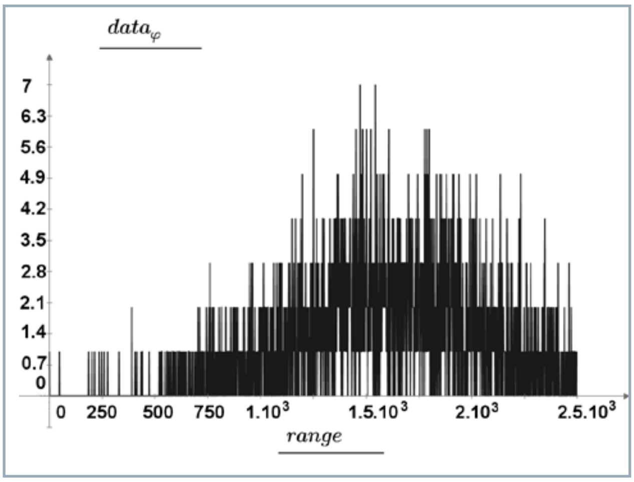

Fig. 6 Histogram of generated critical overturning angles φ_{cr}

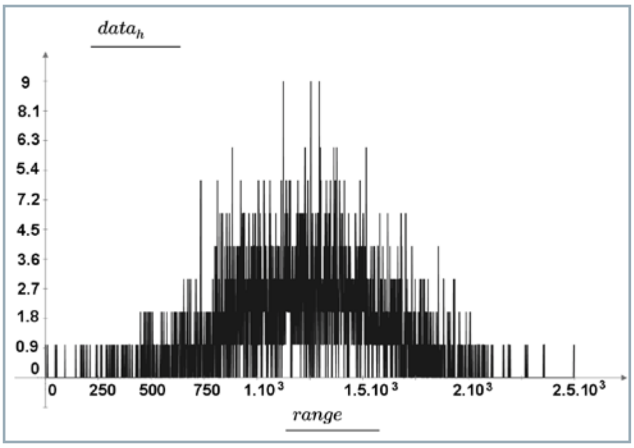

Fig. 7 Histogram of generated COG heights h

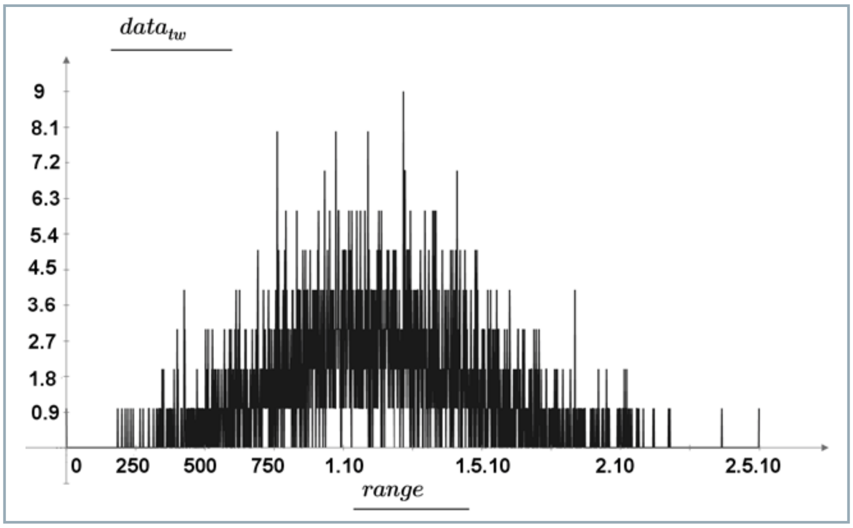

Fig. 8 Histogram of simulated track width *tw*

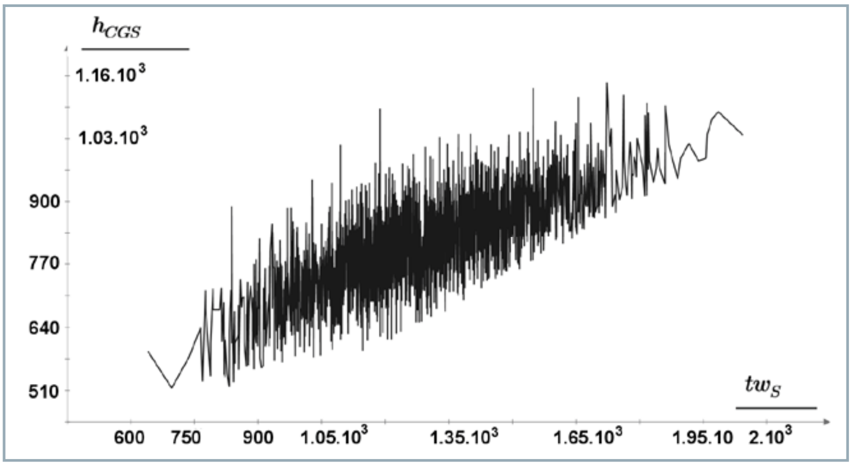

Fig. 9 Generated relationship between track width and COG height

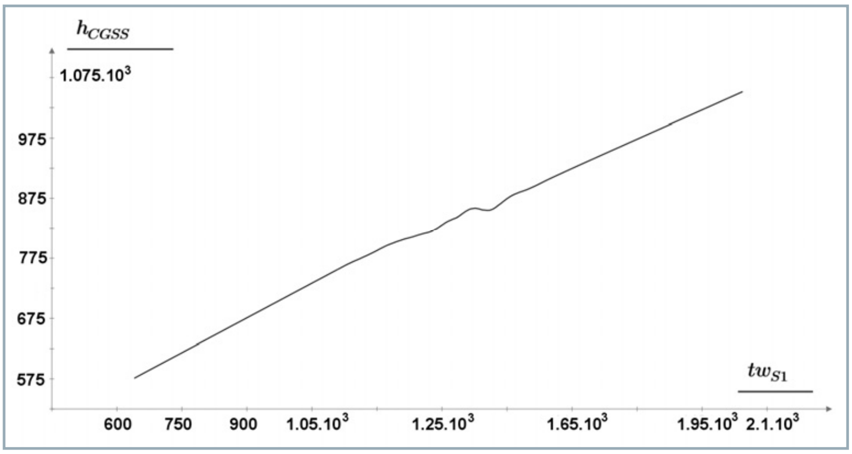

Fig. 10 Smoothed track width and COG height

The functional dependency of simulated track width tw_S and COG height h_{CGS} is depicted in Fig. 9, with a strict stochastic form. From this dependency, any usable conclusions cannot be predicted. Deriving the approximation function is too difficult. For that reason, we tried to smooth the function shape and derive the approximation function. Mathcad® Prime has the following inbuilt functions for smoothing the datasets: *ksmooth*; *medsmooth*; *supsmooth*.

The *supsmooth* function uses a symmetric *k* nearest neighbour linear least-squares fitting procedure to make a series offline segment through data. Unlike *ksmooth* which uses a fixed bandwidth for all data, *supsmooth* will adaptively choose different bandwidths for different portions of data (Mathcad Help, 2015). The function *supsmooth* according to Eq. (8) has been used:

$$h_{CGSS} := supsmooth\,(tw_{S1}, h_{CGS1}) \quad (8)$$

where: h_{CGSS} – COG height (mm), simulated and smoothed; tw_{S1} – track width (mm), simulated; h_{CGS1} – COG height (mm), simulated

The smoothed chart is depicted in Fig. 10. The relationship between simulated track width and COG height has a quasi-linear nature. The fourth order polynomial approximation regression function by function *polyfitc* can be set up as follows:

$$h_{CG} = -8 \cdot 10^{-11} \cdot tw^4 + 5 \cdot 10^{-7} \cdot tw^3 - 0.0013 \cdot tw^2 + 1.7413 \cdot tw - 161.2 \quad (9)$$

The goodness of fit of the plot of smoothed data (Fig. 10) is $R^2 = 0.9988$.

Verification of derived approach

For verification purposes, tractors' parameters shown in Table 2 have been used. Predicted COGs based on Eq. (9) and are listed in Table 3. The dataset in Table 2 is obtained from other independent source than the dataset in Table 1. For the input parameters in Eq. (9), the known tractors' track width from Table 2 was used. The intervals of the real verificatory dataset are $tw_R \in \langle 945 \text{ mm}, 1518 \text{ mm} \rangle$ and

Table 2 Verification group of tractors

No.	Track width (mm)	COG height (mm)
1	945	691
2	1078	645
3	1120	656
4	1140	729
5	1182	510
6	1245	728
7	1250	611
8	1286	616
9	1305	654
10	1325	659
11	1518	619

Source: Franceschetti et al., 2019

Table 3 Simulated tractor parameters from Eq. (9)

No.	Track width (mm)	COG height (mm)
1	945	681.551
2	1078	723.54
3	1120	734.918
4	1140	740.057
5	1182	750.299
6	1245	764.37
7	1250	765.425
8	1286	772.765
9	1305	776.464
10	1325	780.234
11	1518	810.661

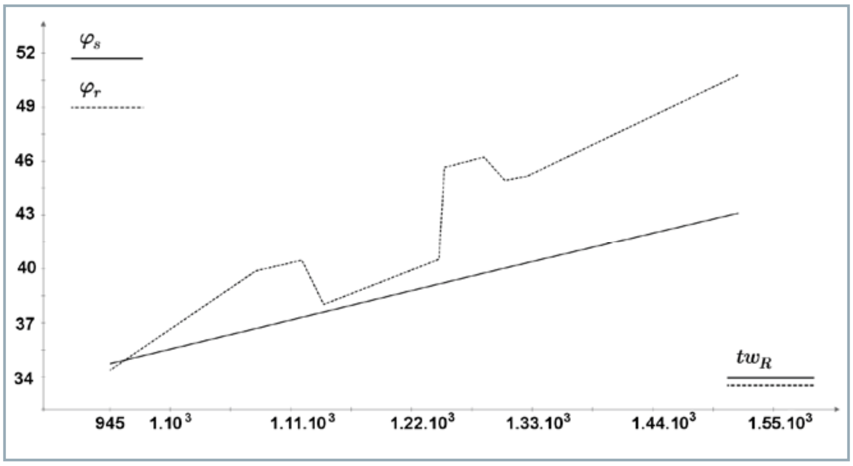

Fig. 11 Simulated (φ_s, deg) and real (φ_r, deg) values of critical overturning angles

$h_{COGR} \in \langle 510\ \text{mm}, 729\ \text{mm} \rangle$ (Table 2). The interval of the calculated data from Eq. (9) is $h_{COGR} \in \langle 681.551\ \text{mm}, 810\ \text{mm} \rangle$. The intervals of the real dataset from Table 1 are $tw \in \langle 1222\ \text{mm}, 1900\ \text{mm} \rangle$ and $h_{COG} \in \langle 830\ \text{mm}, 1340\ \text{mm} \rangle$. Because of the difference of intervals, critical overturning angles were used as the criterion for verification. The key parameter in MCS was the critical overturning angle. From the real dataset and simulated dataset, the real φ_r and simulated φ_s critical overturning angles were calculated according to Eq. (1). The critical overturning angles functional dependency with respect to real track width are depicted in Fig. 11.

To evaluate the process of critical overturning angles approximations, the function *polyfitstat* was used in the following form:

$$pfsa = polyfitstat\,(\varphi_r, \varphi_s, 4, 0.95) \quad (10)$$

where: the order of polynomial has a value equal to 4 as a generated approximation polynomial (Eq. 9), and confidence interval is 0.95 (95%). The result matrix according to Eq. (10) is depicted in Fig. 12 with basic statistical information. The fit of approximation is R^2 = 0.906. The *polyfitstat* matrix provided a few submatrices with partial evaluation results

$$pfsa = \begin{bmatrix} \text{“Regression Analysis”} & \text{“Value”} \\ \text{“Standard Deviation”} & 0.954 \\ \text{“R2”} & 0.906 \\ \text{“Adjusted R2”} & 0.831 \\ \text{“Predicted R2”} & 0 \\ \text{“PRESS”} & 1.59 \cdot 10^3 \\ \text{“Durbin–Watson”} & 1.651 \\ \text{“Coefficients”} & [6 \times 8] \\ \text{“ANOVA”} & [10 \times 6] \\ \text{“Diagnostics”} & [11 \times 9] \end{bmatrix}$$

Fig. 12 Polyfitstat matrix

For approximation, we used these results to evaluate the fit between the observed results and expected results based on the model, which is Eq. (9). A part of the *polyfitstat* submatrix is shown in Fig. 13. The following variables were defined: F_0 = 12.051 –

$$pfsa_{8,1} = \begin{bmatrix} \text{"Source"} & \text{"SSE"} & \text{"DF"} & \text{"MSE"} & \text{"F"} & \text{"P"} \\ \text{"Regression"} & 43.889 & 4 & 10.972 & 12.051 & 0.009 \\ \text{"Lack of Fit"} & 4.552 & 5 & 0.91 & NaN & NaN \\ \text{"Pure Error"} & 0 & 0 & 0 & NaN & NaN \\ \text{"Total"} & 48.442 & 9 & NaN & NaN & NaN \end{bmatrix}$$

Fig. 13 Polyfitstat submatrix

F factor for hypothesis; $\alpha = 0.05$ – the level of significance; $df_{prm} = 4$ – the degrees of freedom for parameters; $df_{err} = 5$ – the degrees of freedom for error.

The critical F value (F_C) was solved according to Eq. (11) as follows:

$$F_C = qF\,(1 - \alpha, df_{prm}, df_{err}) = 5.192 \qquad (11)$$

where: the qF is the F-distribution function (Montgomery, 2013)

The mentioned parameters are taken from the submatrix in Fig. 13. With the Mathcad comparison of inequality $F_0 > F_C = 1$, it was proven that the hypothesis of our model fits the data. The comparison result is equal to 1 (true). For the regression analysis, the *polyfitstat* calculation tool is recommended because the method returns statistical data on a polynomial regression model, fitting the results recorded in matrix **Y** to the data found in matrix **X** (PTC, 2023). The chosen MCM allows users to model the dataset with defined probability distribution. The MCM is an algorithm that uses the uncertainty of problem's inputs to determine the uncertainty of outputs. The algorithm takes a sample from inputs, calculates the output, and repeats this process many times to converge to an output with uncertainty. The MCM has many past applications in decision analysis (Flumerfelt et al., 2019). Another alternative to curve approximation is using the B-Spline curve approximation algorithm. The most effective way to determine curves for practical use is to use a set of control points. These control points can be accompanied by other restrictions for the curve, for example, boundary conditions or conditions for curve continuity (Páleš et al., 2016). However, the mentioned method doesn't give a simple usable approximation polynomial equation. Our approach of prediction of the dislocation of COG with respect to tractor' track width is rare, and our literature review did not find similar published solution to the presented problem. However, the MCM is widely used in multibody dynamics, for example for the computation of torque control of fully actuated non-deterministic multibody systems. The comparison between the time efficiency and accuracy of the polynomial chaos expansion (PCE) method extended with computed control torque law and the traditional MCM indicates that PCE can provide the desired accuracy in a more time effective manner for some engineering problems (Sabet and Poursina, 2017).

Conclusion

The data of dislocations of COG and track width of 31 agricultural tractors are processed for MCS. The tractors have a track width in the interval $\langle 1222 \text{ mm}, 1900 \text{ mm} \rangle$ and COG in the interval $\langle 830 \text{ mm}, 1340 \text{ mm} \rangle$. The test of statistical distribution proved the normal distribution of data. The dataset of the critical overturning angle of stability was calculated from known parameters. The MC model was applied to generate the dataset of COG heights and track widths for 2500 values. From obtained data from MCS, the smoothed dataset was designed and then the approximation function was set up (which was the polynomial of the fourth order). To verify the designed model, tractors' data were taken from a second independent source. The critical overturning angle of stability was calculated from the second source. Then, the critical overturning angle of stability was generated by our model, which is Eq. (9). The fit of simulated data was tested with calculated data from the second source. The model was tested with the *polyfitstat* function from the Mathcad Prime 9.0.0 software and with F-distribution test. The goodness of fit was $R^2 = 0.906$. The F-distribution test proved that the designed model fits the expected dataset.

The F-test is widely used in analysing the variance of two equal populations. The F-test is extensible to three or more groups and so is useful for multiclass classification (Klosterman, 2021). We recommend that the presented model should be used to predict the COG height of the tractor for the machine parameters lying in the recommended intervals of COG height and track width. We recommend the applied algorithms and methods for the estimation of dislocation of the COG of four-wheel agricultural machines with non-articulated frames with respect to the track width intervals mentioned in this article.

Acknowledgment

This research was supported by the European Union funding, under the project Erasmus+ KA2: Fostering Internationalization in Agricultural Engineering in Iran and Russia, project ID 585596-EPP-1-2017-1-DE-EPPKA2-CBHE-JP.

References

BUKAÇI, E. – KORINI, T. – PERIKU, E. – ALLKJA, S. – SHEPERI, P. 2016. Number of iterations needed in Monte Carlo Simulation using reliability analysis for tunnel supports. In International Journal of Engineering Research and Applications, vol. 6, no. 6, pp. 60–64.

DLG (Deutsche Landwirtschafts-Gesellschaft). 2024. Available at: https://www.dlg.org/de/

FLUMERFELT, S. – SCHWARTZ, K. G. – MAVRISM, D. – BRICENO, S. 2019. Complex Systems Engineering: Theory and Practice. Reston, VA : American Institute of Aeronautics and Astronautics, 282 pp. eISBN 978-1-62410-565-4. DOI: https://doi.org/10.2514/4.105654

FRANCESCHETTI, B. – RONDELLI, V. – CIUFFOLI, A. 2019. Comparing the influence of Roll-Over Protective Structure type on tractor lateral stability. In Safety Science, vol. 115, pp. 42–50. DOI: https://doi.org/10.1016/j.ssci.2019.01.028

FUNFSCHILLING, C. – PERRIN, G. 2019. Uncertainty quantification in vehicle dynamics. In Vehicle System Dynamics, vol. 57, no. 7, pp. 1062–1086. DOI: https://doi.org/10.1080/00423114.2019.1601745

HAMDIA, K. M. – GHASEMI, H. 2023. Reliability analysis of the stress intensity factor using multilevel Monte Carlo methods. In Probabilistic Engineering Mechanics, vol. 74, article no. 103497. DOI: https://doi.org/10.1016/j.probengmech.2023.103497

ISO 5005:1977. Earth-moving machinery. Method for locating the centre of gravity.

MAJDAN, R. – ABRAHÁM, R. – TKÁČ, Z. – DRLIČKA, R. – MATEJKOVÁ, E. – KOLLÁROVÁ, K. – MAREČEK, J. 2021. Static lateral stability of tractor with rear wheel ballast weights: comparison of ISO 16231-2 (2015) with experimental data regarding tire deformation. In Applied Sciences, vol. 11, no. 1, article no. 381. DOI: https://doi.org/10.3390/app11010381

KLOSTERMAN, S. 2021. Data Science Projects with Python – A Case Study Approach to Gaining Valuable Insights from Real Data with Machine Learning. 2nd ed., Birmingham, UK : Packt Publishing Ltd., 404 pp. eISBN 978-1-80056-944-7.

MATHCADHELP. 2015. Regression Functions - Mathcad help. Available at: https://www.mathcadhelp.com/regression-functions-12505

MONTGOMERY, D. C. 2013. Design and Analysis of Experiments. 8th ed. Hoboken, NJ : John Wiley & Sons, Inc., 724 pp. ISBN 978-1-118-14692-7

PÁLEŠ, D. – BALKOVÁ, M. – KARANDUŠOVSKÁ, I. 2014. Calculation of failure probability of constantly loaded cantilever beam by Monte Carlo method. In Acta Technologica Agriculturae, vol. 17, no. 3, pp. 80–82. DOI: https://doi.org/10.2478/ata-2014-0018

PÁLEŠ, D. – VÁLIKOVÁ, V. – ANTL, J. – TÓTH, F. 2016. Approximation of vehicle trajectory with B-spline curve. In Acta Technologica Agriculturae, vol. 19, no. 1, pp. 1–5. DOI: https://doi.org/10.1515/ata-2016-0001

PTAK, M. – CZMOCHOWSKI, J. 2023. Using computer techniques for vibration damage estimation under stochastic loading using the Monte Carlo Method for aerospace applications. In Probabilistic Engineering Mechanics, vol. 72, article no. 103452. DOI: https://doi.org/10.1016/j.probengmech.2023.103452

PTC. 2023. Statistics of multivariate polynomial regression. Available at: https://support.ptc.com/help/mathcad/r9.0/en/index.html#page/PTC_Mathcad_Help/statistics_of_multivariate_polynomial_regression.html

PYTHON. 2023. Available at: https://www.spyder-ide.org/

RÉDL, J. – PÁLEŠ, D. 2018. Algorithm of determination of centre of gravity of agricultural machine with error estimation. In Mathematics in Education, Research and Applications (MERAA), vol. 3, no. 2, pp. 95–103.

ROSS, S. M. 2013. Simulation. 5th ed., Cambridge, Massachusetts : Academic Press, 328 pp. ISBN 0124158250

SABET, S. – POURSINA, M. 2017. Computed torque control of fully-actuated nondeterministic multibody systems. In Multibody System Dynamics, vol. 41, pp. 347–365. DOI: https://doi.org/10.1007/s11044-017-9577-4

SONG, C. – KAWAI, R. 2023. Monte Carlo and variance reduction methods for structural reliability analysis: A comprehensive review. In Probabilistic Engineering Mechanics, vol. 73, article no. 103479. DOI: https://doi.org/10.1016/j.probengmech.2023.103479

STEINHAUSER, M. O. 2013. Computer Simulation in Physics and Engineering. 1st ed. Berlin : De Gruyter. 529 pp. ISBN 9783110255904

STN 27 8154:1991. Earth-moving machinery. Method for locating the centre of gravity. (In Slovak).

WALKER, D. 2016. Computational Physics. Herndon, VA : Mercury Learning and Information, 372 pp. ISBN 1942270739