The development of a soil-movement measurement system to create more precise numeric soil models

Árpád Wágner[1] Bence Szabó[2] László Kovács[1] László Grad-Gyenge[1] Kornél Tamás[2]

[1] *Budapest University of Technology and Economics – Department of Automation and Applied Informatics, Magyar tudósok krt. 2., H-1117 Budapest, Hungary*
[2] *Budapest University of Technology and Economics – Department of Machine and Product Design, Műegyetem rkp. 3., H-1111 Budapest, Hungary*

***Abstract*—A well developed and industrialized agriculture system is one of the bases of our modern society, but it introduces a problem, as it also is a large contributor to climate change. Even only the use of unoptimized tools in itself has a multidimensional impact. Pulling such a tool through the soil requires more energy (and thus more fuel) than necessary. Because of its imperfect shape it will inevitably move more soil than needed. Furthermore during soil-tillage the amount of CO2 emissions and the water retention of the soil also depend on the movement of the soil. This last phenomenon also causes a decline in yield and that can increase the level of deforestation.**

By studying the aforementioned soil-tool interactions it is possible to help optimize the tools being used. But testing each revision of a developed tool is not only pricey but also time consuming. In situations like this modeling can come to the rescue. By utilizing the discrete element method (DEM) we can create an adequate numeric model of a given soil type. But calibrating such a model is non-trivial. By measuring the inner movement of the soil during test soil-tool interactions it is possible to adjust to the model until the modeled soil movement closely resembles the measured values, for this the creation of a tracker device was needed, which can be placed into the soil during the interactions and measure how much it moved during it.

In our paper we show how we developed such a device and how we used increasingly more techniques from basic data processing to AI to recreate the device's path. We also describe the development of the soil model and how we plan to utilize our findings to aid the design of better soil tilling tools in the future.

***Index Terms*—IoT, Discrete element method, Soil models, AI**

I. Introduction

As seen in the abstract, our grand goal is to create a measurement and modelling system which would offer a cost-effective way of determining the effectiveness of soil tools. The global aim is to create a numerical model which is accurate enough so that it can be used to test the efficiency in planning of soil tools. We determine the correctness of the model by its capability to recreate movements in the soil which we estimate from data collected during laboratory measurements. This project can be divided into multiple subtasks, detailed below.

- Creating an environment for measurements
- Creating a device which can be put into the soil during measurements and is capable of tracking its own movement during measurements (*DCD – data collection device*)
- Processing the data collected by the DCD to recreate the movement
- Creating a model of the soil
- Developing a method which can automatically calibrate the soil model based on the calculated movement data

In this paper we show the "evolution" of these tasks.

During our literature research we did not find a paper in which a sensor-based device ("active" tracker) would have been used to recreate the movements within the soil during soil-tool interactions, but we did find research papers where certain elements of such a system were studied. For example Dost et al. (2020) [1] used a sensor-based solution to approximate paths of "smart stones" during laboratory landslide experiments, a similar solution was also developed by Akeila et al. (2010) [2], who studied sediment transport in riverbeds. The usage of "passive" trackers is present in some trackers is present in some soil-tool interaction, such as Massah et al. (2020) [3] who put trackers into the top level of the soil and recorded the tillage process with a camera to determine the movement of the trackers during the interaction, while for example Rahman et al. (2005) [9] used multiple passive trackers in multiple layers within the soil and determined their displacement at the end of the tillage process. Milkevych et al. (2018) [4] did not only measure the displacement of passive trackers, but also modelled the soil-sweep interactions using discrete element method (DEM) simulations. There were several other researchers who also used DEM for modelling soil-tool interactions, such as [10], [11], [13], [14], [15] and [12].

II. Data collection device

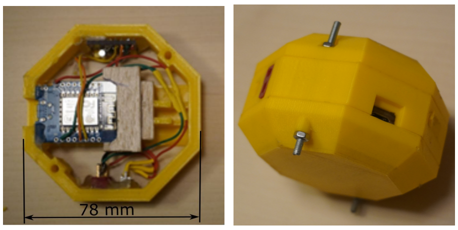
We set out to create a system where besides the usual metrics (like penetration test, force acting on soil-tool) it is also possible to measure the movement of the soil (or at least that of a known object in the soil). For this a device was needed that is able to collect data about its own movement in an environment where it is not possible to physically connect to the device or even to see it.

Based on these requirements the plan was to use accelerometers to gather information during the measurements. Sensors like this exist, but mainly as a part of so called Inertial

This paper was supported by the János Bolyai Research Scholarship of the Hungarian Academy of Sciences and by the ÚNKP Scholarship of Ministry of Human Resources

Measurement Units (IMU) where they are paired with a gyroscope and optionally a magnetometer (in this later case they can also be called MARG (Magnetic, Angular Rate, and Gravity) sensors). Sensors like this can be also used to calculate orientation by utilizing sensor fusion techniques. One of these techniques is the Madgwick filter published by Sebastian O.H. Madgwick [5]. It is capable of using all three possible kinds of sub-sensors a MARG sensor offers but it can also work from accelerometer and gyroscope only. This filter was used by multiple researchers, for example the already mentioned Dost et al. (2020) [1]

The first iteration of the DCD was made using almost exclusively off-the-self components, such as an ESP8266 development board and a Bosch BNO055 sensor breakout panel. Within the 3D printed housing these were connected by hand-soldered wires and kept in place by hot gluing them to their respective places.

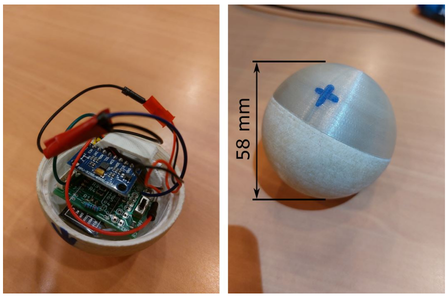


a. b.

Fig. 1. The middle section and the assembled housing of the old DCD

As seen on figure 1, the first version was quite larger than it needed to be. For example, note the padding at the microcontroller unit needed because the originally used unit was larger, furthermore much space was left unused. Another sub-optimal property of this version is the placement of the IMU. As seen on figure 1 it was fixed to a side of the housing, rather than in the middle of the DCD, as it should be.

One of the aims with the second version (see figure 2) was to reduce the size of the DCD. To achieve this, a PCB was designed to hold together the parts needed, making it possible to ditch the rather space-wasting development board. The BNO055 sensor was also replaced in favor of the less capable but faster MPU-9250, but its use still requires a breakout panel. During the design care was taken to place the IMU both in the middle of the board and the centre point of the new spherical housing. Said housing now also has one of its pars 3D printed using semi-transparent material, making it possible to have a color-coded LED-based status feedback.



a. b.

Fig. 2. The housing and the PCB of the new DCD

The summary of the differences between the two DCD versions can be found in table I

TABLE I
COMPARISON OF THE DCD VERSIONS

Property	**Older version**	**New version**	*Comment*
Housing size	78mm (inscribed circle)	58mm (diameter)	
Sensor type	BNO055	MPU-9250	
Orientation data source	Sensor's built-in microprocessor	Madgwick filter used during data processing	
Base MCU	ESP8266		
Communication	WiFi		
Control	Socket based	No control necessary	*Depends on software, so can be changed*
Data collection method	Collect a fixed amount of data when instructed then send back to controller.	Constantly publish measured data to a pre-specified MQTT topic.	*Depends on software, so can be changed*

III. MEASUREMENT ENVIRONMENT

We carried out two kinds of measurements. The first type is done in a larger laboratory soil bin, but is locked to the place of said bin, while the other uses a mobile soil bin but has some disadvantages due to the bin's smaller size.

A. *Laboratory soil bin test*

The first larger measurements were made in the soil bin of the NARIC Institute of Agricultural Engineering. This offers a great environment for the measurements, as the soil bin is large and there is specialized equipment to carry out measurements. As shown in figure 3/a, this laboratory is equipped with a 30 meters long and 2 meters wide soil bin and a guided measuring cart. Figures 3/b and c show how the DCD was buried in the ground.

a. b. c.

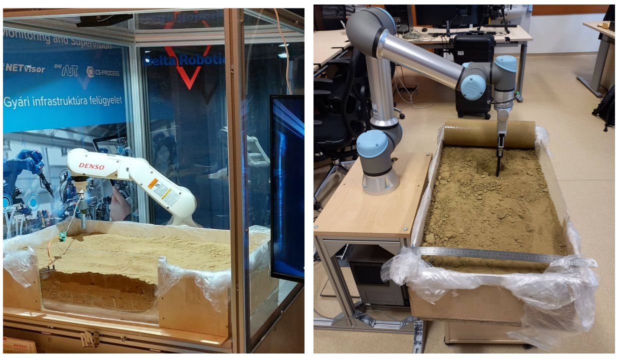
Fig. 3. Early measurements in the NARIC Institute of Agricultural Engineering laboratory soil bin

Multiple measurements were carried out in this setup with one (older style) DCD. During these measurements the DCD was controlled from a laptop, its displacement was determined by measuring its position relative to the soil bin before and after the tillage process as seen on figure 3. Later we determined the paths using the algorithmical method discussed in the IV section.

This environment offers a great way of examining the soil-tool interactions in a scenario that is as close to a real-life setup as possible whilst maintaining the repeatability expected from a laboratory setting. Furthermore it also offers the possibility of using the tool itself which is to be evaluated. On the other hand, this solution is not scalable, as for each measurement travelling and the notification of authorized personnel would be needed, thus another setup was needed.

B. Mobile soil bin

For the better manageability of the measurements a smaller (900×450×190 mm) soil bin was created. This made it possible to carry out "on site" measurements relatively freely and with a small amount of setup.



a. b.

Fig. 4. The soil bin in use during a measurement

This soil bin was filled with dry loamy sand soil. This solution is also called "mobile", as it is possible to relocate it, thus it was used in multiple laboratories using different robotic arms to move the 3D printed example soil tool, as seen on figure 4.

C. Mechanism

As it is later mentioned in section IV, amongst others Artificial Intelligence (AI) was also used as a data processing tool. For this to work properly, it is important to have a sufficient amount of training data, which can be used to teach the AI. An important requirement towards data like this is that both the input and expected output of the AI is needed to be known. For this, a closed-chain mechanism was created which can transform circular motion to a non-circular closed path.

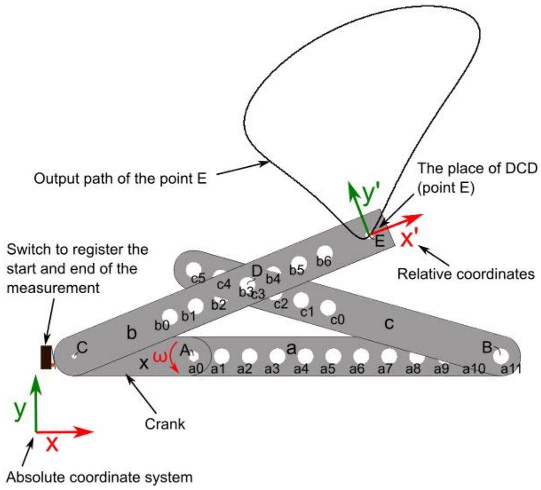


Fig. 5. The mechanism's plan

As seen in figure 5, the links of the mechanism contain multiple holes for joins. These can be used to set up, how the output path of the mechanism (meaning the path the DCD travels) will look like. It is also possible to tilt the board, onto which the mechanism was fixed, making it possible to take into account movements on all three spatial dimensions. Knowing the properties of the mechanism (length of the links, the selected holes) makes it possible to simulate the movement of the DCD when the crank is rotated once. This was simulated using Matlab and the data gathered from it is used as the expected output of the AI, see figure 6 (and of course the input data is what was measured by the DCD).

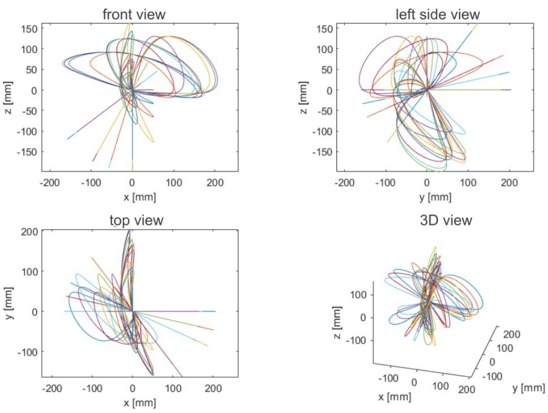


Fig. 6. Some simulated possible movement curves

IV. Data processing

The problem sounds simple. There are given acceleration data and the path they described is needed to be known. This could be done using double integration. The first problem is that the sensor's frame of reference differs from the global one. This can be done easily because IMUs were originally designed to be able to record their own frame of reference. The BNO055 can directly supply orientation data in quaternion format, for the MPU-9250 the usage of orientation filters is needed.

But a much more significant problem arises when calculating the path using double integration. It is known about MEMS acceleration sensors that they are noisy. Because of the integration this noise becomes even more significant, in fact the double integral causes an exponential error, making the results completely unusable.

This first led to the utilization of various data correction methods. First, the data needed to be trimmed to only contain the studied path itself then the acceleration data needed to be smoothened. Taking into account that the recorded data needed to go through various processing steps, a data pipeline focused application was developed. During the planning of this application it was important to make the addition of new processing steps easy, for this the interpreted nature of the Python programming language was utilized.

The data trimming remained a manual task, and the smoothening was done by various filters which could be selected by the user. But these efforts did not prove to be enough, as after the double integration the calculated path still did not resemble the original.

At this time it was noted that the acceleration data were shifted from its desired level. One could see this, because optimally in a closed path, the acceleration data on each axis should start and end at zero (the performance of the system was mainly tested using circles and semicircles), but it was not the case. Because of this a solution was implemented, where the user was able to shift the data by a constant value. This led to much better results, the mentioned circles and semicircles became distinguishable.

For automating this, extra information was needed, namely the endpoint of the traveled path. This was known to be possible to measure in the intended context of use, as there were already multiple researches where simple passive devices were used to record their endpoints during soil-tool interactions, for example Songül et al. [6] Using this extra information, a P-controller-like algorithm was developed. This algorithm iteratively shifts the collected values by a certain amount based on the difference between the intended and currently calculated endpoint.

The aforementioned solution was in use for a while but its precision could not match the desired level. Because of this, the usage of Artificial Intelligence was considered. First neural networks were used, but with those, the desired better accuracy was not achieved, but it was noted that one of the reasons for this can be the limited amount of training data, as the accuracy of these systems fluctuated when the models were retrained. This can happen because the initial (randomized) biases of the networks mattered too much in the end result, indicating an insufficient amount of training data.

By using a different type of AI, namely the gradient boosting technique with the XGBoost library, much better results were achieved. One of the great properties of this technique is the automatic feature selection, meaning that it can decide, which "columns" in the supplied data it deems to be important. Because of this, a large amount of data was supplied to these models, including (but not limited to) the raw sensor data, the data returned by the sensor fusion filters, time, a percentage metric showing, where within the current measurement the data record stands, and maybe even the results returned by the aforementioned "analytical" solution. Another difference was between "wide" and "thin" models. This means a difference in how the different types of data (meaning acceleration, gyroscope data, and other metrics), in case of the "wide" model the X, Y and Z axes are separated into different "columns" (where applicable), while with the "thin" model these data are only categorized by type. The rationale behind the "wide" model is that the sensor's output can be different depending on the axis (for example the measured acceleration on the Y axis tends to be more noisy) while in the case of the "thin" model a greater level of generalization can be expected. The differences between these models are shown on table II.

TABLE II
THE DIFFERENCE BETWEEN THE "WIDE" AND "THIN" MODELS

"Wide" model:

$\mathbf{acc_x}$	$\mathbf{acc_y}$	$\mathbf{acc_z}$	$\mathbf{gyr_x}$	$\mathbf{gyr_y}$	$\mathbf{gyr_z}$	...	**metric m**
acc_{x_1}	acc_{y_1}	acc_{z_1}	gyr_{x_1}	gyr_{y_1}	gyr_{z_1}	...	metric m_1
acc_{x_2} ...	acc_{y_2} ...	acc_{z_2} ...	gyr_{x_2} ...	gyr_{y_2} ...	gyr_{z_2} ...	... ...	metric m_2 ...
acc_{x_n}	acc_{y_n}	acc_{z_n}	gyr_{x_n}	gyr_{y_n}	gyr_{z_n}	...	metric m_n

"Thin" model:

acc	**gyr**	...	**metric m-1**	**metric m**
acc_{x_1}	gyr_{x_1}	...	metric m-1_1	metric m_1
acc_{x_2} ... acc_{x_n}	gyr_{x_2} ... gyr_{x_n}	... ...	metric m-1_2 ... metric m-1_n	metric m_2 ... metric m_n
acc_{y_1}	gyr_{y_1}	...	metric m-1_1	metric m_1
acc_{y_2} ... acc_{y_n}	gyr_{y_2} ... gyr_{y_n}	... ...	metric m-1_2 ... metric m-1_n	metric m_2 ... metric m_n
acc_{z_1}	gyr_{z_1}	...	metric m-1_1	metric m_1
acc_{z_2} ... acc_{z_n}	gyr_{z_2} ... gyr_{z_n}	... ...	metric m-1_2 ... metric m-1_n	metric m_2 ... metric m_n

V. Results

A. Results of the laboratory soil bin measurements

The data gathered from the measurements mentioned in the III-A subsection was later evaluated with the usage of multiple data correction methods and the algorithmical solution mentioned in the IV section. In this case measuring the overall displacement of the DCD is absolutely necessary as the mentioned algorithm is a vital part of the data processing.

A large amount of user interaction is also needed.
This resulted in path curves (figure 7) that closely resemble the expected output, but since the correctness of the output depends on a large amount of variables (including the user's capabilities), a method for evaluating said correctness was not defined.

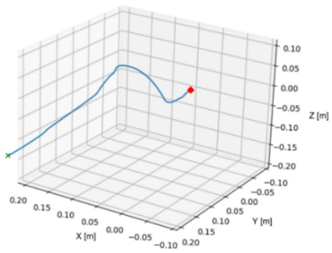
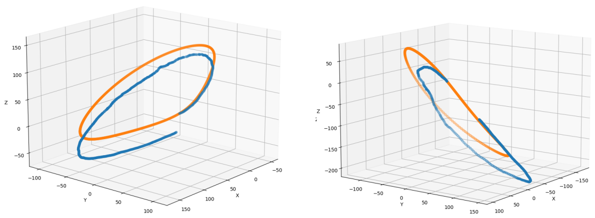


Fig. 7. A result from the algoritmical method

B. Results of the mechanism and AI

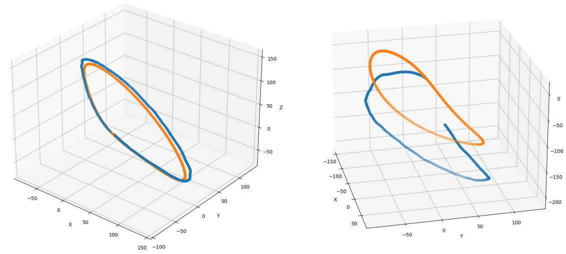
As mentioned in the subsection III-C a mechanism was used to create movement data and a Matlab simulation was used to create the paths belonging to the movement data. This was used as the input for the AI mentioned in section IV.
Regarding the results of the AI it was observed that the wide model worked overall much better than the thin model. Figure 8 shows the performance of the model when the algorithmical solution was not used as an input for the AI, in the **a.** case the prediction was done on training data while in the **b.** case it was done on data similar to the training set but not seen before by the AI.



a. b.

Fig. 8. Performance without algorithmical solution (sizes in mm, orange is the expected values are shown in orange and calculated ones in blue)

In another case the output of the algorithmical solution was also used as an input for the AI. As seen in figure 9/a, it can greatly enhance the AI's performance in case of predictions made on training data but falls behind when used on not seen but similar data (figure 9/b).



a. b.

Fig. 9. Performance with algorithmical solution (sizes in mm, orange is the expected values are shown in orange and calculated ones in blue)

The results show that the AI-based solution can be viable even if the overall displacement needed for the algorithmical solution was not measured, however its existence can enhance the precision of the model and further training of the AI is needed to get better quality results.

VI. Soil model

The numerical soil-tool interaction model is prepare in the YADE software [7]. The DEM model follows the structure of laboratory measurements. The geometry of the soil bin, the simple soil tool, and the tracker in the soil, and the tool speed are the same as during the laboratory measurements. The DEM soil model can be calibrated based on laboratory measurements specified macro-mechanical parameters (in this case average force acting on tool, and the path of the tracer) of soil. However these parameters are not able to directly characterize the DEM model, because the DEM model can only be parameterised by not measurable micro-mechanical relations between two grains. For this purpose the genetic algorithm is applies to find the appropriate micro-mechanical parameters [8], which are able to model the tested soil. It is also under testing what kind of simulation and micro-mechanical parameters have an impact for the path of the modelled tracer.

VII. Conclusion

The main novelty value of our research is a measurement system that can enable us to analyze the inner movements of granular materials. The results from the measurement system can serve as a basis for the calibration of discrete element method models of soil. During our research a laboratory environment for the study of soil-tool interactions was created, example soil tools were planned and manufactured. Artificial intelligence was also used to determine the paths taken by the DCDs during such interactions, but further measurements and developments are needed to enhance the performance of the method.

References

[1] Dost, J. B., Gronz, O., Casper, M. C., Krein, A. (2020). The potential of Smartstone probes in landslide experiments: how to read motion data. Natural Hazards and Earth System Sciences, 20(12), 3501-3519.
[2] Akeila, E., Salcic, Z., Swain, A. (2010). Smart pebble for monitoring riverbed sediment transport. IEEE Sensors Journal, 10(11), 1705-1717.

[3] Massah, J., Etezadi, H., Azadegan, B., Hassan-Beygi, S. R. (2020). Modelling of Soil Displacement Resulting from Sweep during Tillage Operation Using Image Processing. Journal of Agricultural Science and Technology, 22(2), 415-424.
[4] Milkevych, V., Munkholm, L. J., Chen, Y., Nyord, T. (2018). Modelling approach for soil displacement in tillage using discrete element method. Soil and Tillage Research, 183, 60-71.
[5] Madgwick, S. O., Harrison, A. J., Vaidyanathan, R. (2011, June). Estimation of IMU and MARG orientation using a gradient descent algorithm. In 2011 IEEE international conference on rehabilitation robotics (pp. 1-7). IEEE.
[6] Gürsoy, S., Chen, Y., Li, B. (2017). Measurement and modelling of soil displacement from sweeps with different cutting widths. Biosystems engineering, 161, 1-13.
[7] Smilauer, V., Angelidakis, V., Catalano, E., Caulk, R., Chareyre, B., Chevremont, W., ... Yuan, C. (2023). Yade documentation. arXiv preprint arXiv:2301.00611.
[8] Do, H. Q., Aragón, A. M., Schott, D. L. (2018). A calibration framework for discrete element model parameters using genetic algorithms. Advanced Powder Technology, 29(6), 1393-1403.
[9] Rahman, S., Chen, Y., Lobb, D. (2005). Soil movement resulting from sweep type liquid manure injection tools. Biosystems Engineering, 91(3), 379-392.
[10] Tamas, K., Bernon, L. (2021). Role of particle shape and plant roots in the discrete element model of soil–sweep interaction. Biosystems Engineering, 211, 77-96.
[11] Tamás, K. (2018). The role of bond and damping in the discrete element model of soil-sweep interaction. Biosystems engineering, 169, 57-70.
[12] Ucgul, M., Saunders, C., Fielke, J. M. (2018). Comparison of the discrete element and finite element methods to model the interaction of soil and tool cutting edge. Biosystems Engineering, 169, 199-208.
[13] Keppler, I., Hudoba, Z., Oldal, I., Csatar, A., Fenyvesi, L. (2015). Discrete element modeling of vibrating tillage tools. Engineering Computations.
[14] Zachár, A., Keppler, I., Oldal, I. (2016). Investigation of the Applicability and efficiency of Different Mathematical Modeling and Numerical Simulation Methods for Soil-Tool Interaction. Journal of Computational Applied Mechanics, 11(1).
[15] Barr, J. B., Ucgul, M., Desbiolles, J. M., Fielke, J. M. (2018). Simulating the effect of rake angle on narrow opener performance with the discrete element method. Biosystems Engineering, 171, 1-15.