Deep learning based simulation of jack-up rig

Yu Yu Metreveli
Admiral Ushakov Maritime State University, Novorossiysk, Lenin Avenue 93, 353924, Russia
E-mail pim1289@yandex.ru

Abstract. The article illustrates the methodology of jack-up install simulation with the different compound of soils, including spudcan penetration. The core of this method is the combination of the finite element and data-driven approaches: the first - slow, but precise, and the second, based on machine learning proposed the fast solution with results, closing to the first approach.

Keywords: Finite element method; machine learning; neural networks; model reduction; data-driven simulation; spudcans; jack-up rig; offshore geotechnics; penetration resistance profile; punch-through failure; bearing capacity

1. Introduction

The jack-up vessels are widely used for the offshore works like: geological survey, oil and energy industries. Typical jack-up rig is a barge supported and lifted by three or four truss legs with a large spudcan on the end of each.

Jack-ups can be installed on a site independently, without external assistance. For the reaching of sufficient stability from the seabed in any conditions during installation, the ballast tanks are filled with seawater, this process is called preloading [1].

When the jack-up completed its work, the legs are retracted from the seabed, leaving a craters called ‘footprints’ at the site. Jack-ups often return to sites where previous operations have left footprint in the seabed [2].

In the process of jack-up vessel operations it faces a heavy environmental conditions: the axial loads due to the weight of the rig, the lateral and cyclic loads due to continuous waves and wind.

The most dangerous job is the jacking process, especially the sea bottom penetration part, that includes major risks, and therefore, should be analyzed and conducted, carefully [3].

The installation of a jack-up rig is a long time process, so the expand the jack-up operator’s real experience much time is needed. The jack-up rig simulators provide much faster training for operators, and also can reproduce a wide range of events during jack-up works.

The bearing capacity of spudcan foundations for offshore mobile jack-up rigs is currently assessed using methods recommended by ISO [4] and SNAME [5], these recommendations for the punch-through failure are the punching shear and projected area method. In the two methods, the peak resistance is calculated at a specified penetration depth by assuming an initial embedment of spudcan and undisturbed soil strengths. The finite element method (FEM) has been introduced to the punch-through failure analysis, especially the large deformation finite element (LDFE) method in recent years, such as the Remeshing and Interpolating Technique with Small Strain (RITSS), Arbitrary Lagrangian-Eulerian (ALE), and Coupled Eulerian-Lagrangian (CEL) [6].

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
PUBLISHED UNDER LICENCE BY IOP PUBLISHING LTD
The FEM are slow and need to setup the basic parameters of the simulation, these methods cannot be used for interactive simulations as needed for jack-up operator training, but the linear approximations of those can be used for interactive simulation.

The progress of data science allows for the application of data-driven approaches in the field of mathematical and computer modeling. Methods of big data, machine learning, neural networks, are used in science, engineering, chemistry, and pharmacy, etc. [7].

The application of such methods are well known in the simulation of transient processes [8], the simulation of turbulent and laminar flows of liquids and gases [9], in particular for solving the Navier-Stokes equations [10].

As a result of the application of data-driven methods neural networks that implements the necessary physical laws allow the fast and accurate simulation to be performed, which is suitable for training tasks. However, such methods have a natural flaw: the initial data are needed for work. A workaround for this disadvantage is the use of mixed methods, combining traditional modeling (numerical or analytical) and data-driven [11]. The result of this approach will be a fast simulation that is close in accuracy to traditional simulation.

In this article, we will look at a combination of FEM and machine learning for simulating jack-up rig installation and spudcan penetration.

2. Materials and methods

2.1. Spudcan penetration finite element simulation

Spudcan foundations are generally under combined vertical (V), horizontal (H) and moment (M) loads due to the environmental loading (e.g., wind, waves, and currents). Figure 1 shows a platform during installation. The uncertainties and the complex load conditions make the bearing capacity estimation very difficult [12].

The average load resistance profile of a spudcan penetrating in the sand over the clay is shown in figure 2b. The seabed layout is shown in figure 2a. The resistance profile has a peak resistance, marked $q_{peak}$, the depth of penetration on the peak resistance is named $d_{peak}$ in the range of 12-20% of the
upper sand layer thickness. The next point becomes a sharply decreasing of resistance, named post-peak resistance or the $q_{post-peak}$, with the penetration of $d_{post-peak}$. The $q_{peak}$ means the case with the risk of punch-through failure, and the $q_{post-peak}$ will be the post-failure state of spudcan foundation. The difference between $q_{peak}$ and $q_{post-peak}$ called the $q_{diff}$ shows the rank of punch-through failure danger. The punch-through of installed and loaded jack-up ship leads to creasing leg or even overturn the ship [6].

![Diagram](image)

Figure 2. Punch-through failure: (a) Schematic diagram of embedded spudcan; (b) Typical punch-through profile;

The Coupled Eulerian-Lagrangian (CEL) method from the Abaqus software was used to model the penetration of the spudcan into sea bottom soils. The Coupled Eulerian Lagrangian (CEL) method from the Abaqus software was used to model the penetration of the spudcan into sea bottom soils. To create the penetration model in Abaqus is needed to: make the spudcan and surrounding soils geometry, generate the mesh of finite elements, set the boundaries, the loads, and the displacements.

The simulation of the soil layers made from the different density clays is using the Mohr-Coulomb model of plastic fracture. Spudcans are modeled as solid materials, i.e. only Spudcan geometry and legs are involved in the analysis.

The "penalty contact method" including coefficient of friction is used for the algorithm of contact. There are important to set the model geometry, resolution of mesh, and speed of spudcan penetration.

An example of visualizing the result of FEM in the form of penetration depth from bearing capacity is presented in figure 3.
2.2. Subspace and data-driven simulation

The data-driven methods are based on precomputed data, that are typically accurate and computed not in real-time. The simulation of non-linear deformation without model reduction through a neural network is described in [13]. The data-driven methods are also used to simulate fluids [14-15]. Using spatially variable pressure data, a regression forest can predict the behavior of fluid particles [16-17]. This model predicts the dynamics of the individual particle in the mass of the particles, so the performance general requirement is the high level of parallelism, that achieves using GPU. The use of convolutional networks for selective acceleration allows solving incompressible Euler equations for fluid simulation [18].

The main problem of data-driven methods is keeping an optimal combination of performance, resource usage, accuracy, and model adequacy. The method, proposed in this article is close to the specified balance.

The method, illustrated in figure 4 includes four stages. The first stage is to collect accurate and quality FEM simulation data. The second stage is the use of machine learning with a neural network and a novel training method. After this result model is integrated into the jack-up rig simulator.

Principal component analysis (PCA) is one of the main ways to reduce the dimension of data while losing the least amount of information. From a mathematical point of view, this method is an orthogonal linear transformation that maps data from the original feature space to a new space of lower dimensions. A PCA basis is usable for the work with a representative set of deformation examples.

2.3. Training data
An advantage of this method is the compatibility with almost any simulation method for acquiring the data, as the only input to our training procedure is a raw time-series of frame-by-frame vertex positions. In this example, data are taken from a FEM soft as the frames from animation, with the 10⁶ frames for the best result.

2.4. Training

With the gathered data is possible to construct the training dataset, we collect the coordinates of the vertices in each frame \( t \) into one vector \( x_t \in \mathbb{R}^{3c} \), and then we combine these frame-by-frame vertices into one large matrix \( X = [x_0, x_1, \ldots, x_{n-1}] \in \mathbb{R}^{3cxn} \), that describes the states of the modeled object. Also, the states of the external objects in the frame must be taken in count.

In result, there is one large vector for each object, representing the state of the external object \( y_t \in \mathbb{R}^e \), where \( e \) is the number of degrees of freedom. All these vectors are also concatenated into the large matrix \( Y = [y_0, y_1, \ldots, y_{n-1}] \in \mathbb{R}^{exn} \).

After applying a PCA to \( X \) and \( Y \), the subspace representations \( Z \) and \( W \) are constructed to generate subspace a image. \( Z = U(X - \bar{x}_t) \), \( W = V(Y - \bar{y}_t) \), where \( U \in \mathbb{R}^{u \times 3c} \), \( V \in \mathbb{R}^{v \times e} \), \( u \) is the number of subspace bases, \( v \) is the number of bases used to compress the external object representation, \( \bar{x}_t \) and \( \bar{y}_t \) is the mean values. If the PCA procedure requires too much memory, the data must be sampled first. Compression using PCA leads to loss of detail, but the high number (256) of subspace bases is usually enough to preserve most detail.

The development of the model that could predict the vectors states of the models in future frames: \( z_t \) from \( z_{t-1}, z_{t-2} \) and \( w_t \). The simulated objects are usually characterized by inertia with a trend to some average rest state, and the model would be presented as:

\[
\bar{z}_t = \alpha \odot z_{t-1} + \beta \odot (z_{t-1} - z_{t-2}), \tag{1}
\]

The \( \alpha \) and \( \beta \) is the model parameters and \( \odot \) is a component-wise product. This parameters could be found by solving solving the least-squares equation individually for each dimension of \( \alpha \) and \( \beta \):

\[
[\alpha_m \beta_m] = \begin{bmatrix} z_{t,m} & z_{t-1,m} - z_{t-2,m} \end{bmatrix}^\dagger, \tag{2}
\]

where \( \dagger \) denotes the matrix pseudo-inverse.

\( \bar{z}_t \) is a very rough approximation of \( z_t \), without affect from external objects \( w \), and cannot be enough precise model for the training. So the neural network \( \Phi \) should be trained to approximate extra effects of the model:

\[
z_t = \bar{z}_t + \Phi([\bar{z}_t \ z_{t-1} \ w_t]) \tag{3}
\]

The neural network \( \Phi \) is the standard feed-forward, with ten layers and ReLU[19] activation function (except the last layer because it is the output layer). For each layer between input and output is set the

![Figure 4. Schematic of the method: X and Y are training data from FEM software, the Z and W are compressed representations after PCA. The training gives a neural network \( \Phi \) that predicts the compressed state of the object \( z^* \), based on the previous state of object \( z_{t-1} \), and the external objects \( w^* \). The object positions x are computed directly from the model output.](image-url)
number of hidden units equals $1.5 \times$ size of PCA data, which gives a balance for the disk space and the performance.

Training a neural network through iterating over a dataset in order to obtain frame-wise predictions is not a good approach, due to unstable feedback. An alternative is the back propagation method used throughout the entire integration procedure (3), which provides stable long-term predictions.

The training procedure for one training sample is described below, where $\theta$ are network parameters, updatable by using the error, estimated for each step of object subspace state prediction:

$$r_0, r_1 \sim \mathcal{N}(0, r^0)$$  \hspace{1cm} (4)

$$z_0^*, z_1^* \leftarrow z_0 + r_0, z_1 + r_1$$  \hspace{1cm} (5)

For $i \leftarrow 2 \ldots s$ do

$$z_i^* = \alpha \odot z_{i-1}^* + \beta \odot (z_{i-1}^* - z_{i-2}^*)$$  \hspace{1cm} (6)

$L$ is the loss, and computed using mean absolute error:

$$L_{\text{pos}} \leftarrow \|z_{2 \rightarrow s} - z_{2 \rightarrow s}\|_1$$  \hspace{1cm} (7)

$$L_{\text{vel}} \leftarrow \|z_{2 \rightarrow s}z_{1 \rightarrow (s-1)} - z_{2 \rightarrow s}z_{1 \rightarrow (s-1)}\|_1$$  \hspace{1cm} (8)

$$L \leftarrow L_{\text{pos}} + L_{\text{vel}}$$  \hspace{1cm} (9)

Updating network parameters using AmsGrad:

$$\theta \leftarrow \text{AmsGrad}(\theta, \nabla L)$$  \hspace{1cm} (10)

The training procedure works like this: from the small sample of training data $z$ and $w$ took the first two frames $z_0$ and $z_1$, then add a bit of noise $r_0$ and $r_1$ to offset the training trajectory. Then, to predict the next frames the algorithm (equation 6) is running several times, using previous prediction results at each new step. As the full trajectory is predicted the average coordinate and velocity errors are computed, and then passed to AmsGrad [20] optimizer to update the network parameters $\theta$.

Using small training data sets forces the network to make corrections only within the sample window, thereby making the corrections small for all training data. This approach, presented in Figure 5, gives more stable corrections, avoiding overcorrection (typical for training on the whole data).

Repeating this algorithm on 16 frames from 32 frame sample, for 100 epochs or until training converges. The learning rate is 0.0001, learning rate attenuation of 0.999, and a standard noise deviation calculated from the first three components of PCA space. Training requires 48-96 hours, based on the complexity of the parameters and the PCA data size.

Figure 5. The training method illustration. On the left is the standard procedure, that gives an accurate result per frame, the predictions are unstable and give the overcorrection. On the right are proposed method, that targets an accurate prediction per frame, with the moderate corrections

3. Interactive application

The results of this method are visualized in our jack up rig simulator in the figure 6. This interactive application is written on C# with WPF, 3D render used DirectX. Pre-processing and neural network operations in single-threaded C++ code and load the binary network weights obtained during our training procedure.
4. Discussion
The proposed mixed jack-up rig simulation method can be based not only on FEM but also on other modeling methods, as well as on pre-processed data from real rigs. Also, the resulting neural network can be adapted for calculating the jack-up loads and predicting its behavior in real-time during the installation process.

5. Conclusion
The mixture of two modeling approaches: a slow, but precise and the fast, but dependent on data is the novelty and the solution for domains, that used complex models and needs the fast results, like training simulations, predictions, etc.

References
[1] Kim Y H, Hossain M S, Edwards D and Wong P C 2019 Penetration response of spudcans in layered sands Applied Ocean Research 82 236–244 https://doi.org/10.1016/j.apor.2018.11.008
[2] Hossain M S, Stainforth R, Ngo V T, Cassidy M J, Kim Y H and Jun M J 2017 Experimental investigation on the effect of spudcan shape on spudcan-footprint interaction Applied Ocean Research 69 65–75 https://doi.org/https://doi.org/10.1016/j.apor.2017.10.003
[3] Emren V 2015 Three dimensional finite element modeling for the spudcan penetration into clayey seabed.
[4] ISO 19905-1 2016 Petroleum and natural gas industries. Site-specific assessment of mobile offshore units. Jack-ups (ISO 19905-1:2016) (London: BSI standards publication) p 334
[5] SNAME 2008 Guidelines for site specific assessment of mobile jack-up units. T&R Bulletin 5-5A (New Jersey: The Society of Naval Architects and Marine Engeneers) p 366
[6] Zhao J, Jang B-S, Duan M and Liang C 2019 A finite element approach for predicting the full resistance profile of a spudcan deeply penetrating in dense sand overlying clay Applied Ocean Research 87 155–164 https://doi.org/https://doi.org/10.1016/j.apor.2019.03.026
[7] Montánis F J, Chinesta F, Gómez-Bombarelli R and Kutz J N 2019 Data-driven modeling and learning in science and engineering Comptes Rendus Mécanique 347(11) 845–855 https://doi.org/https://doi.org/10.1016/j.crme.2019.11.009
[8] Zhang N, Yuan J and Ng S H 2019 A data-driven online truncation method for transient bias reduction in steady-state simulations Computers & Industrial Engineering 135 723–745 https://doi.org/https://doi.org/10.1016/j.cie.2019.06.034
[9] Chen Q, Wang Y, Wang H and Yang X 2021 Data-driven simulation in fluids animation: A survey Virtual Reality & Intelligent Hardware B 87–104 https://doi.org/https://doi.org/10.1016/j.vrih.2021.02.002
[10] Li Z, Hoagg J B, Martin A and Bailey S C C 2018 Retrospective cost adaptive Reynolds-averaged Navier–Stokes k–ω model for data-driven unsteady turbulent simulations Journal of Computational Physics 357 353–374 https://doi.org/https://doi.org/10.1016/j.jcp.2017.11.037
[11] Holden D, Duong B C, Datta S and Nowrouzezahrai D 2019 Subspace Neural Physics: Fast Data-Driven Interactive Simulation. In Proceedings of the 18th Annual ACM SIGGRAPH/Eurographics Symposium on Computer Animation. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/3309486.3340245

[12] Li L, Li J, Huang J and Gao F-P 2017 Bearing capacity of spudcan foundations in a spatially varying clayey seabed Ocean Engineering 143 97–105 https://doi.org/https://doi.org/10.1016/j.oceaneng.2017.05.026

[13] Luo R, Shao T, Wang H, Xu W, Chen X, Zhou K and Yang Y 2018 NNWarp: Neural Network-Based Nonlinear Deformation IEEE Transactions on Visualization and Computer Graphics 26(4) 1745–1759 https://doi.org/10.1109/TVCG.2018.2881451

[14] Kim B, Azevedo V C, Thuerey N, Kim T, Gross M and Solenthaler B 2019 Deep Fluids: A Generative Network for Parameterized Fluid Simulations Computer Graphics Forum 38(2) 59–70 https://doi.org/10.1111/cgf.13619

[15] Wiewel S, Becher M and Thuerey N 2019 Latent-space Physics: Towards Learning the Temporal Evolution of Fluid Flow Computer Graphics Forum 38 71-82

[16] Ladicky L, Jeong S, Bartolovic N, Pollefeys M and Gross M 2017 Physicsforests: Real-Time Fluid Simulation Using Machine Learning ACM SIGGRAPH 2017 Real Time Live! (New York: Association for Computing Machinery) p 22 https://doi.org/10.1145/3098333.3098337

[17] Ladický L, Jeong S, Solenthaler B, Pollefeys M and Gross M 2015 Data-Driven Fluid Simulations Using Regression Forests ACM Trans. Graph. 34(6) https://doi.org/10.1145/2816795.2818129

[18] Tompson J, Schlachter K, Sprechmann P and Perlin K 2017 Accelerating Eulerian Fluid Simulation With Convolutional Networks. In D. Precup & Y. W. Teh (Eds.), Proc. 34th International Conference on Machine Learning ed Precup D and The Y W (jmlr.org) vol 70 3424–3433

[19] Nair V and Hinton G E 2010 Rectified Linear Units Improve Restricted Boltzmann Machines. Proc. 27th International Conference on International Conference on Machine Learning (Madison: Omnipress) 807–814

[20] Reddi S J, Kale S and Kumar S 2018 On the Convergence of Adam and Beyond Proc. 6th International Conference on Learning Representations (Vancouver: OpenReview.net)