Mixing neural networks and the Newton method for the kinematics of simple cable-driven parallel robots with sagging cables

Ichrak Yahia, Jean-Pierre Merlet, Yves Papegay

Mixing neural networks and the Newton method for the kinematics of simple cable-driven parallel robots with sagging cables

Ichrak Ben Yahia[1] and Jean-Pierre Merlet[1] and Yves Papegay[1]

***Abstract*— Cable-driven parallel robots (CDPR) use cables to move a platform. These cables can be coiled/uncoiled by winches and are all attached to the platform. We are considering here a specific class of CDPR, called N-1 which has N cables that are all attached to the platform at the same point B. With 2 cables we get a planar 2-dof robot while with $N \geq 3$ we get a robot with 3 translational dof. We assume that the cables have elasticity and a mass so that sagging exists. In that case the inverse and direct kinematics (IK, DK) are difficult to solve. As kinematics plays an important role for the robot analysis it is necessary to design fast but exact kinematics procedures. In this paper we consider the 2-1 and 3-1 cases which have a single solution for the kinematic and addresses the use of neural network (NN) to solve the IK and DK. We show that NN provide very approximate results but also that it is possible to design a solving strategy mixing NN and the Newton method to get the exact result in a low computation time. We cannot formally prove that this approach will always work but extensive numerical tests have shown no failure.**

I. INTRODUCTION

Moving a platform over a large workspace, possibly with an heavy load is of interest for many applications such as transfer for elderly people, rehabilitation, maintenance and logistics. Such a robot may be designed with a parallel architecture but one which uses cables instead of rigid legs with linear actuators. Cables offer the advantages of allowing large changes in the leg lengths (thereby providing a large workspace) while retaining the large lifting capacity of parallel robots. In this paper we will consider the so-called $N-1$ cable-driven parallel robot (CDPR) where the N cables are attached at the same point B of the platform allowing to get with 2 cables a 2dof translational planar robot and with $N \geq 3$ a 3dof translational spatial robot. In this paper we will consider the case where N is either 2 or 3.

Still cables have the drawback that they are not rigid, may be elastic and deformable because of the action of their own mass. Such a behavior must be taken into account for the kinematics, which implies to have a cable model. We are especially considering the analysis of a CDPR for design purpose. This implies for example to determine several key-indicators (KI) such as the maximal cable tensions and worst case accuracy over a given workspace. Computing these KIs requires to solve a large number of inverse kinematics (IK, determining the cable lengths to reach a given pose) and direct kinematics (DK, finding the platform pose being given the cable lengths). Hence fast procedures to solve the IK and DK is an essential element for a fast analysis of a CDPR. Furthermore the accuracy of the solution is very important as even a small error may lead to drastic changes on the KIs. If we assume no elasticity and no mass for the cables, then the kinematics of CDPR is identical to the one of parallel robots with rigid legs. Neural network (NN) have been proposed for dealing with this case [1], [2], [3], [4], [5], [6], [7] but with errors that are way too large for a design process. Furthermore the DK has usually several solutions while (NN) are able to approximate only one of them and for CDPR IK and DK are no more a pure geometrical problems but a statico-geometrical one as cable tensions modify the cable shape. As soon as a realistic cable model is used solving the IK and DK of CDPR is much more complex than for parallel robot with rigid legs. Numerical algorithms haven been proposed to find all solutions of the IK and DK [8], [9], [10] but they are computer intensive (several hours of calculation time). Note that as a full CDPR analysis may require millions of IK and DK solving it makes sense to look for a solving approach based on NN as the time overhead of this approach (namely the NN training) will be a small price to pay compared to the time gain for the analysis. A few works have addressed the use of NN for the kinematics of CDPR with simple cable model [11], [12], [13] but the results are very different from the one that will be obtained with real cables. Only recently has been presented a work for a planar CDPR with a realistic cable model [14]. But two major issues have not been addressed in this work:

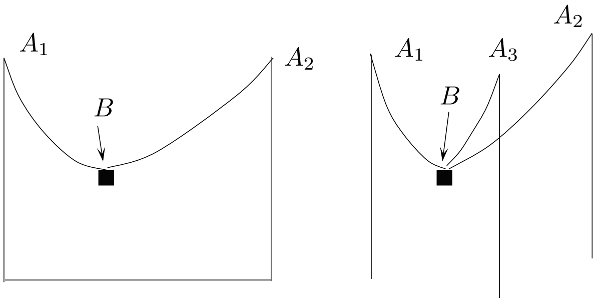

Fig. 1. The 2-1 and 3-1 CDPR

- as the accuracy of the NN prediction is not sufficient for being used in the analysis process can we design a solving strategy based on NN and other means to provide accurate solution(s) ?

*This work has been partly supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002 and by the ANR CRAFT project number ANR-18-CE10-0004-03

[1] All authors are with INRIA Sophia-Antipolis, Hephaistos project-team, France Jean-Pierre.Merlet@inria.fr

- both the IK and DK may have multiple solutions, that have all to be determined

This paper addresses the first issue for CDPR that have a single solution both for the IK and DK, so that the second issue is not relevant, with the purpose of proposing a strategy mixing NN and the Newton method for finding exactly the kinematic solution. It must be mentioned that for both the IK and DK choosing random initial guess does not usually leads to the convergence of the Newton scheme (a non convergence will be denoted as a *failure* of the method and is obtained when the number of Newton iteration is larger than a fixed threshold).

Note that unless otherwise specified length and coordinates units are meters, mass in kg and forces in Newton.

II. Neural networks

As previous works on the subject we will use as neural network model the Multi Layer Perceptron [15] as several platforms propose tools that are designed to build and use such a NN. NN training have been performed in Python using `TensorFlow` and `PyTorch`. Predictions are also performed using this language although a C++ implementation will be faster. Training sets are provided as a cvs file while calculating the error metrics on the verification has been performed in `Maple`. We have the advantage of having the possibility to objectively assess the quality of the result with several metrics (such as mean and maximal errors) that compare it to the exact result. NN have a multitude of parameters whose influence on the quality of the result is unclear. In our case however the training time is low (between 1 and 40 minutes) so that we have been able to explore all combinations for a large number of possible parameters. Preliminary tests have shown that the number of neurons has a low influence on the error metrics as soon as it is large enough so that we set it to 200. Therefore we have considered all NNs with 1 to 6 layers connected by all possible combinations of activation functions among a set of 7 possible functions, leading to the use of 8659 NNs on a cluster. Clearly we cannot present all the results for lack of space and only the most efficient NNs will be mentioned.

III. CABLE MODEL

A cable model that is commonly used for CDPR and has been experimentally verified is a planar model. A reference frame is defined with the axis z being the local vertical. In this frame the extremities of the cable are A, the winch output point with coordinates (0,0) and B with coordinates $x_b > 0, z_b < 0$. The force exerted on the cable at B has as components F_x, F_z. The variables x_b, z_b, F_x, F_z are related by [16]:

$$x_b = F_x(\frac{L_0}{EA_0} + \frac{sinh^{-1}(F_z) - sinh^{-1}((F_z - \frac{\mu g L_0}{F_x})}{\mu g}) \quad (1)$$

$$z_b = \frac{\sqrt{F_x^2 + F_z^2} - \sqrt{F_x^2 + (F_z - \mu g L_0)^2}}{\mu g} + \frac{F_z L_0}{EA_0} - \frac{\mu g L_0^2}{2EA_0} \quad (2)$$

where L_0 is the cable length at rest, E is the Young modulus of the cable material and μ its linear density. In most of our example we will use as cable a Dyneema cable of diameter 4 mm, $E = 100e^9$ N/m^2 and a linear density $\mu = 0.079$ kg/m and all cables are assumed to be identical.

In the sequel of this paper A_i will denote the winch output point of cable i which is supposed to be fixed and whose position is known in the reference frame, B will be the common point of the cable on the platform, L_{0_i} will be the cable length at rest, F_{x_i}, F_{z_i} the horizontal and vertical components of the force exerted on the cable at B.

IV. Maximal number of IK and DK solution

We claim that for both the 2-1 and 3-1 CDPR there is always at most a single solution for both the IK and DK. First let us note that the cable model presented in section III is such that if $E \to \infty$ and $\mu \to 0$, then equations (1,2) are identical to the one obtained for rigid legs. This continuity has been used to provide a solving algorithm for the IK and DK of CDPR [9], [10]. The idea is to start from the solutions of the robot with the same geometry but with N rigid legs but allowing to assume that from 0 to $N-1$ legs have 0 tension. For each of this solution we use a continuation process based on the Newton scheme. We start the process with a very large E and a very small μ and then we move incrementally E, μ toward their real values. Hence the number of solutions for the CDPR cannot exceed the number of solutions determined for the rigid case (e.g. a CDPR with 8 cables cannot have more than 33488 solutions for the DK while a CDPR with 6 cables cannot have more than 63 IK solutions). We will also see later on that a CDPR cannot reach a pose that is exactly below one of the A_i point for the 2-1 and in the vertical plane that include two A_i points. Under that restriction the 2-1 and 3-1 with rigid legs have a single IK and DK solution and therefore the considered CDPR have also a single solution.

V. Kinematics of the 2-1 CDPR

A. Inverse kinematics

For that problem the position of B is fixed and we have to determine the two cable lengths to reach it. But if sagging is taken into account we have to consider as additional unknowns $F_{x_1}, F_{x_2}, F_{z_1}, F_{z_2}$, leading to a total of 6 unknowns. In terms of equations we have 2 equations (1,2) for each cable for a total of 4 equations. We have also to consider the mechanical equilibrium of the system that leads to $F_{x_1} + F_{x_2} = 0$ and $F_{z_1} + F_{z_2} + Mg = 0$, where M is the load mass so that we obtain a square system $\mathcal{S}$ which have no known analytic solution.

1) The Newton method: Provided that we have an estimate of the solution the Newton method may be used to possibly find the solution S_N of $\mathcal{S}$. Clearly as Newton is a numerical method it provides only an approximate solution S_a but without going into the details we are using a specific version that guarantee that there is a single solution S_N that lies inside a ball centered at S_a with radius ϵ, where ϵ can be arbitrarily chosen. In other words we can get S_a as close to S_n as needed so that we may claim that S_a is an *exact* solution.

A Newton estimate may be obtained by considering that the CDPR legs are rigid. In that case the L_0 are simply the distances between A_1, A_2 and B that are denoted ρ_1, ρ_2 while the horizontal and vertical forces $F_{x_1}, F_{x2}, F_{z_1}, F_{z_2}$ exerted on the legs can be determined by a classical static analysis. The set $\{\rho_1, \rho_2, F_{x_1}, F_{x2}, F_{z_1}, F_{z_2}\}$ will be denoted by S_r. Note that this set may provide negative F_{x_i} that are not possible for a cable which can only pull. Furthermore our tests show that the differences between the solution and S_r may be very large so that just running Newton with perturbed S_r will not allow to find the solution.

2) Limiting sagging: Before addressing the IK we have to consider first another problem: if the load mass is too low there may be an important sagging of one of the cable with the lowest point of the cable being below B. This will also occur if the load is coming close to the vertical line going through one of the A_i. Indeed in that case the horizontal force exerted on the load by cable i is becoming very small and thus the other cable has to exert a force that has also a small horizontal component: this can be obtained only if the force at B is almost vertical which implies a large sagging. This effect is evidently dependent upon the load mass M. To show this effect we have considered the output points $A_1(0, 0,), A_2(d = 10, 0)$ (in meter) and the pose (0.2,-5) for B and two load masses (figure 2). If the mass is low cable 2 lowest point is well below B while this does not occur for the larger mass. Having cable below the platform may not be acceptable for some applications. Therefore we consider a given workspace defined by $x \in [x_{min}, d - x_{min}]$, $z \in [-d/2, -1]$ and we are interested in determining the limit M_l on M so that for all $M \geq M_l$ the cables lowest point will be over B. We have noted a further advantage of having $M \geq M_l$: in that case the Newton method initialized with S_r always converges toward the exact solution. For determining M_l we may consider a modified version $\mathcal{S}_M$ of $\mathcal{S}$ where we add M as unknown and a constraint equation that indicates that the altitude of the cable lowest point is z_b (it cannot be given here for lack of space). Solving $\mathcal{S}_M$ is possible but difficult so that we have investigated the use of a NN for determining M_l for arbitrary d, x_{min}.

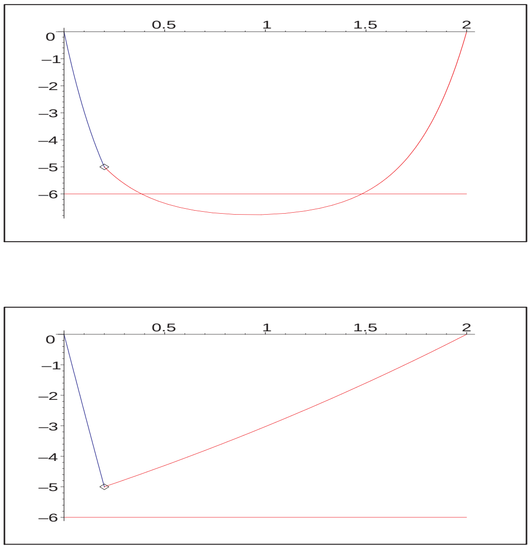

Fig. 2. For the pose (0.2m,-5m) with a load mass M lower than 8.7 kg points of the cable 2 are below B (top) while this does not occur for $M > 8.7kg$ (bottom)

For reader interested in the NN topology we have considered NNs with from 2 to 5 layers, 200 neurons per layer and connected with different activation functions, taken in the set $\{\mathrm{ReLU}, \mathrm{LeakyReLU}, \mathrm{CELU}, \mathrm{Softplud}, \mathrm{ELU}\}$. The Epoch has been experimentally set to 3000 with a batch size of 90. The used optimizer is Adam and the loss function is the MSE loss. We have established a training set of 865 samples by sampling d between 0.2 and 20 meters and various x_{min} value. The training time averaged 4 minutes. We then provided a verification set with 1000 samples to each computed NN and automatically extract the statistical data of the result; mean, minimum, maximum, error on M_l, in absolute and percentage value.

According to the test the estimation of M_l was possible with all activation methods with an accuracy ranging from 0.32% to 1.19% The best estimations were obtained with an NN with 2 layers using the Softplus activation function (accuracy: 0.32%) and with a NN with 5 layers using the CELU activation function (accuracy: 0.37%). Such an accuracy may not be sufficient for low x_{min}. We have then tested if the estimations may be used as initial guess for a Newton method applied on $\mathcal{S}_M$. It appears that they always lead to a convergence of the Newton method, thereby allowing to determine exactly M_l in an average number of iterations respectively of 6.55 and 8.59, the calculation time of the prediction being around 0.3 ms.

These results is our first example of a solving strategy that combines the prediction of a NN with the Newton method in order to get an exact result in a low computation time. Note that the solving strategy has been elaborated in view of the error metrics obtained during the verification test as this will be the case for all the strategies presented in this paper.

We may now address the IK problem that we may restrict to the case where M is lower than M_l. As we have alternate solving algorithms we have designed a training set of 8558 pairs $\{M, \mathbf{X}\}$ of mass and pose to its corresponding IK set $(L_{0_1}, L_{0_2}, F_{x_1}, F_{x_2}, F_{z_1}, F_{z_2})$. As soon as a neural network (NN) has been established we test its robustness and accuracy on a verification set of 2000 samples. As accuracy is extremely important a good performance indicator is the maximal difference between the exact cable lengths and the estimation provided by the NN. Our tests show that the maximal length errors in percentage for L_{0_1} is 7.5% and 15.32% for L_{0_2} with an average value of 0.695% and 2.75%. In terms of lengths the average error is 5.5 and 32.58 cm. Using the NN results is not satisfactory: the DK applied with the predicted result (that provides the pose that will be reached by the CDPR) shows that the distance between the desired pose and the DK solution may goes up to one meter. Hence we have to elaborate a strategy to improve the result. Using the NN result as initial guess for the Newton method leads to the exact solution in about 60% of the cases. We may also use as estimate for a L_0 either the one from S_r or the one coming from a NN, S_n. By using all possible combinations for the estimation of both L_0 we end up with 4 estimates for the pair L_{0_1}, L_{0_2}. Our tests show that Newton always converge to the exact solution for at least one of these combinations (clearly as we have a single IK solution the calculation stops as soon as we get a Newton convergence). They also show that the computation for solving the IK is always less than 3 ms, which is well below the one of any alternate method.

In summary **mixing NN and Newton appears to be a good solving strategy for the IK** although we cannot prove it will always work in which case we will have to resort to the methods proposed in [8], [10].

B. Direct kinematics

For the DK we have to determine the location of the pose x_b, z_b being given L_{0_1}, L_{0_2}. Overall the number of unknowns and constraints is the same than for the inverse kinematics. Preliminary tests have shown that for $d = 20$ and if we limit the maximal value of L_0 to $2d$, then the Newton scheme with S_r as initial guess always converge to the exact solution, whatever the mass is, with a number of iteration between 4 and 13 although we have assumed a cable with a relatively large elasticity (Nylon with $E = 2.7e^9 N/m^2$, 5 mm diameter). Hence using a NN will have an interest only if it leads to a lower computation time.

We designed NNs with 4 and 6 layers, 164 neurons per layer and different activation function, taken in the set $\{\mathrm{ReLU}; \mathrm{CELU}\}$ the optimizer set to Adam and different loss functions. The models were trained with 4400 samples such that $L_0 \leq 1.6d$ with a mass between 1 and 50 kg. They were then verified on 499 samples. As the NN results are not accurate enough they are used as initial guess for the Newton scheme. The results of these tests have shown that the average number of iterations of Newtonś scheme for convergence is 4.048, with a number of iterations between 3 and 12. Let $\mathcal{I}_r, \mathcal{I}_n$ denote the number of iterations before convergence respectively for the initial guesses $\mathcal{S}_r, \mathcal{S}_n$. We found out on the verification set that $\mathcal{I}_n < \mathcal{I}_r$ in 29 cases, $\mathcal{I}_n = \mathcal{I}_r$ in 323 cases and $\mathcal{I}_n > \mathcal{I}_r$ in 147 cases. **Hence using a NN for the direct kinematics is not of interest**.

VI. The 3-1 robot

For the 3-1 robot we have 3 translational dof of the platform with a workspace that is enclosed in the vertical prism $\mathcal{V}$ whose corner lines are the vertical lines going through the A_i. A reference frame is defined so that the coordinates of the A_i points are $A_1(0,0,0)$, $A_2(xa_2,0,0)$ and $A_3(xa_3, ya_3, 0)$. In the examples we will use $xa_2 = 20, xa_3 = 10, ya_3 = 20$ (in meters).

A. Inverse kinematics

For the IK we have as unknowns the 9 $L_{0_i}, F_{x_i}, F_{z_i}$. As for the 2-1 CDPR if we assume rigid legs it is easy to determine the set $S_r = \{\rho_i, F_{x_i}, F_{z_i}\}$ which is a solution of the IK. In terms of equations we get 6 relations from (1, 2). Let denote by $\mathbf{U_j}$ the bidimensional vector whose components are A_jB_x, A_jB_y and by n_x^j, n_y^j the components of the unit vector $\mathbf{U_j}/||\mathbf{U_j}||$ then the mechanical equilibrium imposes to have

$$\sum_{j=1}^{j=3} F_{x_j} n_x^j = 0 \quad \sum_{j=1}^{j=3} F_{x_j} n_y^j = 0 \quad \sum_{j=1}^{j=3} F_{z_j} + Mg = 0 \quad (3)$$

so that we have a total of 9 equations and therefore the IK system $\mathcal{S}$ is square but has no known analytic solution.

Note that B cannot lie in the vertical plane $\mathcal{P}_{ij}$ that include any pair of A_i, A_j. Indeed in that case the cables i, j do not exert any force along the perpendicular $\mathbf{v}$ to the plane and as the third cable k does not lie in this plane its mass exerts a non-0 force along $\mathbf{v}$ unless its length is infinite: hence the mechanical equilibrium cannot be reached. If B is not in the plane $\mathcal{P}_{ij}$, the closer it comes to it, the larger will be the length of cable k. Therefore the CDPR workspace will be strictly enclosed in the prism $\mathcal{V}$ and the workspace border will be determined by the limit we impose on the cable length and/or on the altitude of the cables lowest points.

A major issue for solving the IK with the Newton scheme using S_r as initial guess is that the convergence domain is very sensitive to both the load mass M and to the platform pose. The Newton scheme will not converge if:

- M is very low as the F_x, F_z in S_r are very small
- M is very large as the elasticity of the cable material may lead to large differences between the ρ and L_0

For example we have set $x_b = 10, z_b = -20$ and table I provides, for various values of y_b, the ranges for M for which Newton will converge. Note also that the influence of z_b on the range for M leading to convergence of the Newton scheme is unclear: an increase of z_b may lead to an increase or a decrease for the width of the range.

y_b	10m	9m	5m	3m	1m
M	[1,203]	[1e^{-5},647]	[1e^{-5},99]	[1e^{-5},1.2]	[1e^{-5},0.4]

TABLE I

RANGE FOR THE MASS M IN KG LEADING TO NEWTON CONVERGENCE FOR S_r FOR $x_b = 10m, z_b = -20m$ AND VARIOUS VALUES FOR y_b

We have investigated the use of a NN to solve the IK. For that purpose we have used the equations (3) to reduce the number of unknowns to 5, namely $L_{0_1}, L_{0_2}, L_{0_3}, F_{x_1}, F_{z_1}, F_{z_2}$. We have built a training set of 17776 samples of $x_b, y_b, z_b, M, F_{x_i}, F_{z_i}$ for a mass between 1 and 101 (step size =10), z_b between -20 and -6 (step size =2). In terms of x_b, y_b we have a base set of 1616 points that are investigated for each value of M, z_b. Note that the learning set do not impose any constraint on the cable length (or on the altitude of the lowest cable point) so that we have large changes in the cable lengths: for example the length of cable 3 varies between 6 and 1271 meters.

Regarding the NN we have systematically investigated all NNs with 2, 3 and 4 layers, 200 neurons per layer and an activation function that is the same for all layers, taken in the set $\{\mathrm{ReLU}, \mathrm{LeakyReLU}, \mathrm{CELU}\}$ (other activation functions were tested but none provide better results than the chosen set). The Epoch has been experimentally set to 1500 with a batch size of 90, the optimizer set to Adam and the loss function has been defined as the sum of the MSE loss for each of the variables. The computation time for establishing one NN varies between 20 and 40 minutes, the final loss being between 54 and 560. We then check the NN on a verification set of 1000 randomly chosen poses, all different from the training poses, for which we have calculated the IK exact solution. This allows us to establish statistics data on the NN result such as mean and maximal error on the unknowns (both in absolute value and in absolute percentage value), mean and maximal number of Newton iteration when it converges.

1) Results:

For the verification set Newton with S_r as initial guess does not converge for only 25 samples. The use of the prediction of the NNs ReLU with 3 layers, LeakyRelu with 3 and 4 layers and CELU with 3 layers as initial guess for the Newton scheme allows one to reduce the number Newton failure to 14. Newton failures are obtained because the maximal error on the predicted L_0 is very large (around 200 meters for L_{0_1}!), with a mean error of (4.25, 1.84, 0.24) and the error on F_{z_2} is also very large (around 400 N for M=101 kg).

It may be assumed that if we impose altitude constraint on the lowest cable point (and therefore limit the cable lengths) we may have obtained better result. We have therefore arbitrarily considered the subset of the training set such that none of the lowest cable points have an altitude lower than $z_b - 1$, leading to a training set of 13155 samples. Tests have shown a small improvement as as the number of Newton failure has been reduced to 10. Still we have unacceptable large maximal errors on the L_0 (between 3 and 5 meters) and average errors of (0.35, 0.9, 0.64) meters.

In summary although the NN prediction exhibits better performance to get the exact result than using S_r as initial guess, the result is not satisfying as we still have failure.

2) A possibly winning strategy:

As seen previously one reason for the failure of the Newton scheme using S_n is that we have large errors on both the L_0 and F_{z_2}. But as we have arbitrarily eliminated three unknowns by using the linearity of equations (3) we may wonder if another choice for the three eliminated unknowns may lead to different results. Hence we have built other NNs based on a system with as unknowns $(L_{0_1}, L_{0_2}, L_{0_3}, F_{x_2}, F_{z_1}, F_{z_2})$. We have noted that while the number of Newton failure is similar to the previous model, we have a much lower error on F_{z_2}. This has led us to elaborate an IK solving strategy based on a succession of steps, each step involving the use of the Newton method with different initial guesses:

- use S_r as initial guess
- use S_n of the first model as initial guess
- use S_n of the first model as initial guess except that the F_{z_2} prediction is obtained from the second model
- use S_n of the second model as initial guess

Evidently the process stops as soon as the Newton scheme has converged. Our test on the verification set has shown that we get 0 Newton failure with this strategy (whether we have or not constraint on the lowest cable point). To confirm it we have randomly selected 20 000 poses and mass and run the algorithm for each of them: we have again obtained 0 failure with in all cases at most 3 Newton calls. **Hence by combining the prediction of several NNs and the Newton scheme it is apparently possible to solve exactly the IK** although we cannot prove that it will always work.

B. Direct kinematics

For the DK we have to determine x_b, y_b, z_b being given the 3 L_0. The number of unknowns and equations are the same than for the IK. We may as well use the system of 9 equations or reduce it to a system $\mathcal{S}_2$ of 6 equations by using solving the 3 linear equations (3) for three arbitrarily chosen $F_{x_j}, F_{z_k}, F_{z_l}$. In all cases there is no known analytic solution for the system $\mathcal{S}$. Assuming rigid legs we may also establish the initial guess set S_r.

We have established a training set $\mathcal{T}_1$ of 15131 sets of L_0 by sampling the L_0 between 1 and 31 meters (step size = 2 meters), M varying between 1 and 101 (step size = 10 kg) and providing the exact x_b, y_b, z_b, F_x, F_z. A second set $\mathcal{T}_2$ of 19202 L_0 sets that provides x_b, y_b, z_b but only $F_{x_1}, F_{z_1}, F_{z_2}$ has also been established with the same sampling for the L_0 and M We have also extracted from these set a training set $\mathcal{T}_3$ which provides only x_b, y_b, z_b. A verification set of 1000 samples, all different from the training sets, has been established. Regarding building the NNs we have used the same systematic approach than for the IK for building the various NNs. The computation time for establishing one NN is in the range [20,38] minutes and the final loss lies between 0.022 and 0.19.

Beside using S_r as initial guess for the Newton scheme we have tried 3 solving methods:

1) using the set $\mathcal{T}_1$ to establish a prediction of all the unknowns
2) using the set $\mathcal{T}_2$ to establish a prediction of the reduced system $\mathcal{S}_2$
3) using the set $\mathcal{T}_3$ to establish a prediction of x_b, y_b, z_b

1) Results:

Preliminary tests have shown that method 1 was not working: the Newton scheme almost always never converge with its prediction. Method 3 does not provide a too bad estimation of x_b, y_b, z_b (typically with a maximum error between 0.3 and 0.7 meters) but if we are also interested in the F_x, F_z, the error is very large (typically with an error of at least one order of magnitude).

It was also noted that for the verification set the Newton scheme with S_r as initial guess does not converge in 86.5% of the cases. One of the NN (with 2 layers and LeakyReLU) allows one to reduce the failure case to 69.4%. Similarly to the IK we have also noted large error on F_{z_2}. This suggests to use the same approach than for the IK of having two NN models, both involving F_{z_2}. Let N_i be the prediction of model i and N_j^i the prediction of model j except that the prediction of F_{z_2} is the one of model i. Hence we successively test the Newton scheme with as initial guess N_1, N_2, N_1^2, N_2^1. This approach allows to decrease the Newton failure by 10%, which is not sufficient. We have then decided to generalize the approach. For a given model our systematic training has allowed to establish different NNs with number of layers from 2 to 4 and an activation function in the set $\{\mathrm{CELU}, \mathrm{LeakyReLU}, \mathrm{ReLU}\}$. Hence we have 9 NNs that give the prediction for $x_b, y_b, z_b, F_{x_1}, F_{z_1}, F_{z_2}$ and 9 models for the prediction of $x_b, y_b, z_b, F_{x_2}, F_{z_1}, F_{z_2}$. We then test all 81 combinations of these NNs by using the same strategy than for the IK. This approach works surprisingly quite well: the failure rate goes to 0% (this have then been confirmed for a verification set of 10 000 samples) while the mean number of Newton calls is 3.07 with an average number of Newton iterations of 17.6. The average computation time is between 2-3 ms and around 8 ms in the worst case: this is quite satisfactory for the analysis step. **Hence using the prediction of several NNs and the Newton scheme appears to allows one to solve efficiently and exactly the DK** although we cannot prove it will always work.

VII. Conclusion

In this paper we have considered a realistic cable model, a specific class of CDPR and we have investigated the use of neural networks to solve the inverse and direct kinematics, considered as a tool for the performance analysis of a CDPR. We have shown neural networks, although they provide a prediction of the exact solution with errors that are significantly too high, may be used to design efficient and exact solving methods, provided that they are incorporated in a strategy that combines prediction and the Newton scheme. The strategies that have been presented in this paper have been manually determined according to the error metrics obtained during the verification tests and are therefore possibly ad-hoc with respect to the chosen examples: an issue will be to determine automatically these strategies according to the error metrics obtained from the verification tests. We have also illustrated that neural networks may play a role in determining load mass limit over which kinematics become simpler.

Still we have managed only CDPR that have a single solution for both the inverse and direct kinematics. Next step will be to generalize the proposed approach for the CDPR general cases that have multiple solutions for both the IK and DK. This is a challenging issue as NNs are able to provide only a single solution. A possible approach will be to calculate all kinematics solution for a sample of inputs and by using a continuation to provide a training set for each of the solutions which will allow one to design a specific NN for its branch. Unfortunately this may fail to provide all the solutions as we are unable to determine in advance what will be the maximal number of kinematics solutions and any inputs that leads to this number.

In view of the CDPR analysis point algorithms as proposed in this paper has the drawback that it imposes a sampling of the workspace and therefore cannot guarantee that we will find the real extrema of the KIs. An on-going investigation is to use the result provided by the point algorithm to determine the extrema of the KIs in a small region around the result by using interval analysis, thereby allowing to determine the real extrema of the KIs over a given workspace.

Further research will also investigate the use of recently proposed hardware devoted to IA with a large number of GPU for first decreasing the computation time necessary to establish the NNs and, second, to use the inherent parallelism of interval analysis for extending the point result of the algorithm to a region.

References

[1] R. Achili *et al.*, “A stable adaptive force/position controller for a C5 parallel robot: a neural network approach,” *Robotica*, vol. 30, no. 7, pp. 1177–1187, December 2012.

[2] R. Boudreau, G. Levesque, and S. Darenfed, “Parallel manipulator kinematics learning using holographic neural network models,” *Robotics and Computer-Integrated Manufacturing*, vol. 14, no. 1, pp. 37–44, 1998.

[3] D. Chen, Y. Zhang, and S. Li, “Zeroing neural-dynamics approach and its robust and rapid solution for parallel robot manipulators against superposition of multiple disturbances,” *Neurocomputing*, vol. 275, pp. 845–858, 2018.

[4] R. Kang *et al.*, “Learning the forward kinematics behavior of a hybrid robot employing artifical neural networks,” *Robotica*, vol. 30, no. 5, pp. 847–855, September 2012.

[5] I. Kardan and A. Akbarzadeh, “An improved hybrid method for forward kinematics analysis of parallel robots,” *Advanced Robotics*, vol. 29, no. 6, pp. 401–411, 2015.

[6] L. Sang and M.-C. Han, “The estimation for forward kinematic solution of Stewart platform using the neural network,” in *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Kyongju, October, 17-21, 1999.

[7] C. Yee and K. Lim, “Forward kinematics solution of Stewart platform using neural network,” *Neurocomputing*, vol. 16, no. 4, pp. 333–349, 1997.

[8] J.-P. Merlet, “The forward kinematics of cable-driven parallel robots with sagging cables,” in *2nd Int. Conf. on cable-driven parallel robots (CableCon)*, Duisburg, August, 24-27, 2014, pp. 3–16.

[9] ——, “A new generic approach for the inverse kinematics of cable-driven parallel robot with 6 deformable cables,” in *ARK*, Grasse, June, 27-30, 2016.

[10] ——, “A generic numerical continuation scheme for solving the direct kinematics of cable-driven parallel robot with deformable cables,” in *IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, Daejeon, October, 9-14, 2016.

[11] R. Ahouee, S. Moussavi, and J. Hamedi, “Neuro-fuzzy intelligent control algorithm for cable-driven robots with elastic cables,” in *2nd International Conference on Cybernetics, Robotics and Control*, 2017.

[12] A. Ghasemimi, M. Eghtesad, and M. Farid, “Neural network solution for forward kinematics problem of cable robots,” *J. of Intelligent and Robotic Systems*, vol. 60, pp. 201–215, 2010.

[13] M. Barhaghtalab *et al.*, “On the design of the robust neuro-adaptive controller for cable-driven parallel robots,” *Automatika*, vol. 57, no. 3, pp. 724–735, 2016.

[14] I. Chawla *et al.*, “Neural network-based inverse kineto-static analysis of cable-driven parallel robot considering cable mass and elasticity,” in *5th Int. Conf. on cable-driven parallel robots (CableCon)*, virtual, July, 7-9, 2021.

[15] F. Rosenblatt, “Perceptron simulation experiments,” *Proceedings of the IRE*, vol. 48, no. 3, pp. 301–309, 1960.

[16] H. M. Irvine, *Cable Structures*. MIT Press, 1981.