ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-MOPOST047

DETERMINATION OF THE PHASE-SPACE STABILITY BORDER WITH MACHINE LEARNING TECHNIQUES

F.F. Van der Veken*, R. Akbari, M.P. Bogaert, E. Fol, M. Giovannozzi, A.L. Lowyck, C.E. Montanari, W. Van Goethem, CERN, Meyrin, Switzerland

Abstract

The dynamic aperture (DA) of a hadron accelerator is represented by the volume in phase space that exhibits bounded motion, where we disregard any disconnected parts that could be due to stable islands. To estimate DA in numerical simulations, it is customary to sample a set of initial conditions using a polar grid in the transverse planes, featuring a limited number of angles and using evenly distributed radial amplitudes. This method becomes very CPU intensive when detailed scans in 4D, and even more in higher dimensions, are used to compute the dynamic aperture. In this paper, a new method is presented, in which the border of the phase-space stable region is identified using a machine learning (ML) model. This allows one to optimise the computational time by taking the complex geometry of the phase space into account, using adaptive sampling to increase the density of initial conditions along the border of stability.

INTRODUCTION

When studying the non-linear beam dynamics in a circular hadron ring in numerical simulations, one of the key concepts is that of dynamic aperture (DA). It represents the smallest, simply-connected volume in the $2n$-dimensional phase space that is stable (where n is the number of degrees of freedom), i.e. that exhibits bounded motion over a given time interval [1, 2]. More precisely, it is defined as the radius of the $2n$-sphere with the same volume V_{2n} as the bounded region:

$$\mathrm{DA} = \sqrt[2n]{\frac{\Gamma\left(n+1\right) V_{2n}}{\pi^n}} \,. \tag{1}$$

The DA is a useful quantity, not only in accelerator design, but also because it can be linked to measurable quantities such as the time evolution of the beam intensity [3, 4] or luminosity [5, 6] of a storage ring or collider.

In practice, computational limitations make us consider the DA in two dimensions only, instead of six (the dimension of the phase space of a particle beam). The transverse momenta and longitudinal position coordinates are typically set to zero, while the longitudinal momentum coordinate is typically set at a non-zero value that is deemed representative for the longitudinal beam distribution under consideration.

An important aspect in the definition of DA, is that it excludes disconnected stable islands from the calculation of the volume. To consider this in simulations, traditionally the DA is calculated by sampling initial conditions in a polar grid, over a certain number of angles [2]. The stability border is given, for each angle θ_i, by the largest connected stable amplitude r_i. After integrating r_i over angles, the DA is approximated by [2]:

$$\mathrm{DA} \approx \sqrt{\frac{2}{\pi}\Delta\theta \sum_i^{N_\theta} c_i \, r_i^2} \qquad \Delta\theta = \frac{1}{N_\theta+1}\frac{\pi}{2} \,, \tag{2}$$

where c_i are the constants of the chosen open integration method and N_θ the number of angles. This calculation has an error that scales with $\sim \Delta r \Delta\theta$, hence ideally one would like to keep $\Delta r \sim \Delta\theta$ to minimise the error associated to the DA estimate [2].

It is also clear that the overall strategy to compute the DA of a given accelerator lattice would gain in efficiency if the initial conditions could be chosen to probe with high density the region of the stability border, only. In this paper, we explore an alternative to the polar sampling, where an initial set of particles is sampled uniformly and machine learning (ML) is applied to recognise the stability border at a given number of turns. In a second step, we resample a larger set of particles focused around the border region.

MACHINE LEARNING

To find the stability border at a given number of turns N, we divide the particles into those that survived at least N turns, meaning their motion remains bounded for at least this time, and those that did not, and then train an ML model as a classifier. While in theory it would be feasible to divide the particles in multiple groups for different number of survived turns, it would not be an optimal division to train the ML algorithm on.

We opted to use a support-vector machine (SVM) model [7, 8], which is a supervised learning algorithm that is one of the most robust prediction methods available. Because of its (by default) binary classification, it is well-suited for our particular data structure if split as described in the previous paragraph. Though SVM is by default a linear classifier, it can be used to classify data that are separated by a non-linear boundary by using a kernel transformation: in our case, a radial basis function is most suited due to the radial nature of the data. It is an exponential kernel that maps two vectors $\mathbf{v}$ and $\mathbf{w}$ as a function of the hyperparameter γ:

$$K(\mathbf{v}, \mathbf{w}) = \mathrm{e}^{-\gamma \, \|\mathbf{v}-\mathbf{w}\|} \,, \tag{3}$$

where $\|\cdot\|$ stands for the vector norm. In ML training, the different hyperparameters have to be tuned to get optimal results. Like in classical statistics, a model that is not adequately tuned can lead to under- or over-fitting the data. In our case, there are two hyperparameters to tune: the above-mentioned γ that represents the convolution of the curve

* frederik.van.der.veken@cern.ch

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-MOPOST047

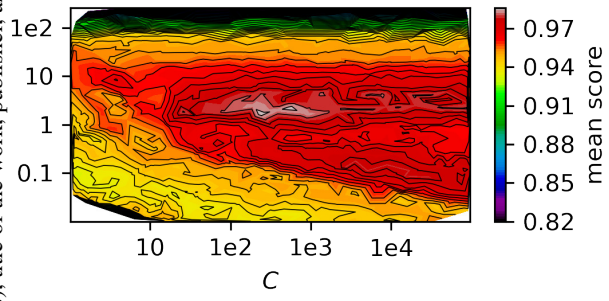

Figure 1: Average CV scoring function of a scan of more than 10^3 values for the hyperparameters C and γ.

(small γ tends to make the curve more circular), and the so-called C-parameter, that represents how strongly misclassified points are penalised.

To tune the hyperparameters, we perform a random scan over the $C - \gamma$ parameter space, and assess each pair's performance using Cross-Validation (CV) [9]. This means that the data set is randomly split into k equally sized partitions, where $k - 1$ partitions are trained upon and the remaining partition is used to test the model's performance. This is then iterated k times, hence for one hyperparameters pair k models are trained and tested. For small data sets it is common to take $k = 5$, but in our case we chose $k = 10$ as the data set is large enough to get reliable testing with only 10% of the data. An example of such a scan is demonstrated in Fig. 1, which shows the average of the 10 models' scores for 10^3 different hyperparameter pair values. It is clear that the model is optimal for $1 \lesssim \gamma < 10$ and $10^2 < C < 10^3$. We observe that the DA does not change much after sampling 2×10^2 pairs of hyperparameters.

BORDER RECOGNITION

Once the model is trained over a well-tuned set of hyperparameters, we want to extract the decision boundary between the two categories as it represents the stability boundary. This is not a trivial task: by its nature, the ML model excels at predicting for any point the category that it belongs to based on the survival criterion; however, it provides no direct information on the boundary.

To get an expression for the boundary, we use a trick: we create a temporary image grid that represents the region of interest, and let the ML model 'colour' each pixel as white or black depending on the predicted category. Then we use standard image recognition techniques to find the contour lines between the two regions. Finally, the points from the contour are interpolated with a spline to obtain a continuous curve parameterised by $u \in [0, 1]$, which can then be integrated to get the area:

$$A = \int_0^1 x(u) \frac{\mathrm{d}y(u)}{\mathrm{d}u} \, \mathrm{d}u \,, \tag{4}$$

from which the DA can be derived using Eq. (1).

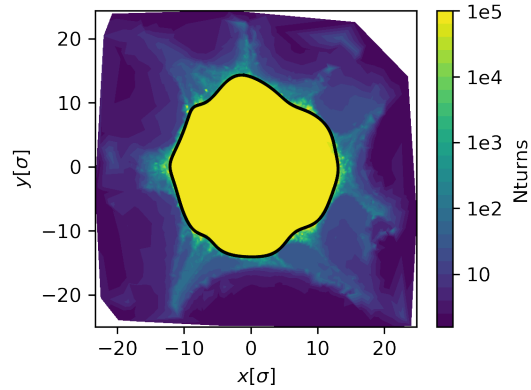

Figure 2: Example of a DA result for the HL–LHC, tracked over 6000 particles after resampling. In black is the DA border as recognised by the ML algorithm.

An example result is shown in Fig. 2, which is a logarithmic plot of the survived number of turns N_{surv}, for a set of 6×10^3 particles after resampling (see below), tracked over the High Luminosity LHC (HL–LHC) lattice [10] including a specific realisation of magnetic field errors. Shown in black is the DA border predicted by the ML model. The DA for this specific example is calculated to be 12.87σ.

RESAMPLING

An adaptive algorithm typically works in several steps. In our case, after we used the uniform sample of 10^3 particles to recognise the minimum and maximum stability borders, defined by the minimum resp. maximum number of survived turns, we can resample particles in between these two borders.[1] However, the turn value that defines the minimum stability border needs to be chosen carefully. If it is too small, the resampling region will be larger than necessary with a sub-optimal resampling efficiency as a consequence. On the other hand, if the minimum turn value is too large, the resampling region will be rather small, making it difficult for the ML model to define two distinct borders. In our case, we have observed that taking $N_{\mathrm{min}} = 20$ turns gives a nice balance between the two. Note that the value of N_{min} does not influence the value of the DA at larger number of turns.

Once we established both borders, we can resample. We want to prioritise particles closest to the maximum border region, as these can rapidly change their survived number of turns for small amplitude steps, due to the chaoticity of the dynamics in this region. Furthermore, the borders we have thus far established are rather crude and might be a little off; hence, to be able to refine them, we want to additionally sample particles at amplitudes larger than the minimum border or smaller than the maximum border.

To achieve this, we first define for each particle a normalised shortest distance d to the maximum border, defined

[1] Note that the minimum border sits at amplitudes larger than the maximum border, as the former contains more particles than the latter.

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-MOPOST047

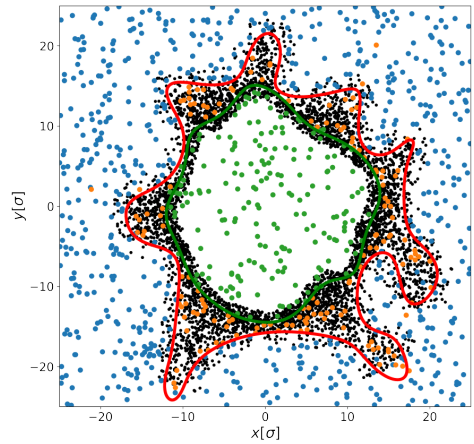

Figure 3: Resampling in action. The minimum and maximum border are given by the red resp. green curves, which are used to sample the 5×10^3 new particles shown in black.

such that $d = 0$ represents a particle on the maximum border, and $d = 1$ a particle on the minimum border. We allow d to take values from $-1/2$ to $+3/2$, representing particles beyond the borders as discussed above. Then we weight the sampling algorithm with a probability that is a function of this distance, skewed towards the maximum border. We achieve the desired sampling density if we use the following weighting function:

$$\sqrt{d + \frac{1}{2}}\, e^{-\left(d+\frac{1}{2}\right)^2} \qquad d \in \left[-\frac{1}{2}, \frac{3}{2}\right] . \tag{5}$$

This is illustrated in Fig. 3, which shows three categories of initial particles: blue for $N_{\text{surv}} < 20$, orange for $20 \leq N_{\text{surv}} < 10^5$, and green for $N_{\text{surv}} = 10^5$. The borders corresponding to $N_{\text{min}} = 20$ and $N_{\text{max}} = 10^5$, as found by the ML model, are the red resp. green curve, and 5×10^3 new particles, sampled following the logic above, are shown in black. These extra points are used in the border calculation as shown in Fig. 2.

PRECISION

The precision of the ML method to calculate the DA is defined by three parameters: the number of tracked particles, the number of configurations used to set the hyperparameters, and the step size of the image grid used to retrieve the decision function. The effect of the last parameter is shown in the top plot of Fig. 4, which shows the relative deviation

$$\delta\text{DA} = \frac{\text{DA}(\Delta x = 1\sigma) - \text{DA}(\Delta x)}{\text{DA}(\Delta x = 1\sigma)} \tag{6}$$

compared to the most crude DA (calculated with a step size of 1σ), as a function of the step size Δx. One can see that already at a step size of $\Delta x \sim 0.02\sigma$ the convergence of the relative deviation is achieved.

Finally, to estimate the overall accuracy of the method, we repeated the same DA calculation 5×10^2 times, with

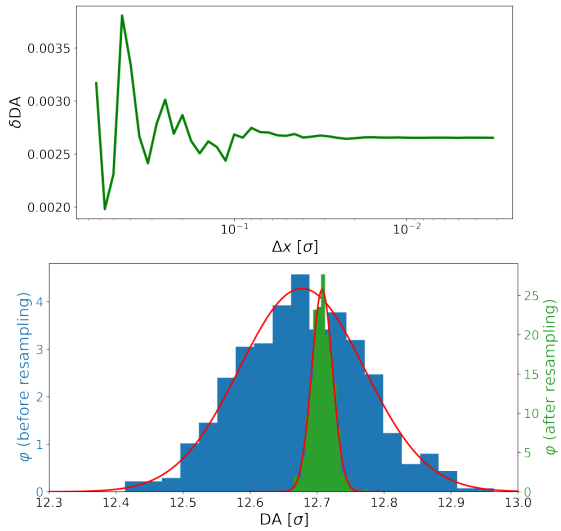

Figure 4: Accuracy assessment of the model. Top: DA precision as a function of the step size of the image grid. Bottom: Histogram of 5×10^2 iterations of the same DA calculation. For the blue plot, each iteration used 10^3 initial conditions, 2×10^2 hyperparameter pairs, and a step size of 0.02σ. The green plot shows the same data after resampling.

each iteration having 10^3 randomly sampled particles, 2×10^2 hyperparameters pairs, and an image grid stepsize of $\Delta x = 0.02\sigma$. This is illustrated in the bottom plot of Fig. 4, which shows a histogram over these iterations (blue plot) with mean 12.681σ and standard deviation 0.093σ. We can conclude that even with only 10^3 initial particles, we obtain a result with a rather satisfying precision. Furthermore, after resampling 5×10^3 extra particles for each iteration (green plot), the precision is strongly enhanced, with mean 12.708σ and standard deviation 0.016σ.

CONCLUSION

We have developed a new method to calculate the DA of a circular accelerator, based on supervised machine learning.

We explored how the new model depends on the different ML hyperparameters, and verified that a randomised search over configurations leads to good results. Furthermore, we inspected the overall performance of the model by making 5×10^2 iterations of the same calculation, confirming a precision compatible with that of traditional methods. Where the latter typically use around 6×10^3 initial particles to achieve this precision, the proposed method does this with 1×10^3 initial particles, only. The clear gain in CPU time that follows from this is very useful for high-volume studies, where multiple different machine configurations are probed.

Finally, we showed how the model can be used to implement an adaptive sampling algorithm that only samples new initial conditions from the region of interest, improving overall accuracy with minimal increase in computing time requirements. This is a necessary improvement for studies that link the DA to physical observables.

ISBN: 978-3-95450-227-1 ISSN: 2673-5490 doi:10.18429/JACoW-IPAC2022-MOPOST047

REFERENCES

[1] F. Schmidt, F. Willeke, and F. Zimmermann, "Comparison of methods to determine long-term stability in proton storage rings. Comparison of methods to prove long term stability in proton storage rings," *Part. Accel.*, vol. 35, 249–256. 9 p, 1991, https://cds.cern.ch/record/218662

[2] E. Todesco and M. Giovannozzi, "Dynamic aperture estimates and phase-space distortions in nonlinear betatron motion," *Phys. Rev. E*, vol. 53, pp. 4067–4076, 4 1996, doi:10.1103/PhysRevE.53.4067

[3] M. Giovannozzi, "Proposed scaling law for intensity evolution in hadron storage rings based on dynamic aperture variation with time," *Phys. Rev. ST Accel. Beams*, vol. 15, p. 024 001, 2 2012, doi:10.1103/PhysRevSTAB.15.024001

[4] A. Bazzani, M. Giovannozzi, E. Maclean, C. Montanari, F. Van der Veken, and W. Van Goethem, "Advances on the modeling of the time evolution of dynamic aperture of hadron circular accelerators," *Phys. Rev. Accel. Beams*, vol. 22, p. 104 003, 10 2019, doi:10.1103/PhysRevAccelBeams.22.104003

[5] M. Giovannozzi and F. Van der Veken, "Description of the luminosity evolution for the CERN LHC including dynamic aperture effects. Part I: the model," *Nucl. Instrum. Methods Phys. Res.*, vol. A905, pp. 171–179, 2018, [Erratum: Nucl. Instrum. Methods Phys. Res. A927,471(2019)], doi:10.1016/j.nima.2019.01.072

[6] M. Giovannozzi and F. F. Van der Veken, "Description of the luminosity evolution for the CERN LHC including dynamic aperture effects. Part II: application to Run 1 data," *Nucl. Instrum. Methods Phys. Res.*, vol. A908, pp. 1–9, 2018, doi:10.1016/j.nima.2018.08.019

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[8] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer New York Inc., 2001, https://hastie.su.domains/ElemStatLearn/

[10] I. Béjar Alonso, O. Brüning, P. Fessia, L. Rossi, L. Tavian, and M. Zerlauth, *High-Luminosity Large Hadron Collider (HL–LHC): Technical design report*. CERN, 2020, doi:10.23731/CYRM-2020-0010