Proving the stability of cycle navigation using capture sets*

Quentin Brateau[a,b], Loïck Degorre[a,c], Fabrice Le Bars[a,d], Luc Jaulin[a,e]

[a] *Lab-STICC UMR 6285, ROBEX team, ENSTA Bretagne, 2 rue François Verny, 29200 Brest France*
[b] *quentin.brateau@ensta-bretagne.org*
[c] *loick.degorre@ensta-bretagne.fr*
[d] *fabrice.le_bars@ensta-bretagne.fr*
[e] *lucjaulin@gmail.com*

Abstract

Navigating Autonomous Underwater Vehicles (AUVs) presents significant challenges due to the absence of traditional localization systems. Cycle navigation emerges as a promising paradigm, enabling reliable navigation using minimal exteroceptive measurements. This approach leverages predefined cyclic trajectories, which are stabilized based on environmental feedback, ensuring frugal and discreet operations without reliance on high computational power or extensive sensor systems. The goal of this work is to prove the stability of the cycle navigation. This work aims to prove the stability of the cycle navigation. As cycle navigation is a non-linear system governed by a discrete inclusion condition, conventional methods have trouble to prove its stability. For this reason, this paper focuses on set methods to prove the stability of cycle navigation using set methods. The stability is proven by exhibiting a positive invariant set, which is a set stable by application of the evolution function of the system. This ensures that the evolution function will not remove states from the positively invariant set. Then, the characterization of the capture basin is an asset when performing cycle navigation, as it represents the set of initial states for the system which leads to the positive invariant set. Once the system reach both the capture basin or the positive invariant set, which are generalized as a capture set, it remains captured forever. This approach not only guarantees the stability of the system in the neighborhood of the equilibrium point, but also establishes that it exists an area in which the stability of the cycle navigation will lead to a stable behavior. This work offers a robust, computationally efficient alternative to traditional stability methods, particularly suited for resource-constrained AUVs, because the underwater environment lacks of suitable, cheap and easy-to-use localization methods, which forces us finding alternative ways to navigate and explore this particular environment.

Keywords: Underwater robotics, Mobile robots and vehicles

1. Introduction

Navigating robots in GNSS-denied environments presents significant challenges, particularly in maintaining stable trajectories without access to traditional localization systems. For Autonomous Underwater Vehicles (AUVs), ensuring reliable navigation is critical for applications such as seabed mapping, environmental monitoring, and search-and-rescue operations. However, as GNSS is unavailable in underwater environments, it becomes convoluted to use classical control laws to navigate effectively. Underwater positioning systems such as Long BaseLine (LBL) or Ultra-Short BaseLine (USBL) are becoming increasingly popular, but these systems come with significant drawbacks. They are expensive, challenging to deploy, and often not easily integrated into small or resource-constrained robots [1, 2]. These limitations underscore the need for alternative navigation paradigms that circumvent the dependency on external, costly infrastructure.

Efforts have been made to develop methods that facilitate the implementation of localization solutions in GNSS-denied environments. For example, one such method involves localizing an AUV using a single acoustic beacon of known position to assist navigation [3]. While these approaches mitigate some of the reliance on complex multi-sensor systems, they still require the deployment of additional hardware on the robot. This can add complexity, weight, and energy demands, which are especially undesirable for small or lightweight AUVs designed for frugal, stealthy operations. Consequently, there remains a pressing need for navigation methods that are inherently robust, computationally efficient, and minimally dependent on external systems or additional hardware.

In classical control theory, the robot's state must first be estimated before it can be controlled towards a reference state. This paradigm works well in environments where reliable state estimation is feasible. However, in GNSS-denied environments, the unavailability of absolute positioning data poses a critical challenge for this approach. The proposed method addresses this issue by fundamentally rethinking the problem. Instead of prioritizing state estimation, the approach begins by having the robot navigate through predefined cyclic trajectories, or cycles, and collecting environmental measurements all along its path. These exteroceptive measurements are then used to adapt and stabilize the cycle's position within the environment. Once the

*This work has been supported by the French Government Defense procurement and technology agency (AID).

cycle is stabilized, the robot naturally converges to a predefined position, effectively solving the state estimation problem as a consequence of its stabilized navigation.

This innovative approach enables navigation in GNSS-denied environments using minimal exteroceptive measurements, ensuring the robot's discretion and frugality [4]. By relying on a simple state machine that adjusts cycle parameters based on collected measurements, the approach eliminates the need for high computational power or complex sensor arrays. This is particularly well-suited for small, autonomous robots operating in resource-constrained environments. Moreover, the proposed navigation paradigm inherently reduces the risk of detection, an important consideration for applications requiring stealth, such as defense or sensitive environmental monitoring.

The central focus of this study is to analyze the stability of such cycles at two distinct levels. First, global stability is examined, where the convergence of the robot's position through successive cycle iterations is proven. This ensures that the robot's trajectory converges reliably to a target region despite environmental disturbances or initial positional inaccuracies. Second, local stability is studied, focusing on the robot's trajectory during a single cycle iteration after convergence. This dual-layered stability analysis provides a comprehensive understanding of the proposed navigation paradigm's robustness and adaptability.

The study of the stability of linear systems is a field that has already been explored and whose results are well known [5]. Non-linear system stability is studied by linearizing the system around an operating point, allowing to use established results for linear systems. This does indeed prove the global system's stability, but only exhibit a neighborhood around an operating point where the linearized system is guaranteed to be stable. Lyapunov methods or invariance principle methods, for instance could be used to study the stability of non-linear systems [6, 7]. However, these methods struggle to deal with discrete inclusion problems. Conventional methods are therefore ill-suited to dealing with non-linear problems, can only prove stability around an operating point, and offer no guarantee of results. The approach proposed in this paper not only provides a guaranteed way of proving that a set of states is stable by applying the system dynamics, but also characterizes the set of starting positions leading to this stability, using capture sets.

Section 2 presents the cycle navigation through experimental results. Section 3 sets out the formalism of the problem, then section 4 proves the stability of the cycle navigation using set methods. Section 5 concludes this paper and presents the perspectives of this new paradigm.

2. Cycle navigation

Stable cycle navigation has proven results both in simulation and in field robotics experiments [1]. Figure 1 shows the BlueBoat [2], an Uncrewed Surface Vehicle (USV) that navigates using cycles on the Guerlédan Lake.

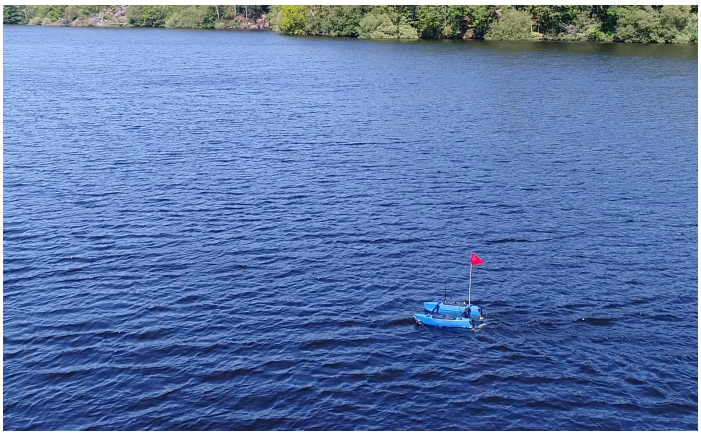


Figure 1: BlueBoat navigating on the Guerlédan Lake using cycle navigation

The robot uses a compass to track its heading, and a simple echosounder to measure depth below the surface. The autonomous boat records GNSS position only for ground truth, and the robot position is not used in the control loop during the experiments. The USV follows a simple timed automaton where durations are controlled relative to the echosounder measurements.

The robot and the timed automaton are synchronized as defined in [8] such that the robot trajectory describes a square. For this purpose, the state machine is a succession of states that guide the robot along straight lines and circle arcs. Then by controlling the duration of some transitions, the position of the cycle could be shifted in the two-dimensional plane.

Figure 2 shows an example of two trajectories of the BlueBoat. These trajectories are plotted on the bathymetric map of the lake. The blue trajectory seems to converge toward a stable cycle, whereas the red one does not seem to converge but rather drifts alongside the isobath.

This experiment shows that the stabilization of a robot using cycle navigation is possible. However, with the same mission script, there is some initial conditions for the USV which are converging toward a stable cycle, and some positions which are not. This naturally leads to the characterization of the set of initial state for the USV that results in the realization of stable cycles. This set of position is the capture basin [9, 10]. Once the robot reaches the capture basin, it remains forever trapped.

Remark. *The capture basin of a predefined cycle is related to the shape of the cycle and the topology of the seabed in the case of the USV scanning the seafloor using an echosounder.*

3. Formalism

3.1. Cycle navigation

Consider a dynamic system of state $\mathbf{x}$ and of input $\mathbf{u}$ following Equation (1).

[1] Video of the experiments at the Guerlédan lake (France) `https://www.youtube.com/watch?v=MDJ6iHYhxyM`

[2] `https://bluerobotics.com`

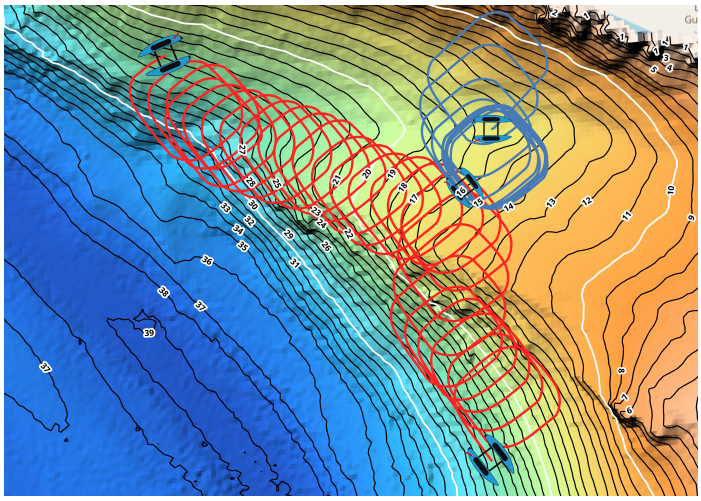


Figure 2: Two trials at the Guerlédan lake of navigation of an USV using cycles. The blue trajectory converges on a stable cycle, while the red trajectory does not.

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{1}$$

Suppose a timed automaton [11] is generating inputs $\mathbf{u}$ for the system as defined in [8]. Each state of the timed automaton is associated to an input $\mathbf{u}$ for the system, and transitions between states are triggered when the clock of the automaton exceeds a reference duration. Duration of some transition can be adjusted using the input parameter of the automaton $\boldsymbol{\omega}$. Figure 3 shows the block diagram of the system controlled by the timed automaton.

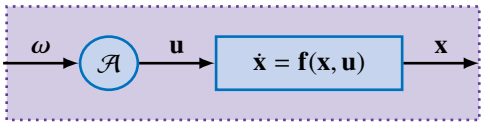


Figure 3: Block diagram of the system controlled by a timed automaton

Previous work has established the abstraction of cycles [8]. This couple of timed automaton and system could be considered as a new system to be controlled, which has the particularity to be discrete. The state of the system $\boldsymbol{\eta}_k$ is now the pose of the robot at the beginning of the k_{th} iteration of the timed automaton, and input is $\boldsymbol{\omega}_k$. Figure 4 summarizes the abstract cycle formalism.

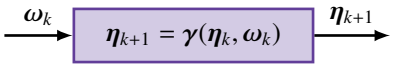


Figure 4: Block diagram of the abstracted cycle

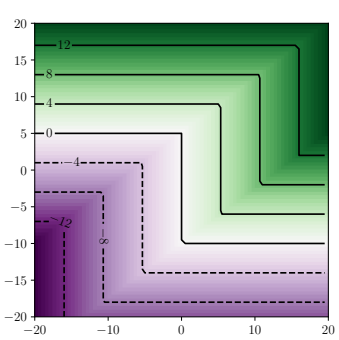
At each iteration of the timed automaton, an input $\boldsymbol{\omega}_k$ is provided. This input will change the duration of the transitions in the timed automaton, and therefore the starting position of the next cycle $\boldsymbol{\omega}_k$.

3.2. Simulated seafloor

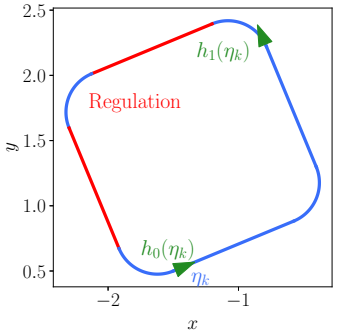
A simulated seafloor representing the trial environment is proposed and shown in Figure 5a. This simulated seafloor represents the measurement equation $\boldsymbol{\sigma}(\boldsymbol{\eta}_k, \boldsymbol{\omega}_k)$ for the studied system as defined in Equation (2).

$$\boldsymbol{\mu}_k = \boldsymbol{\sigma}(\boldsymbol{\eta}_k, \boldsymbol{\omega}_k) \tag{2}$$

At each position of the environment is associated a simulated echosounder measurement. Along the cycle, the robot is taking two measurements $\boldsymbol{\mu} = \begin{bmatrix}\mu_0 & \mu_1\end{bmatrix}^T$ as shown in Figure 5b. μ_0 is taken at the beginning of the timed automaton iteration, at the position $\mathbf{h}_0(\boldsymbol{\eta}_k)$, and μ_1 is taken at the end of the second straight line, at the position $\mathbf{h}_1(\boldsymbol{\eta}_k)$. Measurement positions on the cycle have been chosen to decouple measurements, and regulation segments to avoid redundancy.



(a) Simulated seafloor



(b) Measurement positions and regulated sections

3.3. Cycle stabilization

The system is regulated towards a reference bathymetric measurement $\bar{\boldsymbol{\mu}}$. The error between the measured depths $\boldsymbol{\mu}$ during one cycle and the reference is $\boldsymbol{e} = \bar{\boldsymbol{\mu}} - \boldsymbol{\mu}$. Then, a controller is designed to calculate the required cycle inputs $\boldsymbol{\omega_k}$ stabilizing the cycle on the reference.

During these trials, a Sliding Mode Controller is used [6, 7]. In order to avoid any chattering effect, the usual signum function is replaced with a hyperbolic tangent function at the cost of non-finite time convergence as defined in Equation (3).

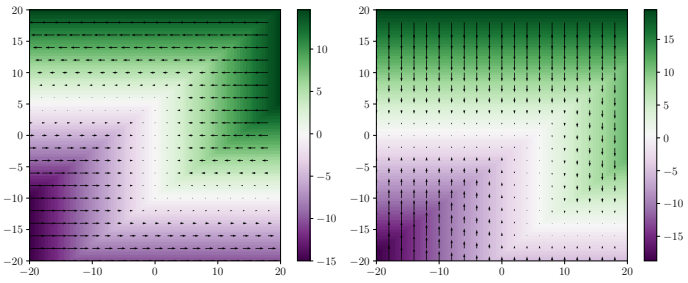
$$\boldsymbol{\omega} = \mathbf{K} \cdot \tanh\left(\frac{\mathbf{e}}{r}\right) \tag{3}$$

Where K is a diagonal control gain matrix and r corresponds to the width of the damping area around the sliding surface. The gain parameters of K are tuned for each bathymetric map. Notably, the sign of these parameters are chosen to adapt to the local depth gradient.

Note that a Sliding Mode Control scheme has been chosen in this work for a better control of the behavior of the cycle between iterations. Notably, the sliding mode approach guarantees a maximum deformation of the cycle between two iterations while maintaining satisfying convergence time. Alternatively, a Proportional (or Proportional Integral) scheme could be chosen. The behavior would be different as, for large depth

error, the cycle would be greatly deformed. This can lead to issues, because a large cycle distortion could shift the cycle so that it leaves the capture basin of the stable cycle.

In this way, over a number of iterations, the initial position of the cycles will move in the plane, and the cycle position follows the vector field shown in Figure 6. Duration ω_0 influences cycle position along x-axis, while duration ω_1 influences position along the y-axis.



(a) Vector field associated to $\boldsymbol{\mu} = \begin{bmatrix}\mu_0 & 0\end{bmatrix}^T$ (b) Vector field associated to $\boldsymbol{\mu} = \begin{bmatrix}0 & \mu_1\end{bmatrix}^T$

Figure 6: Vector field of the displacement over iterations of the cycle

Finally, Figure 7 shows the global vector field followed by the cycle over iterations, which is the sum of vector fields associated to μ_0 and μ_1 measurements shown in Figure 6. The magnitude of the error is represented by the background color, and the displacement applied to the cycle at each iteration by the vector field.

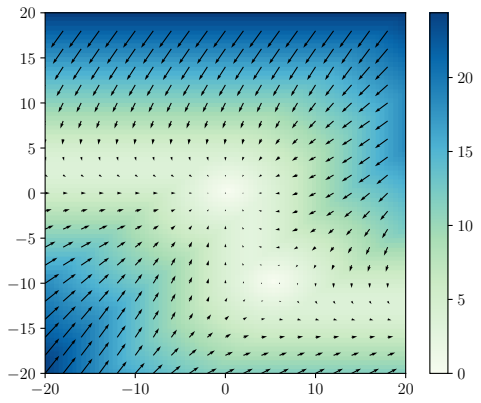


Figure 7: Vector field for the cycle displacement

Figure 7 shows that the vector field seems to have an attractive point around $\boldsymbol{\eta} = \begin{bmatrix}0 & 0\end{bmatrix}$. This is consistent with experimental observations. This hypothesis must now be verified using tools to prove the system's stability.

Remark. *Another equilibrium point seems to be at* $\boldsymbol{\eta} = \begin{bmatrix}5 & -10\end{bmatrix}$, *but this equilibrium seems to be a hyperbolic foxed point as the vector field induce no displacements only when* $\boldsymbol{\eta} = \begin{bmatrix}5 & -10\end{bmatrix}$, *and the vector field tends to move the surrounding points away from each other [12].*

3.4. Discrete inclusion problem

The echosounder measurements μ_i is supposed to belong to the interval $[\mu_i]$, which is modelling the uncertainty associated to the measurement [13, 14]. This uncertainty comes from the sensor accuracy, and from the bathymetric map production methods, which requires a discretization of the mapped area and another sensor with an associated accuracy. Then, the computed error is also an interval $e_i \in [e_i]$ and the vector field is uncertain.

The problem is then model as a discrete inclusion problem and meet Equation (4) [15], as it is possible to have an expression for $\boldsymbol{\gamma}$, but this one is uncertain, and then the real state $\boldsymbol{\eta_{k+1}}$ belongs to $[\boldsymbol{\gamma}](\boldsymbol{\eta}_k)$

$$\boldsymbol{\eta}_{k+1} \in [\boldsymbol{\gamma}](\boldsymbol{\eta}_k) \tag{4}$$

4. Stability analysis

4.1. Set theory stability analysis

Analysis of the cycle stability is decomposed in two stages. First, a positively invariant set must be found [16, 10], i.e. a set $\mathbb{X}_k$ such that the application of the vector field on this set gives a set $\mathbb{X}_{k+1}$ included in the initial set $\mathbb{X}_k$. Thus, this set $\mathbb{X}_k$ represents a stable set of states by application of the vector field, and as soon as the system enters this set, it is captured forever. Then, from this positively invariant set, we need to characterize the capture basin [16, 10] of the dynamic system defined by the vector field, by adding to the positively invariant set the set of points that enter it through the action of the vector field. Thus, as soon as a point enters the capture basin, it is directed into the positively invariant set and remains captured forever.

4.2. Lattice structure of capture sets

A lattice $(\mathcal{E}, \leq)$ is an algebraic structure, which consist of a partially ordered set in which each element has a least upper bound, also called a join, and a greatest lower bound, also called a meet.

Example. $(\mathbb{R}^n, \leq)$ *is a lattice with respect to*

$$\mathbf{x} \leq \mathbf{y}, \Leftrightarrow \forall i \in [\![0, n]\!], x_i \leq y_i$$

Capture sets $(\mathbb{C}, \subseteq)$ have a lattice structure. The join of two capture sets $\mathbb{C}_1$ and $\mathbb{C}_2$ is the intersection $\mathbb{C}_1 \cap \mathbb{C}_2$. The meet of two capture sets $\mathbb{C}_1$ and $\mathbb{C}_2$ is the union $\mathbb{C}_1 \cup \mathbb{C}_2$.

This lattice structure allows to express the smallest and largest elements of capture sets.

Example. *The smallest element of the lattice* $(\mathbb{R}^n, \leq)$ *is the empty set* $\emptyset$, *and the largest is* $\mathbb{R}^n$.

Capture sets can be either positively invariant sets or capture basins. Both these tools will be explained in the next subsections.

4.3. Positive invariant set

The positive invariant set $\mathbb{P}$ is a set stable by application of the evolution equation of the dynamical system, that is $[\boldsymbol{\gamma}](\mathbb{P}) \subseteq \mathbb{P}$ [16, 10, 14].

Positive invariant set has a lattice structure.

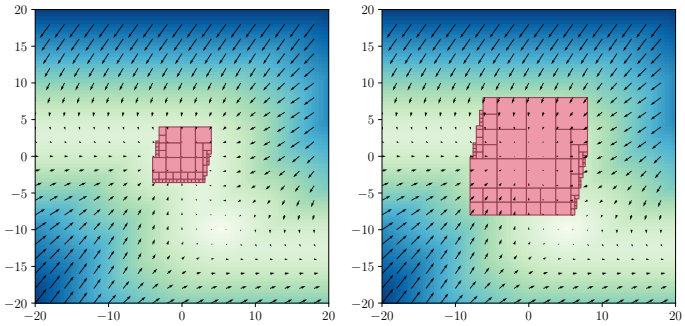
Remark. *The biggest positive invariant set is $\mathbb{R}^n$, and the smallest one is $\emptyset$, because regardless the evolution function $\mathbf{f}$, $\mathbf{f}(\mathbb{R}^n) \subset \mathbb{R}^n$, and $\mathbf{f}(\emptyset) = \emptyset \subset \emptyset$*

A way to compute an inner approximation of this set is to build a sequence of sets $\mathbb{P}_k$ which will converge towards $\mathbb{P}$. $\mathbb{P}_0$ is initialized by a box of the state space around a supposed stable state, and $\mathbb{P}_{k+1}$ is computed as the intersection of $\mathbb{P}_k$ and $[\boldsymbol{\gamma}](\mathbb{P}_k)$. Therefore, each state which belongs to $\mathbb{P}_k$ and which is moved out of this set by the application of the system dynamics is removed of the solution for $\mathbb{P}_{k+1}$. Thus, the set $\mathbb{P}_k$ is iteratively contracted.

$$\begin{cases} \mathbb{P}_0 = \mathbb{P}_0 \\ \mathbb{P}_{k+1} = \mathbb{P}_k \cap [\boldsymbol{\gamma}](\mathbb{P}_k) \end{cases} \tag{5}$$

Then, $\exists n \in \mathbb{N}, \forall k \geq n, [\boldsymbol{\gamma}](\mathbb{P}_k) \subseteq \mathbb{P}_k$. This index n is the minimal index to reach the fix-point, and after this iteration $\mathbb{P}_k$ is positively invariant. $\mathbb{P}_n$ is an inner approximation of a capture basin associated to the starting set $\mathbb{P}_0$.

Figure 8 shows in pink the positive invariant set for the cycle navigation, with two different starting sets. For the example shown in Figure 8a, the starting set $\mathbb{P}_0$ is the box $[-4, 4] \times [-4, 4]$, while it is the box $[-8, 8] \times [-8, 8]$ for the example shown in Figure 8b. It is noticeable that this positive invariant set $\mathbb{P}$ follows the vector field describing the system dynamics shown in Figure 7.



(a) $\mathbb{P}_0 = [-4, 4] \times [-4, 4]$ (b) $\mathbb{P}_0 = [-8, 8] \times [-8, 8]$

Figure 8: Inner approximation of the positive invariant set

The computed set $\mathbb{P}$ is the inner approximation of the largest positive invariant of the starting set $\mathbb{P}_0$. Computing the outer approximation of this set is computationally demanding, as set of state out of $\mathbb{P}$ should be proved to never enter this set.

4.4. Capture basin

The capture basin is a set from which the system cannot escape once it has reached it. This capture basin $\mathbb{B}$ is defined by Equation (6) [16, 10, 14].

$$\mathbb{B} = \{x \in \mathbb{B}_0 \mid \exists t_0 \in \mathbb{R}_+, \forall t \geq t_0, x(t) \in \mathbb{B}\} \tag{6}$$

As $\mathbb{P}$ is positive invariant, then it is already a capture basin [14, 16]. Then this set $\mathbb{P}_n$ will be the starting set to compute the capture basin $\mathbb{B}$ for the dynamical system.

Using the lattice structure, if $\mathbb{B}$ is a capture basin for a dynamical system governed by an evolution function $\mathbf{f}$, then $\forall k \in \mathbb{N}, \mathbf{f}^{-k}(\mathbb{B})$ is also a capture basin. It comes from the fact that $\forall \mathbf{x} \in \mathcal{S}, \mathbf{f}^k(\mathbf{x}) \in \mathbb{B}$.

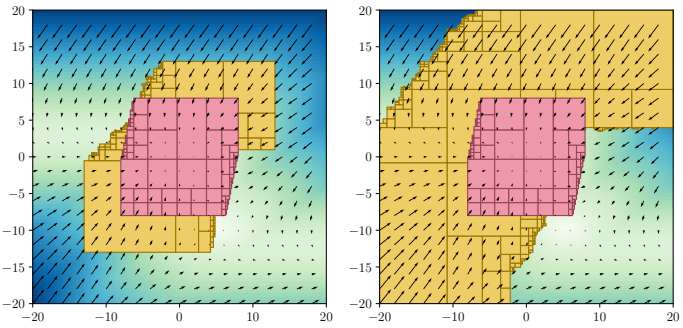
It is then possible to expand a capture basin by using the reciprocal of the evolution function $[\boldsymbol{\gamma}]$ iteratively on a first identified capture basin. By starting from the identified positive invariant set $\mathbb{P}$, it is then possible to compute an inner approximation of a capture basin for the stable cycles by computing the sequence presented in Equation (7).

$$\begin{cases} \mathbb{B}_0 = \mathbb{P}_n \\ \mathbb{B}_{k+1} = \mathbb{B}_k \cup [\boldsymbol{\gamma}]^{-1}(\mathbb{B}_k) \end{cases} \tag{7}$$

The more iterations are performed to determine the basin of attraction, the larger the basin of attraction becomes. This is because, at each iteration, we add the set of points that enter the basin of attraction by the application of the evolution function $[\boldsymbol{\gamma}]$.

Remark. *Unlike the computation of the positively invariant set $\mathbb{P}$, where the computation could only be stopped when the condition $\mathbb{P}_k \subset [\boldsymbol{\gamma}](\mathbb{P}_k)$ was met, the characterization of the capture basin can be stopped whenever desired. The more iterations are performed, the larger is the characterized area, but stopping computations early does not alter the guarantee of the results.*

Figure 9 shows an example of the computed capture basin after $n_1 = 5$ and $n_2 = 20$ iterations from the previously computed positive invariant set $\mathbb{P}$ shown in Figure 8b. States added iteratively to the capture basin is shown in yellow, and the starting positive invariant set is shown in pink. States entering the yellow area will be iteratively displaced in the pink area. Once the system state has reached the pink set, the system state will remain capture forever.



(a) 5 iterations of the vector field (b) 20 iterations of the vector field

Figure 9: Capture basin computation

There is obviously points outside the computed capture basin which could be added to it by applying more iterations of the

vector field. However, each iteration is time-consuming and while the set $\mathbb{B}_k$ inflates, new states could be added at the next iteration.

5. Conclusion

In conclusion, this work analyzes the stability of cycles using set methods, which allows us to obtain guaranteed results. This proves the stability of a discrete system governed by a condition of discrete inclusion.

Firstly, the identification of a positively invariant set for our system has proved the existence of stable cycles. This corroborates the experimental observations made with the BlueBoat on the Guerlédan Lake, with the fact that it is possible to stabilize a robot in a GNSS-denied environment using stable cycles.

Then, by characterizing the capture basin, we were able to determine the set of starting points for the mission, such that the cycle would converge and stabilize around the equilibrium position. This is also the experimental observation shown in Figure 2, where some starting pose for the cycle will lead to a convergence to a stable cycle, and some other will slide and not reach the stability.

The drawback of the presented methods is the difficulty to characterize the set of starting position which will not be stable. Actually, only an inner approximation of the positive invariant set enclosing the equilibrium point and the capture basin are computed. The computation of the outer approximation requires guaranteeing that starting poses for the robot will never reach the capture basin. Therefore, the proposed method does not attempt to characterize the outer approximation.

Finally, this method was proposed on a USV to validate the concept of navigation by stable cycles, and to record GNSS signals in order to have ground truth available. The validation of these methods, and the proof of stability demonstrated in this paper, enable cycles to be used in a GNSS-denied environment. This means that these navigation methods can be applied to underwater robotics, enabling robots to navigate without localization solutions and without getting lost.

References

[1] F. Maurelli, S. Krupiński, X. Xiang, Y. Petillot, Auv localisation: a review of passive and active techniques, International Journal of Intelligent Robotics and Applications 6 (2) (2022) 246–269. doi:10.1007/s41315-021-00215-x.

[2] L. Lionel, Underwater robots part i: Current systems and problem pose, in: A. Lazinica (Ed.), Mobile Robots, IntechOpen, Rijeka, 2006, Ch. 16. doi:10.5772/4697.
URL https://doi.org/10.5772/4697

[3] P. Baccou, B. Jouvencel, V. Creuze, Single beacon acoustic for auv navigation, 2001.

[4] S. Durand, B. Boisseau, N. Marchand, J.-F. Guerrero-Castellanos, Event-Based PID Control: Application to a Mini Quadrotor Helicopter, Journal of Control Engineering and Applied Informatics 20 (1) (2018) 36–47.
URL https://hal.science/hal-01722845

[5] C. L. Phillips, J. M. Parr, E. A. Riskin, T. Prabhakar, Signals, systems, and transforms, Prentice Hall Upper Saddle River, 2003.

[6] H. K. Khalil, Nonlinear systems.

[7] J.-J. E. Slotine, W. Li, Nonlinear applied control, Li, W., Ed (1991).

[8] Q. Brateau, F. Le Bars, L. Jaulin, Navigation without localization using stable cycles, preprint submitted to IEEE ICRA 2025 (Sep. 2024).
URL https://hal.science/hal-04846273

[9] J.-P. Aubin, A survey of viability theory, SIAM Journal on Control and Optimization 28 (4) (1990) 749–788. arXiv:https://doi.org/10.1137/0328044, doi:10.1137/0328044.
URL https://doi.org/10.1137/0328044

[10] P. Saint-Pierre, Approximation of the viability kernel, Appl. Math. Optim. 29 (2) (1994) 187209. doi:10.1007/BF01204182.
URL https://doi.org/10.1007/BF01204182

[11] R. Alur, D. L. Dill, A theory of timed automata, Theoretical computer science 126 (2) (1994) 183–235.

[12] L. Perko, Differential equations and dynamical systems, Vol. 7, Springer Science & Business Media, 2013.

[13] R. E. Moore, R. B. Kearfott, M. J. Cloud, Introduction to Interval Analysis, Society for Industrial and Applied Mathematics, 2009. doi:10.1137/1.9780898717716.
URL https://epubs.siam.org/doi/abs/10.1137/1.9780898717716

[14] F. Blanchini, S. Miani, et al., Set-theoretic methods in control, Vol. 78, Springer, 2008.

[15] V. M. Veliov, Approximations to differential inclusions by discrete inclusions, 1989.
URL https://api.semanticscholar.org/CorpusID:124067217

[16] J.-P. Aubin, A. M. Bayen, P. Saint-Pierre, Viability theory: new directions, Springer Science & Business Media, 2011.