The barycenter-based source seeking [⋆]

Bruno Peixoto [*] **Felipe Pait** [*] **Bruno Angélico** [*]

[*] *Escola Politécnica da Universidade de São Paulo, SP, (e-mail: brunolnetto@gmail.com, pait@usp.br, angelico@usp.br).*

Abstract: This paper explores an application of a quantum theory-inspired derivative-free optimization method on the search of a scalar signal extremum in robotics. The method, which is an alternative to current extremum-seeking techniques, requires local signal readings acquired along with environment exploration. A control function $u(t, x)$ alters the dynamical system trajectory $x(t)$ towards the curve $\hat{\gamma}(t)$ between current tracking coordinates $h_s(x)$ and an estimate of a function's extremum point $\hat{y}_s$. This estimation is updated frequently according to sample data. A numerical simulation validates the proposed method.

Resumo: Este trabalho explora uma aplicação de um método de otimização inspirado na física quântica e livre de derivadas, na busca do extremo de um sinal escalar e aplicado à robótica. O método, o qual é uma alternativa a técnicas atuais de busca extremal, requer leituras locais adquiridas ao longo da exploração do caminho. A função de controle $u(t,\, x)$ altera a trajetória do sistema $x(t)$, dada coordenada inicial x_0 em direção à curva $\hat{\gamma}(t)$ que conecta a coordenada de rastreamento atual $h_s(x)$ e a estimativa do ponto crítico do sinal escalar $\hat{y}_s$. Por fim, uma simulação numérica valida o algoritmo proposto.

Keywords: source seeking, barycenter, optimization, control, dynamics
Palavras-chaves: Busca de fonte, baricentro, otimização, controle, dinâmica

1. INTRODUCTION

The expression *extremum seeking* is a synonym to *optimization.* The term *extremum* refers to either a minimum or maximum coordinate of a function. In the context of control theory, extremum-seeking control is a form of adaptive control without prior knowledge of the mathematical characteristics of the goal function under investigation, other than it having an extremum, as briefly explained on Sastry (2013). A common optimization constraint is that the function is evaluated only on the current coordinate, without knowledge of its derivative. Although the numerical computation of derivatives is possible, a noise-corrupted signal is prone to noise amplification, as discussed on Chapter 1, p. 17, by Sontag (2013).

Åström and Wittenmark (2013) describe the method as the addition of a periodic time-varying signal to the input of the nonlinear dynamic system. Depending on the correlation between the input and output, it is possible to determine the direction toward the extremum. Tan et al. (2010) describes historically the main contributions on the field for both static, i.e., optimization statement, and also adaptive control. Cochran and Krstic (2009) present an application example for the method. It estimates the scalar signal gradient using sinusoidal additive terms on dynamical system input functions. The introduced disturbance modifies the original path so that on average the trajectories tend to follow the trajectory of the goal function's gradient.

In the context of control theory, the search for an extremum in space occurs by a dynamical system whose input function u is weighted by distribution $b(x)$. The resulting summation term alters its vector field $a(x)$ to converge its trajectories, given an initial state x_0. The source seeking agent representation is an affine-form dynamical system $a+bu$. Here and in the computations that follow the x indicating the dependence of the local coordinate is omitted if there is no ambiguity.

In a real-world application, features like noise, terrain unevenness, and obstacles may preclude the extremum position estimation. These are relevant considerations, although not under investigation in the current article for the sake of simplicity.

The main contribution of the current paper is the application of the method first presented at Pait (2018) in substitution to conventional input-output correlation and filtration presented in Tan et al. (2010). The barycenter method can be used for derivative-free optimization of a noise-corrupted function. Its name is inspired by the geometric definition of the barycenter. Its use provides a solution

[⋆] The first author thanks Brenno Lobo, Edimar Peixoto, and Jaqueline Lobo Peixoto for their support during the article writing.

to the problem of source seeking in robotics, discussed in the references above. In comparison to a sinusoidal application of the extremum seeking, an advantage of proposed methods robustness against noise.

In this article, the control law u comes from the *exact linearization algorithm*, presented on Sontag (2013) and Sastry (2013) textbooks, for instance. The resulting input function converges the integral trajectories of the vector field toward the path in direction to estimation $\hat{y}_s$. The states x and input u are respectively in space $\mathcal{X}$ and $\mathcal{U}$, Euclidean in current work. Given hypothesis, the reader may refer to Goldstein et al. (2002) and Pait (2018) for explanations about the barycenter method and classical mechanics.

This paper is organized as follows. Sections 2, 3 and 4 present the building blocks for the proposed solution: the barycenter method, trajectory synthesis method, the seeking agents description, and the control law. The assembly on a flow diagram is in Section 5: it brings the building blocks together in an algorithm for embedded controller programming, depicted in a flow diagram. Section 6 describes briefly the simulation scenario and provides references regarding its results on images. Finally, Section 7 concludes the article with its main considerations and suggestions for further improvements on the barycenter-based source seeking.

2. THE BARYCENTER METHOD SUMMARY

The *barycenter method* belongs to the class of *direct optimization* methods. Such methods require local readings of an objective function, but not any knowledge of its derivatives or functional form. In special, the exponential of this function value weights the sum of respective coordinates. Its calculation represents the weighted average formula. The rationale relies on the exponential function mapping slope: the use of negative exponent assigns a value near zero for high evaluations and vice-versa.

The barycenter formula follows for the batch calculation on equation (1). The parameters ν, function $f(\cdot)$ as well as coordinate points x_n comprehend particular meaning on the method: the variable $\nu \in \mathbb{R}$ intensifies the exponential term: in case of a minimization problem, $\nu > 0$; in opposite, i.e., maximization, it must be $\nu < 0$; function continuity $f(\cdot)$ is rather prefered than discontinuity. However, either noise or discontinuities are allowed as function evaluations. Finally, the coordinates x_n represent discrete control points.

$$\boxed{\hat{x}_n = \frac{\sum\limits_{i=0}^{n} x_i \, e^{-\nu\, f(x_i)}}{\sum\limits_{i=0}^{n} e^{-\nu\, f(x_i)}}} \tag{1}$$

Besides the batch version above, there exists a recursive version of the barycenter method on the work of Pait (2018). The rationale for the method is as follows.

The sum of the weighted distance from estimation point $\hat{x}_n$ to evaluation points x_n by the exponential of negative function value corresponds to a convex and positive objective function concerning the barycenter $\hat{x}_n$. Hence, it allows conventional algebraic optimization techniques. In this case, the distance function minimum occurs on coordinates with zero gradient value. The multiplicative function $F_n(\cdot)$ stands for the expression $\frac{e^{-\nu\, f(x_n)}}{m_{n-1}+e^{-\nu\, f(x_n)}}$. The deterministic barycenter update below presents a similar equation as gradient-based algorithms like quasi-Newton or conjugate descent methods. The probabilistic recursive version of the method justifies this similarity.

$$\boxed{\begin{cases} m_n = m_{n-1} + e^{-\nu\, f(x_n)} \\ \hat{x}_n = \hat{x}_{n-1} + F_n(x_n)\,(x_n - \hat{x}_{n-1}) \end{cases}} \tag{2}$$

The recursive version for the barycenter method presents the difference $x_n - \hat{x}_{n-1}$. It corresponds to the step interval between the current point and the previous barycenter estimation. Based on the central limit theorem, a natural difference choice is a random variable z_n. Among others, figures the Gaussian support distribution with mean value $\bar{z}_n$ and covariance matrix Σ.

The expected value $\mathbb{E}[\Delta\hat{x}_n]$ for the previous statement is presented for clarification below. Here, the subscript indicating the sample ordinality n is omitted. On average, the algorithm steps $\Delta\hat{x}_n$ acquires gradient-like behavior, pertinent to the purpose of source seeking. Pait (2018) provides the main results for the previous statement. The term $\bar{F}_n(\cdot)$ stands for the expression $\frac{m_{n-1}}{m_{n-1}+e^{-\nu\, f(\hat{x}_{n-1}+z)}}\, F_n$, so that $\frac{\partial F}{\partial z} = -\nu\, \bar{F}\, \frac{\partial f}{\partial z}$.

$$\boxed{\mathbb{E}[\Delta\hat{x}_n] = \mathbb{E}[F_n]\bar{z} - \nu\, \Sigma\, \mathbb{E}\left[\bar{F}_n\, \nabla f(\hat{x}_{n-1}+z)\right]}$$

An estimation for the scalar signal source provides information to further explore the space. During space exploration, samples of the scalar signal allow the source estimation improvement. The signal samples occur dependent on the current coordinates of the dynamical system under consideration. The next section describes briefly both general and physical dynamical systems commonly applied in control theory. The main concern of a control theorist is dynamical systems stability, given by the judicious choice of input function u.

3. THE AGENTS' DYNAMICS AND CONTROL FORMULATION

Control theory describes the mathematical area related to the description of a dynamical system given a geometry. From the robotics perspective, mechanical systems are recurrent applications for both modeling and control synthesis. Hence, the class of *mobile robots* is a suitable choice for seeking agents.

According to Lagrange's mechanics interpretation, available on Goldstein et al. (2002), coordinates of a physical system regards the product space of their coordinates q and speeds p. Wheeled robots rely on the rolling wheel movement to perform their motion. They belong to the *nonholonomic class* of dynamical systems, by addition of a constraint contribution to Lagrangian representation of a geodesic.

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} + \frac{\partial R}{\partial \dot{q}} = A^{\intercal}\,\lambda + F_q \tag{3}$$

Physically, the rolling direction corresponds to a motion constraint, dependent on coordinate derivatives $\dot{q}$ and with implicit expression $\psi(q,\,\dot{q}) = 0$. A recurrent constraint description receives the name after the mathematician Pfaff, with form $A(q)\,\dot{q} = 0$ and available on Frobenius (1877). The general solution for the Pfaffian form corresponds to $\dot{q} = B\,p$, with $A\,B = 0$ valid for every $q \in \mathcal{Q}$ and $p \in \mathcal{P}$. Below is the nonlinear equation of resulting dynamical equation. The dependency to both coordinates q and p are ommited for a light notation.

$$\frac{d}{dt}\begin{bmatrix} q \\ p \end{bmatrix} = \begin{bmatrix} B\,p \\ -H^{-1}h \end{bmatrix} + \begin{bmatrix} 0_{n_q \times n_u} \\ H^{-1}Z \end{bmatrix} u \tag{4}$$

Mechanical systems kinematics and dynamics correspond to nonlinear analytical differential equations (i.e. algebraic vector fields dependent on their position). The control law synthesis u modifies the system behavior and allows trajectory tracking and noise suppression. The former term corresponds to stability according to Lyapunov's criterium. The authors Pait and Colón (2006) explore analytical dynamical systems $\dot{x} = a(x)$ on curved spaces, namely *Riemannian manifolds*.

A common control synthesis strategy for mechanical dynamical systems corresponds to the *exact linearization* algorithm. Nonholonomic physical systems require a particular extension of the aforementioned algorithm, namely the *extended exact linearization* algorithm. The resulting control law follows on equation (5).

$$\begin{aligned} \dot{z} &= a_z(x,\,z) + b_z(x,\,z)\,w(t,\,x,\,z) \\ u(t,x,\,z) &= \alpha(x,\,z) + \beta(x,\,z)w(t,\,x,\,z) \end{aligned} \tag{5}$$

The first expression represents added integrators' vector field and the second, the compensation of non-linear terms. Its construction version is available on Sastry (2013) for the kinematic case of a Dubins (1957)'s car. The synthesis algorithm results on a compensator control law according to equation (5), given in general for the sake of brevity. The new control function $w(x,\,z)$ asymptotically leads the distance between tracking trajectory and actual position to zero.

4. SOURCE SEEKING TRAJECTORIES

The trajectory design relies on the approximation statement subject to boundary conditions. Although not necessary, curve continuity is convenient for the task. It is sufficient to compose the trajectory interval $[s_0,\,s_1]$ by the disjoint union of real intervals I_i, for finite real numbers s_0 to s_1. It follows the next two possible explicit options.

The simplest interpolation case of two control points P_0 and P_1 in Euclidean space corresponds to a line. The curve image $P(s)$, for $s \in [s_0,\,s_1]$, corresponds to equation (6).

$$\begin{cases} \begin{bmatrix} P \\ \theta \end{bmatrix}(s) = \begin{bmatrix} P_0 + \frac{P_1 - P_0}{s_1 - s_0}\,s \\ \arg\left(P_1 - P_0\right) \end{bmatrix} \\ P^{(n_d)}(s) = \begin{cases} \frac{P_1 - P_0}{s_1 - s_0} & ,\, n_d = 1 \\ 0 & ,\, n_d > 1 \end{cases} \\ \theta^{(n_d)}(s) = 0,\; n_d > 1 \end{cases} \tag{6}$$

Among nonzero curvature possibilities, it figures the *spline* on Hall and Meyer (1976), the *clothoid* of Vázquez-Méndez and Casal (2016) and the *Bézier curve*, available on Choi et al. (2008). The particular property of these curves refers to point approximation, useful on the obstacle avoidance task.

In the seeking context, the path binds current point and the estimation position for the scalar signal extremum in $\mathcal{X}$ space, immersed on the product space $\mathcal{Q} \times \mathcal{P}$. The dynamical system $\dot{x} = a + b\,u$ requires the design of an input function u in the sense to alter, if possible, its vector field distribution $f(x,\cdot)$. The anew distribution $f(x, u(s,x))$ must, for every trajectory $\phi(s,x_0)$, $s \in \mathbb{R}_{\geq 0}$, $x \in \mathcal{X}$, converge to the developed trajectory $\hat{\gamma}$.

5. THE BARYCENTER-BASED SOURCE SEEKING FLOWCHART

The previous three sections describe the schema building blocks on Fig. 1, brought together in a flow diagram. It corresponds to the visual representation of the barycenter-based source seeking algorithm. The row vector $\begin{bmatrix}\hat{x}_{i,\,j} & \tau_i & \tau_s\end{bmatrix}(\tau)$ represents the variable $p(\tau)$: the source estimation $\hat{x}_{i,\,j}(\tau)$ on coordinate τ, for the subscripts i and j to exploration and sample steps of interval $\Delta\tau_s$ and $\Delta\tau_e$, respectively; the coordinate τ_i initializes each i^{th} interval with amplitude $\Delta\tau = \tau_{i+1} - \tau_i$, such that the coordinates τ_i belongs to an ordered set $\{\tau_i\}_{i \in \mathbb{N}_{\geq 0}}$; and the sample coordinate $\tau_s(\tau)$ within $[\tau_i,\,\tau_{i+1}]$ such that $\tau_s(\tau) = n_e(\tau)\,\Delta\tau_e + n_s(\tau)\,\Delta\tau_s$, for the exploration and sample discrete steps respective to functions $n_e(\tau) = \left\lfloor \frac{\tau}{\Delta\tau_e} \right\rfloor$ and $n_s(\tau) = \left\lfloor \frac{\tau}{\Delta\tau_s} - n_e(\tau)\,\frac{\Delta\tau_e}{\Delta\tau_s} \right\rfloor$. The function $\lfloor\cdot\rfloor$ truncates the number to the closest integer from below.

The variable τ initializes with a non-negative value τ_0, commonly adopted as 0. The initial estimation $\hat{x}_{00}(\tau_0)$ may be the current coordinate $x(0)$ added to a stochastic variable z in the absence of a better guess. The coordinates τ_i and τ_s are both initially τ_0. Thus, a possible initialization vector to the tuple is $p_0 = \begin{bmatrix} x_0 + z & \tau_0 & \tau_0 \end{bmatrix}$. A stochastic innovation term aforementioned provides the exploration property to the barycenter method. Its addition to commands allows vicinity exploration. Other possible additions correspond to control as $u + z_u$, along the trajectory as $\hat{\gamma} + z_\gamma$ and/or to destiny point as $\hat{x}_{i,\,j} + z_{\hat{x}}$. Intermediate points added randomly in a vicinity to the nominal trajectory at coordinates τ_z, $\tau_z \in [\tau_i,\, \tau_{i+1}]$ are also an exploration option.

In the flow diagram, each block describes an application-suitable algorithm. The orange block initializes variables i, j, k, τ_s and $\hat{x}_{i,\,j}$ as described on previous paragraphs. The yellow blocks embed decision-making commands regarding time and source of the scalar signal. The cyan block computes the trajectory $\hat{\gamma}(\tau)$ between current coordinates and estimation of extremum point given interval $\Delta\tau_e$. The green block represents the computation and application of the control signal u based on states $x(\tau)$ and trajectory $\hat{\gamma}(\tau)$.

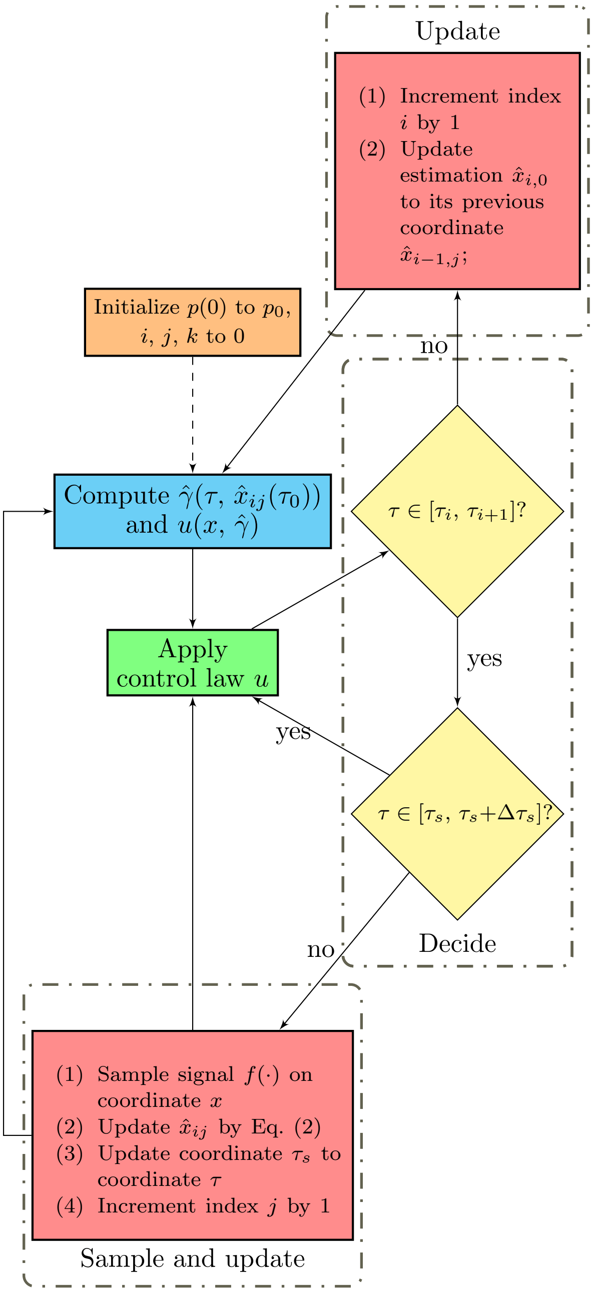

Update
(1) Increment index i by 1
(2) Update estimation $\hat{x}_{i,0}$ to its previous coordinate $\hat{x}_{i-1,j}$;
Initialize $p(0)$ to p_0, i, j, k to 0
Compute $\hat{\gamma}(\tau,\, \hat{x}_{ij}(\tau_0))$ and $u(x,\, \hat{\gamma})$
Apply control law u
no
$\tau \in [\tau_i,\, \tau_{i+1}]$?
yes
$\tau \in [\tau_s,\, \tau_s{+}\Delta\tau_s]$?
yes
no
Decide
(1) Sample signal $f(\cdot)$ on coordinate x
(2) Update $\hat{x}_{ij}$ by Eq. (2)
(3) Update coordinate τ_s to coordinate τ
(4) Increment index j by 1
Sample and update

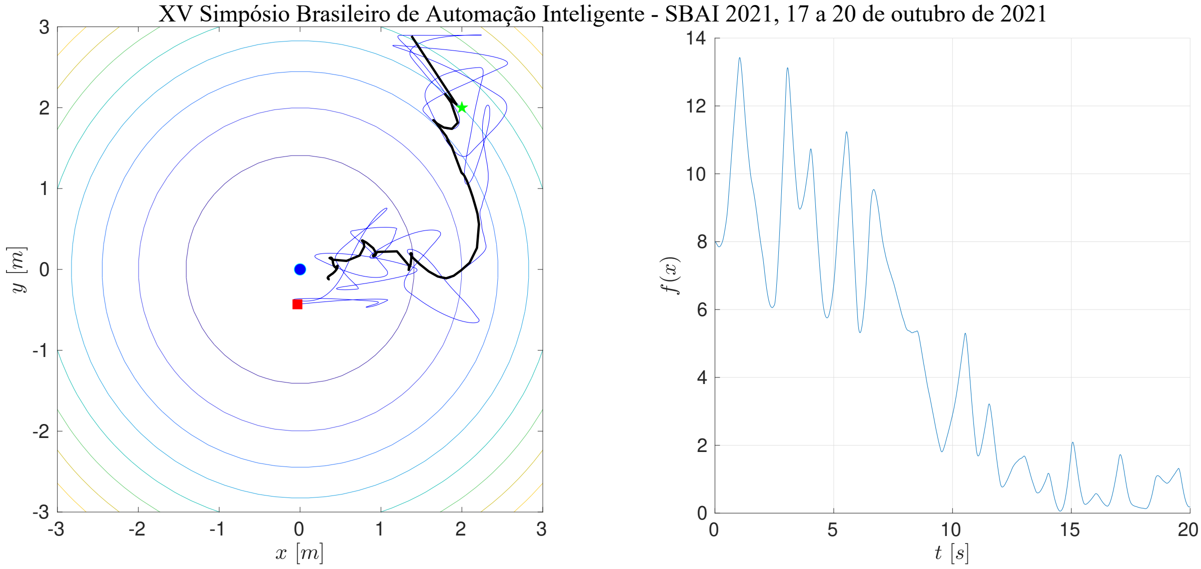
Figure 1. Barycenter-based extremum seeking flow diagram.

Algorithm 1 Pseudocode implementation of flow diagram on Fig. 1

```
Input:  • Initial coordinate $\tau_0$;
        • Trajectory function $\hat{\gamma}(\cdot,\, \cdot)$;
        • Input function $u(\cdot,\, \cdot)$;
        • Signal sampling interval $\Delta\tau_s$;
        • Estimation interval $\Delta\tau_e$.
Output: None
  Assign functions $i(\tau_1)$, $j(\tau_1)$ respectively to expressions $\left\lfloor \frac{\tau_1 - \tau_0}{\Delta\tau_s} - i\,\frac{\Delta\,\tau_e}{\Delta\tau_s} \right\rfloor$ and $\left\lfloor \frac{\tau_1 - \tau_0}{\Delta\tau_e} \right\rfloor$;
  Update
    • Initial indexes $i_0$ and $j_0$ to values $i(\tau_0)$ and $j(\tau_0)$;
    • Indexes $i$ and $j$ to values $i_0$ and $j_0$;
    • Index $k$ to value $j_0$
    • Scalar signal sample coordinate $\tau_s$ to value $\tau_0$;
  extremum estimation $\hat{x}_{i,\,j}(\tau_0)$ to value $x(\tau_0) + z$;

  while $\tau \in \mathbb{R}_{\geq \tau_0}$ do
    Get coordinate value $\tau$
    Update indexes $i$ and $j$ to values $i(\tau)$ and $j(\tau)$;
    Update index $k$ to value $j$;
    Update trajectory coordinate $\hat{\gamma}$ to value $\hat{\gamma}(\tau,\, \hat{x}_{i,\,j}) + z_\gamma$;
    Update discrete control coordinate $u_k$ to value $u(x(\tau), \hat{\gamma}) + z_u$;
    Apply $u + z_u$ as $u_k$ to the input of dynamical system $\dot{x} = a + b\,u$;
    if $\tau \in [\tau_i,\, \tau_{i+1}]$ then
      if $\tau \in \tau_s + [0,\, \Delta\tau_s]$ then
        Continue;
      else
        Update estimation $\hat{x}_{i,j}(\tau_0)$ by Eq. $\hat{x}_{i,j-1} + F_j(x(\tau))(x(\tau) - \hat{x}_{i,j-1})$ and add perturbation $z_{\hat{x}}$;
        Update coordinate $\tau_s$ to coordinate $\tau$;
        Increment index $j$ by 1;
      end
    else
      Increment index $i$ by 1;
      Update estimation $\hat{x}_{i,0}$ to value $\hat{x}_{i-1,j}$;
    end
  end while
```

The sample and exploration intervals $\Delta\tau_s$ and $\Delta\tau_e$ are important values on above description. Their value must be greater than the integration step $\Delta\tau_n$ used for numerical methods. Hence, it is reasonable to choose $\alpha_s\,\Delta\tau_e$ and $\alpha_n\,\Delta\tau_e$ for $\alpha_s, \alpha_n \in (0,1)$ and $\frac{\alpha_s}{\alpha_n} \triangleq k \gg 1$. For notation $\alpha_n = \frac{\Delta\tau_n}{\Delta\tau_e}$, some value α_e between α_n and 1 is a valid choice, e.g. the average $\frac{1+\alpha_n}{2}$ between both



(a) Robot path

(b) Reading signal

Figure 2. Robot path and scalar signal along it. The blue circle designates the extremum; The blue line corresponds to the robot trajectory; the trajectory in black represent the extremum estimation; finally, the green star and red square denotes respectively the initial and final robot positions; finally, the concentric circular lines correspond to level curves of function $f(x) = x^2 + y^2$.

extremes α_n and 1. In this case, the proportionality constant is $\frac{1}{2}(\frac{1}{\alpha_n} + 1)$ i.e. $\frac{k_n+1}{2}$. Finally, due to the strong inequality, also k_n must be much greater than 1 i.e. $k_n \gg 1$.

6. SIMULATIONS

This section describes and depicts the simulation results regarding the flow diagram proposed in Section 5. It exhibits results implementation of the barycenter method for source estimation, a line as trajectory toward it and control of the Lagrangian mechanical representation of an omnidirectional robot as a seeking agent.

Description	Symbol	Unit	Value
Simulation interval	Δt	ms	1
Exploration interval	T_s	ms	100
Planning interval	T_e	ms	500
Poles	λ	-	-10

Figure 3. Time-related parameters

Description	Symbol	Unit	Value
Speed enhancer	ν	-	5
Average value	$\bar{z}_n$	-	$\begin{bmatrix}0 & 0\end{bmatrix}^\intercal$
Standard deviation	σ	m	0.5

Figure 4. Barycenter-related parameters

Description	Symbol	Unit	Value
Average value	$\bar{z}_\tau$	-	$\begin{bmatrix}0 & 0 & 0\end{bmatrix}^\intercal$
Standard deviation	σ_τ	N m	1

Figure 5. Torque-related parameters

The control function u for trajectory tracking emerges from the the exact linearization algorithm, suitable for the task. The synthesis is symbolical and computationally laborious. If not extensive for the current document, the matrices for the chosen dynamical system model on equation (4) and control function below are on section 9 to consult.

The parameters for control tuning and optimization by barycenter method utilizes the values available on Tables 3, 4 and 5. The choice for simulation time constants relies on inertial behaviour of the dynamical system. Exploration and planning intervals does not present a strictive reasoning choice except they must be greater than simulation interval and the former must be lesser of equal to the latter. The poles for the control law may be, for example, greater or equal to $\frac{1}{T_s}$ to allow stationary error. The parameters ν and σ related to the barycenter method as well as torque standard deviation σ_τ are application-oriented.

The Cartesian representation of the trajectories along dynamical system vector field and extremum signal along with it, the estimation source of the scalar signal begin and endpoints for the robot trajectory as well as actual source position are on Figs. 2a and 2b.

For given simulation, Figs. 6 and 7 show respectively trajectory toward it and the control of the Lagrangian mechanical representation of an omnidirectional robot as a seeking agent. Although the scalar signal along the trajectory is not monotonically decreasing, it wanders around the source vicinity. Finally, the control variable u corresponds to the torque τ applied to each wheel individually and described in Fig. 7. For the simulation under

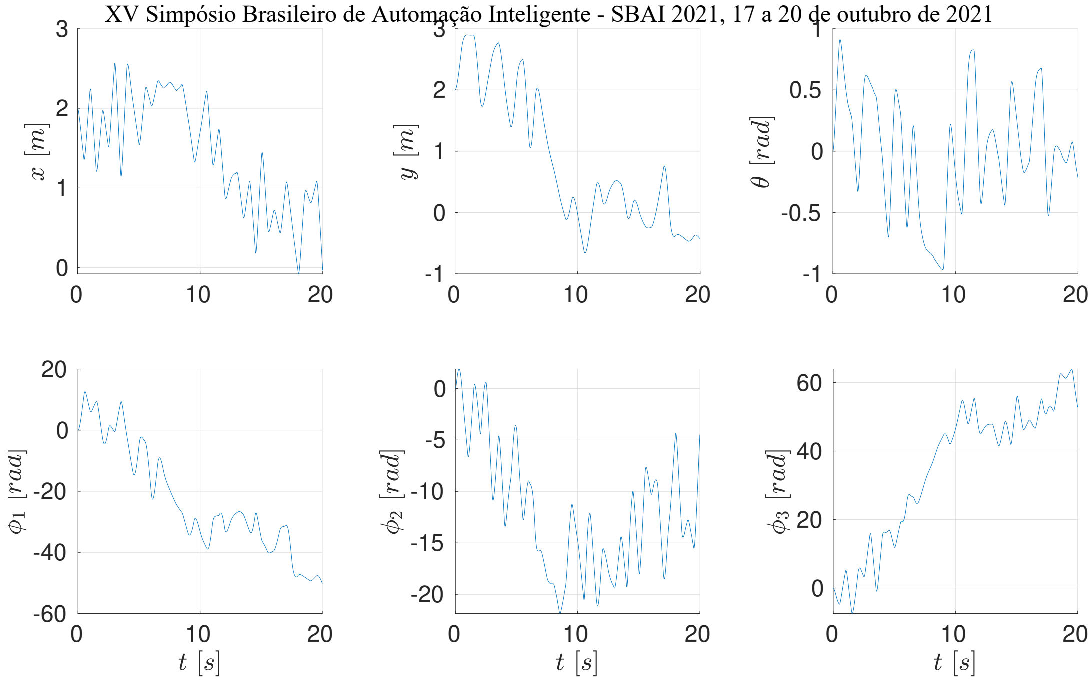


Figure 6. Omnidirectional robot states.

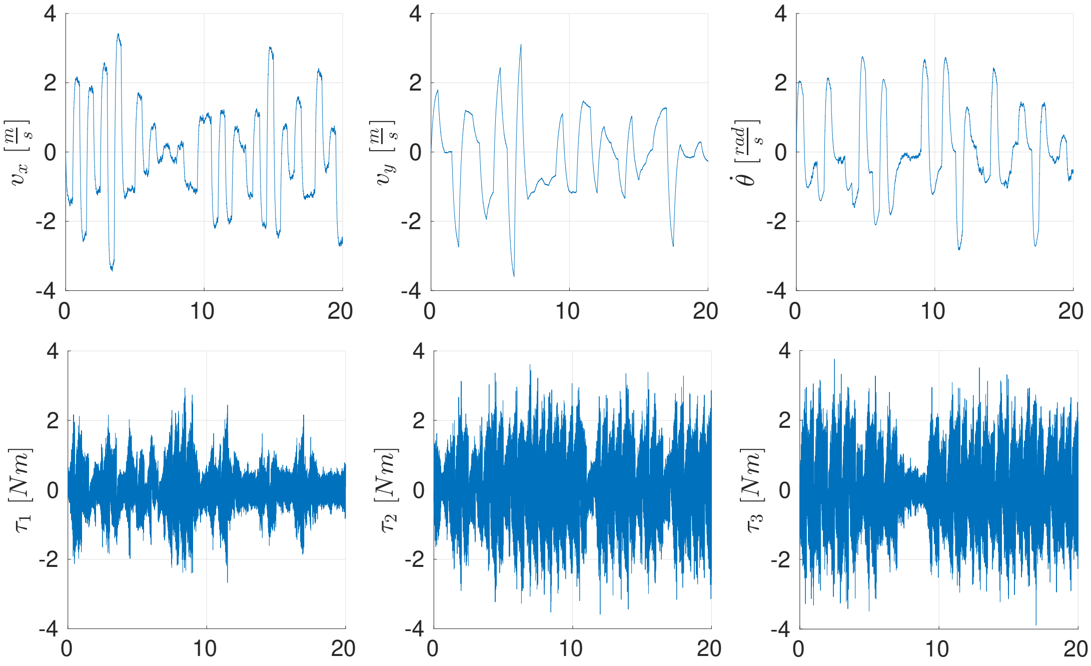


Figure 7. Speeds and torque applied to each wheel of the omnidirectional robot.

analysis, there is the addition of Gaussian noise to the input of the dynamical system.

7. CONCLUSION

The current work presents a recursive algorithm to converge trajectories of a dynamical system given initial coordinate x_0 in the direction of a scalar function critical point. It estimates the source location using the barycenter derivative-free optimization method presented by Pait (2018), which proved suitable for the task at hand.

A control law u alters the dynamical system vector field and converges the trajectories towards the critical point estimation. Although not the only method, the *exact linearization* algorithm corresponds to the equivalent version of a PI controller for nonlinear systems: it brings the tracking output error to zero by pole placement. The synthesis for nonholonomic systems requires an extension, available on Sastry (2013).

The estimation, trajectory synthesis, dynamical system control are the building blocks of the flow diagram in Fig. 1. The algorithm bases on periodic verification and signal sample for the estimation. The simulation in Figs. 6 and 7 depicts the results for hyperparameters on Tables 3, 4 and 5. The noise signal corresponds to a stochastic variable with Gaussian support distribution with zero-mean and standard deviation σ_τ.

Among trajectory choices in direction of the source estimation, the line between points corresponds to the geodesic in euclidean space. Despite the hypothesis of obstacle absence and flat space, actual reality aspects might interfere with the source seeking. Trajectories with non-zero curvature e.g. polynomial-based (e.g. Bézier's curve), infrared, and bump sensors are necessary for obstacle avoidance. Finally, further consideration of distinct vector fields $a_\sigma + b_\sigma\, u_\sigma$ i.e. multiple agents and/or spacely distributed scalar signal sensors y_σ, for index σ within set $\{1, \cdots, n_\sigma\}$, without undue increase in algorithmic complexity or change of flow diagram presented in Fig. 1.

The application of control on dynamical system $\dot{x} = a + b\, u$ does not happen continuously due to available hardware constraints. Alternatively, the control value u is continuous within a discretization interval $\Delta\tau_k$ and equal to u_k, for index k respective to the step count and initially 0. The magnitude of $\Delta\tau_k$ depends on *eigen* properties of the system under analysis and further developed in future work.

Further exploration possibilities are noise addition to source estimation point, reference trajectory as well as intermediate stochastic control points. The additional noise and points introduce stochastic exploration. Since the flow diagram heuristics contains stochastic variables, the expected value for the coordinates of the dynamical system corresponds to the average of results from multiple instances.

8. ACKNOWLEDGMENTS

The first author would like to thank Dr. Lucas Koleff and MSc. Guilherme Selicani for valuable remarks and suggestions.

9. APPENDIX

The chosen seeking agent in the current work corresponds to an omnidirectional mobile robot. Although extensive and comprehensible discussion in Campion et al. (1996) about this class of dynamical systems, the current work explores approach that follows.

For the robot at hand, it is sufficient to define coordinates q and speeds p respective to vectors $\begin{bmatrix} x \; y \; \theta \; \phi_1 \; \phi_2 \; \phi_3 \end{bmatrix}^\intercal$ and $\begin{bmatrix} v_x \; v_y \; \omega \end{bmatrix}^\intercal$. The kinematics description for a robot with constraints requires its Pfaffian equality $A(q)\, \dot{q} = 0$.

For the system under analysis and by hypothesis of a non-slipping wheels, the necessary relation emerges from the velocity projection of each wheel center on equation $\langle {}^0v_i,\, {}^0e_{y_i} \rangle$ along its longitudinal axis equal to the respective translational speed $\dot{\phi}_i\, R$, for variable R equal to the wheel radius length. Below, the i-index are 1, 2 and 3, and the angle θ_i corresponds respectively to 0, $\frac{2\pi}{3}$ and $\frac{4\pi}{3}$ deg. The variable L corresponds to the distance between robot geometric center and the respective wheel.

The velocity ${}^0v_i$ and versor ${}^0e_{y_i}$ correspond respectively to vectors $\begin{bmatrix} \dot{x} \; \dot{y} \; 0 \end{bmatrix}^\intercal + [{}^0e_z]_\times \, {}^i\, e_x\, \dot{\theta}\, L$ and $\mathrm{rot}\,(\theta + \theta_i, {}^0e_z)\, {}^0\, e_y$. The notation $[\cdot]_\times$ stands for the skew-symmetric matricial form for given vectorial entry, which allows the cross product between two vectors. The constraints concatenation above leads to the definition of the required Pfaffian equality and henceforth kinematic equations. The rows of matrix E are given by the resulting row vector of multiplication ${}^0e_{y_i}^\intercal \, \begin{bmatrix} {}^0e_x \; {}^0e_y \; [{}^0e_z]_\times \, {}^i\, e_x\, \mathrm{L} \end{bmatrix}$.

$$\underbrace{\begin{bmatrix} \mathrm{E} & -1_3\, R \end{bmatrix}}_{A(q)} \dot{q} = 0 \Longrightarrow \dot{q} = \underbrace{\begin{bmatrix} 1_3 \\ \frac{1}{R}\, \mathrm{E} \end{bmatrix}}_{B(q)} p \tag{7}$$

The control strategy at hand utilizes the transformation map between spaces $\mathcal{X}$ and $\mathcal{Z}$. For an affine smooth dynamical system, its representation in $\mathcal{Z}$ space composes of a linear controllable portion and a nonlinear non-observable one. The new coordinate system z is given by vector $\begin{bmatrix} x \; v_x \; y \; v_y \; \theta \; \omega \; \phi_1 \; \phi_2 \; \phi_3 \end{bmatrix}^\intercal$, obtained by tracking output derivation $y_s = \begin{bmatrix} x \; y \; \theta \end{bmatrix}^\intercal$ and completion to form a diffeomorphism with original coordinate representation.

By algebraic manipulation, the error equation of z-coordinate $\ddot{\tilde{y}}_s + A_1\, \dot{\tilde{y}}_s + A_0\, \tilde{y}_s = 0_{3\times 1}$ leads to obtain the control function. Henceforth, necessary matrices for control function follow in equations (8). The terms on expression below corresponds to eigenvalues of error decay and fulfill $\lambda_{ji}, \mu_{ji} \in \mathbb{C}^-$ and the operator $\oplus$ represents the direct sum of rings. For matrices, it corresponds to the block diagonal operation. The matrix A_K is Hurwitz stable and equal to matrix $A_\Delta + B_\Delta K$. Finally, the notation $(\cdot)^{-1}$ corresponds to the inverse of a matrix.

$$u(t, x) = \Delta^{-1}(x)\, (y_\delta^\star(t) - A_K \tilde{z}(t,\, x) - \varphi(\, x)) \tag{8}$$

$$\begin{cases} a_{\delta_i} = \begin{bmatrix} -\lambda_{1i}\, \lambda_{2i} \\ -\lambda_{1i} - \lambda_{2i} \end{bmatrix} \\ A_\Delta = a_{\delta_1}^\intercal \oplus a_{\delta_2}^\intercal \oplus a_{\delta_3}^\intercal,\; B_\Delta = 1_3 \\ \Delta = H^{-1}\, Z \Longrightarrow \Delta^{-1} = Z^{-1}\, H \\ \varphi = Z^{-1}(H\, v + h) \\ k_i = -a_{\delta_i} + \begin{bmatrix} -\mu_{1i}\, \mu_{2i} \\ \mu_{1i} + \mu_{2i} \end{bmatrix} \\ v = K\, \tilde{z},\; , \text{ for } K = k_1^\intercal \oplus k_2^\intercal \oplus k_3^\intercal \end{cases} \tag{9}$$

As mentioned in section 3, the robot dynamics requires matrices related to its inertial movement. Their definition are below to consult.

$$\begin{cases} M(q) = \mathrm{diag}\left(\begin{bmatrix} M_{11} & M_{22} & M_{33} & M_{44} \end{bmatrix}\right) & \text{, for } \begin{cases} M_{11} = M_{22} = m_R + 3\, m_r \\ M_{33} = I_R^z + 3\,(I_r^z + m_r\, L^2) + \\ \qquad (I_r^y - I_R^z)\,(\sin^2(\phi_1) + \sin^2(\phi_2) + \sin^2(\phi_3)) \\ M_{44} = I_r^x\, 1_3 \end{cases} \\ \begin{aligned} H(q) &= B^\intercal(q)\, M(q)\, B(q) \\ &= \mathrm{diag}\left(\begin{bmatrix} H_{11} & H_{22} & H_{33} \end{bmatrix}\right) \end{aligned} & \text{, for } \begin{cases} H_{11} \quad = H_{11} = \frac{3}{2}\,\frac{I_r^x}{R^2} + m_R + 3\, m_r \\ H_{33} \quad = I_R^z + 3\,\left(I_r^x\,\left(\frac{L}{R}\right)^2 + I_r^z + m_r\, L^2\right) \\ \qquad I_r^y\,\left(\sin^2(\phi_3) + \sin^2(\phi_1) + \sin^2(\phi_2)\right) - \\ \qquad I_r^z\,(\sin^2(\phi_1) + \sin^2(\phi_2) + \sin^2(\phi_3)) \end{cases} \end{cases} \tag{10}$$

$$U = \begin{bmatrix} 0_3 \\ 1_3 \end{bmatrix} \Longrightarrow Z(\theta) = B^\intercal\, U = \frac{1}{R}\begin{bmatrix} -\sin(\theta) & \cos(\theta) & L \\ -\cos\left(\theta + \frac{\pi}{6}\right) & -\cos\left(\theta - \frac{\pi}{3}\right) & L \\ \sin\left(\theta + \frac{\pi}{3}\right) & -\cos\left(\theta + \frac{\pi}{3}\right) & L \end{bmatrix} \tag{11}$$

$$\begin{cases} \nu(q,\,\dot{q}) = \begin{bmatrix} 0_{2\times 1} \\ \dot{\theta}\,\left(I_r^y - I_r^z\right)\,\left(\dot{\phi}_1\,\sin(2\,\phi_1) + \dot{\phi}_2\,\sin(2\,\phi_2) + \dot{\phi}_3\,\sin(2\,\phi_3)\right) \\ -\frac{I_r^y - I_r^z}{2}\,\dot{\theta}^2\,\sin(2\,\phi_1) \\ -\frac{I_r^y - I_r^z}{2}\,\dot{\theta}^2\,\sin(2\,\phi_2) \\ -\frac{I_r^y - I_r^z}{2}\,\dot{\theta}^2\,\sin(2\,\phi_3) \end{bmatrix} \\ h(q,\,p) = B^\intercal(q)\nu(q,\,B(q)\,p) + B^\intercal(q)M(q)\dot{B}^\intercal(q)\,p \end{cases} \tag{12}$$

REFERENCES

Åström, K.J. and Wittenmark, B. (2013). *Adaptive control*. Courier Corporation.

Campion, G., Bastin, G., and Dandrea-Novel, B. (1996). Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE transactions on robotics and automation*, 12(1), 47–62.

Choi, J.W., Curry, R., and Elkaim, G. (2008). Path planning based on Bézier curve for autonomous ground vehicles. In *Advances in Electrical and Electronics Engineering-IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, 158–166. IEEE.

Cochran, J. and Krstic, M. (2009). Nonholonomic source seeking with tuning of angular velocity. *IEEE Transactions on Automatic Control*, 54(4), 717–731.

Dubins, L.E. (1957). On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of Mathematics*, 79(3), 497–516. URL http://www.jstor.org/stable/2372560.

Frobenius, G. (1877). Über das Pfaffsche problem. *Journal für die reine und angewandte Mathematik*, 82, 230–315.

Goldstein, H., Poole, C., and Safko, J. (2002). Classical mechanics.

Hall, C.A. and Meyer, W. (1976). Optimal error bounds for cubic spline interpolation. *Journal of Approximation Theory*, 16(2), 105–122. doi: https://doi.org/10.1016/0021-9045(76)90040-X. URL https://www.sciencedirect.com/science/article/pii/002190457690040X.

Pait, F. and Colón, D. (2006). On the Lyapunov partial differential equation. In *Proceedings of the 45th IEEE Conference on Decision and Control*, 5102–5107. IEEE.

Pait, F.M. (2018). The barycenter method for direct optimization. *arXiv preprint arXiv:1801.10533*.

Sastry, S. (2013). *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media.

Sontag, E.D. (2013). *Mathematical control theory: deterministic finite dimensional systems*, volume 6. Springer Science & Business Media.

Tan, Y., Moase, W.H., Manzie, C., Nešić, D., and Mareels, I.M. (2010). Extremum seeking from 1922 to 2010. In *Proceedings of the 29th Chinese control conference*, 14–26. IEEE.

Vázquez-Méndez, M.E. and Casal, G. (2016). The clothoid computation: A simple and efficient numerical algorithm. *Journal of Surveying Engineering*, 142(3), 04016005.