Memorias del 2024 Congreso Nacional de Control Automático
8-11 de Octubre, 2024. Ciudad de México, México.


# Gradient Estimate for Continuous-Time Linear Dynamical Systems

**Fernando López-Caamal** * **Ixbalank Torres** **

* *Departamento de Ingeniería Química, Universidad de Guanajuato, Noria Alta, C.P. 36050, Guanajuato, México, email: fernando.lopez.caamal@gmail.com*
** *C. A. Telemática, Departamento de Ingeniería Electrónica, Universidad de Guanajuato, Carr. Salamanca-Valle de Santiago, Km. 3.5+1.8, C.P. 36885, Salamanca, México, email: ixbalank@ugto.mx*

Resumen In this paper we compare two different methodologies for computing the gradient of a linear, continuous-time dynamical system with no parametric uncertainties nor unknown inputs. Such a gradient is computed by performing the partial derivative of a performance function w.r.t. the input of the system. The first of the methodologies uses the model of the dynamical system to construct the gradient; whereas the second one avails of the differentiation of the performance function and the system's input. We compare both approaches via a numerical simulation.

*Keywords:* Model-based gradient, Continuous-time estimation, Linear systems

## 1. INTRODUCTION

Optimisation is an important tool in decision science and engineering. To solve optimisation problems, an objective (output variable) must be firstly identified, a quantitative measure of the performance of the system under study. Optimisation algorithms are iterative, they begin with an initial guess of the decision variable (input variable) and generate a sequence of improved estimates until they terminate at a solution. The strategy used to move from one iterate to the next distinguishes one algorithm from another. Most strategies make use of the values of the objective function and the input-output gradient (Nocedal and Wright, 2000).

The gradient estimation of the output of a dynamical system with respect to its input is an important trait of the behavior of such dynamical systems, as it measures how fast an input drifts the output of the system. In optimisation problem solution, the gradient estimation gives important information about the direction in which the objective function grows or decreases faster.

In particular, Extremum Seeking Control (ESC) is a model-free, real-time optimisation approach that aims to steer the state of dynamical system to an optimal condition. Such a control technique relies on an estimate of the output-input gradient via a high-pass filter in order to compute an input that drifts the state closer to such an optimal condition (Ariyur and Krstic, 2003; Dochain et al., 2011). In general, such strategies do not require the knowledge of the underlying dynamical model, but only an online measurement of the objective function and the input of the dynamical system. Some modifications of the ESC rely on the actual estimate of the gradient; see for instance (Torres-Zuniga et al., 2021) where the gradient is computed by differentiating the output and the input with respect to time and then the parametric gradient is calculated as the quotient of both differentiated signals; in (Guay and Dochain, 2017) the gradient is parameterised and such parameters are online estimated by means of a Lipschitz projection operator; in (Feudjio Letchindjio et al., 2019) the gradient is parameterised and such parameters are estimated online through a recursive estimation algorithm.

In this paper, we provide a closed form expression for the gradient of a linear, continuous-time dynamical system, for which we know the model. Such an expression relies completely on the model, the (estimated) states of the system, along with the input applied to such system. By means of a numerical simulation, we compare the performance of such estimation of the gradient with a model-free technique, based on sliding modes differentiators (López-Caamal and Moreno, 2019).

This article is organized as follows, in Section 2 the problem to address is formulated and the gradient estimation approach is presented; in Section 3 simulation results of the gradient estimation approach are presented and compared with a gradient estimation strategy previously proposed by the authors; finally, in Section 4 conclusions about the work developed are discussed.

https://doi.org/10.58571/CNCA.AMCA.2024.074

## 2. OUTPUT-INPUT GRADIENT

Let us consider a time-invariant SISO linear plant in continuous time

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{b}u(t) \tag{1a}$$

$$y(t) = \mathbf{c}\mathbf{x}(t), \tag{1b}$$

along with the performance function

$$J(t) = g(t, y, u). \tag{1c}$$

Here $\mathbf{x} : \mathbb{R}_+ \to \mathbb{R}^n$ and $u, y, J : \mathbb{R}_+ \to \mathbb{R}$. The matrices $\mathbf{A}, \mathbf{b}, \mathbf{c}$ have appropriate dimensions. We further assume that the pair $(\mathbf{A}, \mathbf{c})$ is observable.

The gradient of $J(t)$ w.r.t. $u(t)$ is defined as

$$\begin{aligned}\sigma &:= \pm\frac{\partial J}{\partial u}\\ &= \pm\frac{\frac{\mathrm{d}J(t)}{\mathrm{d}t}}{\frac{\mathrm{d}u(t)}{\mathrm{d}t}},\end{aligned} \tag{2}$$

whenever the derivatives exist and $\frac{\mathrm{d}u(t)}{\mathrm{d}t} \neq 0$. The $\pm$ sign in the expression above is considered since in some cases such a gradient is used to maximise or to minimise the objective function. However, such a control scheme is out of the scope of the present work.

### 2.1 Model-based approach

In this first approach, we use of the model to derive the gradient estimate. Since

$$\frac{\mathrm{d}J}{\mathrm{d}t} = \frac{\partial g(t, y, u)}{\partial t} + \frac{\partial g(t, y, u)}{\partial y}\mathbf{c}(\mathbf{A}\mathbf{x} + \mathbf{b}u) + \frac{\partial g(t, y, u)}{\partial u}\dot{u} \tag{3a}$$

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \dot{u}(t), \tag{3b}$$

the gradient in (2) becomes

$$\sigma = \pm\left(\frac{\partial g}{\partial u} + \frac{\frac{\partial g}{\partial t} + \frac{\partial g}{\partial y}\mathbf{c}(\mathbf{A}\mathbf{x} + \mathbf{b}u)}{\dot{u}}\right). \tag{4}$$

However, the state $\mathbf{x}(t)$ is not known. Given the observability of $(\mathbf{A}, \mathbf{c})$ we can design an observer and use the estimate of $\mathbf{x}$ to attain an estimate of the gradient. We, however, consider the function $u(t)$ and its time derivative known.

For the sake of simplicity, let the observer be a Luenberger Observer:

$$\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{x}}(t) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{b}u(t) - \mathbf{L}(\hat{y} - y) \tag{5a}$$

$$\hat{y}(t) = \mathbf{c}\hat{\mathbf{x}}(t). \tag{5b}$$

Let us now define the error estimation as $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$. This way, the error dynamics can be expressed as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{e}(t) = (\mathbf{A} - \mathbf{L}\mathbf{c})\mathbf{e}(t).$$

The stability of the observation error origin may be attained by choosing an appropriate $\mathbf{L}$, provided that the pair $(\mathbf{A}, \mathbf{c})$ is observable. To compute the observer gain $\mathbf{L}$, let us now consider the following candidate Lyapunov function

$$V(\mathbf{e}) = \mathbf{e}^\top\mathbf{P}\mathbf{e}, \tag{6}$$

with $\mathbf{P} = \mathbf{P}^\top$ (Scherer and Weiland, 2004). The derivative of $V(\mathbf{e})$ with respect to time is given by

$$\dot{V}(\mathbf{e}) = \mathbf{e}^\top\left(\mathbf{A}^\top\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{c}^\top\mathbf{L}^\top\mathbf{P} - \mathbf{P}\mathbf{L}\mathbf{c}\right)\mathbf{e}.$$

Clearly, (6) is a Lyapunov function if and only if

$$\mathbf{P} > 0, \tag{7a}$$

$$\mathbf{A}^\top\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{c}^\top\mathbf{Q}^\top - \mathbf{Q}\mathbf{c} < 0, \tag{7b}$$

with $\mathbf{Q} = \mathbf{P}\mathbf{L}$. This way the asymptotic stability of the estimation error is assured and therefore, the estimated $\hat{\mathbf{x}}$ asymptotically will converge to the true state $\mathbf{x}$. By solving LMIs (7) for $\mathbf{P}$ and $\mathbf{Q}$, the observer gain is then computed as $\mathbf{L} = \mathbf{P}^{-1}\mathbf{Q}$.

Thus, when using the observed states, the estimate of the gradient becomes

$$\hat{\sigma}_1 = \pm\left(\frac{\partial g}{\partial u} + \frac{\frac{\partial g}{\partial t} + \frac{\partial g}{\partial y}\mathbf{c}(\mathbf{A}\hat{\mathbf{x}} + \mathbf{b}u)}{\dot{u}}\right). \tag{8}$$

### 2.2 Differentiator-based approach

A different way to estimate the gradient is by means of (2) and by differentiating the performance function and the input, to obtain

$$\hat{\sigma}_2 = \pm\frac{\hat{\omega}_1}{\hat{\omega}_2}, \tag{9}$$

where $\hat{\omega}_i$ are the estimates of

$$\boldsymbol{\omega} := \begin{pmatrix}\dot{J}\\ \dot{u}\end{pmatrix}; \text{ furthermore} \tag{10}$$

$$\boldsymbol{\theta} := \begin{pmatrix}J\\ u\end{pmatrix}. \tag{11}$$

An estimate of the derivative of $\boldsymbol{\theta}$ may be given by (López-Caamal and Moreno, 2019)

$$\begin{aligned}\dot{\hat{\boldsymbol{\theta}}}(t) &= -k_1\boldsymbol{\phi}_1\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right) + \hat{\boldsymbol{\omega}}(t)\\ \dot{\hat{\boldsymbol{\omega}}}(t) &= -k_2\boldsymbol{\phi}_2\left(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta}\right),\end{aligned} \tag{12}$$

where $\hat{\boldsymbol{\theta}}$ ( $\hat{\boldsymbol{\omega}}$, resp.) denotes the estimation of $\boldsymbol{\theta}$ ($\boldsymbol{\omega}$ , resp.), in addition,

$$\boldsymbol{\phi}_1(\mathbf{z}) := \left(\eta\,||\mathbf{z}(t)||_2^{-p} + \beta + \gamma\,||\mathbf{z}(t)||_2^{q}\right)\mathbf{z}, \quad \boldsymbol{\phi}_1(\mathbf{0}) := \mathbf{0},$$
$$\boldsymbol{\phi}_2(\mathbf{z}) := \left(\eta(1-p)\,||\mathbf{z}(t)||_2^{-p} + \beta + \gamma(1+q)\,||\mathbf{z}(t)||_2^{q}\right)\boldsymbol{\phi}_1,$$

where $\mathbf{z}(t) := \hat{\boldsymbol{\theta}}(t) - \boldsymbol{\theta}(t)$ and $||\mathbf{z}(t)||_2 := \sqrt{\mathbf{z}^\top(t)\mathbf{z}(t)}$. In addition, $\eta, \beta, \gamma > 0$, $\frac{1}{2} \geq p > 0$, and $q > 0$. To ensure that the estimate converges in finite-time to the actual derivatives, one requires the actual signals derivative to be bounded, i.e. $|\dot{\omega}_i| < \Delta,\ \Delta \in \mathbb{R}_+$. Moreover, one requires the matrix

$$\boldsymbol{\Gamma} = \begin{pmatrix} -\kappa_1 & 1 \\ -\kappa_2 & 0 \end{pmatrix}$$

to be negative definite. This is attained, for instance, when $\kappa_1, \kappa_2 > 0$.

### 2.3 Discretisation of the derivatives

To compare both approaches we compute the gradient in (2) via the ratio of the finite differences of the performance function and input; that is

$$\hat{\sigma}_3 = \pm\frac{J([k+1]\tau) - J(k\tau)}{u([k+1]\tau) - J(k\tau)}. \tag{13}$$

Here $k$ and $\tau$ denotes the $k$th time-step and sampling period, respectively.

In addition, to assess the performance of the model-based approach and the one based on the differentiator, we define

$$\epsilon_i(t) := \log_{10}\left|\frac{\hat{\sigma}_i(t) - \hat{\sigma}_3(t)}{\hat{\sigma}_3(t)}\right|,\ \hat{\sigma}_3(t) \neq 0. \tag{14}$$

In the following section, we compare these three approaches for a linear plant via numeric simulation.

## 3. SIMULATION RESULTS

In this section, we consider the following linear system in (1) with the following matrices

$$\mathbf{A} = \begin{pmatrix} -1 & 8 & 2 \\ 1 & -3 & 0 \\ -3 & -5 & -3 \end{pmatrix}$$
$$\mathbf{b} = (0\ 0\ 1)^\top$$
$$\mathbf{c} = (0\ 0\ 1)$$

Along with the objective function

$$J(t) = y^2(t),$$

and input

$$u(t) = 10t + \cos(3t),$$

whose time derivative is

$$\dot{u}(t) = 10 - 3\sin(3t).$$

### 3.1 Model-based observer

For this approach one requires the state estimation obtained via (5) with gain $\mathbf{L}$ computed by solving LMIs (7) using CVX solver (Grant and Boyd, 2014). This way, the observer gain obtained is

$$\mathbf{L} = 1 \times 10^3\,(1.8382\ \ -1.4159\ \ 0.0680)^\top.$$

The estimates states may be found in Figure 1. Now, with the definitions above and with Eq. (8) the estimated gradient becomes

$$\hat{\sigma}_1 = \frac{2y(t)\mathbf{c}\,(\mathbf{A}\hat{\mathbf{x}} + \mathbf{b}u(t))}{10 - 3\sin(3t)}.$$

### 3.2 Differentiator-based approach

For this approach let

$$\{\eta, \beta, \gamma\} = \{15, 5, 5\}$$
$$\{p, q\} = \{0.5, 0.5\}$$
$$\{\kappa_1, \kappa_2\} = \{3, 3\}.$$

The gradient estimation in (9) is computed with the differentiator in (12).

### 3.3 Discretisation of derivatives

In this case, we use a time step of $\tau = 2 \times 10^{-3}$. For all cases, the numerical solution was obtained via Matlab's ode45 with relative tolerance of $10^{-3}$ and an absolute one of $10^{-4}$.

### 3.4 Outcome assessment

Figures 2 and 3 shows the comparison of the three approaches for the computation of the gradient. Despite the transient of the observer and the differentiator, one may see that the results are very similar to that of the one obtained with the discretisation of the derivatives in (13).

In turn, Figure 4 shows the relative, absolute error as computed with Equation (14). Please notice that, although such an error is very similar, the error obtained with $\hat{\sigma}_2$ exhibits chattering given that it is computed via the ratio of derivatives obtained with sliding mode approaches.

Now, to assess the effect of a perturbation on the estimation of the gradient, we consider an additive perturbation in the input

$$u_p(t) = 0.75\cos(10t), \tag{15}$$

which is unknown to the observer and to Eq. (8). Figures 5 and 6 show how the model-based estimation of the gradient, $\hat{\sigma}_1$, deteriorates when such an unknown input is applied. The differentiator-based gradient estimation, $\hat{\sigma}_2$, practically keeps its accuracy given that it does not rely on the knowledge of $u(t)$, but on its online measurement.

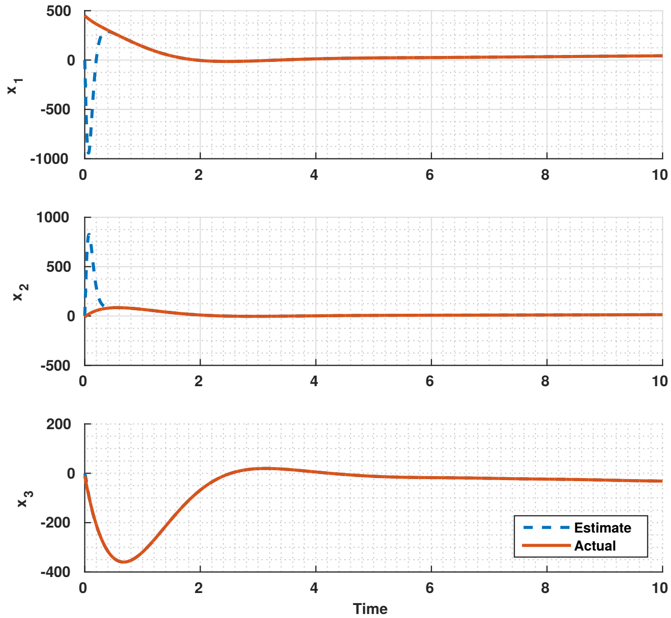


Figura 1. Estimated states via the Luenberger observer in (5).

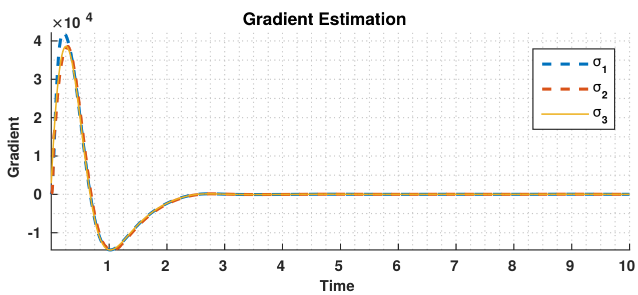


Figura 2. Estimated gradient. The blue discontinuous line, $\sigma_1$, is the model-based estimation in (8); whereas, the red discontinuous line, $\sigma_2$, is the gradient estimation in (9) obtained via the differentiators. In turn, the finite-difference estimation of the gradient, $\sigma_3$, in (13) is the yellow, continuous line.

## 4. CONCLUSIONS

Three different approaches to compute the gradient of a continuous-time, linear dynamical system are considered and compared via a numerical simulation.

The first approach provides a closed formula for the gradient, but it requires the mathematical model of the system along with the knowledge of the input and output and the state estimates. In this case, the estimation error of the gradient is driven by the underpinning estimation error of the state. Furthermore, should the system had uncertainties, the state estimates would not be accurate, which would deteriorate the gradient estimate, as may be

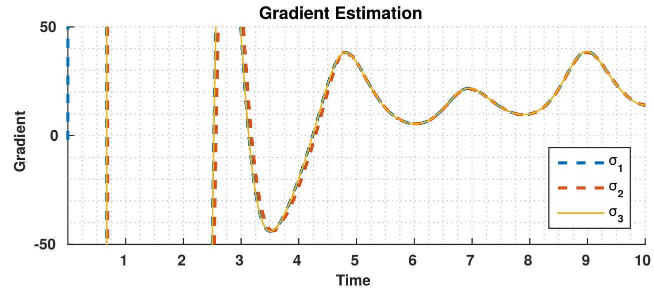


Figura 3. Detail of the estimated gradient. The organisation of the figure is the same as for Figure 2.

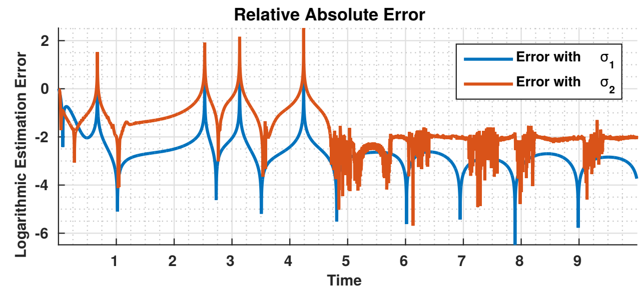


Figura 4. Relative, absolute error defined in (14) of the gradient estimations w.r.t. the finite difference estimation.

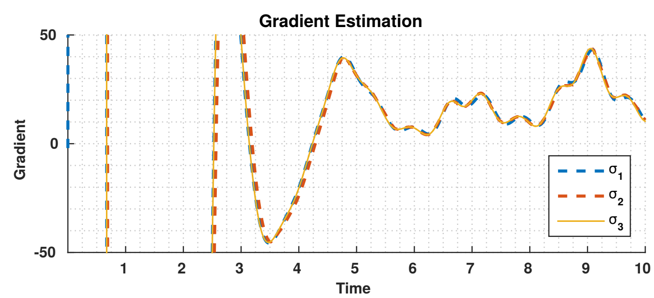


Figura 5. Detail of the estimated gradient with an additive perturbation in the input.

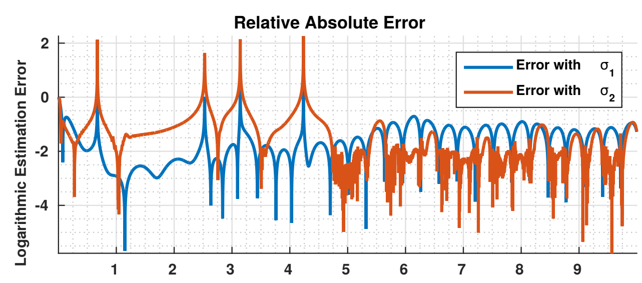


Figura 6. Relative, absolute error defined in (14) of the gradient estimations w.r.t. the finite difference estimation, considering the perturbation in the input in (15).

seen in Figures 4 and 6. The effect of uncertainties on the estimation, however, may be minimized by designing a robust Luenberger observer.

In contrast, the second approach relies only on the online measurements of the input and the output to estimate its first derivatives via a sliding modes based differentiator; the gradient is then computed by the ratio of such derivatives, which might have some chattering, thus exhibiting such trait in the gradient estimate along with a larger estimation error of the gradient as may bee seen from Figure 4. However, the presence of an unknown perturbation in the input did not yield a significant effect on this estimation.

In turn, the last method requires the sampling of the data one step of time ahead in order to provide the estimation of the gradient, thus not providing an online estimate of the gradient. If the application allows it, the use of previous samples instead of future samples could mitigate this problem.

## REFERENCIAS

Ariyur, K.B. and Krstic, M. (2003). *Real-time optimization by extremum-seeking control*. John Wiley & Sons.

Dochain, D., Perrier, M., and Guay, M. (2011). Extremum seeking control and its application to process and reaction systems: A survey. *Mathematics and Computers in Simulation*, 82(3), 369–380.

Feudjio Letchindjio, C.G., Dewasme, L., Deschenes, J.S., and Vande Wouwer, A. (2019). An extremum seeking strategy based on block-oriented models: Application to biomass productivity maximization in microalgae cultures. *Industrial & Engineering Chemistry Research*, 58(30), 13481–13494.

Grant, M. and Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. `https://cvxr.com/cvx`.

Guay, M. and Dochain, D. (2017). A proportional-integral extremum-seeking controller design technique. *Journal of Process Control*, 77, 61–67.

López-Caamal, F. and Moreno, J.A. (2019). Generalised multivariable supertwisting algorithm. *International Journal of Robust and Nonlinear Control*, 29(3), 634–660.

Nocedal, J. and Wright, S. (2000). *Numerical optimization*. Springer, 2nd. edition edition.

Scherer, C. and Weiland, S. (2004). *Linear Matrix Inequalities in Control*. Delft University of Technology.

Torres-Zuniga, I., Lopez-Caamal, F., Hernandez-Escoto, H., and Alcaraz-Gonzalez, V. (2021). Extremum seeking control and gradient estimation based on the super-twisting algorithm. *Journal of Process Control*, 105, 223–235.

https://doi.org/10.58571/CNCA.AMCA.2024.074