IDETC 2023 • 19th MSNDC

Recursive Least-Squares Based Constraint Enforcement Algorithm for Multibody Systems

Renato Maia Matarazzo Orsino*
Éverton Lins de Oliveira**

Mech. Eng. Dept., Escola Politécnica – University of São Paulo
*Offshore Mechanics Lab. (LMO)
**Dynamics and Control Lab. (LDC)

1 Motivation
2 RLS based constraint enforcement algorithm
3 Application to the rectangular Bricard mechanism
4 Final remarks

1 Motivation
2 RLS based constraint enforcement algorithm
3 Application to the rectangular Bricard mechanism
4 Final remarks

Motivation

- Modeling and numerical simulation of constrained multibody systems are still challenging tasks, **even with the great variety of methodologies** already available in the literature.
- **Parameterizing** the description of motions of all the bodies in terms of **minimal** numbers of generalized coordinates and/or quasi-velocities is **not always possible and not always desirable**.
- The most immediate strategy in these cases would involve the direct numerical treatment of the resulting system of **differential algebraic equations (DAEs)** which can be **numerically intricate**.
- In order to avoid these difficulties, many **constraint enforcement methods** have been proposed, including:
  - Elimination of undetermined multipliers implicitly (e.g. Maggi's equation, orthogonal projection operators based strategies) or explicitly (e.g. Udwadia-Kalaba equation).
  - Use of minimal number of quasi-velocities only (e.g. Gibbs-Appell equation, Kane's method).
  - Penalty methods, augmented Lagrangian approaches.

Motivation

- **Constraint violation stabilization and suppression approaches** as well as **strategies for properly scaling constraint equations** have also been proposed to obtain better numerical performance when some **constraint enforcement methods** are adopted.
- Some works in the literature focus on reviewing and comparing these methods, approaches and strategies, e.g. [1, 2, 3, 5].
- Do we really need another constraint enforcement algorithm? What about **reinterpreting** and **adapting** an existing recursive least-squares (RLS) algorithm so that it becomes applicable as a **constraint enforcement strategy** able to overcome the difficulties that motivated the proposition of several other algorithms along the years?

Modular Modeling Methodology (MMM)

A recursive approach based on a hierarchycal conception

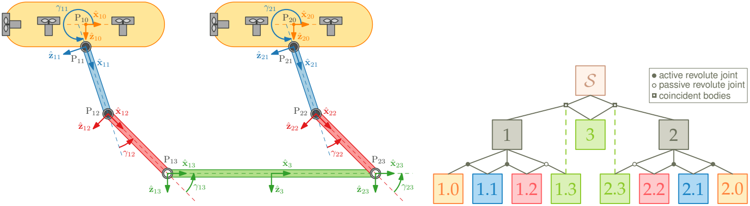


Figure: Hierarchical conception of a cooperative transportation task using I-AUVs [6].

In the MMM [7, 8, 9] approach, the recursive enforcement of constraints is based on the assembly conception of the system, leading to a hierarchy that defines in which step a given block of constraint equations is enforced.

1 Motivation
2 RLS based constraint enforcement algorithm
3 Application to the rectangular Bricard mechanism
4 Final remarks

State variables and constraint equations

Consider a **ν-DOF** multibody system with **bilateral constraints** whose state is parameterized by:

- **generalized coordinates** $\mathbf{q} \in \mathbb{R}^n$, $n \geq \nu$;
- **quasi-velocities** $\mathbf{v} \in \mathbb{R}^m$, $m \geq \nu$.

From purely kinematic relations, there must be a function $\mathbf{g}$ which allows to express $\dot{\mathbf{q}}$ in terms of the **state** $(t, \mathbf{q}, \mathbf{v})$:

$$\dot{\mathbf{q}} = \mathbf{g}(t, \mathbf{q}, \mathbf{v}) \tag{1}$$

Other kinematic relations lead to the **constraint equations** of the system:

$$\tilde{\mathbf{q}}(t, \mathbf{q}) = \mathbf{0} \qquad \textbf{configuration level} \rightarrow \textbf{holonomic} \tag{2}$$

$$\tilde{\mathbf{v}}(t, \mathbf{q}, \mathbf{v}) = \mathbf{0} \qquad \textbf{velocity level} \tag{3}$$

Eq. (3) should necessarily **include** the **time derivatives of the configuration level constraints**, but might also include constraints that only exist at velocity level, which are **non-holonomic**.

Equations of motion

Acceleration level constraints can be obtained from the time derivatives of the velocity level ones[1]:

$$\tilde{\mathbf{a}} = \mathbf{H}(t, \mathbf{q}, \mathbf{v})\mathbf{a} - \mathbf{y}(t, \mathbf{q}, \mathbf{v}) = \mathbf{0} \quad \text{with} \quad \mathbf{a} = \dot{\mathbf{v}} \tag{4}$$

Thus, **at a given state** $(t, \mathbf{q}, \mathbf{v})$, the value of $\mathbf{a}$ may be **uniquely determined** as the solution of the system constituted by the dynamic equations and the constraint ones at the acceleration level, i.e.:

$$\begin{cases} \mathbf{M}(t, \mathbf{q})\mathbf{a} + \mathbf{H}^\top(t, \mathbf{q}, \mathbf{v})\boldsymbol{\lambda} = \mathbf{f}(t, \mathbf{q}, \mathbf{v}) \\ \tilde{\mathbf{a}} = \mathbf{H}(t, \mathbf{q}, \mathbf{v})\mathbf{a} - \mathbf{y}(t, \mathbf{q}, \mathbf{v}) = \mathbf{0} \end{cases} \tag{5}$$

The existence of a **unique solution** for $\mathbf{a}$ requires $\begin{bmatrix} \mathbf{M} \\ \mathbf{H} \end{bmatrix}$ to be **full-rank**.

[1] Typically, velocity level constraints are linear in $\mathbf{v}$, so that $\mathbf{H} = \mathbf{H}(t, \mathbf{q})$.

Gauss' Principle of Least Constraint

These **equations of motion** can be interpreted as the ones arising from the problem of finding the value of $\mathbf{a}$ that **minimizes the Gaussian deviation function** (assuming $\mathbf{M}$ to be **positive-definite**[2]):

$$Z(\mathbf{a}, \boldsymbol{\lambda}) = \frac{1}{2}(\mathbf{a} - \mathbf{P}_0\mathbf{f})^\top \mathbf{P}_0^{-1}(\mathbf{a} - \mathbf{P}_0\mathbf{f}) - \boldsymbol{\lambda}^\top(\mathbf{y} - \mathbf{H}\mathbf{a}), \quad \text{with} \quad \mathbf{P}_0^{-1} = \mathbf{M} \tag{6}$$

Noticing that a single multiplier λ_{r+1} is defined for each individual constraint equation, and denoting by $\mathbf{h}_{r+1}^\top$ the $(r+1)$-th row of the Jacobian matrix $\mathbf{H}$):

$$Z(\mathbf{a}, \boldsymbol{\lambda}) = \frac{1}{2}(\mathbf{a} - \mathbf{P}_0\mathbf{f})^\top \mathbf{P}_0^{-1}(\mathbf{a} - \mathbf{P}_0\mathbf{f}) - \sum_{r=0}^{c-1} \lambda_{r+1}(y_{r+1} - \mathbf{h}_{r+1}^\top \mathbf{a}) \tag{7}$$

[2] If the **positive definiteness** of $\mathbf{M}$ cannot be assured, choose a positive-definite weighting matrix $\mathbf{W}$ and reset (Udwadia and Wanichanon, 2013; Orsino, 2020):

$$\mathbf{M} \leftarrow \mathbf{M} + \mathbf{H}^\top\mathbf{W}\mathbf{H}$$

$$\mathbf{f} \leftarrow \mathbf{f} + \mathbf{H}^\top\mathbf{W}\mathbf{y}$$

The new $\mathbf{M}$ will be **positive-definite** and the equations of motion so obtained will be the same.

Recursive least-squares algorithm (RLS)

Given a **training set** $\{(\mathbf{x}_i, y_i); i = 1, 2, \ldots, c\}$ with vector inputs $\mathbf{x}_i \in \mathbb{R}^m$ and scalar outputs $y_i \in \mathbb{R}$, find a vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^m$ such that using $\hat{y}_i = \mathbf{x}_i^\top \boldsymbol{\theta}$ as an estimation of y_i minimizes the **objective function**[3]:

$$J(\boldsymbol{\theta}) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \mathbf{P}_0^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + \sum_{r=0}^{c+1} \frac{1}{2\sigma_{r+1}^2}\left(y_{r+1} - \mathbf{x}_{r+1}^\top \boldsymbol{\theta}\right)^2 \tag{8}$$

This problem has the following recursive solution (repeat for $r = 0, \ldots, c-1$):

$$\boldsymbol{\theta}_{r+1} \leftarrow \boldsymbol{\theta}_r + \frac{y_{r+1} - \mathbf{x}_{r+1}^\top \boldsymbol{\theta}_r}{\mathbf{x}_{r+1}^\top \mathbf{P}_r \mathbf{x}_{r+1} + \sigma_{r+1}^2} \mathbf{P}_r \mathbf{x}_{r+1} \tag{9}$$

$$\mathbf{P}_{r+1} \leftarrow \mathbf{P}_r - \frac{1}{\mathbf{x}_{r+1}^\top \mathbf{P}_r \mathbf{x}_{r+1} + \sigma_{r+1}^2} \mathbf{P}_r \mathbf{x}_{r+1} \mathbf{x}_{r+1}^\top \mathbf{P}_r \tag{10}$$

[3] If the errors $\varepsilon_i = y_i - \hat{y}_i = y_i - \mathbf{x}_i^\top \boldsymbol{\theta}$ are modeled as independently distributed zero-mean Gaussian noises, the σ_i^2 is the corresponding variance. In any case, $1/(2\sigma_i^2)$ can always be interpreted as a weighting factor.

Recursive least-squares algorithm (RLS)

When the number of training points is less than the number of parameters of the model, i.e. $c < m$, an alternative objective function can be proposed:

$$J(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \frac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)^\top \mathbf{P}_0^{-1}(\boldsymbol{\theta} - \boldsymbol{\theta}_0) + \sum_{r=0}^{c+1} \lambda_{r+1}\left(y_{r+1} - \mathbf{x}_{r+1}^\top \boldsymbol{\theta}\right) \tag{11}$$

This latter expression for J corresponds to an **infinite error penalty** version of the previous one, since:

$$\lim_{\sigma_i \to 0} \left(y_i - \mathbf{x}_i^\top \boldsymbol{\theta}\right) = 0 \quad \Leftrightarrow \quad \lim_{\sigma_i \to 0} \frac{y_i - \mathbf{x}_i^\top \boldsymbol{\theta}}{2\sigma_i^2} = \lambda_i \text{ (finite)} \tag{12}$$

The recursive solution to this problem can be found by taking $\sigma_{r+1} \to 0$:

$$\boldsymbol{\theta}_{r+1} \leftarrow \boldsymbol{\theta}_r + \frac{y_{r+1} - \mathbf{x}_{r+1}^\top \boldsymbol{\theta}_r}{\mathbf{x}_{r+1}^\top \mathbf{P}_r \mathbf{x}_{r+1}} \mathbf{P}_r \mathbf{x}_{r+1} \tag{13}$$

$$\mathbf{P}_{r+1} \leftarrow \mathbf{P}_r - \frac{1}{\mathbf{x}_{r+1}^\top \mathbf{P}_r \mathbf{x}_{r+1}} \mathbf{P}_r \mathbf{x}_{r+1} \mathbf{x}_{r+1}^\top \mathbf{P}_r \tag{14}$$

RLS based constraint enforcement algorithm

The quasi-acceleration vector $\mathbf{a} = \dot{\mathbf{v}}$ corresponding to the solution of the equations of motion of a **constrained multibody system** at a given state $(t, \mathbf{q}, \mathbf{v})$ is the one that minimizes the Gaussian deviation function:

$$Z(\mathbf{a}, \boldsymbol{\lambda}) = \frac{1}{2}(\mathbf{a} - \mathbf{P}_0\mathbf{f})^\top \mathbf{P}_0^{-1}(\mathbf{a} - \mathbf{P}_0\mathbf{f}) - \sum_{r=0}^{c-1} \lambda_{r+1}(y_{r+1} - \mathbf{h}_{r+1}^\top \mathbf{a}) \tag{15}$$

Let $\mathbf{a}_0 = \mathbf{P}_0\mathbf{f}$ and define $\mathbf{a}_r$ as the value of the quasi-acceleration obtained if only the constraints up to the r-th one were considered. The value of $\mathbf{a}_{r+1}$ can be obtained through a recursion which adds to $\mathbf{a}_r$ a correction term **proportional the violation** of the $(r+1)$-st constraint equation:

$$\mathbf{a}_{r+1} \leftarrow \mathbf{a}_r - \underbrace{\frac{\mathbf{h}_{r+1}^\top \mathbf{a}_r - y_{r+1}}{\mathbf{h}_{r+1}^\top \mathbf{P}_r \mathbf{h}_{r+1}}}_{\text{amount of violation (scalar)}} \underbrace{\mathbf{P}_r \mathbf{h}_{r+1}}_{\text{drift direction}} \tag{16}$$

Special properties of the proposed recursive algorithm

P1 Insensitivity to the scaling of constraint equations: the effect of scaling a given constraint equation by a factor β will be cancelled out within the calculation of its own correction terms, not affecting the terms associated to any of other constraints.

$$\mathbf{a}_{r+1} \leftarrow \mathbf{a}_r - \frac{\beta\left(\mathbf{h}_{r+1}^\top \mathbf{a}_r - y_{r+1}\right)}{\left(\beta\mathbf{h}_{r+1}\right)^\top \mathbf{P}_r \left(\beta\mathbf{h}_{r+1}\right)} \mathbf{P}_r \left(\beta\mathbf{h}_{r+1}\right) \tag{17}$$

$$\mathbf{P}_{r+1} \leftarrow \mathbf{P}_r - \frac{1}{\left(\beta\mathbf{h}_{r+1}\right)^\top \mathbf{P}_r \left(\beta\mathbf{h}_{r+1}\right)} \mathbf{P}_r \left(\beta\mathbf{h}_{r+1}\right)\left(\beta\mathbf{h}_{r+1}\right)^\top \mathbf{P}_r \tag{18}$$

P2 The drift direction corresponding to a redundant constraint equation is a zero vector: indeed, due to the recursion involved in the definition of $\mathbf{P}_{r+1}$ it can be stated that its null space is given by $\mathrm{span}\left(\{\mathbf{h}_1, \ldots, \mathbf{h}_r, \mathbf{h}_{r+1}\}\right)$; in other words, $\mathrm{rank}(\mathbf{P}_{r+1})$ corresponds to the **number of degrees of freedom** left after the enforcement of the first $(r+1)$ constraints.

Constraint violation suppression at position and velocity levels

Even with an accurate value for the quasi-acceleration vector $\mathbf{a}$ that does not violate any of the acceleration level constraints, the numerical integration itself would induce some drift for the position and velocity level constraints. Noticing that the recursion for $\mathbf{a}$ can be written in the form:

$$\mathbf{a}_{r+1} \leftarrow \mathbf{a}_r - \frac{\tilde{a}_{r+1}}{s_{r+1}} \mathbf{z}_{r+1}, \quad \text{with} \quad \begin{cases} \mathbf{z}_{r+1} = \mathbf{P}_r \mathbf{h}_{r+1} \\ s_{r+1} = \mathbf{h}_{r+1}^\top \mathbf{z}_{r+1} \end{cases} \tag{19}$$

with each $\mathbf{z}_{r+1}$ denoting the drift direction associated to the $(r+1)$-th constraint equation $\tilde{a}_{r+1} = \mathbf{h}_{r+1}^\top \mathbf{a} - y_{r+1} = 0$, the following recursions can be proposed for suppressing the velocity and position constraint violations:

$$\mathbf{v}_{r+1} \leftarrow \mathbf{v}_r - \frac{\tilde{v}_{r+1}}{s_{r+1}} \mathbf{z}_{r+1} \tag{20}$$

$$\mathbf{q}_{r+1} \leftarrow \mathbf{q}_r - \frac{\tilde{q}_{r+1}}{s_{r+1}} \mathbf{G} \mathbf{z}_{r+1}, \quad \text{with} \quad \mathbf{G} = \frac{\partial \mathbf{g}}{\partial \mathbf{v}} \tag{21}$$

Remember that, from Eq. (1), $\dot{\mathbf{q}} = \mathbf{g}(t, \mathbf{q}, \mathbf{v})$.

RLS based constraint enforcement algorithm

$\mathbf{P}_0 \leftarrow \mathbf{M}^{-1}$

$\mathbf{a}_0 \leftarrow \mathbf{P}_0\mathbf{f}$

for $r = 0, \ldots, c-1$ (the order of the constraint equations can be sorted)

$\mathbf{z}_{r+1} \leftarrow \mathbf{P}_r \mathbf{h}_{r+1}$

$s_{r+1} \leftarrow \mathbf{h}_{r+1}^\top \mathbf{z}_{r+1}$

$\mathbf{P}_{r+1} \leftarrow \mathbf{P}_r - (1/s_{r+1})\mathbf{z}_{r+1}\mathbf{z}_{r+1}^\top$

$\tilde{a}_{r+1} \leftarrow \mathbf{h}_{r+1}^\top \mathbf{a}_r - y_{r+1}$

$\mathbf{a}_{r+1} \leftarrow \mathbf{a}_r - (\tilde{a}_{r+1}/s_{r+1})\mathbf{z}_{r+1}$

$(\Delta t, \mathbf{q}_0, \mathbf{v}_0) \leftarrow$ single_integration_step$(t, \mathbf{q}, \mathbf{v}, \mathbf{a}_c)$

for $r = 0, \ldots, c-1$ calculate $\mathbf{z}_{r+1}, s_{r+1}, \mathbf{P}_{r+1}, \tilde{q}_{r+1}, \tilde{v}_{r+1}$ and do

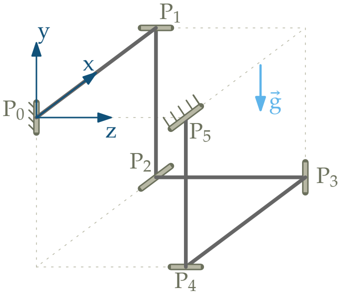
$\mathbf{q}_{r+1} \leftarrow \mathbf{q}_r - (\tilde{q}_{r+1}/s_{r+1})\mathbf{G}\mathbf{z}_{r+1}$

$\mathbf{v}_{r+1} \leftarrow \mathbf{v}_r - (\tilde{v}_{r+1}/s_{r+1})\mathbf{z}_{r+1}$

if $t \leq t_{\text{end}}$ do $(t, \mathbf{q}, \mathbf{v}) \leftarrow (t + \Delta t, \mathbf{q}_c, \mathbf{v}_c)$ **and repeat the algorithm**

1 Motivation
2 RLS based constraint enforcement algorithm
3 Application to the rectangular Bricard mechanism
4 Final remarks

Rectangular Bricard mechanism



- A problem from the IFToMM multibody benchmark library [4].
- Uniform 1 m, 1 kg bars with square cross-sections of width 0.1 m.
- $n = 10$ generalized coordinates:

$$\mathbf{q} = (x_1, y_1, x_2, y_2, z_2, x_3, y_3, z_3, y_4, z_4)$$

- $m = 10$ quasi-velocities trivially defined:

$$\mathbf{v} = \dot{\mathbf{q}}$$

- $c = 10$ holonomic constraint equations:
  - 5 for enforcing the constant length of the bars.
  - 5 for enforcing that each joint axes remain orthogonal to the adjacent bars.
- $\nu = 1$ degree-of-freedom – the system is redundantly constrained, but the redundant constraint equation cannot be immediately identified.

Rectangular Bricard mechanism

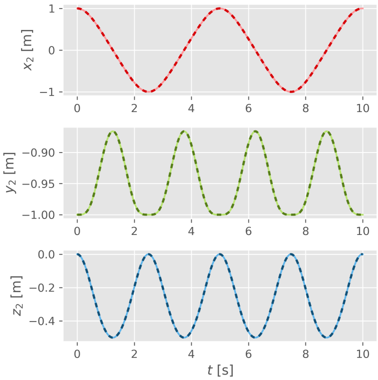


Figure: Coordinates of point P_2 – pastel-colored and solid: simulations with the proposed algorithm; darker and dashed: benchmark results [4].

Rectangular Bricard mechanism

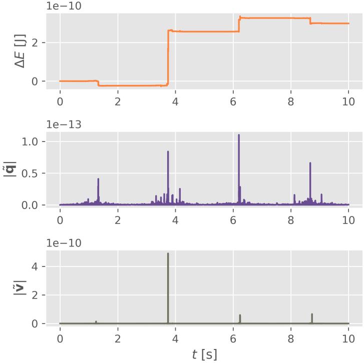


Figure: Variation of mechanical energy and violations of constraints at configuration and velocity levels – for instance, the value of $|\Delta E| \leq 4 \times 10^{-10}$ J, in the benchmark [4], $|\Delta E|$ goes up to 8×10^{-4} J.

1 Motivation
2 RLS based constraint enforcement algorithm
3 Application to the rectangular Bricard mechanism
4 Final remarks

Final remarks

- A **novel constraint enforcement strategy** for multibody systems, based on a **well known recursive least-squares algorithm**, is proposed.
- Each constraint equation provides an **individual correction term** for the **quasi-acceleration vector** of the system.
- Within the computation of this correction term, the **direction** in which the **constraint drift** occurs is identified, which allows the propostion of an extra correction step in which **position and velocity level constraint violations can be suppressed** right after a numerical integration step.
- The effect of **scaling** a given equation is **directly filtered out** within the calculation of the corresponding correction term, not affecting or being affected significantly by other constraint equations.
- In future works, the analogy with the classical RLS formulation can be explored for treating problems in which **imperfect constraints** are admissible (in which a non-zero value for σ_r can be considered).

Final remarks

- A draft paper entitled *"Recursive algorithms for constraint enforcement and drift suppression and its applications for the modeling of multibody systems"* will be submitted in the next couple of weeks.
- The paper will present the full derivation of the proposed algorithm along with its application to other benchmark problems. This paper also includes a block-wise form of the recursive algorithm.
- The Wolfram Mathematica (symbolic derivation of the equations of motion) and Julia language (numerical simulation using the proposed recursive approach) routines used in the case study can be found in the GitHub repository **renatoorsino/RLS-CEA**.
- A copy of this presentation is also available in this repository as part of its documentation.

References I

- Augustynek, K., Urbaś, A.: Numerical investigation on the constraint violation suppression methods efficiency and accuracy for dynamics of mechanisms with flexible links and friction in joints.
  Mech. Mach. Theory **181**(March 2023), 105211 (2023).
  DOI 10.1016/j.mechmachtheory.2022.105211

- Bauchau, O.A., Laulusa, A.: Review of Contemporary Approaches for Constraint Enforcement in Multibody Systems.
  J. Comput. Nonlinear Dyn. **3**(1), 1–8 (2008).
  DOI 10.1115/1.2803258

- Blajer, W.: Methods for constraint violation suppression in the numerical simulation of constrained multibody systems - A comparative study.
  Comput. Methods Appl. Mech. Eng. **200**(13-16), 1568–1576 (2011).
  DOI 10.1016/j.cma.2011.01.007

References II

- Francisco González, González, M.: Rectangular Bricard mechanism (2014)
- Laulusa, A., Bauchau, O.A.: Review of Classical Approaches for Constraint Enforcement in Multibody Systems.
  J. Comput. Nonlinear Dyn. **3**(1), 1–12 (2008).
  DOI 10.1115/1.2803257
- de Oliveira, É.L., Orsino, R.M.M., Donha, D.C.: Modular modeling and coordination control scheme for an underwater cooperative transportation performed by two I-AUVs.
  Control Eng. Pract. **125**(April), 105198 (2022).
  DOI 10.1016/j.conengprac.2022.105198

References III

- Orsino, R.M.M.: A contribution on modeling methodologies for multibody systems.
  Ph.D. thesis, Universidade de São Paulo (2016)
- Orsino, R.M.M.: Recursive modular modelling methodology for lumped-parameter dynamic systems.
  Proc. R. Soc. A Math. Phys. Eng. Sci. **473**(2204), 20160891 (2017).
  DOI 10.1098/rspa.2016.0891
- Orsino, R.M.M.: Extended constraint enforcement formulations for finite-DOF systems based on Gauss's principle of least constraint.
  Nonlinear Dyn. **101**(4), 2577–2597 (2020).
  DOI 10.1007/s11071-020-05924-9

Thank you!

reorsino@usp.br