**Memorias del XVI Congreso Latinoamericano
de Control Automático, CLCA 2014
Octubre 14-17, 2014. Cancún, Quintana Roo, México**

Efficient computation of extended Lie brackets

García Ramírez, E. * **Márquez Martínez, L. A.** *

* *Centro de Investigación Científica y de Educación Superior de Ensenada, Carretera Ensenada-Tijuana 3918, Zona Playitas, 22860 B.C. México (e-mail: edugar@cicese.edu.mx, lmarquez@cicese.mx).*

Abstract: The recently developed geometric theory for nonlinear time delay systems has used as a central operation an extension of the Lie bracket, which is of fundamental importance in the case of nonlinear system without delay. However, the operations involved with these new results require a considerable amount of computational resources. In this work a recursive strategy that reduce the amount of operations in the iterative extended Lie bracket computations is presented. This result is applied in the integrability test of a codistribution.

Keywords: Nonlinear time-delay systems, geometric control, extended Lie bracket

1. INTRODUCTION

In recent years new tools, based on the theory of noncommutative rings, were developed with the aim of dealing with open problems of analysis and control of nonlinear time-delay systems (NTDS). This new theory includes an extension of the geometrical framework control theory, and the operation known as Lie bracket (Califano et al. (2011)). The Lie bracket is a cornerstone for the theory of nonlinear systems without delays in several applications like the equivalence with linear systems, disturbance decoupling (Isidori (1995)), non-interacting control (Isidori (1995)), stability of switched nonlinear systems (Margaliot and Liberzon (2004)), nonlinear filtering (Hazewinkel and Marcus (1982)) and so on. Using the extended geometrical tools for NTDS, necessary and sufficient conditions for the solution of problems like equivalence with a linear system (Califano and Moog (2011); Califano et al. (2010); Califano and Moog (2012)), linearization by input-output injection (Califano et al. (2013)), and feedback linear equivalence (Califano and Moog (2011)) were presented. The results presented in the cited articles that involve the extended Lie bracket, require a big quantity of computations which may difficult, in practice, the implementation of such tool. To overcome this difficulty a software implementation of the extended Lie bracket by means of a computer algebra system was developed (Gárate-García et al. (2013)). In this work a recursive strategy which reduces the number of the extended Lie bracket operations is presented. This result reduces the time in the computation algorithms, like the integrability of a submodule test (Califano and Moog (2011)), important for the analysis of nonlinear time-delay systems.

⋆ Sponsor and financial support acknowledgment goes here. Paper titles should be written in uppercase and lowercase letters, not all uppercase.

2. PRELIMINARIES AND NOTATION

Consider the nonlinear dynamical time-delay system with commensurable delays represented by the equations

$$\begin{aligned}\dot{x}(t) &= F(x(t), x(t-1), \ldots, x(t-s)) + \\ &\sum_{j=0}^{s} G(x(t), x(t-1), \ldots, x(t-s))u(t-j) \\ y(t) &= H(x(t), x(t-1), \ldots, x(t-s)).\end{aligned} \tag{1}$$

The following notation is taken from Califano et al. (2011): $\mathcal{K}$ denotes the field of meromorphic functions of the symbols $\{x(t-i), u(t-i), \ldots, u^{[k]}(t-i), i, k \in \mathbb{N}\}$; d is the differential operator that maps elements from $\mathcal{K}$ to $\mathcal{E} = \{dx(t-i), du(t-i), \ldots, du^{[k]}(t-i), i, k \in \mathbb{N}\}$; δ is the time-shift operator defined over meromorphic functions as follows: $a(\cdot), f(\cdot) \in \mathcal{K},\ \delta(a(t)df(t)) = a(t-1)\delta df(t) = a(t-1)df(t-1)$. Using the time-shift operator δ as indeterminate, the non commutative Euclidean (left) ring of polynomials with coefficients over $\mathcal{K}$ is denoted as $\mathcal{K}(\delta]$; $\mathbb{R}(\delta)$ is the ring of polynomials in δ with coefficients in $\mathbb{R}$. Denoting $deg(\cdot)$ the degree in δ of its arguments, the elements of $\mathcal{K}(\delta]$ may be written as $\alpha(\delta] = \sum_{i=0}^{r_\alpha} \alpha_i(t)\delta^i$, with $\alpha_i \in \mathcal{K}$, and $r_\alpha = deg(\alpha(\delta])$. Addition and multiplication on this ring are defined by

$$\alpha(\delta] + \beta(\delta] = \sum_{i=0}^{max\{r_\alpha, r_\beta\}} (\alpha_i(t) + \beta_i(t))\delta^i, \tag{2}$$

and

$$\alpha(\delta]\beta(\delta] = \sum_{i=0}^{r_\alpha} \sum_{j=0}^{r_\beta} \alpha_i(t)\beta_j(t-i)\delta^{i+j}. \tag{3}$$

A matrix $A(\mathbf{x}_{[s]}, \delta) \in \mathcal{K}^{n\times n}(\delta]$ is called unimodular if it has a polynomial inverse.

Let us define $(\mathbf{x}_{[s]}) = (x(t), x(t-1), \ldots, x(t-s))$, with $(\mathbf{z}_{[s]})$, and $(\mathbf{u}_{[s]})$ in the same vein, and $(\bar{\mathbf{u}}_{[s],k}) = (\mathbf{u}_{[s]}, \dot{\mathbf{u}}_{[s]}, \ldots, \mathbf{u}_{[s]}^{(k)})$. Consider also $\psi(\mathbf{x}_{[s]}) \mid_{\mathbf{x}_{[l]}(-j)} := \psi(x(t-j), x(t-j-1), \ldots, x(t-j-l))$. Then it is possible to rewrite the equation (1) as

$$\dot{x}(t) = F(\mathbf{x}_{[s]}) + \sum_{j=0}^{s} G(\mathbf{x}_{[s]})u(t-j) \qquad (4)$$

$$y(t) = H(\mathbf{x}_{[s]}). \qquad (5)$$

The corresponding differential form representation is given by

$$d\dot{x} = f(\mathbf{x}_{[s]}, \mathbf{u}_{[s]}, \delta)dx + g(\mathbf{x}_{[s]}, \delta)du \qquad (6)$$

where

$$\begin{aligned} f(\mathbf{x}_{[s]}, \mathbf{u}_{[s]}, \delta) &= \sum_{i=0}^{s} \left(\frac{\partial F(\mathbf{x}_{[s]})}{\partial x(t-i)} + \sum_{j=0}^{s} u(t-j) \frac{\partial G(\mathbf{x}_{[s]})}{\partial x(t-i)} \right) \delta^i \qquad (7) \\ g(\mathbf{x}_{[s]}, \delta) &= \sum_{j=0}^{s} G(\mathbf{x}_{[s]})\delta^j \end{aligned}$$

Let $\mathbf{r}_1(\mathbf{x}, \mathbf{u}, \delta) = \sum_{j=0}^{s} \mathbf{r}_1^j(\mathbf{x}, \mathbf{u})\delta^j$, and $\mathbf{r}_2(\mathbf{x}, \mathbf{u}, \delta) = \sum_{j=0}^{s} \mathbf{r}_2^j(\mathbf{x}, \mathbf{u})\delta^j$.

The extended Lie bracket $[\mathbf{r}_1^k(\cdot, \mathbf{u}), \mathbf{r}_2^k(\cdot, \mathbf{u})]_{E_i} \in \mathbb{R}^{(i+1)n}$, $i \geq 0$ is defined as:

$$\begin{aligned} &[\mathbf{r}_1^k(\cdot, \mathbf{u}), \mathbf{r}_2^l(\cdot, \mathbf{u})]_{E_i} = \\ &\sum_{j=0}^{min(k,l,i)} ([\mathbf{r}_1^{k-j}(\cdot, \mathbf{u}), \mathbf{r}_2^{l-j}(\cdot, \mathbf{u})]_{E_0})^T \mid_{(x(-j), \mathbf{u}(-j))} \frac{\partial}{\partial x(t-j)} \end{aligned} \qquad (8)$$

with:

$$\begin{aligned} [\mathbf{r}_1^k(\cdot, \mathbf{u}), \mathbf{r}_2^l(\cdot, \mathbf{u})]_{E_0} &= \sum_{i=0}^{k} \frac{\partial \mathbf{r}_2^l(\mathbf{x}, \mathbf{u})}{\partial x(t-i)} \mathbf{r}_1^{k-i}(\mathbf{x}(-i), \mathbf{u}(-i)) \\ &- \sum_{i=0}^{l} \frac{\partial \mathbf{r}_1^k(\mathbf{x}, \mathbf{u})}{\partial x(t-i)} \mathbf{r}_2^{k-i}(\mathbf{x}(-i), \mathbf{u}(-i)) \end{aligned} \qquad (9)$$

where $l, k, i \in \{0, 1, \ldots\}$. The extended Lie bracket definition is associated with the infinite dimensional system

$$\begin{aligned} \dot{x}(t) &= F(\mathbf{x}_{[s]}) + \sum_{j=0}^{s} G(\mathbf{x}_{[s]})u(t-j) \\ \dot{x}(t-1) &= F(\mathbf{x}_{[s]}(-1)) + \sum_{j=0}^{s} G(\mathbf{x}_{[s]}(-1))u(t-j-1) \\ \dot{x}(t-2) &= F(\mathbf{x}_{[s]}(-2)) + \sum_{j=0}^{s} G(\mathbf{x}_{[s]}(-2))u(t-j-2) \\ &\vdots \end{aligned} \qquad (10)$$

Some properties of the extended Lie bracket are:

- $[\mathbf{r}_1^k(\cdot, \mathbf{u}), \mathbf{r}_2^l(\cdot, \mathbf{u})]_{E_i} = -[\mathbf{r}_2^l(\cdot, \mathbf{u}), \mathbf{r}_1^k(\cdot, \mathbf{u})]_{E_i}$. Skew symmetry.
- let $k \leq l, k \leq \gamma, \quad \gamma \leq 0$
$$\begin{aligned} &[\mathbf{r}_1^k, \mathbf{r}_2^l]_{E_\gamma} = [\mathbf{r}_1^k, \mathbf{r}_2^l]_{E_k} = \\ &\sum_{j=0}^{k} ([\mathbf{r}_1^{k-j}(\cdot, \mathbf{u}), \mathbf{r}_2^{l-j}(\cdot, \mathbf{u})]_{E_0})^T \mid_{(x(-j), \mathbf{u}(-j))} \frac{\partial}{\partial x(t-j)} \end{aligned} \qquad (11)$$
- let $\hat{\gamma} \geq k + ps \geq 0, \ k \leq l$
$$\begin{aligned} &\left[\mathbf{r}_1^{k+ps}, \mathbf{r}_2^{l+ps}\right]_{E_{\hat{\gamma}}} = \left[\mathbf{r}_1^{k+ps}, \mathbf{r}_2^{l+ps}\right]_{E_{k+ps}} = \\ &\sum_{j=0}^{k} ([\mathbf{r}_1^{k-j}(\cdot, \mathbf{u}), \mathbf{r}_2^{l-j}(\cdot, \mathbf{u})]_{E_0})^T \mid_{(x(-\vartheta))} \frac{\partial}{\partial x(t-\vartheta)}, \\ &\vartheta = j + ps, \ k \geq s \end{aligned} \qquad (12)$$
- let $k \leq l, k \leq \gamma, \quad \gamma \leq 0$
$$[\mathbf{r}_1^k, \mathbf{r}_2^l]_{E_\gamma} = [\mathbf{r}_1^k, \mathbf{r}_2^l]_{E_k} = 0, \qquad l - k > 2s \qquad (13)$$

The concept of integrability of a distribution is fundamental in the solution of control problems (like linear equivalence) for nonlinear systems without delays. The next definition of integrability is useful in the analysis of nonlinear time-delay systems (Califano and Moog (2011)).

Definition. The submodule Δ, nonsingular locally around $\mathbf{x}^0$, is integrable if there exist $n - j$ independent functions $\lambda_l(\mathbf{x}_{[s]})$, $l \in [1, n-j]$ such that $rank \frac{\partial \lambda(\mathbf{x})}{\partial x(t)} = n - j$ and

$$\sum_{p=0}^{\gamma} \frac{\delta \lambda_l(\mathbf{x})}{\partial x(t-j)} \delta^p \sum_{k=s}^{s} r_i^k \delta^k = 0, \quad l \in [1, n-j], \forall i \in [i, j] \qquad (14)$$

3. MAIN RESULT

Regardless the infinite dimensionality of time-delay systems, due to the extended Lie bracket properties, it is possible to characterize the extended Lie bracket operation using a set of finite dimensional distributions defined as:

$$\begin{aligned} \Delta_i' = span_{\mathcal{K}}\{ &\sum_{l=0}^{min(\gamma,i)} (\mathbf{r}_k^{\gamma-l}(\mathbf{x}(-l)))^T \frac{\partial}{\partial x(t-l)}, \\ &k \in [1, j], \gamma = 0, \ldots, i+s\}, i \geq 0. \end{aligned} \qquad (15)$$

To perform the extended Lie bracket operator over a set of elements that spans this distribution, it is possible to compute $[\mathbf{r}_1^j, \mathbf{r}_2^{q+j}]_{E_i}$, $p \leq q + j, i \leq q, \ j = 0, \ldots, p$ several times until all the required operations are done.

Now, we present the main result of this work.

Proposition 1. Consider the set of operations defined by the extended Lie bracket $\varepsilon_j = [\mathbf{r}_1^j, \mathbf{r}_2^{q+j}]_{E_i}$, $p \leq q, i \leq q, \ j = 0, \ldots, p$. It is possible to compute ε_j using the next iterative expression

$$\varepsilon_{j+1} = [\mathbf{r}_1^j, \mathbf{r}_2^{q+j}]_{E_0} \frac{\partial}{\partial x(t)} + \varepsilon_j \mid_{\mathbf{x}(-i)}, \quad \varepsilon_0 = 0 \qquad (16)$$

Proof. It comes from the definition of the extended Lie bracket

$$\begin{aligned} &[\mathbf{r}_1^j(\cdot), \mathbf{r}_2^{q+j}(\cdot)]_{E_i} = \\ &\sum_{\kappa=0}^{j} ([\mathbf{r}_1^{j-\kappa}(\cdot), \mathbf{r}_2^{q+j-\kappa}(\cdot)]_{E_0})^T \mid_{(x(-\kappa))} \frac{\partial}{\partial x(t-\kappa)} = \\ &([\mathbf{r}_1^j(\cdot), \mathbf{r}_2^{q+j}(\cdot)]_{E_0})^T \frac{\partial}{\partial x(t)} + \\ &\sum_{\kappa=1}^{j} ([\mathbf{r}_1^{j-\kappa}(\cdot), \mathbf{r}_2^{q+j-\kappa}(\cdot)]_{E_0})^T \mid_{(x(-\kappa))} \frac{\partial}{\partial x(t-\kappa)} \end{aligned} \qquad (17)$$

Remark. Note that, instead of performing j operations in the form of equation (9), a time-shift in each step is computed saving a total of $p(p+1)/2$ operations.

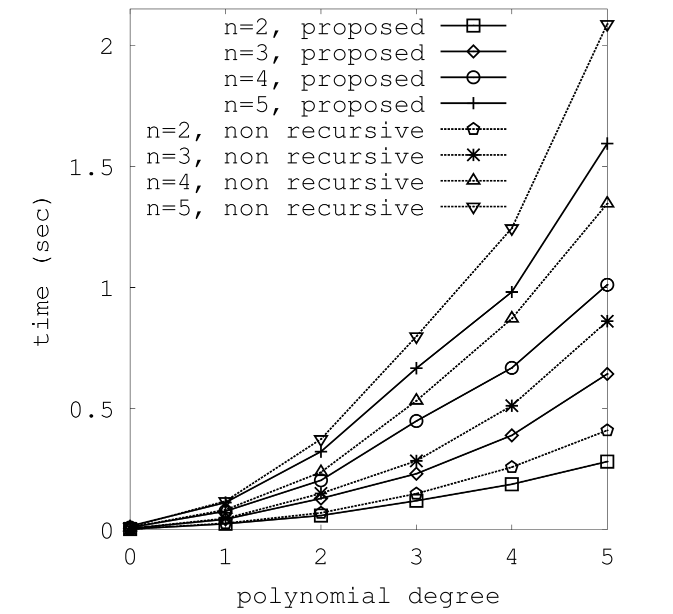

Fig. 1. Comparison with the proposed algorithm

It is possible to visualize the result of the Proposition 1 using the set of vectors:

$$\left\{[\mathbf{r}_1^0,\mathbf{r}_2^{\rho}]_{E_i},\ldots,[\mathbf{r}_1^{p-1},\mathbf{r}_2^{q-1}]_{E_i},\ [\mathbf{r}_1^{p},\mathbf{r}_2^{q}]_{E_i}\right\} = \left\{\begin{pmatrix}[\mathbf{r}_1^0,\mathbf{r}_2^{\rho}]_{E_0}\\ 0\\ 0\\ \vdots\\ 0\\ 0\end{pmatrix}, \begin{pmatrix}[\mathbf{r}_1^1,\mathbf{r}_2^{\rho+1}]_{E_0}\\ [\mathbf{r}_1^0,\mathbf{r}_2^{\rho}]_{E_0}(-1)\\ 0\\ \vdots\\ 0\\ 0\end{pmatrix},\ldots,\right. \left.\begin{pmatrix}[\mathbf{r}_1^{p-1},\mathbf{r}_2^{q-1}]_{E_0}\\ [\mathbf{r}_1^{p-2},\mathbf{r}_2^{q-3}]_{E_0}(-1)\\ [\mathbf{r}_1^{p-3},\mathbf{r}_2^{q-4}]_{E_0}(-2)\\ \vdots\\ [\mathbf{r}_1^0,\mathbf{r}_2^{\rho}]_{E_0}(-p+1)\\ 0\end{pmatrix}, \begin{pmatrix}[\mathbf{r}_1^{p},\mathbf{r}_2^{q}]_{E_0}\\ [\mathbf{r}_1^{p-1},\mathbf{r}_2^{q-1}]_{E_0}(-1)\\ [\mathbf{r}_1^{p-2},\mathbf{r}_2^{q-3}]_{E_0}(-2)\\ \vdots\\ [\mathbf{r}_1^1,\mathbf{r}_2^{\rho+1}]_{E_0}(-p+1)\\ [\mathbf{r}_1^0,\mathbf{r}_2^{\rho}]_{E_0}(-p)\end{pmatrix}\right\} \tag{18}$$

with $\rho = q - p$. Note that the space generated by this vectors has the same structural properties than the distribution Δ_i'.

3.1 Performance test

For the performance test a set of vectors, elements of a coditributions, were randomly generated. The structure of this vectors is in the form $\mathbf{r}_1(\mathbf{x},\mathbf{u},\delta) = \sum_{j=0}^{s}\mathbf{r}_1^j(\mathbf{x},\mathbf{u})\delta^j$ with the polynomial degree $s = 0,\ldots,5$, and dimension $n = 2,\ldots 4$. The algorithm that generate this vector is included in the Appendix A. In the Figure 1, a comparison between the computation the proposed recursive solution, and the non recursive solution is presented.

3.2 Example.

Consider the submodule

$$\Delta = span_{\mathcal{K}(\delta]}\left\{\left(2x_1(t)x_2(t-3)\delta^2 - 4x_1(t)\delta\right)\frac{\partial}{\partial x_1(t)} - \left((x_2(t)x_2(t-3)\delta - 2x_2(t)\right)\frac{\partial}{\partial x(t)}\right\}. \tag{19}$$

now consider

$$\begin{aligned}\Delta_2' = {} & span_{\mathcal{K}}\left\{2x_2(t)\frac{\partial}{\partial x_2(t)}\right\} + span_{\mathcal{K}}\left\{-4x_1(t)\frac{\partial}{\partial x_1(t)}\right.\\ & \left.-x_2(t)x_2(t-2)\frac{\partial}{\partial x_2(t)} + 2x_2(t-1)\frac{\partial}{\partial x_2(t-1)}\right\}\\ & +span_{\mathcal{K}}\left\{-x_2(t-1)x_2(t-3)\frac{\partial}{\partial x_2(t-1)}\right.\\ & +2x_2(t-2)\frac{\partial}{\partial x_2(t-2)} - 4x_1(t-1)\frac{\partial}{\partial x_1(t-1)}\\ & \left.2x_1(t)x_2(t-3)\frac{\partial}{\partial x_1(t)}\right\}\\ & +span_{\mathcal{K}}\left\{-x_2(t-2)x_2(t-4)\frac{\partial}{\partial x_2(t-2)}\right.\\ & -4x_1(t-2)\frac{\partial}{\partial x_1(t-2)}\\ & \left.+2x_1(t-1)x_2(t-4)\frac{\partial}{\partial x_1(t-1)}\right\}\\ & +span_{\mathcal{K}}\left\{2x_1(t-2)x_2(t-5)\frac{\partial}{\partial x_1(t-2)}\right\}\end{aligned} \tag{20}$$

note that $[r^0,r^1]_{E_0} = [r^2,r^3]_{E_0} = [r^3,r^4]_{E_0} = [r^4,r^5]_{E_0} = 0$, and $[r^1,r^2]_{E_0} = 2\,x_2\,(t-2)\ x_2\,(t)\,\frac{\partial}{\partial x_2(t)}$ using the recursive equation (16) it is possible to compute $[r^0,r^1]_{E_2} = [r^4,r^5]_{E_2} = 0$, $[r^1,r^2]_{E_2} = 2\,x_2\,(t-2)\ x_2\,(t)\,\frac{\partial}{\partial x_2(t)}$, $[r^2,r^3]_{E_2} = 2\,x_2\,(t-3)\ x_2\,(t-1)\,\frac{\partial}{\partial x_2(t-1)}$, and $[r^3,r^4]_{E_2} = 2\,x_2\,(t-4)\ x_2\,(t-2)\,\frac{\partial}{\partial x_2(t-2)}$. Note that, following the Theorem 2 presented in Califano et al. (2011), using this algorithm it is possible to determinate that the distribution (19) is integrable. In fact, the function $x_1(t)x_2^2(t-1)$ fulfill the integrability definition.

4. CONCLUSION

In the present paper, an efficient recursive solution that reduces the computational effort to calculate the extended Lie bracket was presented. This recursive solution may be applied, for example, in applications of control theory that involve the concept of integrability of a codistribution. The same algorithm may be used also in problems like equivalence with a linear system, normal form equivalence, and linearization via input-output injection. The proposed methodology could be used in other applications, like the computation of the series expansion of an element that belongs to a distribution Δ_i with time-delay. Finally, the proposed recursive algorithm shows a saving of time, with respect to the available one, that increases with the dimension of the vectors.

ACKNOWLEDGEMENTS

First author would like to thank Prof. C.Califano and Prof. C. Moog for useful discussions.

REFERENCES

Califano, C., Marquez-Martinez, L., and Moog, C. (2010). On linear equivalence for time-delay systems. In *American Control Conference (ACC), 2010*, 6567–6572. Baltimore, Maryland, USA. doi:10.1109/ACC.2010.5531404.

Califano, C., Marquez-Martinez, L., and Moog, C. (2011). Extended lie brackets for nonlinear time-delay systems. *Automatic Control, IEEE Transactions on*, 56(9), 2213–2218. doi:10.1109/TAC.2011.2157405.

Califano, C. and Moog, C. (2011). Feedback linear equivalence for nonlinear time delay systems. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, 4158–4163. Orlando, Florida, USA. doi:10.1109/CDC.2011.6161082.

Califano, C. and Moog, C. (2012). Canonical forms of time-delay systems. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 3862–3867. Maui, Hawai, USA. doi:10.1109/CDC.2012.6426805.

Califano, C., Marquez-Martinez, L.A., and Moog, C.H. (2013). Linearization of time-delay systems by input-output injection and output transformation. *Automatica*, 49(6), 1932 – 1940.

Gárate-García, A., Márquez-Martínez, L., Cuesta-García, J., and García-Ramírez, E. (2013). A computer algebra system for analysis and control of nonlinear time-delay systems. *Advances in Engineering Software*, 65(0), 138 – 148.

Hazewinkel, M. and Marcus, S.I. (1982). On lie algebras and finite dimensional filtering. *Stochastics*, 7(1-2), 29–62.

Isidori, A. (1995). Nonlinear Control Systems. Springer. 3^{rd} Ed. 549 p.

Margaliot, M. and Liberzon, D. (2004). A lie-algebraic condition for stability of switched nonlinear systems. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 5, 4619–4624 Vol.5. Bahamas. doi: 10.1109/CDC.2004.1429512.

Appendix A. TEST VECTORS GENERATOR

The next algorithm was used to generate the vectors for the performance test in Section 3.1.

Algorithm 1 Vector generator for the test

Input: s polynomial degree, n system dimension
Output: $\mathbf{r}(\mathbf{x_{[s]}}) \in \mathcal{K}^{n\times 1}(\delta]$

STEP 1: $\mathbb{R}^n(\delta) \ni r(\delta) \leftarrow \sum_{i=o}^{s} r_i\delta^i$ $\backslash\backslash$ $r_i \in \mathbb{R}^n$ *with entries randomly generated using "random" function from Maxima v5.27.0.*
STEP 2:
for $i = 1$ **thru** n **do**
  for $j = 1$ **thru** n **do**
    $a_{ij}(\mathbf{x}_{[s]}) \leftarrow \rho(-k, k) * x_{\rho(1,n)}(t - \rho(0, s))$
  end for
end for
$\backslash\backslash$ *where* $\rho(a, b) := random(a, b)$, *and* $a_{ij}(\mathbf{x}_{[s]})$ *are the entries of the matrix* $A(\mathbf{x}_{[s]}) \in \mathcal{K}^{n\times n}$
STEP 3: $\mathbf{r}(\mathbf{x_{[s]}}) \leftarrow A(\mathbf{x}_{[s]}).r(\delta)$

return $\mathbf{r}(\mathbf{x_{[s]}}) \in \mathcal{K}^n(\delta]$ $\backslash\backslash$ *the output is a column vector with entries randomly generated in* $\mathcal{K}^n(\delta]$
END