Linear Impulsive Systems: Lie-Algebraic Techniques and Computational Tools

A research project to the faculty of the School of Electrical Engineering and Computer Science
Russ College of Engineering and Technology, Ohio University

In partial fulfillment of the requirements for the degree of Master of Science

Nghi M. V. Nguyen
August 15, 2018

Contents

List of Figures

Abstract

This research project will study the uniform exponential stability (UES) analysis for a class of linear impulsive systems using Lie algebraic techniques. The goal of the project is the computation of a Levi decomposition of the impulsive Lie algebra in which computational tools will be developed in a particular manner such that all algorithms must preserve nice numbers-integers or rational numbers and better illustrate the underlying theory and algorithms. The traditional MATLAB commands do not meet this objective. Computational tools will be designed to satisfy this aim.

1 Introduction

An important class of hybrid systems is impulsive systems that consist of a mixture of continuous-time differential equations and discrete-time difference equations. In impulsive systems the continuous-time state variables undergo instantaneous changes or jumps caused by the nature of the systems or by the impulsive control laws (digital controllers). This phenomenon is of interest to scientists and engineers in a variety of applications. Of particular interests are sampled-data control systems and communication networks.

In this research project we investigate the uniform exponential stability (UES) analysis for a class of linear impulsive systems using Lie algebraic techniques. UES involves an exponentially decaying norm bound on the state trajectories that is uniform with respect to the initial time and sequence of impulse times. In [1] the author has successfully established the impulsive Lie algebra which is associated with the linear impulsive system and theoretically obtained sufficient conditions for the UES. From this paper, the solvability of the impulsive Lie algebra implies the UES of the system. Solvability means that every member of the Lie algebra can be represented as an upper triangular matrix in a common basis. In the case that the impulsive Lie algebra is not solvable, the Levi decomposition is employed and analyzed in which if a Levi factor in the decomposition is compact, then the system is also UES. This research project is the follow-up of these sufficient conditions in [1] in which we concentrate on computational aspects of the Levi decomposition.

Our contribution of this research is the development of MATLAB-based computational tools for the construction of the Lie algebra and related objects that preserve "nice" numbers (e.g. integers or rational numbers) in order to support the development of tutorial examples that illustrate the underlying theory.

The organization of this research report is arranged as follows; Section 1.1 will introduce linear impulsive systems, and Section 1.2 is about quadratic Lyapunov functions. Chapter 2 will provide essential mathematical background. The main part of the project will be described in Chapter 3. Chapter 4 presents an application of these tools to linear impulsive systems. Finally, Chapter 5 is the conclusion.

1.1 Linear Impulsive Systems

In this research work we investigate the uniform exponential stability analysis, using Lie-algebraic techniques, for the following class of linear impulsive systems

$$\dot{x}(t) = A_{\mathcal{C}}x(t) + B_{\mathcal{C}}u(t) \quad t \notin \mathcal{T}$$
$$x(\tau_k) = A_{\mathcal{I}}x(\tau_k^-) + B_{\mathcal{I}}u[k] \quad \tau_k \in \mathcal{T}, \; k \in \mathbb{Z}$$
$$y(t) = C_{\mathcal{C}}x(t) + D_{\mathcal{C}}u(t)$$
$$y[k] = C_{\mathcal{I}}x(\tau_k^-) + D_{\mathcal{I}}u[k]$$

in which $x(t)$ is a continuous time state governed by the differential equation that undergoes jumps (instantaneous changes) at discrete times which are assumed to be a strictly increasing sequence in the set $\mathcal{T}$. Also, $u(t)$ and $u[k]$ are continuous-time and discrete-time control inputs, respectively, and $y(t)$ and $y[k]$ are continuous-time and discrete-time outputs, respectively. The matrices $A_{\mathcal{C}}$, $B_{\mathcal{C}}$, $C_{\mathcal{C}}$, and $D_{\mathcal{C}}$ are system data that describe the continuous-time dynamics whereas the matrices $A_{\mathcal{I}}$, $B_{\mathcal{I}}$, $C_{\mathcal{I}}$, and $D_{\mathcal{I}}$ are used to specify the discrete-time dynamics.

We restrict our stability analysis to a class of homogeneous linear impulsive systems with respect to an arbitrary sequence of impulse times. Specifically, we consider the linear impulsive system as described below

$$\dot{x}(t) = A_{\mathcal{C}}x(t) \qquad t \notin \mathcal{T} \tag{1}$$
$$x(\tau_k) = A_{\mathcal{I}}x(\tau_k^-) \quad \tau_k \in \mathcal{T}$$

where $\mathcal{T}$ is an arbitrary sequence of strictly increasing impulse times.

The sate transition matrix for the linear impulsive system (1) is given by $\Phi(t_f, t_0) = e^{A_{\mathcal{C}}[t_f - t_0]}$ when $\mathcal{T} \cap (t_0, t_f] = \emptyset$ (no impulse effects from t_0 to t_f). Otherwise,

$$\Phi(t_f, t_0) = e^{A_{\mathcal{C}}[t_f - \tau_{\kappa(t_f)}]} A_{\mathcal{I}} \left(\prod_{j=\kappa(t_0)+1}^{\kappa(t_f)-1} e^{A_{\mathcal{C}}[\tau_{j+1} - \tau_j]} A_{\mathcal{I}} \right) e^{A_{\mathcal{C}}[\tau_{\kappa(t_0)+1} - t_0]}$$

where the function κ is a subsequent bookkeeping function from $\mathbb{R}$ to $\mathbb{Z}$ defined as

$$\kappa(t) = \sup\{k \in \mathbb{Z} \mid \tau_k \in \mathcal{T} \text{ and } \tau_k \le t\}$$

Definition 1. Uniform Exponential Stability (UES). (see [2]) The system (1) is said to be uniformly exponentially stable if there exist positive constants k, λ such that the state response satisfies

$$\|x(t)\| \le k e^{-\lambda(t-t_0)}\|x(t_0)\|, \quad \forall t \ge t_0$$

for all initial times t_0, all initial states $x(t_0)$, and all impulse time sets $\mathcal{T}$.

Example 1. (From [1]) Let us consider a two-dimensional impulsive system (1) with data given by

$$A_{\mathcal{C}} = \begin{bmatrix} -\mu & -\omega/\rho \\ \omega\rho & -\mu \end{bmatrix}, \qquad A_{\mathcal{I}} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

where μ, ω, and ρ are positive. The continuous-time dynamics are exponentially stable and the impulsive dynamics produce an instantaneous 90-degree counterclockwise rotation of the state trajectory at the impulse times.

We demonstrate that for the case $\rho \neq 1$ there exist values of μ and ω for which the impulsive system is not uniformly exponentially stable. Specifically, there exists an initial state and a sequence of impulse times that produce an unbounded state trajectory. Without loss of generality, we assume $\rho > 1$; the case $\rho < 1$ can be addressed in a similar way.

Let us first analyze the continuous-time dynamics. The eigenvalues of $A_{\mathcal{C}}$ are $-\mu \pm j\omega$, thus the continuous-time dynamics produce a state trajectory that spirals to the origin for any initial state. Specifically, for the initial state $x_0 = [\ 1 \quad 0\ ]^T$, the state trajectory is given by

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = e^{-\mu t} \begin{bmatrix} \cos(\omega t) \\ \rho \sin(\omega t) \end{bmatrix}, \quad t \geq 0$$

which consists of two parts. The first part is the exponentially decaying factor $e^{-\mu t}$. The second part defines a periodic trajectory with period $T = 2\pi/\omega$ that rotates in the counterclockwise direction along the ellipse

$$x_1^2 + \left(\frac{x_2}{\rho}\right)^2 = 1$$

For $\rho > 1$, the major axis of the ellipse lies along the x_2-axis and the minor axis of the ellipse lies along the x_1-axis. As the elliptical portion of the trajectory transitions from the minor axis to the major axis over a quarter of the period, its norm is expanded by the factor ρ. Specifically, the norm of the given initial state equals 1 and the norm of

$$\begin{bmatrix} \cos(\omega T/4) \\ \rho \sin(\omega T/4) \end{bmatrix} = \begin{bmatrix} \cos(\pi/2) \\ \rho \sin(\pi/2) \end{bmatrix} = \begin{bmatrix} 0 \\ \rho \end{bmatrix}$$

is equal to ρ. Over the next quarter period, the state transitions to

$$\begin{bmatrix} \cos(\omega T/2) \\ \rho \sin(\omega T/2) \end{bmatrix} = \begin{bmatrix} \cos(\pi) \\ \rho \sin(\pi) \end{bmatrix} = \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

whose norm once again equals 1. This process repeats. The expansion of the elliptical trajectory as it transitions from minor axis to major axis over a quarter of a period is immediately followed by a contraction over the next quarter of a period.

Even with the exponentially decaying factor, the norm of the continuous-time trajectory may still expand as it transitions minor axis to major axis. This occurs when

expansion by the factor $\rho > 1$ over a quarter of a period overcomes the exponential decay over that interval, that is, when $\rho e^{-\mu T/4} = \rho e^{-\mu\pi/(2\omega)} > 1$ or $\mu < \frac{2\omega}{\pi}\ln\rho$. When this is immediately followed by the ensuing contraction over the next quarter of a period along with the same exponential decay, the net effect over half a period is a decay in norm by the factor $e^{-\mu T/2} = e^{-\mu\pi/\omega}$. Ultimately, the continuous-time trajectory is norm bounded by $\rho e^{-\mu t}$ for all $t \geq 0$ and the trajectory spirals to the origin. This is depicted in Figure 1 for the parameter values

$$\rho = 3, \quad \mu = 0.25, \quad \omega = 1$$

for which $\frac{2\omega}{\pi}\ln\rho = 0.2581 > \mu$.

The continuous-time behavior described above for $\mu < \frac{2\omega}{\pi}\ln\rho$ leads to an instability for the impulsive system when the impulsive effects are timed to occur when the trajectory reaches the x_2−axis. In this situation, the net expansion as the trajectory transitions from the x_1-axis to the x_2−axis according to the continuous-time dynamics is immediately followed by an instantaneous norm-preserving rotation back to the x_1−axis produced by the impulsive dynamics, thus bypassing the contraction phase of the continuous-time dynamics. This results in an uninterrupted sequence of expansions yielding an unbounded trajectory for the impulsive system.

We conclude that for $\rho > 1$ and μ, ω satisfying $\mu < \frac{2\omega}{\pi}\ln\rho$, the impulsive system fails to be uniformly exponentially stable. Simulations illustrate this phenomenon for the values of ρ, μ, and ω given above. For the same initial state $x(0) = x_0 = [\; 1 \quad 0\;]^T$ and the impulse time set

$$\mathcal{T} = \{\tau_k = kT/4 = k\pi/2, \; k = 1, ..., 10\}$$

the phase portrait of the impulsive system is shown in Figure 2. The individual state variable responses are plotted in Figures 3 and 4. These responses depict an unbounded trajectory for the impulsive system for the specified parameter values, initial state, and impulse time set.

Furthermore, when $\rho = 1$, the impulsive system is uniformly exponentially stable for any $\mu > 0$ and $\omega > 0$ (for any initial state and any sequence of impulse times). Indeed, this is a special case of the case $\rho \neq 1$ in which there is no expansion factor. The amount $e^{-\mu t}$ of the exponentially decaying factor is always less than one for any $\mu > 0$ and $\omega > 0$, and for all $t \geq 0$. Also, the impulse effects do not change the contraction phase of the trajectories. Thus, from the analysis above for $\rho \neq 1$, the trajectories of the impulsive system spiral towards the origin without any expansion.
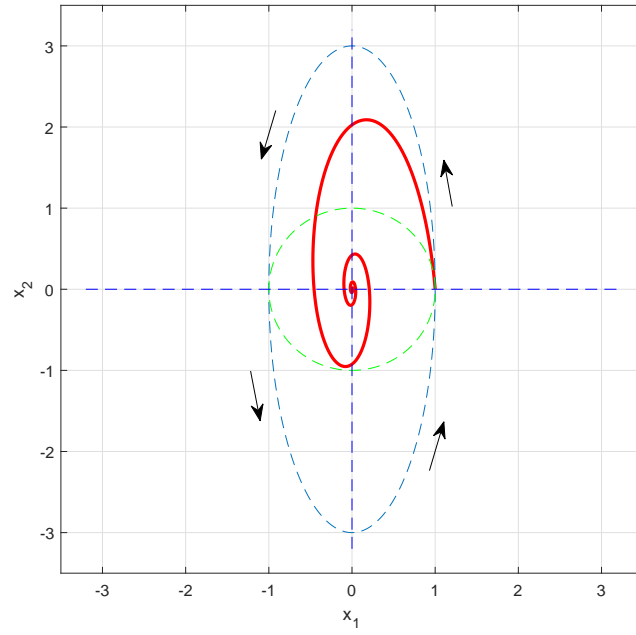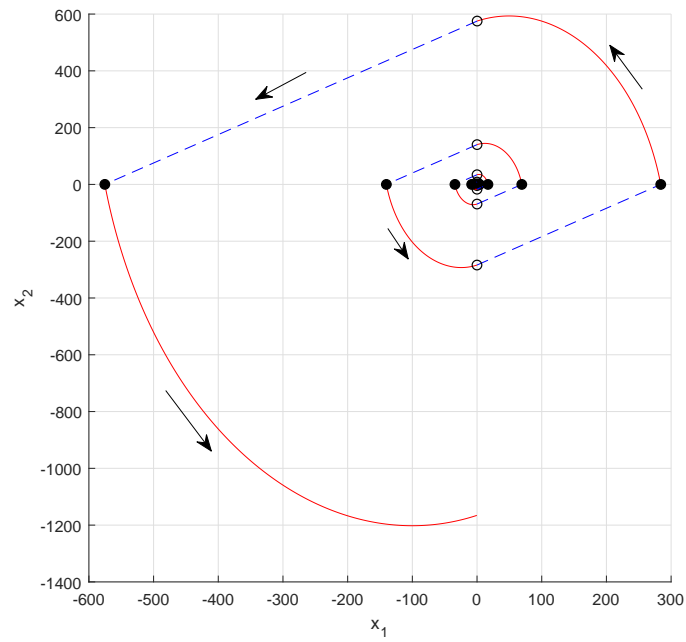
Figure 1: Phase Portrait without impulse effects
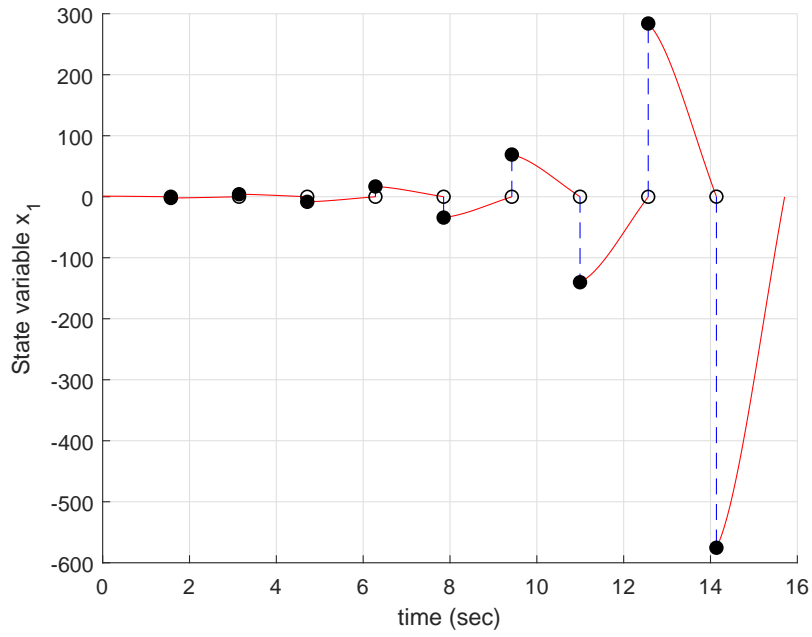

Figure 2: Phase Portrait with impulse effects

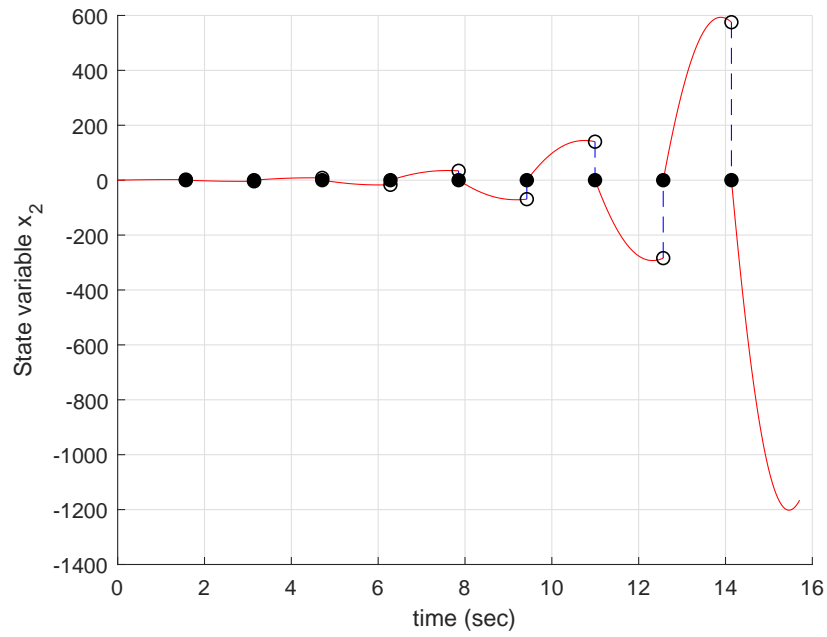Figure 3: Initial Value Response $x_1(t)$ vs t

Figure 4: Initial Value Response $x_2(t)$ vs t

9

1.2 Quadratic Lyapunov Functions

Definition 2. A symmetric matrix P is called a positive definite (semidefinite) matrix if all of its leading principal minors are positive (nonnegative). Equivalently, all eigenvalues of P are positive (nonnegative), or $x^T P x > 0$ ($x^T P x \geq 0$), $0 \neq x \in \mathbb{R}^n$. If $-x^T P x > 0$ ($-x^T P x \geq 0$), $0 \neq x \in \mathbb{R}^n$, P is called negative definite (semidefinite).

Example 2. $P = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 5 & 1 \\ -1 & 1 & 5 \end{bmatrix}$ is symmetric positive definite as verified bellow

$$p_{11} = 1 > 0, \quad p_{22} = \det \begin{bmatrix} 1 & -1 \\ -1 & 5 \end{bmatrix} = 4 > 0, \quad p_{33} = \det \begin{bmatrix} 1 & -1 & -1 \\ -1 & 5 & 1 \\ -1 & 1 & 5 \end{bmatrix} = 16 > 0.$$

Definition 3. For the continuous-time system $\dot{x}(t) = \mathrm{A}x(t)$, the function $V(x) = x^T P x$ is a quadratic Lyapunov function (QLF) for the system if P is symmetric positive definite and $\mathrm{A}^T P + P\mathrm{A} \leq 0$ (negative semidefinite).

Definition 4. The derivative of $V(x) = x^T P x$ along the trajectories of the continuous dynamics is defined by

$$\dot{V}(x) = \frac{\partial V}{\partial x}(x)\mathrm{A}(x) = x^T[\mathrm{A}^T P + P\mathrm{A}]x$$

$\dot{V}(x)$ is also a quadratic form expressed in terms of the symmetric matrix $\mathrm{A}^T P + P\mathrm{A}$.

Definition 5. For the discrete-time system $x[k+1] = \mathrm{A}x[k]$, the function $V(x) = x^T P x$ is a quadratic Lyapunov function (QLF) for the system if Q is symmetric positive definite and $\mathrm{A}^T P\mathrm{A} - P \leq 0$.

Definition 6. The difference of $V(x) = x^T P x$ along the trajectories of the discrete-time system $x[k+1] = \mathrm{A}x[k]$ is defined by

$$\Delta V(x) = x^T[\mathrm{A}^T P\mathrm{A} - P]x$$

The system (1) can be viewed as a mixed combination of continuous and discrete dynamics. Hence, it is natural to establish a Lyapunov-type criterion for the uniform exponential stability of (1). Precisely, the result is presented in the following theorem.

Theorem 1. *(see [2]) If there exists a common symmetric positive definite matrix P that satisfies the inequalities*

$$\mathrm{A}_{\mathcal{C}}^T P + P\mathrm{A}_{\mathcal{C}} < 0 \tag{2}$$
$$\mathrm{A}_{\mathcal{I}}^T P\mathrm{A}_{\mathcal{I}} - P < 0$$

Then, the system (1) is uniformly exponentially stable. Let $V(x) = x^T P x$. Its derivative along the trajectories satisfies

$$\dot{V}[x(t)] < 0, \ t \in [\tau_k, \tau_{k+1}), \quad V[x(\tau_k)] - V[x(\tau_k^-)] < 0, \quad \forall \ t_0, \ \forall \ \mathcal{T}.$$

Thus, $V(x)$ can be thought of as a common quadratic Lyapunov function for (1).

2 Mathematical Background

In this section we will present fundamental concepts and important results in Lie algebra theory that relate to the study of uniform exponential stability of linear impulsive systems.

2.1 Matrix Lie Algebras

Definition 7. Let $\mathbb{F}$ be a field ($\mathbb{C}$ or $\mathbb{R}$). An $\mathbb{F}$-vector space L along with a bilinear function $[\cdot, \cdot]$, the Lie bracket, is called a Lie algebra if $(\mathsf{L}, [\cdot, \cdot])$ satisfies the following postulates

$$[X, X] = 0 \quad \forall X \in \mathsf{L},$$
$$[X, [Y, Z]] + [Y, [Z, X]] + [Z, [X, Y]] = 0 \quad \forall X, Y, Z \in \mathsf{L}.$$

Example 3. Let us consider the vector space of all $n \times n$ matrices over $\mathbb{C}$, denoted by $\mathfrak{gl}(n, \mathbb{C})$, with the Lie bracket defined by

$$[X, Y] = XY - YX$$

in which XY is the normal product of the two matrices X and Y. It is not difficult to check this Lie bracket satisfies the two postulates in Definition 6. Thus, $\mathfrak{gl}(n, \mathbb{C})$ is a Lie algebra.

Definition 8. The Lie algebra $\mathfrak{gl}(n, \mathbb{C})$ is called the general linear algebra.

Definition 9. Let X be a fixed element of a Lie algebra L. The adjoint operator ad_X is a linear operator from L into L defined by

$$\mathrm{ad}_X : \mathsf{L} \longrightarrow \mathsf{L}$$
$$\mathrm{ad}_X Y = [X, Y]$$

Definition 10. Let L be a Lie algebra. A Lie subalgebra S of L is a vector subspace of L that is closed under the Lie bracket, i.e.,

$$[X, Y] \in \mathsf{S}, \quad \forall X, Y \in \mathsf{S}.$$

A Lie subalgebra of the general linear algebra $\mathfrak{gl}(n, \mathbb{C})$ is called a matrix Lie algebra.

Example 4. (see [9]) The subspace of all upper triangular matrices, $\mathsf{b}(n,\mathbb{C})$, and the subspace of all strictly upper triangular matrices, $\mathsf{n}(n,\mathbb{C})$, of $\mathfrak{gl}(n,\mathbb{C})$ are matrix Lie algebras.

Example 5. The special Lie algebra $\mathfrak{sl}(n,\mathbb{C})$, defined by

$$\mathfrak{sl}(n,\mathbb{C}) = \{X \in \mathfrak{gl}(n,\mathbb{C}) \mid \mathsf{tr}(X) = 0\}$$

is also a Lie subalgebra of $\mathfrak{gl}(n,\mathbb{C})$ and so is a matrix Lie algebra.

Definition 11. An ideal I of a Lie algebra L is a subspace of L such that $[\mathsf{L},\mathsf{I}] \subset \mathsf{I}$. Obviously, I is also a Lie subalgebra of L.

Example 6. The special Lie algebra $\mathfrak{sl}(n,\mathbb{C})$ is an ideal of $\mathfrak{gl}(n,\mathbb{C})$. The Lie algebra $\mathsf{n}(n,\mathbb{C})$ is an ideal of $\mathsf{b}(n,\mathbb{C})$. But $\mathsf{b}(n,\mathbb{C})$ is not an ideal of $\mathfrak{gl}(n,\mathbb{C})$.

Let L be a d-dimensional Lie algebra and let $\mathfrak{B} = \{X_1, X_2, \ldots, X_d\}$ denote a basis for L (as a vector space). It is possible to express Lie brackets of each pair of basis vectors (elements of $\mathfrak{B}$) as linear combinations of themselves. In other words, there exist scalars $c_{k,j}^i$ for which

$$[X_k, X_j] = \sum_{i=1}^{d} c_{k,j}^i X_i, \quad 1 \leq k,j \leq d$$

Definition 12. The scalars $c_{k,j}^i$ in the expression above are called the structure constants of L. The number of structure constants is d^3. Also, the structure constants do depend on the basis of L.

For any fixed k, $1 \leq k \leq d$, let us rewrite the expression above in operator form as

$$\mathrm{ad}_{X_k} X_j = \sum_{i=1}^{d} c_{k,j}^i X_i, \quad 1 \leq j \leq d$$

This defines a matrix representation for the adjoint operator ad_{X_k} as below.

Definition 13. A matrix representation of ad_{X_k} is $[\mathrm{ad}_{X_k}] = [c_{k,j}^i]$ in which i is the row index and j is the column index.

Definition 14. (see [9]) The Killing form on a Lie algebra L is the symmetric bilinear form $\mathcal{K} : \mathsf{L} \times \mathsf{L} \to \mathbb{F}$ defined by

$$\mathcal{K}(X,Y) = \mathsf{tr}(\mathrm{ad}_X \circ \mathrm{ad}_Y), \ X, Y \in \mathsf{L}$$

in which ad_X and ad_Y denote the adjoint operators with respect to X and Y, respectively. Also, the Killing form satisfies the invariance property

$$\mathcal{K}(X,[Y,Z]) = \mathcal{K}([X,Y],Z), \ X,Y,Z \in \mathsf{L}$$

Example 7.(From [12]) Let L be a 2-dimensional Lie algebra with a basis $\{X_1, X_2\}$. Define the Lie bracket on the basis as

$$[X_1, X_2] = X_1$$

Recall that $[X_1, X_1] = 0$ and $[X_2, X_2] = 0$. To find the Killing form on L, we first calculate adjoint operators with respect to X_1 and X_2. From Definition 9, we have

$$\text{ad}_{X_1} X_1 = [X_1, X_1] = 0, \quad \text{ad}_{X_1} X_2 = [X_1, X_2] = X_1$$

$$\text{ad}_{X_2} X_1 = [X_2, X_1] = -X_1, \quad \text{ad}_{X_2} X_2 = [X_2, X_2] = 0$$

Thus, the matrix representations for ad_{X_1} and ad_{X_2} with respect to the given basis are

$$[\text{ad}_{X_1}] = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad [\text{ad}_{X_2}] = \begin{bmatrix} -1 & 0 \\ 0 & 0 \end{bmatrix}$$

We now compute the Killing form

$$\mathcal{K}(X_1, X_1) = \text{tr}(\text{ad}_{X_1} \circ \text{ad}_{X_1}) = 0$$
$$\mathcal{K}(X_1, X_2) = \text{tr}(\text{ad}_{X_1} \circ \text{ad}_{X_2}) = 0$$
$$\mathcal{K}(X_2, X_1) = \text{tr}(\text{ad}_{X_2} \circ \text{ad}_{X_1}) = 0$$
$$\mathcal{K}(X_2, X_2) = \text{tr}(\text{ad}_{X_2} \circ \text{ad}_{X_2}) = 1$$

The matrix representation for $\mathcal{K}$ is $[\mathcal{K}] = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ which is positive semi-definite.

Next, we introduce the concept of orthogonal subspaces of L with respect to the Killing form.

Definition 15. Let $\mathcal{K}$ be a Killing form on L and S be a subspace of L. The orthogonal (perpendicular) subspace to S is defined by

$$\mathsf{S}^\perp = \{X \in \mathsf{L} \mid \mathcal{K}(X, S) = 0, \ \forall S \in \mathsf{S}\}$$

We note that if I is an ideal of L, then the orthogonal subspace $\mathsf{I}^\perp$ to I is also an ideal of L as a consequence of the invariance property of the Killing form (see [9]).

2.2 Impulsive Lie Algebra

We now briefly introduce the impulsive Lie algebra associated with the impulsive system (1). Let $\mathcal{X} = \mathbb{C}^n$ be the underlying vector space. The impulsive Lie algebra L for system (1) is a Lie subalgebra of the general matrix algebra $\mathfrak{gl}(n, \mathbb{C})$ generated by the following set (see [1])

$$\mathfrak{G} = \{\mathrm{A}_k = \mathrm{A}_\mathcal{I}^{-k} \mathrm{A}_\mathcal{C} \mathrm{A}_\mathcal{I}^k, \quad k \geq 0\}$$

13

in which $A_{\mathcal{C}} \in \mathfrak{gl}(n, \mathbb{C})$, and $A_{\mathcal{I}}$ is assumed to be invertible or $A_{\mathcal{I}} \in GL(n, \mathbb{C})$, the general linear group. Specifically, L is the $\mathbb{C}$-linear span of elements of $\mathfrak{G}$ of the form

$$[A_{k_r}, [A_{k_{l-1}}, [..., [A_{k_2}, A_{k_1}]...]]], \ A_{k_j} \in \mathfrak{G}, \ j = 1, ..., r \text{ and } r > 0.$$

Geometrically, L can be characterized in terms of a subspace of $\mathfrak{gl}(n, \mathbb{C})$ that is invariant with respect to certain maps. For this purpose we employ the adjoint operator and the adjoint map that have this property.

Definition 16. Let $A \in \mathfrak{gl}(n, \mathbb{C})$ and $g \in GL(n, \mathbb{C})$. The adjoint operator ad_A and the adjoint map Ad_g are defined respectively by

$$\mathrm{ad}_A : \mathfrak{gl}(n, \mathbb{C}) \longrightarrow \mathfrak{gl}(n, \mathbb{C})$$

$$\mathrm{ad}_A X = [A, X]$$

$$\mathrm{Ad}_g : \mathfrak{gl}(n, \mathbb{C}) \longrightarrow \mathfrak{gl}(n, \mathbb{C})$$

$$\mathrm{Ad}_g X = g X g^{-1}$$

With respect to these two maps for $A_{\mathcal{C}} \in \mathfrak{gl}(n, \mathbb{C})$ and $A_{\mathcal{I}} \in GL(n, \mathbb{C})$, L is the smallest *subspace* of $\mathfrak{gl}(n, \mathbb{C})$ that is invariant under both $\mathrm{ad}_{A_{\mathcal{C}}}$ and $\mathrm{Ad}_{A_{\mathcal{I}}}$ and contains $\mathrm{span}\{A_{\mathcal{C}}\}$. From [1], L can be recursively computed by the following algorithm

$$L_0 = \langle \mathrm{Ad}_{A_{\mathcal{I}}} \mid \mathrm{span}\{A_{\mathcal{C}}\} \rangle$$
$$L_k = L_{k-1} + \langle \mathrm{Ad}_{A_{\mathcal{I}}} \mid \mathrm{ad}_{A_{\mathcal{C}}} L_{k-1} \rangle \quad k \geq 1$$

in which the notation $\langle \mathrm{Ad}_{A_{\mathcal{I}}} \mid \mathrm{span}\{A_{\mathcal{C}}\} \rangle$ means the minimal (smallest) subspace that is invariant under the map $\mathrm{Ad}_{A_{\mathcal{I}}}$ and contains $\mathrm{span}\{A_{\mathcal{C}}\}$.

2.3 Solvable, Nilpotent, and Abelian Lie Algebras

Definition 17. Let L be a Lie algebra. The derived series of L is the series of terms

$$L^{(k)} = [L^{(k-1)}, L^{(k-1)}], \quad k > 1.$$

The derived series of L is a decreasing series of ideals, i.e., $L \supset L^{(1)} \supset L^{(2)} \supset ...$

Definition 18. The Lie algebra L is said to be solvable if there exists an integer $r \geq 1$ such that $L^{(r)} = 0$.

Example 8. The Lie algebra $L = \mathfrak{b}(n, \mathbb{C})$ of upper triangular matrices over $\mathbb{C}$ is a solvable algebra.

Definition 19. The lower central series of a Lie algebra L is the series of terms

$$L^1 = [L, L]$$
$$L^k = [L, L^{k-1}], \quad k > 1.$$

The lower central series of L is also a decreasing series of ideals i.e., $\mathsf{L} \supset \mathsf{L}^1 \supset \mathsf{L}^2 \supset ...$

The first term in both the derived and lower central series $\mathsf{L}^{(1)} = \mathsf{L}^1 = [\mathsf{L}, \mathsf{L}]$ is called the derived algebra.

Definition 20. The Lie algebra L is said to be nilpotent if there exists an integer $r \geq 1$ such that $\mathsf{L}^r = 0$. If L is nilpotent then L is solvable.

Example 9. The Lie algebra $\mathsf{L} = \mathsf{n}(n, \mathbb{C})$ of strictly upper triangular matrices over $\mathbb{C}$ is a nilpotent algebra. Furthermore, if $n \geq 2$ then $\mathsf{b}(n, \mathbb{C})$ is solvable but not nilpotent.

Definition 21. A Lie algebra L is said to be Abelian or commutative if $[\mathsf{L}, \mathsf{L}] = 0$. Obviously, if L is Abelian then L is nilpotent and solvable.

Definition 22. Let L be a Lie algebra. The center of L is an ideal of L defined by

$$\mathcal{C}(\mathsf{L}) = \{x \in \mathsf{L} \mid [x, \mathsf{L}] = 0\}$$

Example 10. The center $\mathcal{C}(\mathsf{L})$ is an Abelian Lie algebra.

Definition 23. Let L be a Lie algebra. The radical of L, denoted by $\mathsf{Rad}(\mathsf{L})$, is the maximal solvable ideal of L, i.e. the unique solvable ideal that contains all other solvable ideals.

Definition 24. A Lie algebra L is said to be semisimple if it has no non-zero solvable ideal, i.e. its only solvable ideal is zero.

Remark 1. If the radical of L is equal to L, L is solvable, and if $\mathsf{Rad}(\mathsf{L})$ is zero, then L is semisimple.

Remark 2. The Killing form of a semisimple Lie algebra L is nondegenerate, i.e. $\mathcal{K}(X, X) \neq 0$ for all nonzero $X \in \mathsf{L}$.

Definition 25. A semisimple Lie algebra L is compact if its Killing form is negative definite, i.e. $\mathcal{K}(X, X) < 0$ for all nonzero $X \in \mathsf{L}$.

Definition 26. If the radical of a Lie algebra L is also its center, i.e. $\mathsf{Rad}(\mathsf{L}) = \mathcal{C}(\mathsf{L})$, then L is said to be reductive.

The following theorem actually provides an algorithm for computing the radical.

Theorem 2. *(see [7,8]) The radical of a Lie algebra L can be characterized through the Killing form as*

$$\mathsf{Rad}(\mathsf{L}) = \{X \in \mathsf{L} \mid \mathcal{K}(X, Y) = 0, \ \forall \ Y \in [\mathsf{L}, \mathsf{L}]\} = [\mathsf{L}, \mathsf{L}]^{\perp}$$

2.4   Levi Decomposition

The structure of solvable Lie algebras is somewhat simple (similar to simultaneously upper triangular matrices) but not every Lie algebra is solvable in general. Fortunately, for any Lie algebra L, there exists a semisimple subalgebra S of L such that L can be decomposed as a direct sum of its radical and S. This is the content of Levi's Theorem.

Theorem 3. *(**Levi's Theorem**)(see [7,10]) For any Lie algebra L, there exists a semisimple subalgebra S of L such that L = Rad(L) ⊕ S. S is called a Levi factor or a Levi subalgebra of L.*

The remainder of this document will present algorithms and related knowledge for computing a Levi decomposition of the impulsive Lie algebra L in the reductive case.

3   Algorithms

This section includes the main content of the document. We first introduce Kronecker product of two matrices. Then matrix representations for ad_A and Ad_g follow. In the last subsection a sequence of algorithms will be described in which the algorithm for computing a Levi factor is the last one.

3.1   Kronecker Product

Let A be an m_1-by-n_1 matrix and **B** an m_2-by-n_2 (See [11]).

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n_1} \\ a_{21} & a_{22} & \dots & a_{2n_1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m_11} & a_{m_12} & \dots & a_{m_1n_1} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n_2} \\ b_{21} & b_{22} & \dots & b_{2n_2} \\ \vdots & \vdots & \ddots & \vdots \\ b_{m_21} & b_{m_22} & \dots & b_{m_2n_2} \end{bmatrix}.$$

The Kronecker product of A and **B** is an m_1m_2-by-n_1n_2 block matrix denoted by $A \otimes \mathbf{B}$ and defined by

$$A \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \dots & a_{1n_1}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \dots & a_{2n_1}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m_11}\mathbf{B} & a_{m_12}\mathbf{B} & \dots & a_{m_1n_1}\mathbf{B} \end{bmatrix}$$

Specifically, each block (m_2-by-n_2) of $A \otimes \mathbf{B}$ is made up by multiplying **B** by each element of A.

Example 11. $A = \begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix}$, $B = \begin{bmatrix} 2 & 0 \\ 1 & -1 \end{bmatrix}$. The Kronecker product of A and B is

$$A \otimes B = \left[\begin{array}{c|c} 1 \cdot \begin{bmatrix} 2 & 0 \\ 1 & -1 \end{bmatrix} & -1 \cdot \begin{bmatrix} 2 & 0 \\ 1 & -1 \end{bmatrix} \\ \hline 0 \cdot \begin{bmatrix} 2 & 0 \\ 1 & -1 \end{bmatrix} & 2 \cdot \begin{bmatrix} 2 & 0 \\ 1 & -1 \end{bmatrix} \end{array} \right] = \begin{bmatrix} 2 & 0 & -2 & 0 \\ 1 & -1 & -1 & 1 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 2 & -2 \end{bmatrix}$$

3.2 Matrix Representation for ad_A and Ad_g

Let us consider the vector space of all $n \times n$ matrices over $\mathbb{C}$, $\mathfrak{gl}(n, \mathbb{C})$ or $\mathbb{C}^{n \times n}$ and the vector space of all $n^2 \times 1$ vectors over $\mathbb{C}$, $\mathbb{C}^{n^2}$. If we rearrange each $n \times n$ matrix to $n^2 \times 1$ vector by putting each column of the matrix into a single column one after another, then this arrangement yields a one-to-one relation and preserves the structure between the two spaces $\mathbb{C}^{n \times n}$ and $\mathbb{C}^{n^2}$. Precisely, this relation is an isomorphism (e.g. a bijective map that preserves the structure of the two spaces) between the two spaces as verified below. Let denote $\mathrm{vec}(\cdot)$ the isomorphism.

$$\mathrm{vec} : \mathbb{C}^{n \times n} \longrightarrow \mathbb{C}^{n^2}$$

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \longmapsto \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \\ \hline \vdots \\ \hline a_{1n} \\ a_{2n} \\ \vdots \\ a_{nn} \end{bmatrix}$$

It is obvious that $\mathrm{vec}(\cdot)$ is one-to-one and onto. Let α, $\beta \in \mathbb{C}$ and A, B $\in \mathbb{C}^{n \times n}$.

$$\mathrm{vec}(\alpha \mathrm{A} + \beta \mathrm{B}) = \mathrm{vec}\left(\alpha \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} + \beta \begin{bmatrix} b_{11} & \cdots & b_{1n} \\ b_{21} & \cdots & b_{2n} \\ \vdots & \ddots & \vdots \\ b_{n1} & \cdots & b_{nn} \end{bmatrix} \right)$$

$$= \mathrm{vec}\left(\begin{bmatrix} \alpha a_{11} + \beta b_{11} & \cdots & \alpha a_{1n} + \beta b_{1n} \\ \alpha a_{21} + \beta b_{21} & \cdots & \alpha a_{2n} + \beta b_{2n} \\ \vdots & \ddots & \vdots \\ \alpha a_{n1} + \beta b_{n1} & \cdots & \alpha a_{nn} + \beta b_{nn} \end{bmatrix} \right)$$

$$\mathrm{vec}(\alpha \mathrm{A} + \beta \mathrm{B}) = \begin{bmatrix} \alpha a_{11} + \beta b_{11} \\ \vdots \\ \alpha a_{n1} + \beta b_{n1} \\ \hline \vdots \\ \hline \alpha a_{1n} + \beta b_{1n} \\ \vdots \\ \alpha a_{nn} + \beta b_{nn} \end{bmatrix} = \begin{bmatrix} \alpha a_{11} \\ \vdots \\ \alpha a_{n1} \\ \hline \vdots \\ \hline \alpha a_{1n} \\ \vdots \\ \alpha a_{nn} \end{bmatrix} + \begin{bmatrix} \beta b_{11} \\ \vdots \\ \beta b_{n1} \\ \hline \vdots \\ \hline \beta b_{1n} \\ \vdots \\ \beta b_{nn} \end{bmatrix} = \alpha \begin{bmatrix} a_{11} \\ \vdots \\ a_{n1} \\ \hline \vdots \\ \hline a_{1n} \\ \vdots \\ a_{nn} \end{bmatrix} + \beta \begin{bmatrix} b_{11} \\ \vdots \\ b_{n1} \\ \hline \vdots \\ \hline b_{1n} \\ \vdots \\ b_{nn} \end{bmatrix}$$

$$= \alpha \mathrm{vec}(\mathrm{A}) + \beta \mathrm{vec}(\mathrm{B}).$$

Conversely, the inverse of $\text{vec}(\cdot)$, denoted by $\text{vec}^{-1}(\cdot)$ maps each $n^2 \times 1$ vector back to an $n \times n$ matrix.

Recall that for $A \in \mathfrak{gl}(n, \mathbb{C})$, the adjoint operator ad_A from $\mathfrak{gl}(n, \mathbb{C})$ into $\mathfrak{gl}(n, \mathbb{C})$ is defined by

$$\text{ad}_A X = [A, X], \quad X \in \mathfrak{gl}(n, \mathbb{C}),$$

and for $g \in \text{GL}(n, \mathbb{C})$, the adjoint map Ad_g from $\mathfrak{gl}(n, \mathbb{C})$ into $\mathfrak{gl}(n, \mathbb{C})$ defined by

$$\text{Ad}_g X = gXg^{-1}, \quad X \in \mathfrak{gl}(n, \mathbb{C}).$$

With the isomorphism $\text{vec}(\cdot)$ and Kronecker product we now can introduce a matrix representation for ad_A and Ad_g as in the following theorem.

Theorem 4. *(See [8]) The matrix representations (with respect to a certain basis of $\mathfrak{gl}(n, \mathbb{C})$) for ad_A and Ad_g are formulated by*

$$[\text{ad}_A] = I_n \otimes A - A^T \otimes I_n \quad and \quad \text{ad}_A X = \text{vec}^{-1}\left([I_n \otimes A - A^T \otimes I_n]\text{vec}(X)\right),$$

$$[\text{Ad}_g] = g^{-T} \otimes g \quad and \quad \text{Ad}_g X = \text{vec}^{-1}\left([g^{-T} \otimes g]\text{vec}(X)\right), \quad X \in \mathbb{C}^{n \times n}.$$

Remark 3. $[\text{Ad}_{g^{-1}}] = g^T \otimes g^{-1} = [\text{Ad}_g]^{-1}$.

Example 12. $A = \begin{bmatrix} 1 & -1 \\ 0 & 2 \end{bmatrix}$, $g = \begin{bmatrix} 1 & 0 \\ 2 & -1 \end{bmatrix}$. Using these formula gives matrix representations for ad_A and Ad_g as

$$[\text{ad}_A] = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad [\text{Ad}_g] = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 2 & -1 & 4 & -2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & -2 & 1 \end{bmatrix}$$

We also have

$$[\text{Ad}_{g^{-1}}] = g^T \otimes g^{-1} = \begin{bmatrix} 1 & 0 & 2 & 0 \\ 2 & -1 & 4 & -2 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & -2 & 1 \end{bmatrix} = [\text{Ad}_g]^{-1}$$

3.3 Algorithm for The Impulsive Lie Algebra

Rewrite the linear impulsive system as below

$$\dot{x}(t) = A_{\mathcal{C}}x(t) \qquad t \notin \mathcal{T}$$
$$x(\tau_k) = A_{\mathcal{I}}x(t) \qquad \tau_k \in \mathcal{T}$$

Let replace $A_{\mathcal{C}}$ by A, $A_{\mathcal{I}}$ by g and recall that L can be iteratively computed as

$$\mathsf{L}_0 = \langle \mathrm{Ad}_g \mid \mathrm{span}\{A\}\rangle$$
$$\mathsf{L}_k = \mathsf{L}_{k-1} + \langle \mathrm{Ad}_g \mid \mathrm{ad}_A \mathsf{L}_{k-1}\rangle \quad k \geq 1$$

We first introduce an algorithm for computing a minimal subspace that is invariant under a linear map and contains a given subspace. Let $\mathcal{X}$ be a vector space, $\mathrm{T} : \mathcal{X} \to \mathcal{X}$ a linear map and $\mathcal{B}$ a subspace of $\mathcal{X}$ ($\mathcal{B}$ is normally the image space of a certain linear map). Let denote $\langle \mathrm{T} \mid \mathcal{B}\rangle$ the mentioned invariant subspace. An algorithm for computing $\langle \mathrm{T} \mid \mathcal{B}\rangle$ is described in Algorithm 1, using the following recursion.

$$M_0 = \mathcal{B}$$
$$M_k = M_{k-1} + \mathrm{T}M_{k-1}$$

Remark 4. Recall that for an $n \times n$ matrix A and an $n \times 1$ vector b if

$$q = \mathrm{rank}\, [\, b \quad Ab \quad A^2 b \quad \ldots \quad A^{n-1}b \,] \leq n$$

then the first q columns $\{b, Ab, A^2 b, \ldots, A^{q-1}b\}$ are linearly independent.

Algorithm 1 - $\langle \mathrm{T} \mid \mathcal{B}\rangle$ - Minimal T-invariant containing $\mathcal{B}$

Input: Matrix T and matrix B (each column is a basis vector of $\mathcal{B}$)
   Initialization. $M_0 = B$, $k = 1$.
repeat
      Step 1. Compute $\mathrm{T}M_{k-1}$.
      Step 2. Augment M_{k-1} by $\mathrm{T}M_{k-1}$ to form $[M_{k-1} \quad \mathrm{T}M_{k-1}]$.
      Step 3. Compute a basis matrix M_k for $[M_{k-1} \quad \mathrm{T}M_{k-1}]$.
      $k = k + 1$.
until $\mathrm{rank}(M_k) = \mathrm{rank}(M_{k-1})$.
Output: A basis matrix M for $\langle \mathrm{T} \mid \mathcal{B}\rangle$

We now can compute the impulsive Lie algebra L. We first convert matrices A and g into column vectors by using the operator $\mathrm{vec}(\cdot)$. Next, we apply Algorithm 1 for calculating $\langle \mathrm{Ad}_g \mid \mathrm{span}\{\mathrm{vec}(A)\}\rangle$ and then repeat the process. The details are described in Algorithm 2 as below.

Algorithm 2 - Impulsive Lie algebra L

Input: Matrix A and matrix g (invertible)

    Step 1. Compute matrix representations for $\mathrm{Ad_g}$ and $\mathrm{ad_A}$ using Theorem 2.

    Step 2. Compute $\mathsf{L}_0 = \langle \mathrm{Ad_g} \mid \mathrm{span}\{A\} \rangle$ by using Algorithm 1.

    Initialization. $\mathsf{L}_0 = \mathsf{L}_0$, $k = 1$.

repeat

    Step 3. Compute $\mathrm{ad_A}\mathsf{L}_{k-1}$ then apply Algorithm 1 to calculate

$$M_{k-1} = \langle \mathrm{Ad_g} \mid \mathrm{ad_A}\mathsf{L}_{k-1} \rangle$$

    Step 4. Augment L_{k-1} by M_{k-1}.

    Step 5. Compute a basis matrix L_k for $[\mathsf{L}_{k-1} \quad M_{k-1}]$ ($\mathsf{L}_k = \mathsf{L}_{k-1} + M_{k-1}$).

    $k = k + 1$.

until $\mathsf{L}_k = \mathsf{L}_{k-1}$.

Output: A basis matrix for L, $\mathfrak{B} = \{X_1, X_2, \ldots, X_d\}$

3.4   Algorithm for Radical

Recall from Theorem 2 that the radical of L is calculated by

$$\mathsf{Rad}(\mathsf{L}) = \{X \in \mathsf{L} \mid \mathcal{K}(X, Y) = 0, \ \forall \ Y \in [\mathsf{L}, \mathsf{L}]\}$$
$$= [\mathsf{L}, \mathsf{L}]^{\perp}$$

or the radical is the orthogonal subspace to $[\mathsf{L}, \mathsf{L}]$ with respect to the Killing form.

Algorithm 3 - The radical of L

Input: A basis $\mathfrak{B} = \{X_1, X_2, \ldots, X_d\}$ for Lie algebra L

    Step 1. Compute the derived algebra $[\mathsf{L}, \mathsf{L}]$ of L by finding all Lie brackets $[X_k, X_j]$ for $X_k, X_j \in \mathfrak{B}$, $1 \leq i \leq d-1$, $i \leq j \leq d$. Then obtain a basis $\mathfrak{B}^{(1)} = \{Y_1, Y_2, \ldots, Y_{d_1}\}$ for $[\mathsf{L}, \mathsf{L}]$.

    Step 2. Compute all matrices $[\mathrm{ad}_{X_k}]$ and $[\mathrm{ad}_{Y_j}]$ for ad_{X_k} with $X_k \in \mathfrak{B}$ and ad_{Y_j} with $Y_j \in \mathfrak{B}^{(1)}$, respectively using Definition 12.

    Step 3. Compute a matrix $[\mathcal{K}]$ for the Killing form $\mathcal{K} = \mathrm{tr}(\mathrm{ad}_{X_k} \circ \mathrm{ad}_{Y_j})$, $X_k \in \mathfrak{B}$, $Y_j \in \mathfrak{B}^{(1)}$ following Step 2.

    Step 4. Compute the radical $\mathsf{Rad}(\mathsf{L})$ to be the orthogonal complement of $[\mathsf{L}, \mathsf{L}]$, with respect to $[\mathcal{K}]$, i.e.,
$$\mathsf{Rad}(\mathsf{L}) = [\mathsf{L}, \mathsf{L}]^{\perp}$$

Output: A basis for $\mathsf{R} = \mathsf{Rad}(\mathsf{L})$, $\mathfrak{B}_{\mathsf{R}} = \{R_1, R_2, \ldots, R_n\}$

3.5 Algorithm for Levi Factor

Recall from Levi's Theorem that $L = Rad(L) \oplus S$. In this research project we will restrict to the case that the impulsive Lie algebra is reductive ($Rad(L) = \mathcal{C}(L)$). We follow [6] in order to compute a Levi factor for L. Specifically, the algorithm is presented in Algorithm 4 below.

Algorithm 4 - Levi factor of L

Input: A basis $\mathfrak{B} = \{X_1, X_2, \ldots, X_d\}$ for Lie algebra L

Step 1. Compute a basis $\mathfrak{B}_R = \{R_1, R_2, \ldots, R_n\}$ for the radical $R = Rad(L)$.

Step 2. Compute a complement subspace to R in L, let $\mathfrak{B}_C = \{Z_1, Z_2, \ldots, Z_m\}$ be a basis for the complement.

Step 3. Update a new basis for L, $\mathfrak{B}_C \cup \mathfrak{B}_R$. Then compute structure constants for L with respect to the new basis and compute matrices $[\text{ad}_{X_i}]$ with $X_i \in \mathfrak{B}_C \cup \mathfrak{B}_R$.

Step 4. Obtain a basis $\bar{\mathfrak{B}}_R = \{\bar{Z}_1, \bar{Z}_2, \ldots, \bar{Z}_m\}$ for the quotient algebra L/R. The structure constants for L and the matrices $[\text{ad}_{X_k}]$ in Step 3 induce structure constants c_{ij}^k and matrices $[\text{ad}_{\bar{Z}_i}]$ for L/R, in which $\bar{Z}_i \in \bar{\mathfrak{B}}_R$, $1 \leq i \leq m$.

Step 5. For each i, $1 \leq i \leq m$, set the equation

$$Y_i = Z_i + \sum_{j=1}^{n} a_{ij} R_j \tag{3}$$

Since L is reductive, this equation is equivalent to

$$\sum_{k=1}^{m} \sum_{p=1}^{n} c_{ij}^k a_{kp} R_p = [Z_i, Z_j] - \sum_{k=1}^{m} c_{ij}^k Z_k, \quad 1 \leq i, j \leq m \tag{4}$$

where c_{ij}^k are from Step 4 because Y_i and $\bar{Z}_i$ must have the same Lie brackets. There are $m(m-1)/2$ equations with mn unknowns a_{kp} in (4).

Step 5.1. Compute the right-hand side of (4) and transform its left-hand side into a matrix form with respect to the unknowns a_{kp}, $1 \leq k \leq m$, $1 \leq p \leq n$.

Step 5.2. For each value of the right-hand side, there is a corresponding equation with the unknowns a_{kp}, $1 \leq k \leq m$, $1 \leq p \leq n$.

Step 5.3. Obtain a system of linear equations from Step 5.2. Then solve it for a_{kp}, $1 \leq k \leq m$, $1 \leq p \leq n$. Note that from Theorem 3 (Levi's Theorem) this system must has a solution.

Step 6. Substitute a_{kp} back into (3) in Step 5 to obtain $\{Y_1, Y_2, \ldots, Y_m\}$.

Output: A basis for a Levi factor S, $\mathfrak{B}_S = \{Y_1, Y_2, \ldots, Y_m\}$

Example 13. (From [6]) We now consider an example for computing a Levi factor

of a Lie algebra following Algorithm 4. Let L be the Lie algebra with basis

$$\mathfrak{B} = \{X_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; X_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; X_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; X_4 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\}$$

First, we compute the radical $\mathsf{Rad}(\mathsf{L})$ of L by using Algorithm 3. Running this Algorithm yields $\mathsf{Rad}(\mathsf{L})$, spanned by

$$\mathfrak{B}_{\mathsf{R}} = \{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}\}$$

Direct calculation shows $[\mathfrak{B}, \mathfrak{B}_{\mathsf{R}}] = 0$ or $[\mathsf{L}, \mathsf{Rad}(\mathsf{L})] = 0$. Hence, the radical is also the center of L, $\mathsf{Rad}(\mathsf{L}) = \mathcal{C}(\mathsf{L})$, and the Lie algebra is reductive. Next, implementing Algorithm 4 produces a Levi factor S of L, spanned by

$$\mathfrak{B}_{\mathsf{S}} = \{\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; \begin{bmatrix} 0 & 2 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}\}$$

It is obvious from the bases of the Lie algebra, the radical, and the Levi factor that $\mathsf{L} = \mathsf{Rad}(\mathsf{L}) \oplus \mathsf{S}$.

4 Application to Impulsive Systems

Throughout this section we assume that the eigenvalues of $A_{\mathcal{C}}$ lie in the open left-half of the complex plane and the eigenvalues of $A_{\mathcal{I}}$ lie in the interior of the unit circle in the complex plane.

Theorem 5. *([1, Theorem 4.4]) If the impulsive Lie algebra is solvable, then the linear impulsive system (1) is UES.*

Example 14. Let us consider a 3-dimensional linear impulsive system in the form of (1) with data matrices $A_{\mathcal{C}}$ and $A_{\mathcal{I}}$ given as below

$$A_{\mathcal{C}} = \begin{bmatrix} -2 & 0 & 1 \\ 1 & -1 & -1 \\ 0 & 0 & -1 \end{bmatrix}, \qquad A_{\mathcal{I}} = \begin{bmatrix} -1/2 & -1 & 1 \\ 1/2 & 1 & 0 \\ -1/2 & -1/2 & 0 \end{bmatrix}$$

In this example, we first apply the Algorithms in Section 3 to compute the impulsive Lie algebra associated with the system and its radical. The two will turn out to be equal, so the Lie algebra is solvable. Then, a conclusion of the uniform exponential stability of the system will be drawn accordingly. Also, the common Lyapunov function will be calculated for the case that the impulsive Lie algebra is solvable. Lastly, we will

implement simulation for the system with respect to a random sequence of impulse times and then plot the state trajectories and 3-dimensional state portrait so as to illustrate the behavior of the system.

First, applying Algorithm 2 produces the three-dimensional impulsive Lie algebra L with a basis given by

$$\mathfrak{B} = \{ \begin{bmatrix} 1 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -2 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \}$$

Next, by applying Algorithm 3 we obtain a basis for the radical as below

$$\mathfrak{B}_\mathsf{R} = \{ \begin{bmatrix} 1 & -1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & -2 & 0 \\ 1 & 3 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \}$$

We see $\mathfrak{B}_\mathsf{R} = \mathfrak{B}_\mathsf{L}$. Thus, $\mathsf{R} = \mathsf{L}$, or the impulsive Lie algebra is solvable, and then Levi factors are zero. It follows from Theorem 5 that the impulsive system is uniformly exponentially stable. Furthermore, the solvability of L, according to Theorem 4.4 in [2], ensures the existence of a real positive definite symmetric P that satisfies the inequalities in Theorem 1. Following Lemma 4.2 in [2] for calculating the matrix P yields

$$P = \begin{bmatrix} 9 & 8 & 0 \\ 8 & 8 & 0 \\ 0 & 0 & 16 \end{bmatrix}$$

We will show that P satisfies the inequality (2). Indeed, the transposes of $A_\mathcal{C}$ and $A_\mathcal{I}$ are

$$A_\mathcal{C}^T = \begin{bmatrix} -2 & 1 & 0 \\ 0 & -1 & 0 \\ 1 & -1 & -1 \end{bmatrix}, \qquad A_\mathcal{I}^T = \begin{bmatrix} -1/2 & 1/2 & -1/2 \\ -1 & 1 & -1/2 \\ 1 & 0 & 0 \end{bmatrix}$$

Then

$$M = -A_\mathcal{C}^T P - P A_\mathcal{C} = \begin{bmatrix} 20 & 16 & -1 \\ 16 & 16 & 0 \\ -1 & 0 & 32 \end{bmatrix}$$

$$N = -A_\mathcal{I}^T P A_\mathcal{I} + P = \begin{bmatrix} 19/4 & 7/2 & 1/2 \\ 7/2 & 3 & 1 \\ 1/2 & 1 & 7 \end{bmatrix}$$

The matrix M is symmetric positive definite, or $-M = A_\mathcal{C}^T P + P A_\mathcal{C}$ is negative definite as verified bellow

$$m_{11} = 20 > 0, \quad m_{22} = \det \begin{bmatrix} 20 & 16 \\ 16 & 16 \end{bmatrix} = 64 > 0, \quad m_{33} = \det \begin{bmatrix} 20 & 16 & -1 \\ 16 & 16 & 0 \\ -1 & 0 & 32 \end{bmatrix} = 2032 > 0.$$
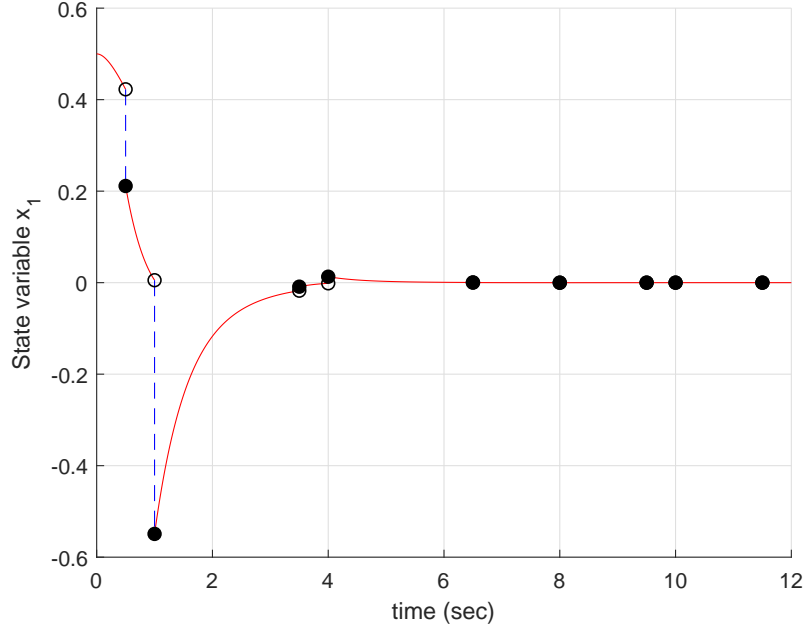
23

Figure 5: Initial Value Response $x_1(t)$ vs t

Similarly, all leading principal minors of N are positive. Thus, N is symmetric positive definite, or $-N = A_{\mathcal{I}}^T P A_{\mathcal{I}} - P$ is also negative definite.

This implies, from Theorem 1, a common quadratic Lyapunov function for (1) is given by

$$V(x) = x^T P x = 9x_1^2 + 8x_2^2 + 16x_3^2 + 16x_1 x_2$$

Finally, we generate simulation for the system with respect to a random set of impulse times and provide the state trajectory, the state portrait, and the Lyapunov function plots as in the following. The initial state is $x_0 = \begin{bmatrix} 0.5 \\ 0.5 \\ 1 \end{bmatrix}$.

As seen in the figures (Figure 5-9), the behavior of the state trajectory graphically supports the conclusion of uniform exponential stability of the system.
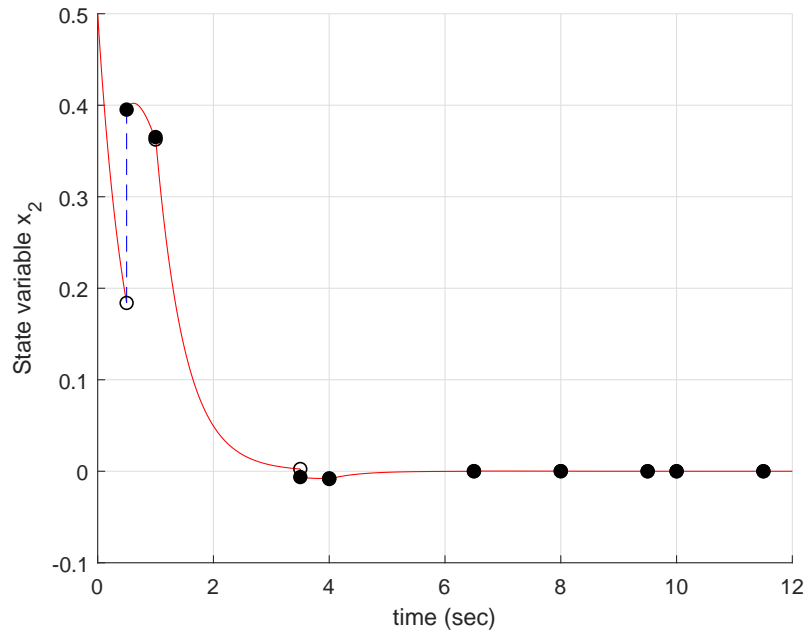
24

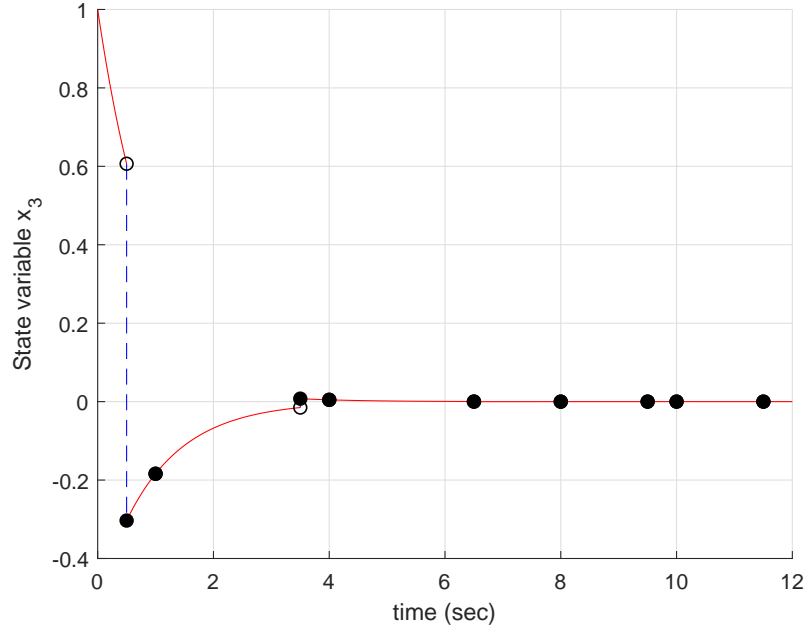Figure 6: Initial Value Response $x_2(t)$ vs t

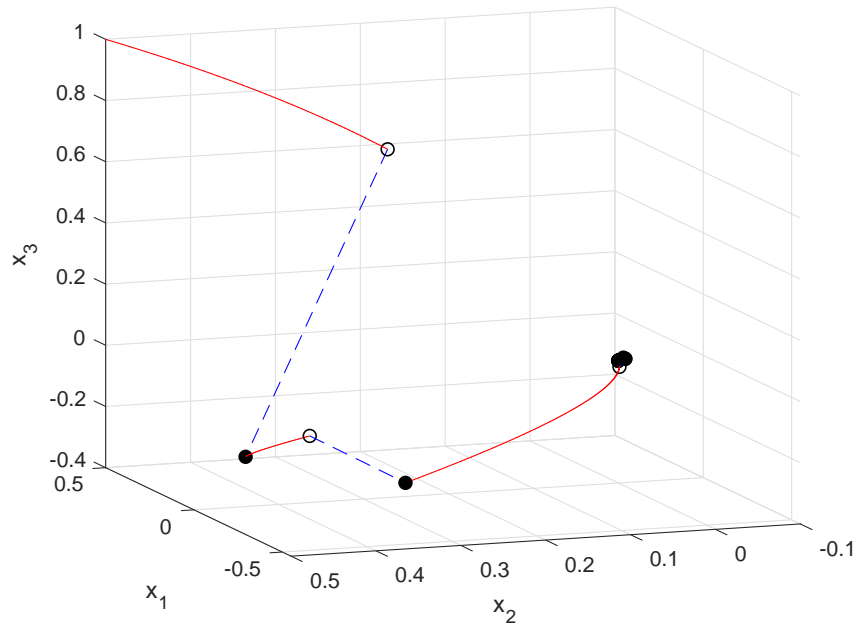Figure 7: Initial Value Response $x_3(t)$ vs t
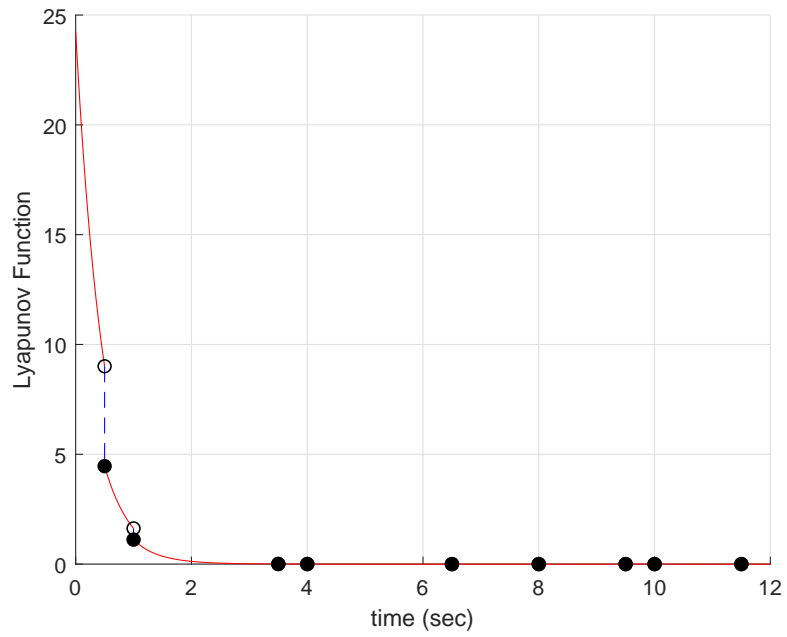
25

Figure 8: Phase Portrait

Figure 9: Lyapunov Function $V[x(t)]$ vs t

Theorem 6. *([1, Theorem 4.5]) Let* $\mathsf{L} = \mathsf{Rad}(\mathsf{L}) \oplus \mathsf{S}$ *denote a Levi decomposition of the impulsive Lie algebra. If the Levi factor* S *is compact, then the linear impulsive system (1) is UES.*

Example 15. In this example we also consider a 3-dimensional linear impulsive system in the form of (1) with data matrices $A_{\mathcal{C}}$ and $A_{\mathcal{I}}$ given by

$$A_{\mathcal{C}} = \begin{bmatrix} -1/4 & -1/4 & 0 \\ 2 & -1/4 & -2 \\ 0 & 1/4 & -1/4 \end{bmatrix}, \qquad A_{\mathcal{I}} = \begin{bmatrix} 1/4 & -1/2 & 1/4 \\ 1/4 & 0 & -1/4 \\ 1/4 & 1/2 & 1/4 \end{bmatrix}$$

Similar to Example 14, we follow the Algorithms in Section 3 to compute the impulsive Lie algebra and its radical and Levi factor. We also create simulation for the system with respect to a set of impulse times and then plot the state trajectories and 3-dimensional state portrait. In the case that the Levi factor is not compact, we make a conclusion of the UES according to the behavior of the state trajectories and state portrait of the system.

First, applying Algorithm 2 produces the four-dimensional impulsive Lie algebra L with a basis given by

$$\mathfrak{B} = \{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 3 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \}$$

Next, by applying Algorithm 3 we obtain the one-dimensional radical with a basis below

$$\mathfrak{B}_{\mathsf{R}} = \{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \}$$

Direct calculation shows $[\mathfrak{B}, \mathfrak{B}_{\mathsf{R}}] = 0$ or $[\mathsf{L}, \mathsf{Rad}(\mathsf{L})] = 0$. Hence, the radical is also the center of L, and so the Lie algebra is reductive. Next, implementing Algorithm 4 gives us a Levi factor S of L, generated by

$$\mathfrak{B}_{\mathsf{S}} = \{ \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \}$$

If the Killing form of the Levi factor is not negative definite, then the Levi factor is not compact. Running the MATLAB function for computing Killing form (see Appendix A) produces a matrix of the Killing form of the Levi factor as

$$[\mathcal{K}_{\mathsf{S}}] = \begin{bmatrix} 0 & 0 & 8 \\ 0 & 32 & 0 \\ 8 & 0 & 0 \end{bmatrix}$$

27

It is clear that $[\mathcal{K}_\mathsf{S}]$ is not negative definite (as the 1-by-1 leading principal minor is zero, and eigenvalues of $[\mathcal{K}_\mathsf{S}]$ are -8, 8, 32). Hence, $\mathcal{K}_\mathsf{S}$ is not negative definite. This implies, from Definition 25, that the Levi factor S is not compact.

Since the Levi factor is not compact, we have no conclusion (based on Theorem 6) of whether or not the system is UES. However, the result can be drawn from the behavior of the state trajectories and the state portrait of the system. The system is stimulated and plotted with respect to the initial state and the set of impulse times as given below respectively

$$x_0 = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix},$$

$$\mathcal{T} = \{k\frac{\pi}{2}, \ k = 1, ..., 10\}$$

From the graphs (Figure 10-13), the behavior of the state trajectories graphically indicates that the system is not uniformly exponentially stable.
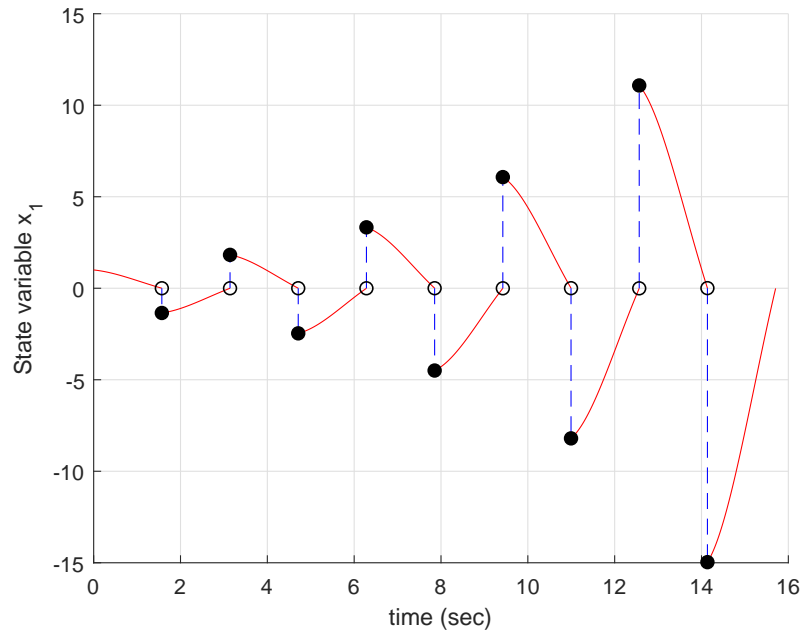
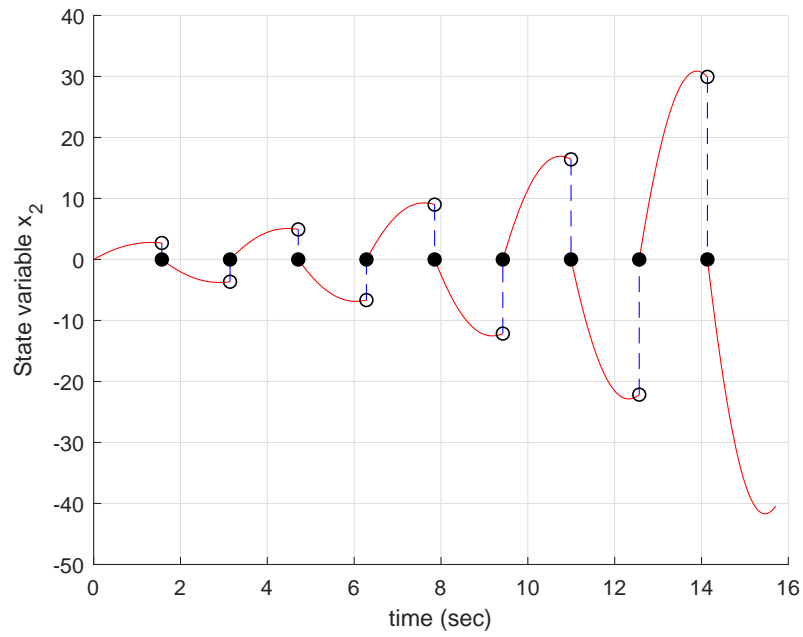Figure 10: Initial Value Response $x_1(t)$ vs t

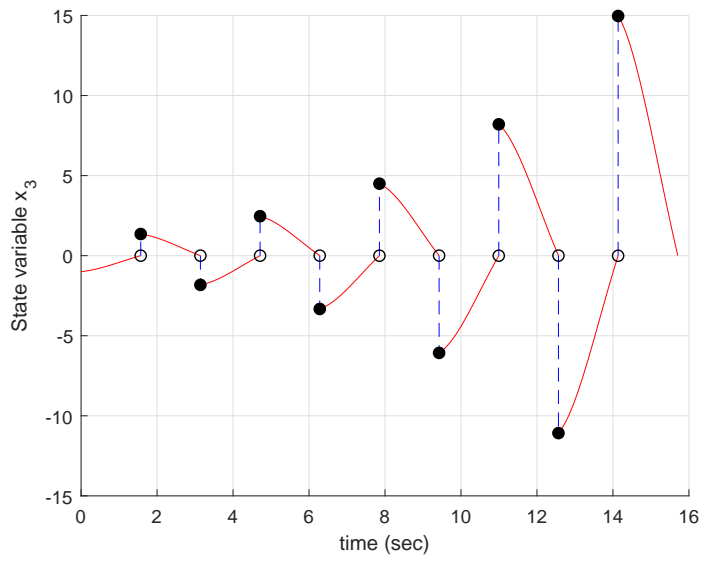Figure 11: Initial Value Response $x_2(t)$ vs t

29

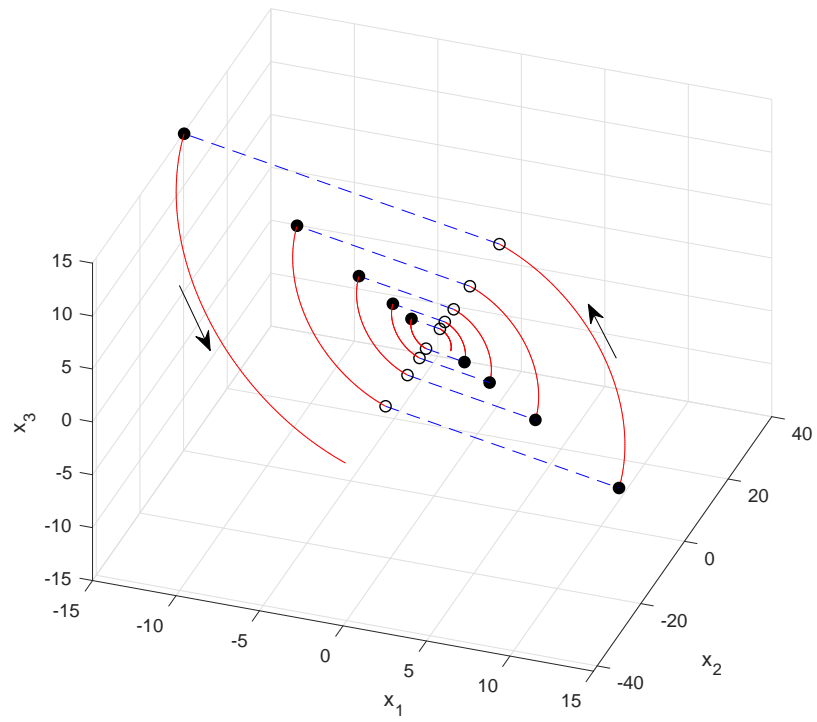Figure 12: Initial Value Response $x_3(t)$ vs t

Figure 13: Phase Portrait

30

5 Conclusion

This research project has investigated uniform exponential stability for linear impulsive systems using Lie algebraic techniques in which we have focused on the computation of the Levi decomposition of the impulsive Lie algebra. The goal of the project is mainly intended to develop computational tools for executing the Levi decomposition using MATLAB. Our primary contribution is the development of convenient computational tools which can provide tutorial values and thus effectively support academic ideas and uses. For this purpose, the computational tools have been developed in the particular manner that all algorithms must preserve "nice" numbers-integers or rational numbers and better illustrate the underlying theory and algorithms presented in the project. The traditional MATLAB commands do not meet these aims and objectives. We have designed computational tools that satisfy the goal. In particular, our accomplishments involve the following:

1. Algorithm and MATLAB implementation for a complement to a subspace in a given subspace and solving matrix equations that preserve nice numbers (integers and rationals),

2. Algorithm and MATLAB implementation for the minimal invariant subspace containing a specified subspace

3. Algorithm and MATLAB implementation for the impulsive Lie algebra,

4. Algorithm and MATLAB implementation for the derived Lie algebra of any Lie algebra,

5. Algorithm and MATLAB implementation for structure constants for any Lie algebra,

6. Algorithm and MATLAB implementation for Killing form of any Lie algebra,

7. Algorithm and MATLAB implementation for the radical of any Lie algebra,

8. Algorithm and MATLAB implementation for a Levi factor of the Lie algebra in the reductive case.

However, the computational tools for the case that the radical is not the center have not been completed. This suggests topics for further work. Specifically, our future research would concentrate on the two more general cases. The first case is that the radical is Abelian but not equal to the center and the second is that the radical is not required to be Abelian.

References

[1] D. A. Lawrence. Lie-algebraic stability conditions for stability of linear impulsive systems. In *Proceedings of the 2013 American Control Conference*, pages 3185-3190, 2013.

[2] D. A. Lawrence. On the uniform exponential stability of linear impulsive systems. In *Proceedings of the 2017 American Control Conference*, pages 1250-1256, 2017.

[3] N. Nguyen and D. A. Lawrence. On the Lie algebra of a linear impulsive systems. In *Proceedings of the 2018 American Control Conference*, pages 430-435, 2018.

[4] D. Liberzon, J. P. Hespanha, and S. Morse. Stability of switched systems: a Lie-algebraic condition. *Systems & Control Letters*, 37:117-122, 1996.

[5] A. A. Agrachev and D. Liberzon. Lie-algebraic stability criteria for switched systems. *SIAM Journal on Control and Optimization*, 40(1):253-269, 2001.

[6] T. Yang. *Impulsive Systems and Control: Theory and Applications*. Nova Science Publishers, 2001.

[7] M. Dagli. *Levi Decomposition of Lie Algebras; Algorithm for its Computation*. MS. thesis, Iowa State University, Department of Mathematics, 2004.

[8] H. Zassenhaus. *Lie Groups, Lie Algebras and Representation Theory*. Presses University Montreal, 1981.

[9] K. Erdmann and M. J Wildon. *Introduction to Lie Algebras*. Springer-Verlag, London, 2006.

[10] W. A. de Graaf. *Lie Algebras Theory and Algorithms*. Elsevier Science, 2000.

[11] A. Graham. *Kronecker Products and Matrix Calculus with Applications*. Ellis Horwood Limited, Chichester, 1981.

[12] W.-H Steeb. *Problems and Solutions in Introductory and Advanced Matrix Calculus*. World Scientific, 2006.

[13] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2000.

Appendix A MATLAB CODE

```
% COMPUTING THE MINIMAL INVARIANT SUBSPACES

function M = min_inv(T,B)
% T is a linear transformation (matrix), B is a matrix
% Initialization

M1 = B;
[R, jb] = rref(M1);
r1 = length(jb); % r1 = rank(M1);
condition = 1;
while condition
   r0 = r1;
   M  = M1;
   M1 = [B T*M];
   [R1, jb1] = rref(M1);
   r1 = length(jb1); % r1 = rank(M1);
   condition = (r1 > r0);
end
[R2, jb2] = rref(M');
r2 = length(jb2);
M = R2(1:r2,:)';


% COMPUTING IMPULSIVE LIE ALGEBRA

function L = Lie(A,g)
ad_A = ad_A_of(A);
Ad_g_inv = Ad_g_inv_of(g);

L_0 = min_inv(Ad_g_inv, vec(A));
L_1 = add(L_0, min_inv(Ad_g_inv, ad_A*L_0));

N1 = L_0;
[R, jb] = rref(N1);
r1 = length(jb); % r1 = rank(N1);
condition = 1;

while condition
   r0 = r1;
   N  = N1;
   N1 = add(L_0, min_inv(Ad_g_inv, ad_A*N));
```

```
    [R1, jb1] = rref(N1);
    r1 = length(jb1);
    condition = (r1 > r0);
end

[R2,jb2] = rref(N');
r2 = length(jb2);
L = R2(1:r2,:)'; % In matrix form


%COMPUTING THE DERIVED ALGEBRA

function L_pr  = der_al(L)
% L is a matrix whose columns constitute a basis for L
k = 1;
[R,jb] = rref(L);
d = length(jb);  %d = rank(L);

if d == 1
    L_pr = []; % L is Abelian
else
    for i=1:d-1
        for j=i+1:d
        B = bracket(vec_inv(L(:,i)),vec_inv(L(:,j)));
        STORAGE(:,k) = vec(B);
        k = k+1;
        end
    end
[R1,jb1] = rref(STORAGE');
r = length(jb1);   % r = rank(STORAGE);
    if r == 0
        L_pr = []; % L is Abelian
    else
        L_pr = R1(1:r,:)';
    end
end


% COMPUTING STRUCTURE CONSTANTS

function ad_X = str_cons(L,L1)
k = 1;
[r1,d1] = size(L);  % d1 = the dimension of L;
[r2,d2] = size(L1); % d2 = the dimension of L1
```

```
% Calculating structure constants
for j=1:d2
    for i=1:d1
        Z = bracket(vec_inv(L1(:,j)),vec_inv(L(:,i)));
        STORAGE(:,k) = coor(L,Z);
        k = k+1;
    end
end

% Calculating adjoint representation of L1
[row,c] = size(STORAGE);
r = 1;
m = 1;
n = d1;
while n <= c
    C(:,:,r) = STORAGE(:,m:n);
    r = r+1;
    m = n+1;
    n = r*d1;
end
ad_X = C; % Return a three dimensional array



% COMPUTING THE KILLING FORM

function Kill = Kill_form(L,L1)
ad_X = str_cons(L,L);
ad_Y = str_cons(L,L1);

[r,c] = size(L); % c is the dimension of L
[r1,c1] = size(L1); % c1 is the dimension of L1

% Finding Killing form with respect to L and L_pr
for j=1:c1
    for i=1:c
        K(j,i) = trace(ad_X(:,:,i)*ad_Y(:,:,j));
    end
end
Kill = K; % Return a matrix


% COMPUTING THE RADICAL OF A LIE ALGEBRA L
```

```matlab
function Radical = rad(L)
L1 = der_al(L); % The derived algebra of L
[r,c] = size(L);
[r1,c1] = size(L1);
if c1 == 0
    Radical = L;  % L is solvable
elseif c1 == c
    Radical = []; % L is semisimple
else
Kill = Kill_form(L,L1); % The Killing form

% Calculating coefficients alpha
alpha = null(Kill,'r');
    % if norm(alpha) == 0; or if alpha = zeros(size(alpha))
    %    Radical = [];
    % else

% A set of basis vectors for the radical is linear combinations
% of basis vectors X_i in L and alpha
[r_alpha,c_alpha] = size(alpha);
for n = 1:c_alpha
    b = 0; % b is a c-by-1 vector
        for k = 1:c
        b = b + alpha(k,n)*L(:,k);
        end;
    STORAGE(:,n) = b;
    % Rad(:,:,n) = vec_inv(b);
end;

Radical = STORAGE;  % Return a matrix
end


% COMPUTING A LEVI FACTOR

function Levi_factor = Levi(L)
R = rad(L);
[r,c] = size(L);
[r1,c1] = size(R);
if c1 == c
    Levi_factor = []; % L is solvable
elseif c1 == 0
    Levi_factor = L;  % L is semisimple
else
```

36

```
% The complement of R in L is
Com = comp(L,R);
L1 = [Com R];
% Find structure constants for the quotient space L/R
% L/R <--> {x1_bar, x2_bar ... xm_bar} ~ C <--> {x1, x2 ... xm}
ad_X = str_cons(L1,L1); % Recalculate a set of structure constants for L
                        % with respect to a new basis [C R]
[r,c,p] = size(ad_X);
   for k = 1:p-1
       B = ad_X(:,:,k);
       [r_b,c_b] = size(B);
       % ad_X_bar are induced map from ad_X
       ad_X_bar(:,:,k) = B(1:r_b-1,1:c_b-1); % Delete last columns and last rows
                                             % of ad_Xk
   end

   % STEP 1 (RIGHT-HAND SIDE): Calculating the right-hand side tube b_ij of
   % the EQUATION
   % and STEP 3 (LEFT-HAND SIDE): Calculating the Matrix in the left-hand side of
   % the EQUATION
   [r1,c1,p1] = size(ad_X_bar);
   column = 1;
   for k = 1:p1-1                       % k is the index of page
       for j = k+1:p1                   % j is the index of column
           Xk = vec_inv(Com(:,k));
           Xj = vec_inv(Com(:,j));
           tube(:,column) = vec(bracket(Xk,Xj)) - Com*ad_X_bar(:,j,k);
           % RIGHT-HAND SIDE
           % There are p1(p1-1)/2 = m(m-1)/2
           b = ad_X_bar(:,j,k);
           Matrix_trans(:,column) = b; % LEFT-HAND SIDE
           column = column+1;
       end
   end

% STEP 2(LEFT-HAND SIDE): Calculating the Yij (scalars) in the left-hand side
% of the EQUATION
   Yij_trans = mat_sol(R,tube); % Yij is a column vector
   Yij = Yij_trans';

   Matrix = Matrix_trans';
```

```matlab
    alpha = mat_sol(Matrix,Yij); % alpha is a matrix in which column i goes
                               % with r_i

    % STEP 4: Computing a basis for a Levi factor using the following
    % transformations yi = xi + sum(apha_ji*r_j), i = 1...m
    [r_alpha,c_alpha] = size(alpha);
    for i = 1:r_alpha
        yi = Com(:,i) + R*alpha(i,:)';
        Levi_factor(:,i) = yi;
      end
end


% SIMULATION

function [simulation, time] = simimp(A,g,Delta,t0,x0)
% A and g are data matrices
% Tau is a set of impulse times given in a row vector
% t0 is an nitial time
% x0 is an initial state

sysA  = ss(A, [], eye(size(A)), []);

d = length(Delta);
Tau(1) = t0;
for j = 2:d+1
    T = Tau(j-1) + Delta(j-1);
    Tau(j) = T;
end
Tau;

time  = [];
impulse_state = [];

xp    = x0;

l     = length(Tau);

for k = 1:l-1
    tt = 0:0.01:Delta(k);
    xx = initial(sysA, xp, tt)';
    impulse_state = [impulse_state xx];
    time = [time tt+Tau(k)];
```

```matlab
    xn = xx(:, end);
    xp = g * xn;
end


simulation = impulse_state;


% SOLVING MATRIX EQUATION AX = B

function [X_particular, S_0, matrix_solver] = mat_sol(A,B)
% A is a matrix and B is a matrix
[R,jb] = rref(A);
r_0 = length(jb);

A_aug = [A B];
[R1,jb] = rref(A_aug);
r_aug = length(jb);

if r_0 == r_aug
% We just solve the system when it is consistent, i.e. r = r_aug
% Determine the pivot variables and the particular solution
    [r,c] = size(A);
    [r1,c1] = size(B);
    X_p = zeros(c,1); % initialize

    k=1;
    for i = 1:c1
        X_p(jb) = R1(1:r_aug,c+i);
        X_par(:,k) = X_p;
        k = k+1;
    end
    X_particular = X_par;
    % Determine the parameterized set of the associated
    % homogeneous equation
    S_0 = null(A,'r');

    % Determine the general solution for the nonhomogeneous equation
    matrix_solver = [X_par S_0];

else disp('The equation is not consistent');
    [X_particular, S_0, matrix_solver] = [];
end
```