1 Basic Calls of Toolbox of Lie-Derivatives

Nonlinear controller and observer design require often certain types of Lie derivatives. They are total derivatives of tensor fields along a vector field. The drivers provided for calculating Lie derivatives of scalar, vector, covector fields and gradient of Lie derivatives of scalar or vector field are prototyped in the header <lie_tool.h>.

We consider computational problems occurring in controller and observer design for nonlinear control systems

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u, \quad \mathbf{y} = \mathbf{h}(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0 \in \Omega \tag{1.1}$$

with the scalar input u, the state $\mathbf{x}$, and the output $\mathbf{y}$, where the vector fields $\mathbf{f} : \boldsymbol{\Omega} \to \mathcal{R}^{\mathbf{n}}$, $\mathbf{g} : \boldsymbol{\Omega} \to \mathcal{R}^{\mathbf{n}}$ and $\mathbf{h} : \boldsymbol{\Omega} \to \mathcal{R}^{\mathbf{m}}$ (It can be a scalar field h, wenn $m = 1$) are defined on an open subset $\Omega \subseteq \mathcal{R}^n$.

1.1 Lie derivatives of a scalar field

Consider the initial value problem

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \quad y = h(\mathbf{x}), \quad \mathbf{x}(0) = \mathbf{x}_0 \in \Omega \tag{1.2}$$

with the vector field $\mathbf{f} : \boldsymbol{\Omega} \to \mathcal{R}^{\mathbf{n}}$. To compute the Lie derivatives of the scalar field $h : \boldsymbol{\Omega} \to \mathcal{R}$ along the vector field $\mathbf{f}$

$$L^k_{\mathbf{f}} h(\mathbf{x}_0) = (L^0_{\mathbf{f}} h(\mathbf{x}_0), \cdots, L^d_{\mathbf{f}} h(\mathbf{x}_0)), \quad \text{for} \quad k = 0, \cdots, d, \tag{1.3}$$

we give the tape numbers of active sections of $\mathbf{f}$ and h, the number of independent variables n, the initial coefficients $\mathbf{x}_0$ and the highest derivative degree d. The values of the Lie derivatives will be given in the variable result with dimension $d + 1$.

```
int lie_scalarc(Tape_F, Tape_H, n, x0, d, result)
short Tape_F;              // tape identification of vector field f
short Tape_H;              // tape identification of scalar field h
short n;                   // number of independent variables n and m = 1
double x0[n];              // values of independent variables x0
short d;                   // highest derivative degree d
double result[d+1];        // resulting Lie derivatives of a scalar field
```

To compute Lie derivative of smooth map $\mathbf{h} : \boldsymbol{\Omega} \to \mathcal{R}^{\mathbf{m}}$ with $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \cdots, h_m(\mathbf{x}))$ along the same vector field $\mathbf{f}$

$$L^k_{\mathbf{f}} \mathbf{h}(\mathbf{x}_0) = (L^k_{\mathbf{f}} h_1(\mathbf{x}_0), \cdots, L^k_{\mathbf{f}} h_m(\mathbf{x}_0)), \quad \text{for} \quad k = 0, \cdots, d, \tag{1.4}$$

we will give additionally the number of dependent variables m.

```
int lie_scalarcv(Tape_F, Tape_H, n, m, x0, d, result)
short Tape_F;              // tape identification of vector field f
short Tape_H;              // tape identification of vector field h
short n;                   // number of independent variables n
short m;                   // number of dependent variables m
double x0[n];              // values of independent variables x0
short d;                   // highest derivative degree d
double result[m][d+1];     // resulting Lie derivatives of
                           // vectorial scalar fields
```

1.2 Gradients of Lie derivatives of a scalar field

To compute the gradients of the Lie derivatives of a scalar field h

$$dL_{\mathbf{f}}^{k}h(\mathbf{x}_0) = (dL_{\mathbf{f}}^{0}h(\mathbf{x}_0), \cdots, dL_{\mathbf{f}}^{d}h(\mathbf{x}_0)), \quad \text{for} \quad k = 0, \cdots, d, \tag{1.5}$$

we call

```
int lie_gradientc(Tape_F, Tape_H, n, x0, d, result)
short Tape_F;              // tape identification of vector field f
short Tape_H;              // tape identification of scalar field h
short n;                   // number of independent variables n and m = 1
double x0[n];              // values of independent variables x0
short d;                   // highest derivative degree d
double result[n][d+1];     // resulting gradients of Lie derivatives of a scalar field
```

and for calculating the jacobians of the Lie derivatives of scalar fields $\mathbf{h} : \mathbf{\Omega} \to \mathcal{R}^{\mathbf{m}}$ with $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \cdots, h_m(\mathbf{x}))$

$$dL_{\mathbf{f}}^{k}\mathbf{h}(\mathbf{x}_0) = (dL_{\mathbf{f}}^{k}h_1(\mathbf{x}_0), \cdots, dL_{\mathbf{f}}^{k}h_m(\mathbf{x}_0)), \quad \text{for} \quad k = 0, \cdots, d, \tag{1.6}$$

we use

```
int lie_gradientcv(Tape_F, Tape_H, n, m, x0, d, result)
short Tape_F;              // tape identification of vector field f
short Tape_H;              // tape identification of vector field h
short n;                   // number of independent variables n
short m;                   // number of dependent variables m
double x0[n];              // values of independent variables x0
short d;                   // highest derivative degree d
double result[m][n][d+1];  // resulting jacobians of Lie derivatives of
                           // vectorial scalar fields
```

1.3 Lie derivatives of a covector field

For programs, there are no differences between a vector field $\mathbf{f} : \mathbf{\Omega} \to \mathcal{R}^{\mathbf{n}}$ and its covector field $\omega : \mathbf{\Omega} \to (\mathbf{R}^{\mathbf{n}})^*$. For convenience we also use the same character $\mathbf{h}$, but we define it as a covector field $\mathbf{h} : \mathbf{\Omega} \to (\mathbf{R}^{\mathbf{n}})^*$. So Lie derivatives of the covector field $\mathbf{h}$ along the vector field $\mathbf{f}$

$$L_{\mathbf{f}}^{k}\mathbf{h}(\mathbf{x}_0) = (L_{\mathbf{f}}^{k}h_1(\mathbf{x}_0), \cdots, L_{\mathbf{f}}^{k}h_n(\mathbf{x}_0)), \quad \text{for} \quad k = 0, \cdots, d, \tag{1.7}$$

can be computed by

```
int lie_covectorv(Tape_F, Tape_H, n, x0, d, result)
short Tape_F;              // tape identification of vector field f
short Tape_H;              // tape identification of covector field h
short n;                   // number of independent variables n
double x0[n];              // values of independent variables x0
short d;                   // highest derivative degree d
double result[n][d+1];     // resulting Lie derivatives of a covector field
```

1.4 Lie derivatives of a vector field (Lie brackets)

We can calculate iterated Lie derivatives (Lie brackets) of the vector field $\mathbf{g} : \mathbf{\Omega} \rightarrow \mathcal{R}^{\mathbf{n}}$ along the vector field $\mathbf{f} : \mathbf{\Omega} \rightarrow \mathcal{R}^{\mathbf{n}}$

$$L^k_{\mathbf{f}}\mathbf{g}(\mathbf{x}_0) = ad^k_{\mathbf{f}}\mathbf{g}(\mathbf{x}_0) = (ad^k_{\mathbf{f}}g_1(\mathbf{x}_0), \cdots, ad^k_{\mathbf{f}}g_n(\mathbf{x}_0)), \quad \text{for} \quad k = 0, \cdots, d, \tag{1.8}$$

using

```
int lie_bracketv(Tape_F, Tape_G, n, x0, d, result)
short Tape_F;              // tape identification of vector field f
short Tape_G;              // tape identification of vector field g
short n;                   // number of independent variables n
double x0[n];              // values of independent variables x0
short d;                   // highest derivative degree d
double result[n][d+1];     // resulting Lie derivatives of a vector field
```

2 Overloaded Calls of Toolbox of Lie-Derivatives

In this section, the several versions of Lie derivatives routines, which utilize the overloading capabilities of C++, are listed. They are also prototyped in the header <lie_tool.h>.

2.1 The Scalar Case

This procedure computes the Lie derivatives of a scalar field h as (1.3).

```
int lie_scalar(Tape_F, Tape_H, n, x0, d, result)
short Tape_F;              // tape identification of vector field f
short Tape_H;              // tape identification of scalar field h
short n;                   // number of independent variables n and m = 1
double x0[n];              // values of independent variables x0
short d;                   // highest derivative degree d
double result[d+1];        // resulting Lie derivatives of a scalar field
```

and this procedure computes the gradients of Lie derivatives of h as (1.5).

```
int lie_gradient(Tape_F, Tape_H, n, x0, d, result)
short Tape_F;              // tape identification of vector field f
short Tape_H;              // tape identification of vector field h
short n;                   // number of independent variables n and m = 1
double x0[n];              // values of independent variables x0
short d;                   // highest derivative degree d
double result[n][d+1];     // resulting gradients of Lie derivatives of
                           // a scalar field
```

2.2 The Vector Case

This procedure computes the Lie derivatives of $\mathbf{h}$ as (1.4)

```
int Lie_scalar(Tape_F, Tape_H, n, m, x0, d, result)
short Tape_F;              // tape identification of vector field f
short Tape_H;              // tape identification of scalar field h
```

```
short n;                    // number of independent variables n
short m;                    // number of dependent variables m
double x0[n];               // values of independent variables x0
short d;                    // highest derivative degree d
double result[m][d+1];      // resulting gradients of Lie derivatives of
                            // vectorial scalar fields
```

and this procedure computes the jacobians of Lie derivatives of **h** as (1.6)

```
int Lie_gradient(Tape_F, Tape_H, n, m, x0, d, result)
short Tape_F;               // tape identification of vector field f
short Tape_H;               // tape identification of vector field h
short n;                    // number of independent variables n
short m;                    // number of dependent variables m
double x0[n];               // values of independent variables x0
short d;                    // highest derivative degree d
double result[m][n][d+1];   // resulting jacobians of Lie derivatives of
                            // vectorial scalar fields
```

Although we don't have to differentiate between scalar or vector cases for calculating the Lie derivates of a covector field and the Lie brackets, we still list them here. They could be compared with (1.7) and (1.8).

```
int lie_covector(Tape_F, Tape_H, n, x0, d, result)
short Tape_F;               // tape identification of vector field f
short Tape_H;               // tape identification of covector field h
short n;                    // number of independent variables n
double x0[n];               // values of independent variables x0
short d;                    // highest derivative degree d
double result[n][d+1];      // resulting Lie derivatives of a covector field
```

```
int lie_bracket(Tape_F, Tape_G, n, x0, d, result)
short Tape_F;               // tape identification of vector field f
short Tape_G;               // tape identification of vector field g
short n;                    // number of independent variables n
double x0[n];               // values of independent variables x0
short d;                    // highest derivative degree d
double result[n][d+1];      // resulting Lie derivatives of a vector field
```