**Основы теории управления**
**Лабораторная работа №4**
**Наблюдатель Люенбергера**

**Цель работы:** исследование оценки вектора состояния наблюдателем Люенбергера полного порядка.

1. Подготовка к выполнению лабораторной работы.

   1.1. Взять скрипт с именем `lab_otu_modal_do.m`, сделанный на предыдущей лабораторной работе, в котором реализовать весь процесс задания параметров моделирования и обработки результатов.

   1.2. Задать передаточную функцию непрерывной системы с использованием функциии `zpk`(), установив один ноль системы, равный единице, четыре полюса равные -3, -4, -5, -6 и коэффициент усиления равный 100. Используя функцию `ss`() преобразовать передаточную функцию к минимальному описанию в переменных состояния и получить матрицы $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ в соответствующие именам переменные. Построить переходной процесс и сделать выводы о динамике системы, в том числе отметить величины основных критериев качества.

   Примечание. Минимальная реализация получается передачей в функцию `ss`() строкового параметра 'minimal'.

2. Подготовка схемы моделирования для исследования наблюдателя Люенбергера полного порядка.

   2.1. Создать в Simulink схему моделирования непрерывной системы, заданную в виде структурной схемы описания системы в переменных состояния, сохранив модель в файл с именем `lab_otu_observer.slx` согласно схеме на рис. 1. Имена всех блоков должны в точности соответствовать схеме. Блоки, реализующие матрицы $\mathbf{A} - \mathbf{BI}$, где $\mathbf{I}$ — это подсистемы (Subsystem), пример которой приведен на рис. 2. Блок подсистемы можно создать используя меню New → Subsystem. Матрицы задаются с использованием блоков `Constant`, источник шума - `Band − Limited White Noise`.

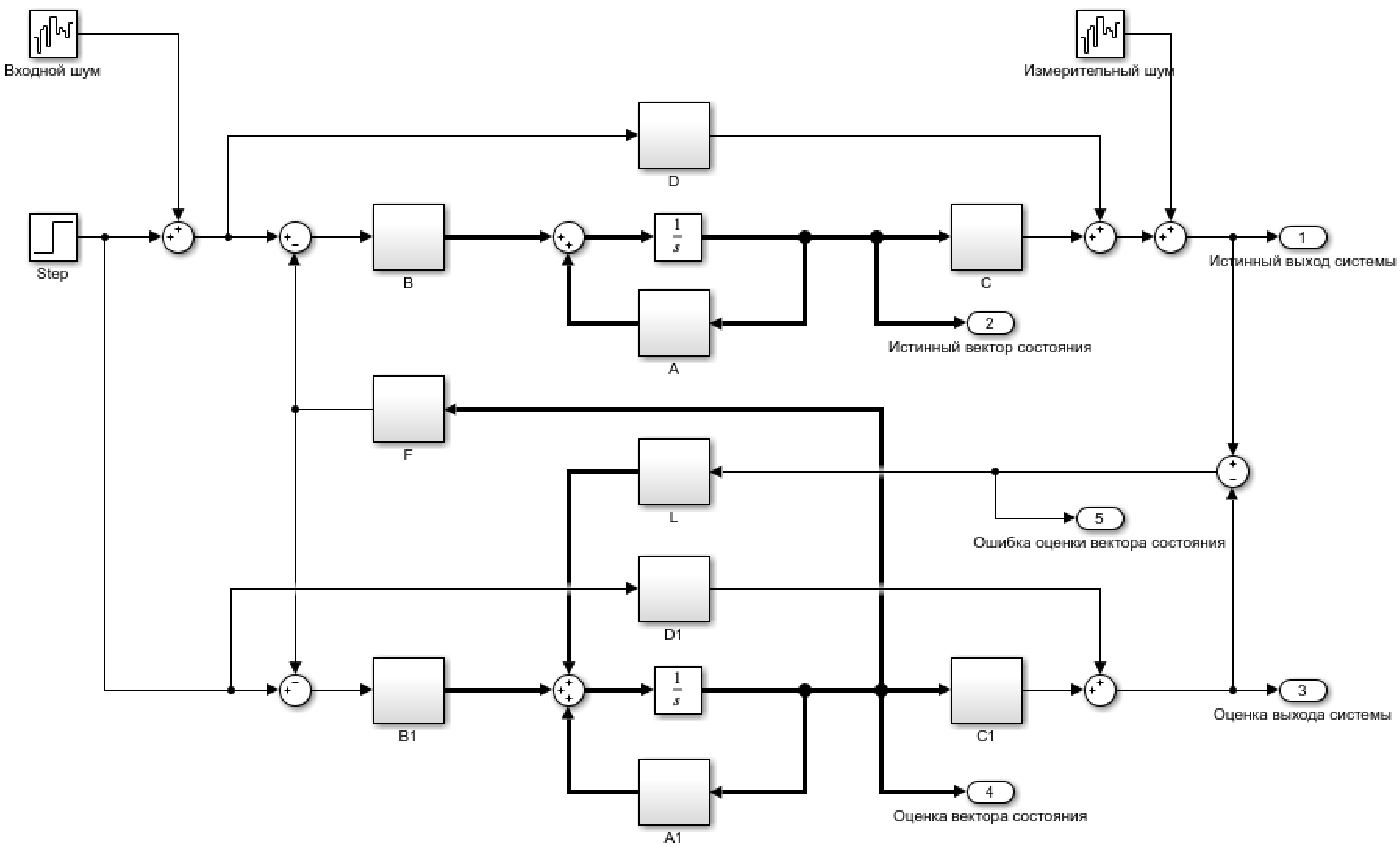


Рис. 1: Схема моделирования системы с модальным управлением и наблюдателем

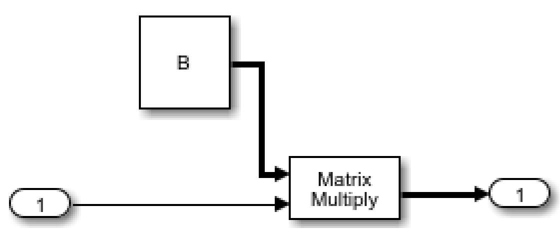


Рис. 2: Схема моделирования блока подсистемы

   2.2. Дополнить скрипт `lab_otu_modal_do.m` кодом, реализующим настройку всех параметров модели в Simulink, запуска моделирования и построения графиков. Для этого первоначально задать в скрипте переменные $T = 0.01$ — период дискретизации, а также `cov_Input` $= 1e - 6$ и `cov_Output` $= 1e - 6$ - дисперсии входных и измерительных шумов. Затем с использованием функции `set_param()` установить следующие параметры (все задаются в виде строк) блоков модели и подсистем:

- Блок `Step`:
  - `SampleTime` в значение T;
    *(период дискретизации блока, заданный переменной T)*
  - `Time` в значение 1.0;
    *(время срабатывания ступеньки в секундах)*
  - `Before` в значение 0.0;
    *(магнитуда до ступеньки)*
  - `After` в значение 1.0.
    *(магнитуда ступеньки)*
- Блоки входного и измерительного шумов:
  - `Ts` в значение T;
    *(период дискретизации блока, заданный переменной T)*
  - `Cov` в значение `cov_Input` или `cov_Output`.
    *(дисперсия каждого блока)*
- Блоки `Constant` описания матриц в подсистемах:
  - `Value` в значение A или B или C или D или F или L.
    *(значения матриц, соответствующие имеющимся в коде переменным, а блоки матриц наблюдателя* `A1`, `B1`, `C1`, `D1` *теми же значениями A, B, C, D)*

Примечание. У блоков `Step` и все `Constant`, использованных для описания матриц, обязательно надо установить параметр `VectorParams1D` в значение `off`, которое также задается строкой. Это необходимо для того, чтобы при моделировании все сигналы воспринимались в каждый момент времени как вектора соответствующей размерности. Например, выход подсистемы `A` — это вектор состояния из четырех элементов в каждый момент времени $t_k$, и если этой настройки не сделать, то в итоге будет не вектор-столбец, а будет вектор строка и будет нарушена размерность перемножаемых векторов и матриц, в результате чего система не сможет быть смоделирована.

2.3. Матрицу $\mathbf{L}$ — матрицу усиления наблюдателя сформировать также методом модального управления, с использованием функции `place()`, но теперь для пары матриц $\mathbf{A}$ и $\mathbf{C}$ с теми же полюсами $\mathbf{p} = \{-9, -6, -7, -8\}$.

3. Исследование наблюдателя Люенбергера.

3.1. Используя функцию sim() провести моделирование схемы в Simulink, а результат сохранить в переменную DATA, имеющую достаточно сложную структуру. Выделить следующим образом переменные, полученные в результате моделирования:

```
t = DATA.tout;
Y = DATA.yout{1}.Values.Data;
X = DATA.yout{2}.Values.Data;
Y_est = DATA.yout{3}.Values.Data;
X_est = DATA.yout{4}.Values.Data;
Y_err = DATA.yout{5}.Values.Data;
```

Примечание. X и Y — вектора состояния и выхода самой систеты, а X_est и Y_est — то, что вычисляется в наблюдателе. Y_err — разность между выходом системы и выходом наблюдателя.

Примечание. Номера 1, 2, 3, 4, 5 — номера портов вывода, если созданная схема соответствует, приведенной на рис. 1 в точности, все переменные будут на своих местах, если нет, то номера портов на схеме можно изменить открыв параметры блока каждого порта вывода.

Примечание. В различных версиях MATLAB и при различных настройках, структура DATA может иметь разлиную конфигурацию. Показанный выше пример получения конкретных значений может быть на практике реализован и по другом.

3.2. Построить следующие графики, все подписать, добавить легенды:

- На одной канве два вертикально расположенных графика. На верхнем — четыре кривых истинного вектора состояния, внизу четыре кривых оцененного вектора состояния. Все как функции времени.
- На одной канве три вертикально расположенных графика. На верхнем — кривая выхода системы, на среднем — кривая выхода наблюдателя, на нижнем — разность между выходом системы и выходом наблюдателя.
- На одной канве вертикальную диаграмму, состоящую из четырех столбцов - четырех интегральных нормированных оценок каждого элемента вектора состояния. Для ее вычисления необходимо найти разницу между X и X_est, найти норму по каждому вектору-столбцу и поделить на количество точек в сигнале (векторе времени). Последнее делает результат сравнимым независимо от длительности сигнала, потому что интегральная ошибка при увеличении длительности сигнала будет возрастать, а при нормировке — перестанет. Подобная ошибка — ошибка оценки каждого элемента вектора состояния на один такт измерений.

3.3. Провести моделирование системы для трех случаев значений матрицы $\mathbf{L}$. В первом случае матрица $\mathbf{L}$ определяется модальным управлением с желаемыми полюсами замкнутой системы наблюдения $\mathbf{p} = \{-9, -6, -7, -8\}$, а во втором и третьем равными $0.2\,\mathbf{p}$ и $2\,\mathbf{p}$. При этом желаемые полюса замкнутой системы управления остаются для всех трех испытаний одинаковыми, соответствующими $\mathbf{p} = \{-9, -6, -7, -8\}$, т.е.

матрица $\mathbf{F}$ не меняется. По результатам сделать выводы о влиянии значений полюсов наблюдателя и величин элементов матрицы $\mathbf{L}$ на динамику и на точность оценки вектора состояния. Сделать необходимые пояснения, в том числе и с использованием графиков.

3.4. Вернуть базовую конфигурацию при которой матрица $\mathbf{L}$ определяется модальным управлением с желаемыми полюсами замкнутой системы наблюдения $\mathbf{p} = \{-9, -6, -7, -8\}$. Провести три эксперимента с различными значениями дисперсий входных и измерительных шумов:

- $\texttt{cov\_Input} = \texttt{1e} - \texttt{6}$ и $\texttt{cov\_Output} = \texttt{1e} - \texttt{6}$.
- $\texttt{cov\_Input} = \texttt{1e} - \texttt{3}$ и $\texttt{cov\_Output} = \texttt{1e} - \texttt{6}$.
- $\texttt{cov\_Input} = \texttt{1e} - \texttt{6}$ и $\texttt{cov\_Output} = \texttt{1e} - \texttt{5}$.

По результатам сделать сравнительный анализ о влиянии уровней входных и измерительных шумов на точность оценки вектора состояния и динамику всей системы в целом.

**Вопросы для подготовки к защите лабораторной работы**

1. Зачем нужна оценка вектора состояния?
2. Что такое наблюдатель Люенбергера?
3. В чем достоинства и недостатки использования наблюдателя Люенбергера?
4. Каким образом определяется матрица усиления наблюдателя?