

Control Systems Lab Report

Your Name

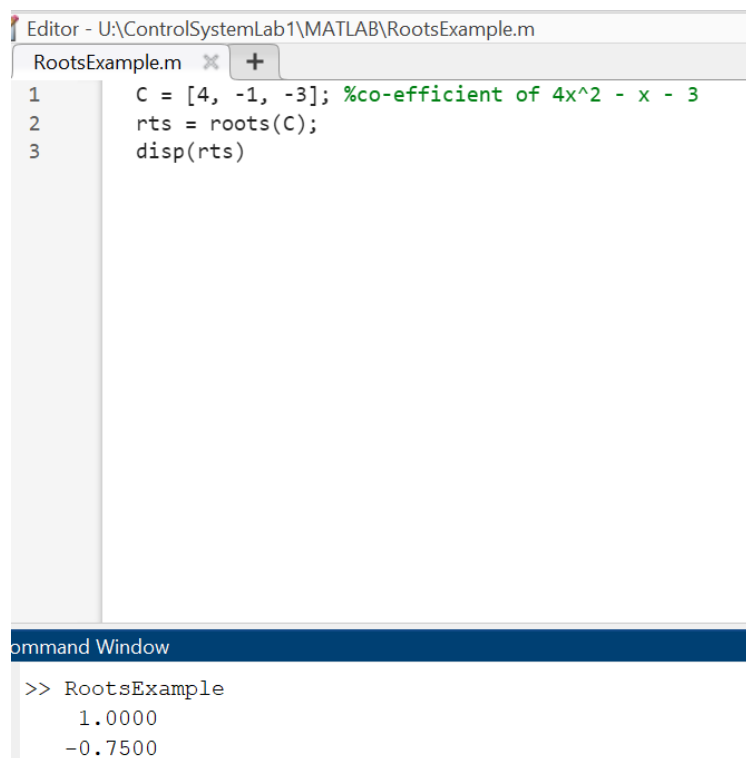
October 1, 2024

1 Introduction

This lab report summarizes various MATLAB functions used in control systems analysis and design. Each function plays a significant role in handling polynomials, transfer functions, and system responses.

2 MATLAB Functions

2.1 roots



The image shows a MATLAB environment. The Editor window displays a script named 'RootsExample.m' with the following code:

```
1 C = [4, -1, -3]; %co-efficient of 4x^2 - x - 3
2 rts = roots(C);
3 disp(rts)
```

The Command Window shows the output of the script:

```
>> RootsExample
    1.0000
   -0.7500
```

Figure 1: Function: **roots**

The **roots** function finds the roots of a polynomial given its coefficients. It is useful for determining the stability of control systems.

2.2 poly

The **poly** function generates a polynomial with specified roots. This is useful for constructing system transfer functions from desired pole locations.

```

>> help roots
roots Find polynomial roots.
roots(C) computes the roots of the polynomial whose coefficients
are the elements of the vector C. If C has N+1 components,
the polynomial is C(1)*X^N + ... + C(N)*X + C(N+1).

Note: Leading zeros in C are discarded first. Then, leading relative
zeros are removed as well. That is, if division by the leading
coefficient results in overflow, all coefficients up to the first
coefficient where overflow occurred are also discarded. This process is
repeated until the leading coefficient is not a relative zero.

Class support for input c:
    float: double, single

See also poly, residue, fzero.

Documentation for roots
Other uses of roots

```

Figure 2: Help Roots

2.3 polyval

The `polyval` function evaluates a polynomial for a given set of values. It is useful for analyzing system outputs at specific input values.

2.4 tf

The `tf` function creates a transfer function model from numerator and denominator coefficients. It is essential for representing dynamic systems.

2.5 conv

The `conv` function computes the convolution of two sequences. This is useful for determining the output of a system in response to a given input.

2.6 pzmap

The `pzmap` function generates a pole-zero map of a transfer function. This visualization helps in analyzing system stability and performance.

2.7 impulse

The `impulse` function computes the impulse response of a system. It is vital for understanding how systems respond to sudden inputs.

2.8 step

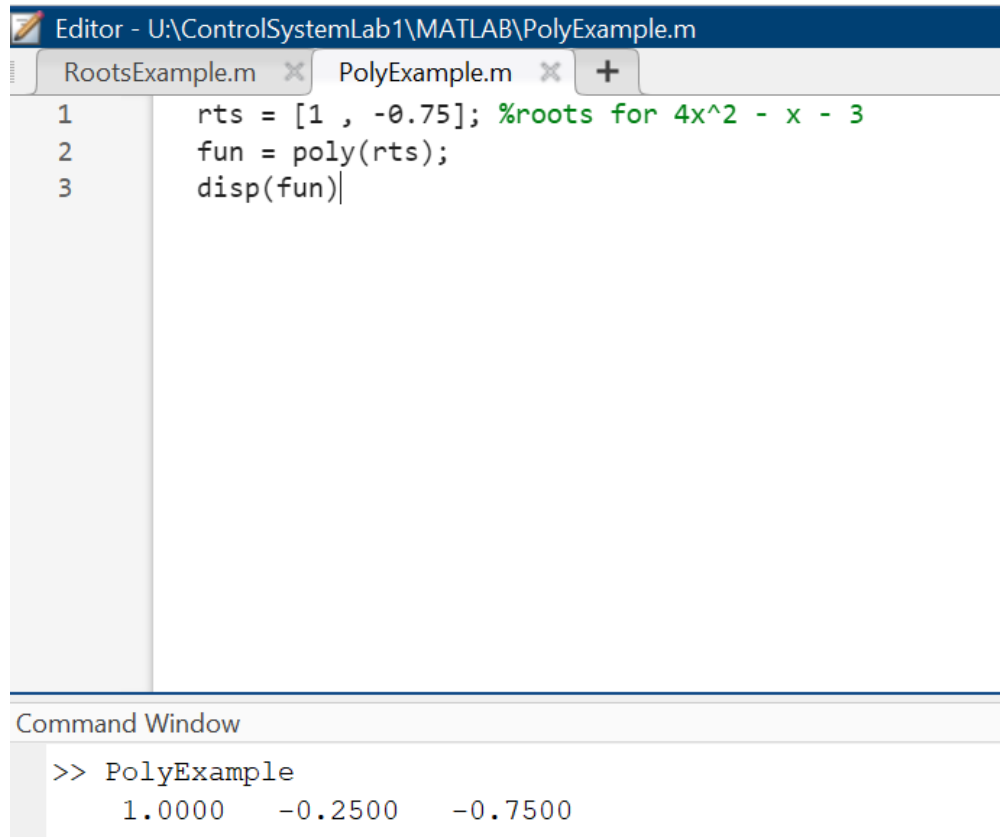
The `step` function computes the step response of a system. It helps evaluate how a system reacts to a step input over time.

2.9 residue

The `residue` function performs partial fraction decomposition. This is useful for simplifying transfer functions into summable terms.

2.10 series

The `series` function connects two transfer functions in series. This is important for cascading system responses.



The image shows a MATLAB Editor window with a file named 'PolyExample.m' open. The code in the editor is as follows:

```
1 rts = [1 , -0.75]; %roots for 4x^2 - x - 3
2 fun = poly(rts);
3 disp(fun)
```

Below the editor is the Command Window, which shows the execution of the script:

```
>> PolyExample
      1.0000   -0.2500   -0.7500
```

Figure 3: Function: poly

2.11 parallel

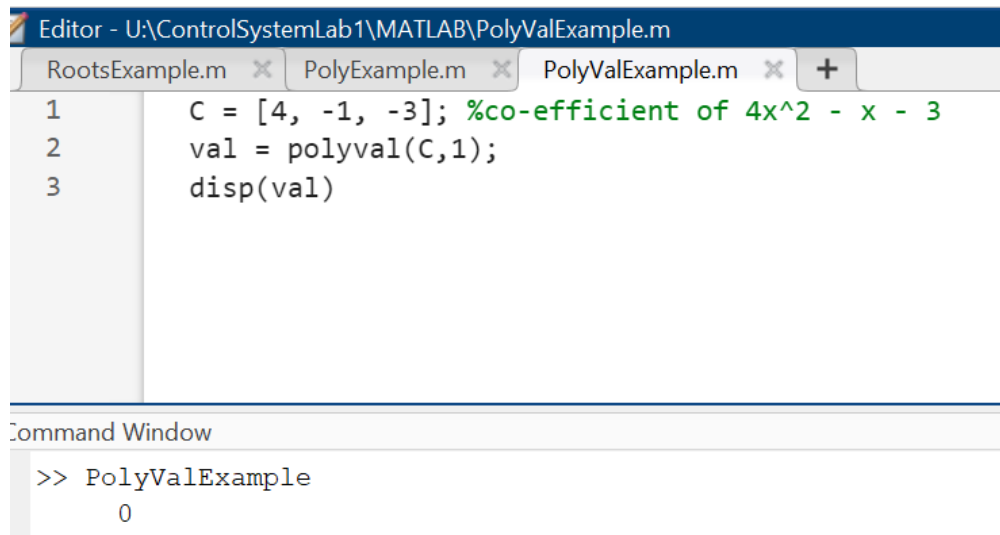
The `parallel` function combines two transfer functions in parallel. This is useful for systems with multiple pathways.

2.12 feedback

The `feedback` function computes the closed-loop transfer function of a system with feedback. This is essential for analyzing control system stability.

3 Conclusion

This report has presented an overview of key MATLAB functions used in control systems analysis. Each function has its unique application, and understanding them is crucial for effective system design and analysis.



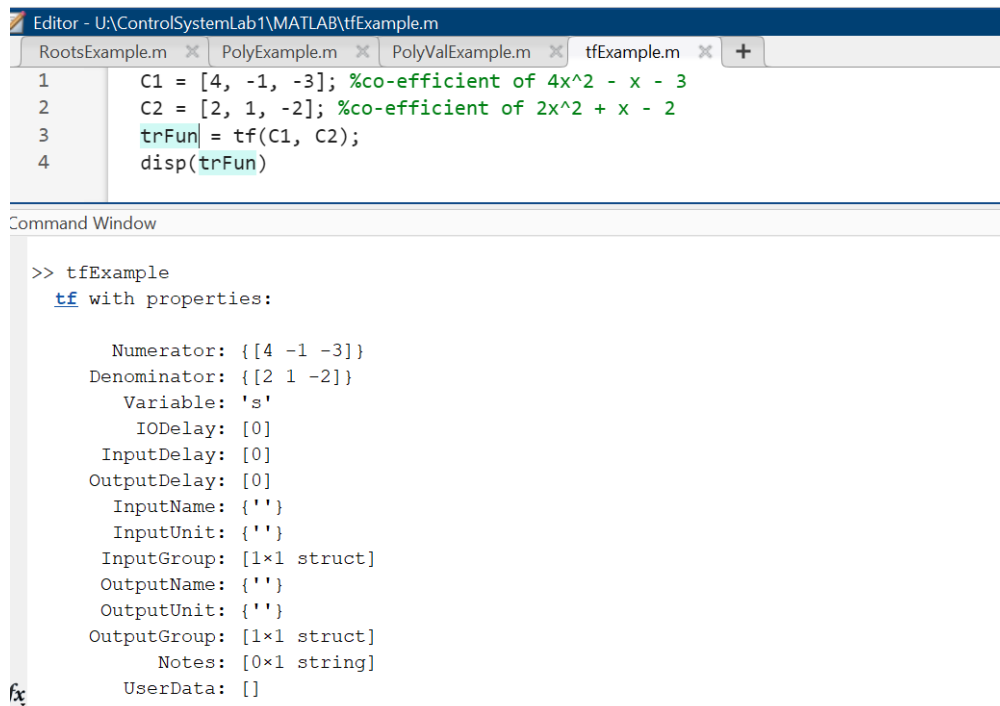
The image shows a MATLAB Editor window with the file `PolyValExample.m` open. The code in the editor is as follows:

```
1 C = [4, -1, -3]; %co-efficient of 4x^2 - x - 3
2 val = polyval(C,1);
3 disp(val)
```

Below the editor is the Command Window, which shows the execution of the script:

```
>> PolyValExample
0
```

Figure 4: Function: `polyval`



The image shows a MATLAB Editor window with the file `tfExample.m` open. The code in the editor is as follows:

```
1 C1 = [4, -1, -3]; %co-efficient of 4x^2 - x - 3
2 C2 = [2, 1, -2]; %co-efficient of 2x^2 + x - 2
3 trFun = tf(C1, C2);
4 disp(trFun)
```

Below the editor is the Command Window, which shows the execution of the script:

```
>> tfExample
tf with properties:
    Numerator: {[4 -1 -3]}
    Denominator: {[2 1 -2]}
    Variable: 's'
    IODelay: [0]
    InputDelay: [0]
    OutputDelay: [0]
    InputName: {''}
    InputUnit: {''}
    InputGroup: [1x1 struct]
    OutputName: {''}
    OutputUnit: {''}
    OutputGroup: [1x1 struct]
    Notes: [0x1 string]
    UserData: []
```

Figure 5: Function: `tf`

```

Editor - U:\ControlSystemLab1\MATLAB\convExample.m
RootsExample.m x PolyExample.m x PolyValExample.m x tfExample.m x convExample.m x +
1      C1 = [4, -1, -3]; %co-efficient of 4x^2 - x - 3
2      C2 = [2, 1, -2]; %co-efficient of 2x^2 + x - 2
3      product = conv(C1, C2);
4      disp(product)

Command Window
>> convExample
      8      2     -15      -1      6

>>

```

Figure 6: Function: conv

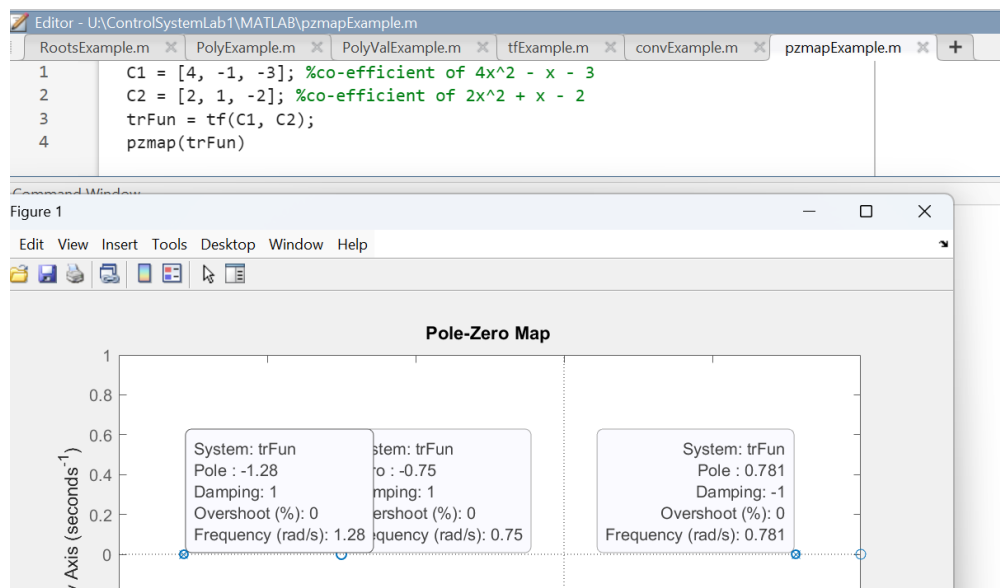


Figure 7: Function: pzmap

`\section{Introduction}`

This lab report summarizes various MATLAB functions used in control systems analysis and design. Examples include polynomials, transfer functions, and system responses.

`\section{MATLAB Functions}`

`\subsection{roots}`

```
\begin{figure}[h]
  \centering
  \includegraphics[width=0.8\textwidth]{placeholder-image} % Replace with your image file
  \caption{Function: \texttt{roots}}
  \label{fig:roots}
```

```
Console output
'geometry' driver: auto-detecting
'geometry' detected driver: pdftex

LaTeX Error: File 'placeholder-image' not found.

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...

.25 ...cs[width=0.8\textwidth]{placeholder-image}
      % Replace with your image...

?
```

Figure 8: Function: `impulse`

`\section{Introduction}`

This lab report summarizes various MATLAB functions used in control systems analysis and design. Examples include polynomials, transfer functions, and system responses.

`\section{MATLAB Functions}`

`\subsection{roots}`

```
\begin{figure}[h]
  \centering
  \includegraphics[width=0.8\textwidth]{placeholder-image} % Replace with your image file
  \caption{Function: \texttt{roots}}
  \label{fig:roots}
```

```
Console output
'geometry' driver: auto-detecting
'geometry' detected driver: pdftex

LaTeX Error: File 'placeholder-image' not found.

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...

.25 ...cs[width=0.8\textwidth]{placeholder-image}
      % Replace with your image...

?
```

Figure 9: Function: `step`

`\section{Introduction}`

This lab report summarizes various MATLAB functions used in control systems analysis and design. E.g. polynomials, transfer functions, and system responses.

`\section{MATLAB Functions}`

`\subsection{roots}`

```
\begin{figure}[h]
  \centering
  \includegraphics[width=0.8\textwidth]{placeholder-image} % Replace with your image file
  \caption{Function: \texttt{roots}}
  \label{fig:roots}
```

```
Console output
'geometry*' driver: auto-detecting
'geometry*' detected driver: pdftex

LaTeX Error: File 'placeholder-image' not found.

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...

.25 ...cs[width=0.8\textwidth]{placeholder-image}
      % Replace with your imag...

?
```

Figure 10: Function: `residue`

`\section{Introduction}`

This lab report summarizes various MATLAB functions used in control systems analysis and design. E.g. polynomials, transfer functions, and system responses.

`\section{MATLAB Functions}`

`\subsection{roots}`

```
\begin{figure}[h]
  \centering
  \includegraphics[width=0.8\textwidth]{placeholder-image} % Replace with your image file
  \caption{Function: \texttt{roots}}
  \label{fig:roots}
```

```
Console output
'geometry*' driver: auto-detecting
'geometry*' detected driver: pdftex

LaTeX Error: File 'placeholder-image' not found.

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...

.25 ...cs[width=0.8\textwidth]{placeholder-image}
      % Replace with your imag...

?
```

Figure 11: Function: `series`

`\section{Introduction}`

This lab report summarizes various MATLAB functions used in control systems analysis and design. Examples include polynomials, transfer functions, and system responses.

`\section{MATLAB Functions}`

`\subsection{roots}`

```
\begin{figure}[h]
  \centering
  \includegraphics[width=0.8\textwidth]{placeholder-image} % Replace with your image file
  \caption{Function: \texttt{roots}}
  \label{fig:roots}
```

```
Console output
'geometry*' driver: auto-detecting
'geometry*' detected driver: pdftex

LaTeX Error: File 'placeholder-image' not found.

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...

.25 ...cs[width=0.8\textwidth]{placeholder-image}
      % Replace with your image...

?
```

Figure 12: Function: `parallel`

`\section{Introduction}`

This lab report summarizes various MATLAB functions used in control systems analysis and design. Examples include polynomials, transfer functions, and system responses.

`\section{MATLAB Functions}`

`\subsection{roots}`

```
\begin{figure}[h]
  \centering
  \includegraphics[width=0.8\textwidth]{placeholder-image} % Replace with your image file
  \caption{Function: \texttt{roots}}
  \label{fig:roots}
```

```
Console output
'geometry*' driver: auto-detecting
'geometry*' detected driver: pdftex

LaTeX Error: File 'placeholder-image' not found.

See the LaTeX manual or LaTeX Companion for explanation.
Type H <return> for immediate help.
...

.25 ...cs[width=0.8\textwidth]{placeholder-image}
      % Replace with your image...

?
```

Figure 13: Function: `feedback`