Michael Korhummel

MECA 482-01

10/10/2021

Homework 2 – Introductory concepts

1. **Please provide:**
   a. **Capabilities Database (project requirements)**

Table 1. Capabilities Database (project requirements)

1.	**REQUIRMENTS**
1.1	**Introduction/Objective:** To design an inverted pendulum that balances vertically (up-right) by rotating on the horizontal plane.
1.2	**Stakeholder needs:** • Pendulum should be able to react and respond to sudden forces or disturbances in a reasonable manner • Pendulum should return to its normal “state” after responding to a disturbance • Have the pendulum balance with some kind of actuator so the pendulum can stay upright. • Stay within given time frame and budget • Be of a proper size for application
1.3	**System Requirements:** • Motion control about x-axis • Motion control about y-axis • Closed loop system to respond to perturbation and disturbances • Keep pendulum vertical through closed loop feedback and control system design
1.4	**Hardware Requirements:** • Motors, actuators to respond to perturbation and disturbances • Rails/Chassis to build pendulum and support and house: • Motors, actuators • Encoder • Power source • Electrical wiring, sensors • Controller
1.5	Implement and operational viewpoint of the project requirements
1.6	Implement a logical/functional viewpoint of project requirements

b. Operational Viewpoint

Furuta Pendulum Operational Viewpoint/Schematic

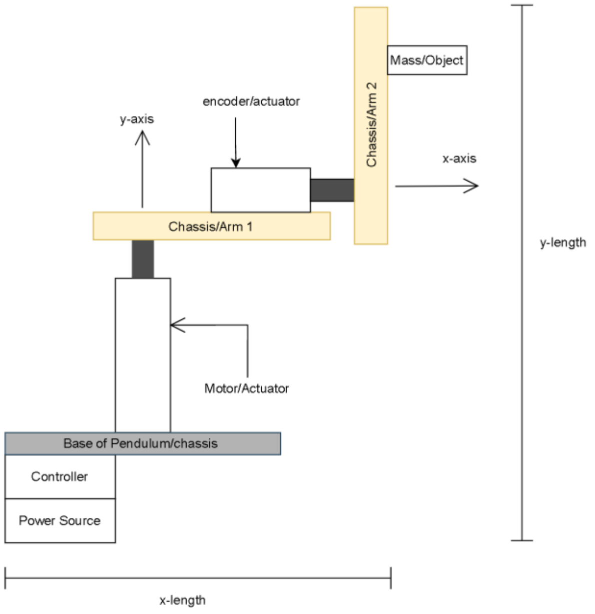


Figure 1. – Side View

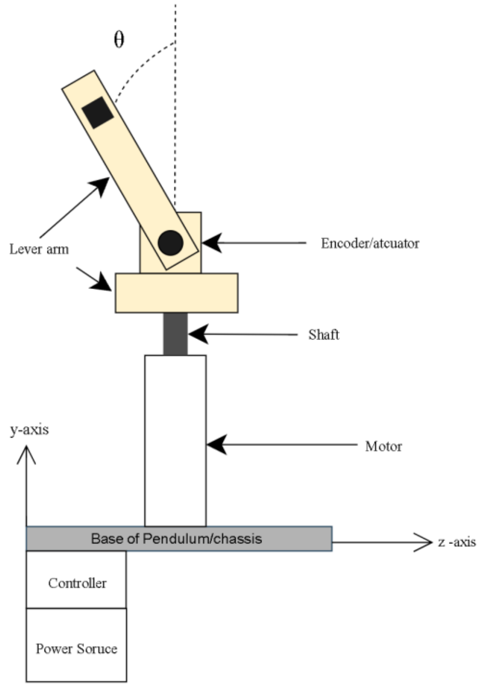


Figure 2. Front View

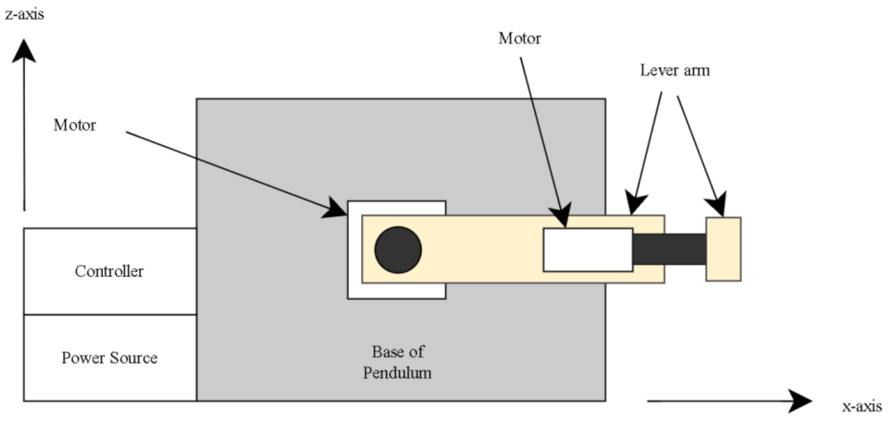


Figure 3. Top View

The inputs for the controller can be programmed through a computer or human machine interface (HMI) to program and test control system.

c. Logical/functional viewpoint

Furuta Pendulum Logical/functional viewpoint

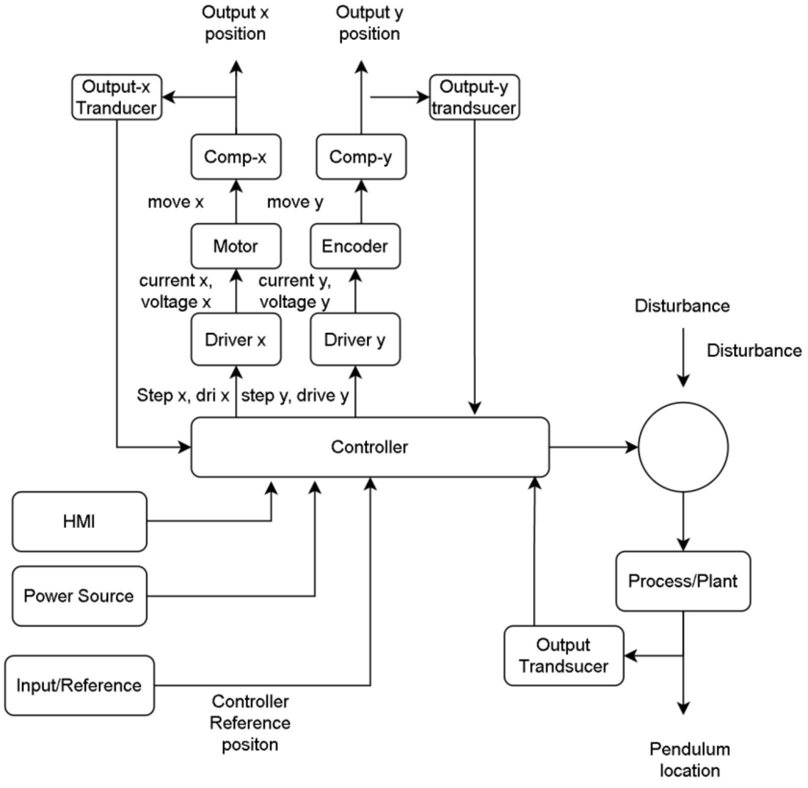
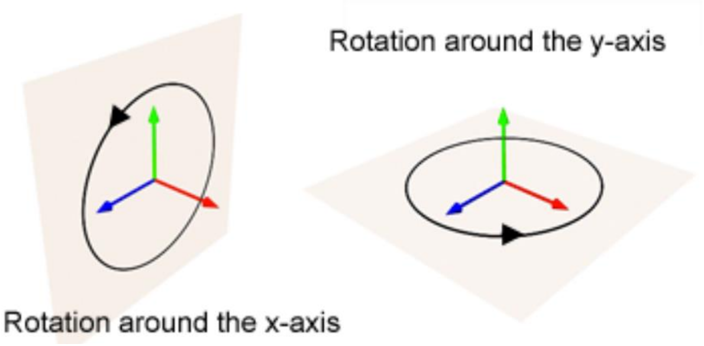


Figure 4. Logical/functional Viewpoint

The logical/functional viewpoint was based off of the motor/actuator moving about the y-axis and the encoder moving about the x-axis. Since the shafts are moving about the y-axis or x-axis, the motors can be controlled based on where about the x-axis or y-axis they are. An input/reference will be used so the motors where the reference about the x or y axis is. If there is a disturbance, the output will change and the motors can then adjust their positions to get back the input/reference position. The output of the x position and output of the y positon are related to the output of the pendulum positon as seen in the logical diagram.



2. **Part of the logical/function diagram that corresponds to the control system that I am designing.**

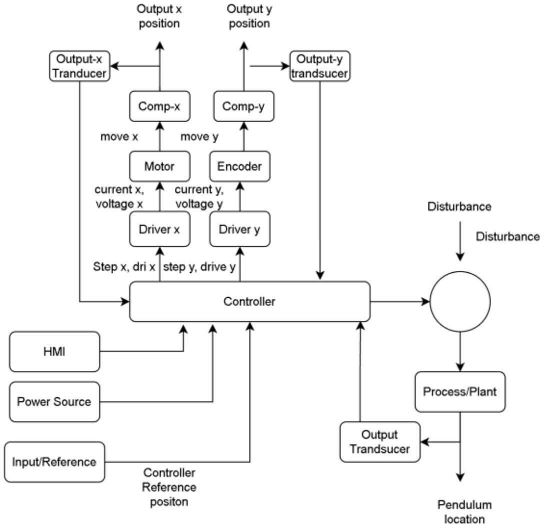


Figure 4. Logical/functional Viewpoint

Based off of figure 4, the logical/functional viewpoint for the furuta pendulum, the part that corresponds to the area of focus is the control around the x and y axis of the encoder and motor. From the operational viewpoint of the diagram, it appears that the motors can be controlled based upon where about the x-axis or y-axis.

3. Task plan by dates/responsibilities of the team members which facilitates the deliverables by the deadline.

Deliverables:

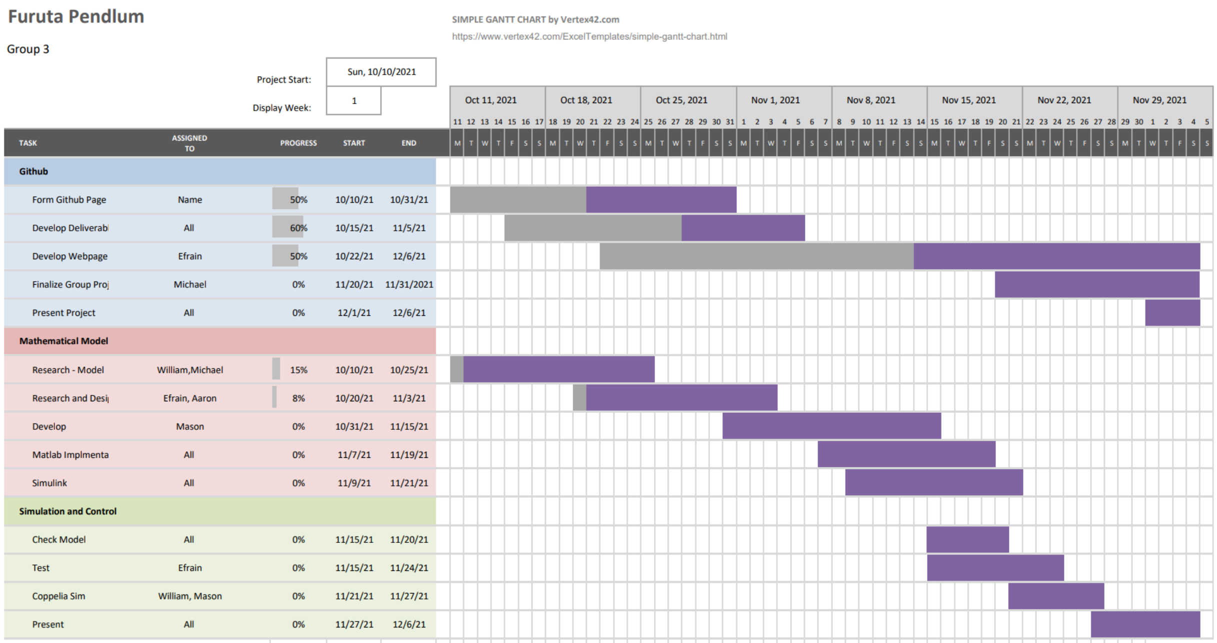
a. The team is expected to present their project (~ 5mins) and build a web page which contains all deliverables in GitHub.

b. The mathematical model of the system must be delivered -preferably in Python or MATLAB. - The control system should be provided preferable in Simulink, State-flow, or with a high-level programming language (e.g., C, C++, or Python). However, the team must show that the control algorithm will give the design requirements for the target system. In other words, one cannot build the system solely hands-on approach similar to the videos below:

   i. Pan and Tilt Mechanism

c. The system will have a simulation with the control system and mathematical model by connecting Coppelia Sim to MATLAB, Simulink, or whatever the programming landscape is used.

d. If the system contains hardware, the design of hardware should consist the necessary architectural explanations such as hardware and software relationships.



Furuta Pendlum

Group 3

SIMPLE GANTT CHART by Vertex42.com
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

Project Start: Sun, 10/10/2021

Display Week: 1

TASK	ASSIGNED TO	PROGRESS	START	END
Github				
Form Github Page	Name	50%	10/10/21	10/31/21
Develop Deliverabl	All	60%	10/15/21	11/5/21
Develop Webpage	Efrain	50%	10/22/21	12/6/21
Finalize Group Proj	Michael	0%	11/20/21	11/31/2021
Present Project	All	0%	12/1/21	12/6/21
Mathematical Model				
Research - Model	William,Michael	15%	10/10/21	10/25/21
Research and Desi	Efrain, Aaron	8%	10/20/21	11/3/21
Develop	Mason	0%	10/31/21	11/15/21
Matlab Implmenta	All	0%	11/7/21	11/19/21
Simulink	All	0%	11/9/21	11/21/21
Simulation and Control				
Check Model	All	0%	11/15/21	11/20/21
Test	Efrain	0%	11/15/21	11/24/21
Coppelia Sim	William, Mason	0%	11/21/21	11/27/21
Present	All	0%	11/27/21	12/6/21

Furuta Pendlum

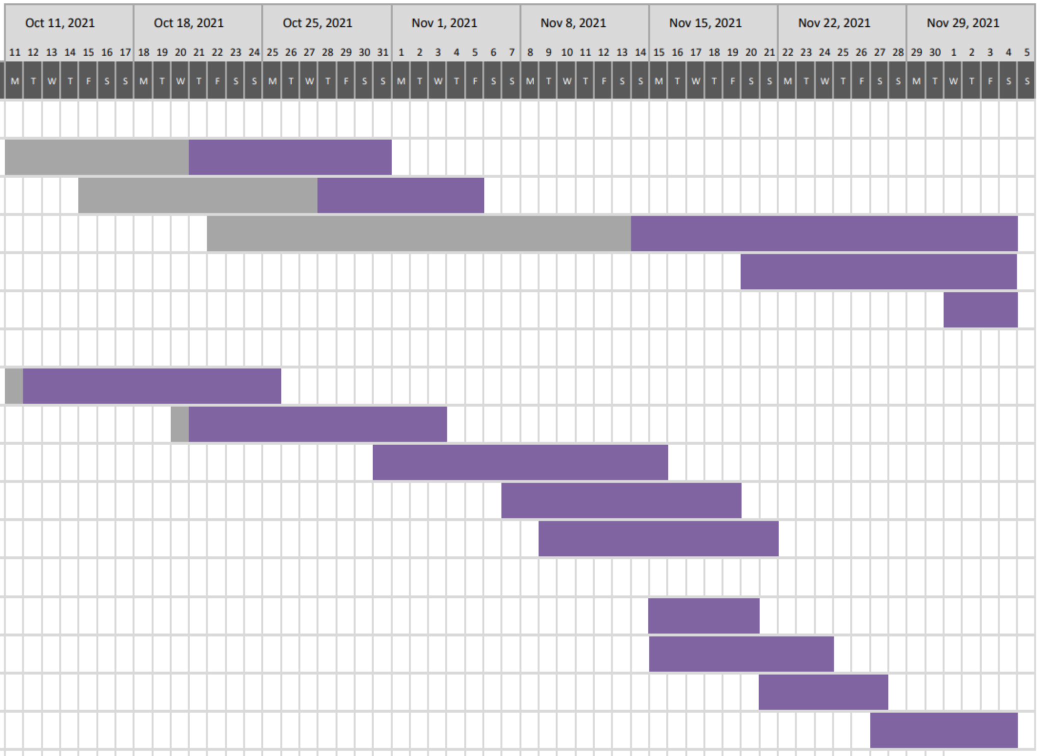
Group 3

SIMPLE GANTT CHART by Vertex42.com

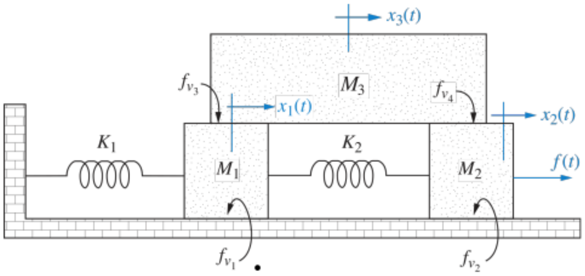
https://www.vertex42.com/ExcelTemplates/simple-gantt-chart.html

Project Start:	Sun, 10/10/2021
Display Week:	1

TASK	ASSIGNED TO	PROGRESS	START	END
Github				
Form Github Page	Name	50%	10/10/21	10/31/21
Develop Deliverabl	All	60%	10/15/21	11/5/21
Develop Webpage	Efrain	50%	10/22/21	12/6/21
Finalize Group Proj	Michael	0%	11/20/21	11/31/2021
Present Project	All	0%	12/1/21	12/6/21
Mathematical Model				
Research - Model	William,Michael	15%	10/10/21	10/25/21
Research and Desi	Efrain, Aaron	8%	10/20/21	11/3/21
Develop	Mason	0%	10/31/21	11/15/21
Matlab Implmenta	All	0%	11/7/21	11/19/21
Simulink	All	0%	11/9/21	11/21/21
Simulation and Control				
Check Model	All	0%	11/15/21	11/20/21
Test	Efrain	0%	11/15/21	11/24/21
Coppelia Sim	William, Mason	0%	11/21/21	11/27/21
Present	All	0%	11/27/21	12/6/21



4. Please find the transfer functions for the following system based on its provided equations in s-domain:



$$[M_1s^2 + (f_{v_1} + f_{v_3})s + (K_1 + K_2)]X_1(s) - K_2X_2(s) - f_{v_3}sX_3(s) = 0$$

$$-K_2X_1(s) + [M_2s^2 + (f_{v_2} + f_{v_4})s + K_2]X_2(s) - f_{v_4}sX_3(s) = F(s)$$

$$-f_{v_3}sX_1(s) - f_{v_4}sX_2(s) + [M_3s \ \ + (f_{v_3} + f_{v_4})s]X_3(s) = 0$$

a. How many different transfer functions would you consider?

Based off of the given equation and the number of degrees of freedom, it is possible to consider 3 transfer functions based off the given information. The equation for a transfer function is as follows:

$$G(s) = \frac{Output}{Input}$$

With our external force, or input being $F(t)$ the three possible outputs can be as followed: $x_1(t)\,, x_2(t)\,, x_3(t)$. With 3 different outputs it is possible to have 3 different transfer functions based on the outputs.

Going from time domain to frequency domain:

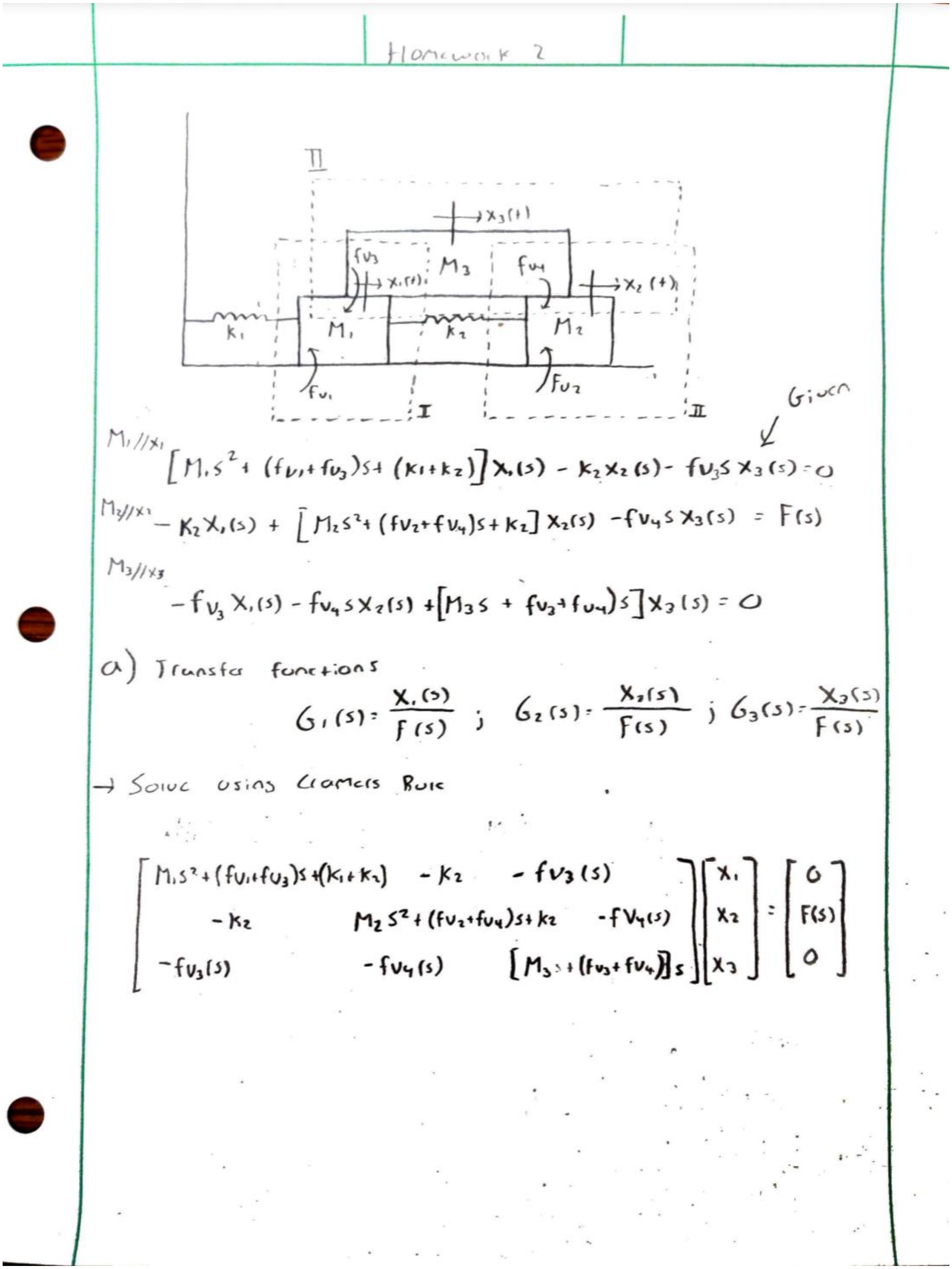
$$x(t) \to x(s)$$

$$G_1(s) = \frac{x_1(s)}{F(s)}$$

$$G_2(s) = \frac{x_2(s)}{F(s)}$$

$$G_3(s) = \frac{x_3(s)}{F(s)}$$

b. By using Cramer's rule (by hand) without any computation tool.


Homework 2
II
$x_3(t)$
f_{v_3}
M_3
f_{v_4}
$x_1(t)$
$x_2(t)$
K_1
M_1
K_2
M_2
f_{v_1}
F_{v_2}
I
II
Given

$M_1//x_1$

$$[M_1 s^2 + (f_{v_1}+f_{v_3})s + (K_1+K_2)]X_1(s) - K_2 X_2(s) - f_{v_3} s\, X_3(s) = 0$$

$M_2//x_2$

$$-K_2 X_1(s) + [M_2 s^2 + (f_{v_2}+f_{v_4})s + K_2]X_2(s) - f_{v_4} s\, X_3(s) = F(s)$$

$M_3//x_3$

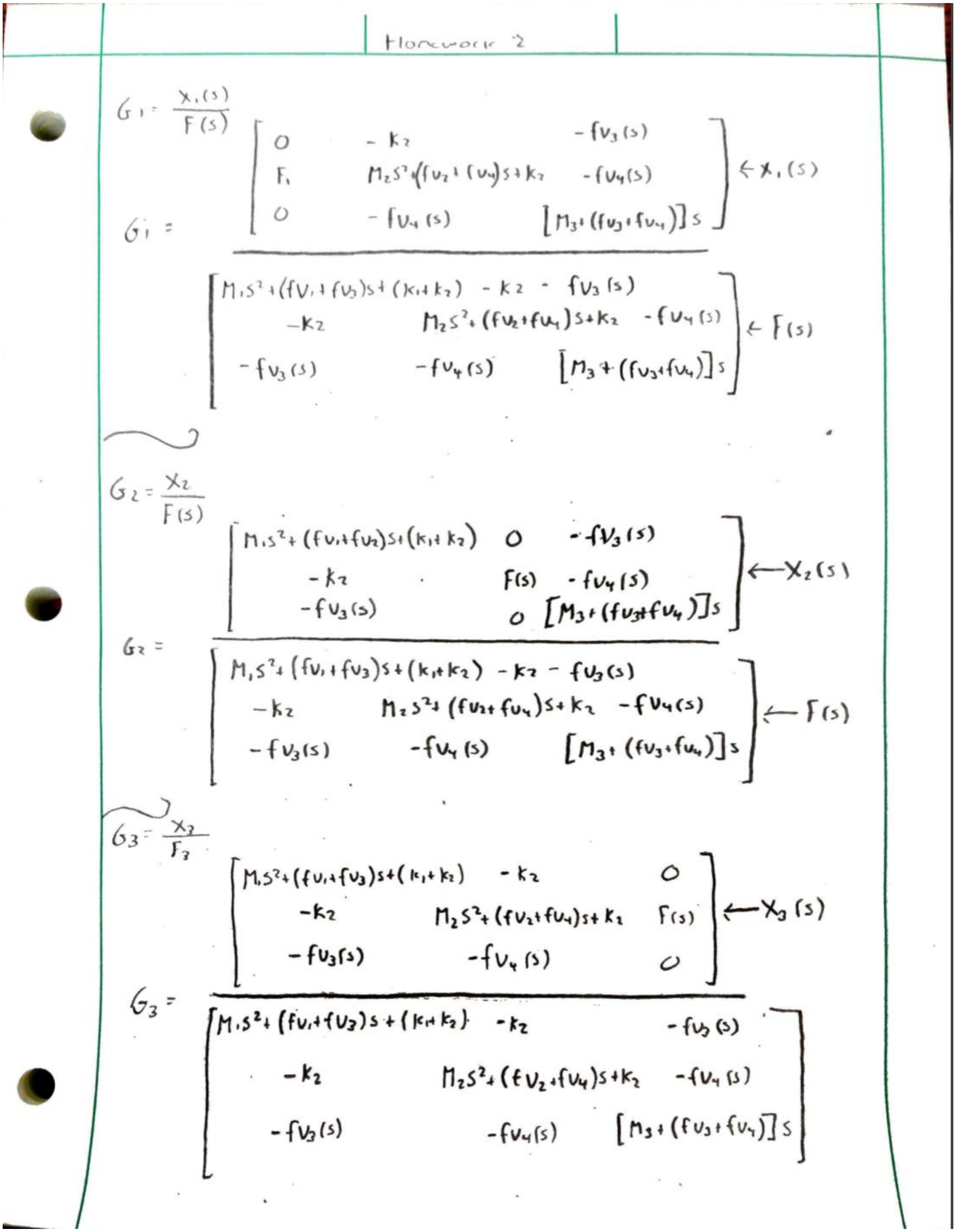
$$-f_{v_3} X_1(s) - f_{v_4} s X_2(s) + [M_3 s + f_{v_3}+f_{v_4})s]X_3(s) = 0$$

a) Transfer functions

$$G_1(s) = \frac{X_1(s)}{F(s)} \; ; \quad G_2(s) = \frac{X_2(s)}{F(s)} \; ; \quad G_3(s) = \frac{X_3(s)}{F(s)}$$

→ Solve using Cramers Rule

$$\begin{bmatrix} M_1 s^2 + (f_{v_1}+f_{v_3})s + (K_1+K_2) & -K_2 & -f_{v_3}(s) \\ -K_2 & M_2 s^2 + (f_{v_2}+f_{v_4})s + K_2 & -f_{v_4}(s) \\ -f_{v_3}(s) & -f_{v_4}(s) & [M_3 s + (f_{v_3}+f_{v_4})]s \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} = \begin{bmatrix} 0 \\ F(s) \\ 0 \end{bmatrix}$$



Homework 2

$$G_1 = \frac{X_1(s)}{F(s)}$$

$$G_1 = \frac{\left[\begin{matrix} 0 & -k_2 & -fv_3(s) \\ F_1 & M_2S^2+(fv_2+fv_4)s+k_2 & -fv_4(s) \\ 0 & -fv_4(s) & [M_3+(fv_3+fv_4)]s \end{matrix}\right] \leftarrow X_1(s)}{\left[\begin{matrix} M_1s^2+(fV_1+fv_3)s+(k_1+k_2) & -k_2 & -fv_3(s) \\ -k_2 & M_2S^2+(fv_2+fv_4)s+k_2 & -fv_4(s) \\ -fv_3(s) & -fv_4(s) & [M_3+(fv_3+fv_4)]s \end{matrix}\right] \leftarrow F(s)}$$

$$G_2 = \frac{X_2}{F(s)}$$

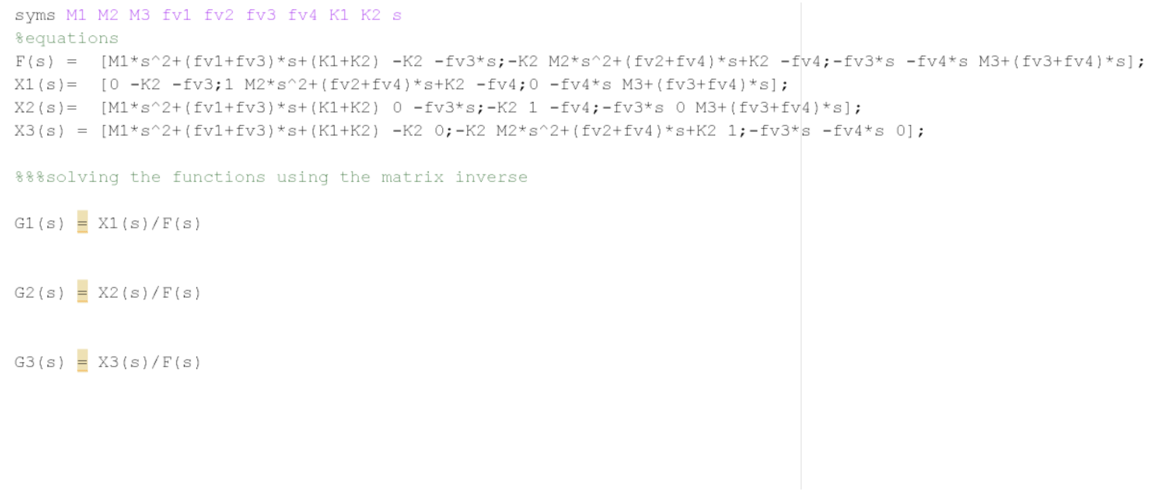
$$G_2 = \frac{\left[\begin{matrix} M_1s^2+(fv_1+fv_2)s+(k_1+k_2) & 0 & -fV_3(s) \\ -k_2 & F(s) & -fv_4(s) \\ -fv_3(s) & 0 & [M_3+(fv_3+fv_4)]s \end{matrix}\right] \longleftarrow X_2(s)}{\left[\begin{matrix} M_1s^2+(fv_1+fv_3)s+(k_1+k_2) & -k_2 & -fv_3(s) \\ -k_2 & M_2s^2+(fv_2+fv_4)s+k_2 & -fv_4(s) \\ -fv_3(s) & -fv_4(s) & [M_3+(fv_3+fv_4)]s \end{matrix}\right] \longleftarrow F(s)}$$

$$G_3 = \frac{X_3}{F_3}$$

$$G_3 = \frac{\left[\begin{matrix} M_1S^2+(fv_1+fv_3)s+(k_1+k_2) & -k_2 & 0 \\ -k_2 & M_2S^2+(fV_2+fv_4)s+k_2 & F(s) \\ -fv_3(s) & -fV_4(s) & 0 \end{matrix}\right] \longleftarrow X_3(s)}{\left[\begin{matrix} M_1s^2+(fv_1+fU_3)s+(k_1+k_2) & -k_2 & -fv_3(s) \\ -k_2 & M_2S^2+(fV_2+fv_4)s+k_2 & -fv_4(s) \\ -fV_3(s) & -fv_4(s) & [M_3+(fv_3+fv_4)]s \end{matrix}\right]}$$

c. By using MATLAB matrix inverse

First, open a command window and type in the given transfer functions found using cramers rule



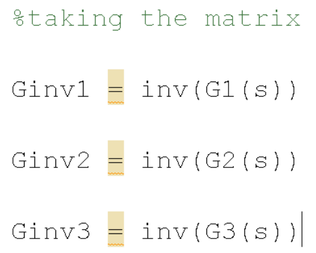
```
syms M1 M2 M3 fv1 fv2 fv3 fv4 K1 K2 s
%equations
F(s) =  [M1*s^2+(fv1+fv3)*s+(K1+K2) -K2 -fv3*s;-K2 M2*s^2+(fv2+fv4)*s+K2 -fv4;-fv3*s -fv4*s M3+(fv3+fv4)*s];
X1(s)=  [0 -K2 -fv3;1 M2*s^2+(fv2+fv4)*s+K2 -fv4;0 -fv4*s M3+(fv3+fv4)*s];
X2(s)=  [M1*s^2+(fv1+fv3)*s+(K1+K2) 0 -fv3*s;-K2 1 -fv4;-fv3*s 0 M3+(fv3+fv4)*s];
X3(s) = [M1*s^2+(fv1+fv3)*s+(K1+K2) -K2 0;-K2 M2*s^2+(fv2+fv4)*s+K2 1;-fv3*s -fv4*s 0];

%%%solving the functions using the matrix inverse

G1(s) = X1(s)/F(s)


G2(s) = X2(s)/F(s)


G3(s) = X3(s)/F(s)
```

The following transfer functions are as followed: G_1, G_2, G_3

```
G1(s) =

[
-(K2^2*M3 + M2*fv3^2*s^3 + fv2*fv3^2*s^2 + fv3^2*fv4*s^2 + K2*fv3^2*s
+ K2^2*fv3*s + K2^2*fv4*s + 2*K2*fv3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 +
K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 - fv1*fv4^2*s^2 +
fv1*fv4^2*s^3 - fv3*fv4^2*s^2 + fv3*fv4^2*s^3 - K1*fv4^2*s -
K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 + M1*M2*M3*s^4
+ K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 + K2*M1*fv4*s^3 +
K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 + M2*M3*fv1*s^3 +
M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 + M1*M2*fv4*s^5 +
K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 +
K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 +
M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 +
M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 +
M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 +
fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s + K1*K2*fv4*s +
K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s + K2*M3*fv3*s +
K2*M3*fv4*s - K2*fv3*fv4*s),
-(K2^2*M3 + K1*K2*M3 + fv3^2*fv4*s^2 + K2*fv3^2*s + K2^2*fv3*s +
K2^2*fv4*s + K2*M1*M3*s^2 + K2*M1*fv3*s^3 + K2*M1*fv4*s^3 +
K2*fv1*fv3*s^2 + K2*fv1*fv4*s^2 + K2*fv3*fv4*s^2 + M1*fv3*fv4*s^3 +
fv1*fv3*fv4*s^2 + K1*K2*fv3*s + K1*K2*fv4*s + K2*M3*fv1*s +
K2*M3*fv3*s + K1*fv3*fv4*s + K2*fv3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 +
K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 - fv1*fv4^2*s^2 +
fv1*fv4^2*s^3 - fv3*fv4^2*s^2 + fv3*fv4^2*s^3 - K1*fv4^2*s -
K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 + M1*M2*M3*s^4
+ K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 + K2*M1*fv4*s^3 +
K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 + M2*M3*fv1*s^3 +
M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 + M1*M2*fv4*s^5 +
K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 +
K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 +
M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 +
M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 +
M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 +
fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s + K1*K2*fv4*s +
K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s + K2*M3*fv3*s +
K2*M3*fv4*s - K2*fv3*fv4*s),
-(K2^2*fv4 + M2*fv3^2*s^3 + K1*K2*fv3 + K1*K2*fv4 + fv2*fv3^2*s^2 +
fv3^2*fv4*s^2 + K2*fv3^2*s + K2^2*fv3*s + K1*M2*fv3*s^2 +
K2*M1*fv3*s^2 + K2*M1*fv4*s^2 + K2*M2*fv3*s^2 + M1*M2*fv3*s^4 +
M1*fv2*fv3*s^3 + M2*fv1*fv3*s^3 + M1*fv3*fv4*s^3 + fv1*fv2*fv3*s^2 +
fv1*fv3*fv4*s^2 + K1*fv2*fv3*s + K2*fv1*fv3*s + K2*fv1*fv4*s +
K2*fv2*fv3*s + K1*fv3*fv4*s + 2*K2*fv3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2
+ K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 - fv1*fv4^2*s^2 +
fv1*fv4^2*s^3 - fv3*fv4^2*s^2 + fv3*fv4^2*s^3 - K1*fv4^2*s -
K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 + M1*M2*M3*s^4
+ K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 + K2*M1*fv4*s^3 +
K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 + M2*M3*fv1*s^3 +
M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 + M1*M2*fv4*s^5 +
K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 +
K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 +
M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 +
M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 +
M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 +
fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s + K1*K2*fv4*s +
K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s + K2*M3*fv3*s +
K2*M3*fv4*s - K2*fv3*fv4*s)]

[ (K2^2*M3 - fv4^2*s + K2*M3 + fv4^2*s^2 + K2*fv4^2*s^2 + K2*fv3*s +
K2*fv4*s + M3*fv2*s + M3*fv4*s + M2*M3*s^2 + K2^2*fv3*s - K2*fv4^2*s +
K2^2*fv4*s + M2*fv3*s^3 + M2*fv4*s^3 + fv2*fv3*s^2 + fv2*fv4*s^2 +
fv3*fv4*s^2 + K2*M2*M3*s^2 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
K2*fv2*fv3*s^2 + K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + K2*M3*fv2*s +
K2*M3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s), (K2^2*M3 +
K2*M3 + K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 + K2*fv3*s + K2*fv4*s - K1*fv4^2*s + K2^2*fv3*s -
K2*fv4^2*s + K2^2*fv4*s + fv3*fv4*s^2 + K1*M2*M3*s^2 + K2*M1*M3*s^2 +
K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + 2*K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s)/(K1*K2*M3 +
K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 -
fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 + fv3*fv4^2*s^3 -
K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 +
M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 +
K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 +
M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 +
M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 +
K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 +
K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 +
M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 +
M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 +
fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s +
K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s +
K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s),
(K2*fv4 + K2^2*fv4 + K2*fv3*s + K2^2*fv3*s + M2*fv3*s^3 + fv2*fv3*s^2
+ fv3*fv4*s^2 + K2*M2*fv3*s^3 + K2*fv2*fv3*s^2 +
K2*fv3*fv4*s^2)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3
+ M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s)]
```

```
[
(K2*fv3^2*s^2 + M2*fv3^2*s^4 + fv2*fv3^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 + fv3^2*fv4*s^3 + M2*M3*fv3*s^3 + K2*fv3*fv4*s^2 +
M3*fv2*fv3*s^2 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv2*fv3*fv4*s^3 +
K2*M3*fv3*s)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s),
(s*(fv4*fv3^2*s^2 + K2*fv3^2*s + K2*fv4*fv3*s + K2*M3*fv3))/(K1*K2*M3
+ K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 -
fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 + fv3*fv4^2*s^3 -
K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 +
M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 +
K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 +
M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 +
M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 +
K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 +
K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 +
M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 +
M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 +
fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s +
K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s +
K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s), (K1*K2*M3 + K1*fv4^2*s^2 +
K2*fv3^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 +
M2*fv3^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 + fv2*fv3^2*s^3 -
fv3*fv4^2*s^2 + fv3*fv4^2*s^3 + fv3^2*fv4*s^3 - K1*fv4^2*s -
K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 + M1*M2*M3*s^4
+ K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 + K2*M1*fv4*s^3 +
K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 + M2*M3*fv1*s^3 +
M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 + M1*M2*fv4*s^5 +
K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 +
K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 +
M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 +
M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 +
M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 +
fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s + K1*K2*fv4*s +
K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s + K2*M3*fv3*s +
K2*M3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s)]


G2(s) =

[ (K2^2*M3 + K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s + K2^2*fv3*s - K2*fv4^2*s + K2^2*fv4*s +
K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 + M1*M2*M3*s^4 +
K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 + K2*M1*fv4*s^3 +
K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 + M2*M3*fv1*s^3 +
M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 + M1*M2*fv4*s^5 +
K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 +
K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 +
M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 +
M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 +
M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 +
fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s + K1*K2*fv4*s +
K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s + K2*M3*fv3*s +
K2*M3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s),
(K2^2*M3 + K1*K2*M3 + K2^2*fv3*s + K2^2*fv4*s + K2*M1*M3*s^2 +
K2*M1*fv3*s^3 + K2*M1*fv4*s^3 + K2*fv1*fv3*s^2 + K2*fv1*fv4*s^2 +
K2*fv3*fv4*s^2 + K1*K2*fv3*s + K1*K2*fv4*s + K2*M3*fv1*s +
K2*M3*fv3*s)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s),
(K2^2*fv4 + K1*K2*fv4 + K2^2*fv3*s + K2*M1*fv4*s^2 + K2*fv1*fv4*s +
K2*fv3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s)]

[
-(K2^2*M3 - K2*M3 + K2*fv4^2*s^2 + fv3*fv4^2*s^2 - K2*fv3*s - K2*fv4*s
- fv3*fv4*s + K2^2*fv3*s + K2^2*fv4*s + K2*M2*M3*s^2 + K2*M2*fv3*s^3 +
K2*M2*fv4*s^3 + K2*fv2*fv3*s^2 + K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 +
M2*fv3*fv4*s^3 + fv2*fv3*fv4*s^2 + K2*M3*fv2*s + K2*M3*fv4*s +
K2*fv3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s), (K1*M3 -
K2^2*M3 + K2*M3 - M1*fv4^2*s^3 - fv1*fv4^2*s^2 - fv3*fv4^2*s^2 +
K1*fv3*s + K1*fv4*s + K2*fv3*s + K2*fv4*s + M3*fv1*s + M3*fv3*s +
M1*M3*s^2 - K1*fv4^2*s - K2^2*fv3*s - K2*fv4^2*s - K2^2*fv4*s +
M1*fv3*s^3 + M1*fv4*s^3 + fv1*fv3*s^2 + fv1*fv4*s^2 + fv3*fv4*s^2 -
K2*fv3*fv4*s^2 - K2*fv3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2
- M1*fv4^2*s^3 + M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 -
fv3*fv4^2*s^2 + fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2
+ K2*M1*M3*s^2 + K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 +
K2*M1*fv3*s^3 + K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 +
K2*M2*fv4*s^3 + M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 +
M1*M3*fv4*s^3 + M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 +
K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 +
K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 +
M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 +
M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 +
M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 +
fv2*fv3*fv4*s^3 + K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s +
K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s -
K2*fv3*fv4*s),
-(K2^2*fv4 - K2*fv4 - K1*fv4 + M1*fv4^2*s^3 + K1*K2*fv4 +
fv1*fv4^2*s^2 + fv3*fv4^2*s^2 - K2*fv3*s - fv1*fv4*s - fv3*fv4*s +
K1*fv4^2*s + K2^2*fv3*s + K2*fv4^2*s - M1*fv4*s^2 + K1*M2*fv4*s^2 +
K2*M1*fv4*s^2 + K2*M2*fv3*s^3 + K2*M2*fv4*s^2 + M1*M2*fv4*s^4 +
K2*fv2*fv3*s^2 + K2*fv3*fv4*s^2 + M1*fv2*fv4*s^3 + M2*fv1*fv4*s^3 +
M2*fv3*fv4*s^3 + fv1*fv2*fv4*s^2 + fv2*fv3*fv4*s^2 + K1*fv2*fv4*s +
K2*fv1*fv4*s + K2*fv2*fv4*s + K2*fv3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 +
K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 - fv1*fv4^2*s^2 +
fv1*fv4^2*s^3 - fv3*fv4^2*s^2 + fv3*fv4^2*s^3 - K1*fv4^2*s -
K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 + M1*M2*M3*s^4
+ K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 + K2*M1*fv4*s^3 +
K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 + M2*M3*fv1*s^3 +
M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 + M1*M2*fv4*s^5 +
K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 +
K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 +
M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 +
M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 +
M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 +
fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s + K1*K2*fv4*s +
K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s + K2*M3*fv3*s +
K2*M3*fv4*s - K2*fv3*fv4*s)]

[
(K2*fv4^2*s^2 + fv3*fv4^2*s^2 + K2*fv3*fv4*s^2 +
K2*M3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s),
(s*(M1*fv4^2*s^3 + K1*M3*fv4 + K2*M3*fv4 + fv1*fv4^2*s^2 +
fv3*fv4^2*s^2 + K1*fv4^2*s + K2*fv4^2*s + M1*M3*fv4*s^2 +
M1*fv3*fv4*s^3 + fv1*fv3*fv4*s^2 + K1*fv3*fv4*s + K2*fv3*fv4*s +
M3*fv1*fv4*s + M3*fv3*fv4*s))/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2
- M1*fv4^2*s^3 + M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 -
fv3*fv4^2*s^2 + fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2
+ K2*M1*M3*s^2 + K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 +
K2*M1*fv3*s^3 + K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 +
K2*M2*fv4*s^3 + M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 +
M1*M3*fv4*s^3 + M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 +
K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 +
K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 +
M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 +
M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 +
M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 +
fv2*fv3*fv4*s^3 + K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s +
K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s -
K2*fv3*fv4*s), (K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 + M1*fv4^2*s^4
+ fv1*fv4^2*s^3 + fv3*fv4^2*s^3 + K1*M2*M3*s^2 + K2*M1*M3*s^2 +
K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + 2*K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s)/(K1*K2*M3 +
K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 -
fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 + fv3*fv4^2*s^3 -
K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 +
M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 +
K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 +
M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 +
M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 +
K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 +
K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 +
M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 +
M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 +
fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s +
K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s +
K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s)]


G3(s) =

[ (K1*K2*M3 + K1*fv4^2*s^2 + K2*fv3^2*s^2 + K2*fv4^2*s^2 -
M1*fv4^2*s^3 + M1*fv4^2*s^4 + M2*fv3^2*s^4 - fv1*fv4^2*s^2 +
fv1*fv4^2*s^3 + fv2*fv3^2*s^3 - fv3*fv4^2*s^2 + fv3*fv4^2*s^3 +
fv3^2*fv4*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + 2*K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s)/(K1*K2*M3 +
K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 -
fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 + fv3*fv4^2*s^3 -
K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 +
M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 +
K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 +
M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 +
M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 +
K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 +
K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 +
M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 +
M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 +
fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s +
K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s +
K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s),
(fv3*s^2*(K1*fv4 + K2*fv3 + K2*fv4 + fv1*fv4*s + fv3*fv4*s +
M1*fv4*s^2))/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s),
(K2*fv3^2*s^2 + M2*fv3^2*s^4 + fv2*fv3^2*s^3 + fv3^2*fv4*s^3 +
K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K2*M2*fv3*s^3 + M1*M2*fv3*s^5 +
K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv3*fv4*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 + M1*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv3*fv4*s^3 + K1*K2*fv3*s)/(K1*K2*M3 +
K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 -
fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 + fv3*fv4^2*s^3 -
K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 +
M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 +
K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 +
M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 +
M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 +
K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 +
K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 +
M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 +
M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 +
fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s +
K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s +
K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s)]

[
(fv3*fv4^2*s^2 + K2*fv3*s + K2*fv4*s + K2*fv4^2*s + M2*fv3*s^3 +
fv2*fv3*s^2 + fv3*fv4*s^2 + M2*fv3*fv4*s^3 + fv2*fv3*fv4*s^2 +
K2*fv3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 +
M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2
+ K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 +
K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 +
M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 +
M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 +
K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 +
M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3
+ K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s
+ K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s), (K1*K2*M3 +
K1*fv4^2*s^2 + K2*fv4^2*s^2 + M1*fv4^2*s^4 + fv1*fv4^2*s^3 +
fv3*fv4^2*s^3 + K1*fv4*s + K2*fv3*s + K2*fv4*s + M1*fv4*s^3 +
fv1*fv4*s^2 + fv3*fv4*s^2 + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2
+ M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 +
K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 +
M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 +
M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 +
K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 +
K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 +
M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 +
M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 +
fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s +
K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s +
K2*M3*fv3*s + K2*M3*fv4*s)/(K1*K2*M3 + K1*fv4^2*s^2 + K2*fv4^2*s^2 -
M1*fv4^2*s^3 + M1*fv4^2*s^4 - fv1*fv4^2*s^2 + fv1*fv4^2*s^3 -
fv3*fv4^2*s^2 + fv3*fv4^2*s^3 - K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2
+ K2*M1*M3*s^2 + K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 +
K2*M1*fv3*s^3 + K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 +
K2*M2*fv4*s^3 + M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 +
M1*M3*fv4*s^3 + M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 +
K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 +
K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 + K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 +
M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 +
M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 +
M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 +
fv2*fv3*fv4*s^3 + K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s +
K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s -
K2*fv3*fv4*s), (K1*K2 + M1*fv4^2*s^3 + K1*K2*fv4 + fv1*fv4^2*s^2 +
fv3*fv4^2*s^2 + K1*fv2*s + K2*fv1*s + K2*fv2*s + K1*fv4*s + K2*fv3*s +
K2*fv4*s + K1*M2*s^2 + K2*M1*s^2 + K2*M2*s^2 + M1*M2*s^4 + K1*fv4^2*s
+ K2*fv4^2*s + M1*fv2*s^3 + M2*fv1*s^3 + M1*fv4*s^3 + M2*fv3*s^3 +
fv1*fv2*s^2 + fv1*fv4*s^2 + fv2*fv3*s^2 + fv3*fv4*s^2 + K1*M2*fv4*s^2
+ K2*M1*fv4*s^2 + K2*M2*fv4*s^2 + M1*M2*fv4*s^4 + M1*fv2*fv4*s^3 +
M2*fv1*fv4*s^3 + M2*fv3*fv4*s^3 + fv1*fv2*fv4*s^2 + fv2*fv3*fv4*s^2 +
K1*fv2*fv4*s + K2*fv1*fv4*s + K2*fv2*fv4*s + K2*fv3*fv4*s)/(K1*K2*M3 +
K1*fv4^2*s^2 + K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 -
fv1*fv4^2*s^2 + fv1*fv4^2*s^3 - fv3*fv4^2*s^2 + fv3*fv4^2*s^3 -
K1*fv4^2*s - K2*fv4^2*s + K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 +
M1*M2*M3*s^4 + K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 +
K2*M1*fv4*s^3 + K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 +
M2*M3*fv1*s^3 + M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 +
M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 +
K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 +
K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 +
M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 +
M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 +
fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 + K1*K2*fv3*s +
K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s +
K2*M3*fv3*s + K2*M3*fv4*s - K2*fv3*fv4*s)]

[
```

To find the inverse, simply take the inverse of the transfer function.



```
%taking the matrix inverse

Ginv1 = inv(G1(s))

Ginv2 = inv(G2(s))

Ginv3 = inv(G3(s))
```

```
Ginv1 =

[ (K2^2*M3 + K2*M3 + K1*K2*M3 + K1*fv4^2*s^2 + K2*fv3^2*s^2 +
K2*fv4^2*s^2 - M1*fv4^2*s^3 + M1*fv4^2*s^4 + M2*fv3^2*s^4 -
fv1*fv4^2*s^2 + fv1*fv4^2*s^3 + fv2*fv3^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 + fv3^2*fv4*s^3 + K2*fv3*s + K2*fv4*s - K1*fv4^2*s
+ K2^2*fv3*s - K2*fv4^2*s + K2^2*fv4*s + fv3*fv4*s^2 +
K1*M2*M3*s^2 + K2*M1*M3*s^2 + K2*M2*M3*s^2 + M1*M2*M3*s^4 +
K1*M2*fv3*s^3 + K2*M1*fv3*s^3 + K1*M2*fv4*s^3 + K2*M1*fv4*s^3 +
K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + M1*M3*fv2*s^3 + M2*M3*fv1*s^3 +
M1*M2*fv3*s^5 + M1*M3*fv4*s^3 + M2*M3*fv3*s^3 + M1*M2*fv4*s^5 +
K1*fv2*fv3*s^2 + K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 +
K2*fv1*fv4*s^2 + K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 +
K2*fv2*fv4*s^2 + 2*K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 +
M1*fv2*fv3*s^4 + M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 +
M3*fv2*fv3*s^2 + M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 +
M1*fv3*fv4*s^4 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv1*fv2*fv3*s^3 + fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 +
fv2*fv3*fv4*s^3 + K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s +
K2*M3*fv1*s + K2*M3*fv2*s + K1*M3*fv4*s + K2*M3*fv3*s +
K2*M3*fv4*s)/(K2*M3 + K2*fv3*s + K2*fv4*s + fv3*fv4*s), K1 + K2
+ fv1*s + fv3*s + M1*s^2, (K2*fv3 + K2^2*fv3 + K2^2*fv4 +
M2*fv3^2*s^3 + K1*K2*fv3 + K1*K2*fv4 + fv2*fv3^2*s^2 +
fv3^2*fv4*s^2 - K2*fv3*s + K2*fv3^2*s + K1*M2*fv3*s^2 +
K2*M1*fv3*s^2 + K2*M1*fv4*s^2 + K2*M2*fv3*s^2 + M1*M2*fv3*s^4 +
M1*fv2*fv3*s^3 + M2*fv1*fv3*s^3 + M1*fv3*fv4*s^3 +
fv1*fv2*fv3*s^2 + fv1*fv3*fv4*s^2 + K1*fv2*fv3*s + K2*fv1*fv3*s
+ K2*fv1*fv4*s + K2*fv2*fv3*s + K1*fv3*fv4*s +
2*K2*fv3*fv4*s)/(K2*M3 + K2*fv3*s + K2*fv4*s + fv3*fv4*s)]
[
-(K2^2*M3 - fv4^2*s + K2*M3 + fv4^2*s^2 + K2*fv4^2*s^2 +
K2*fv3*s + K2*fv4*s + M3*fv2*s + M3*fv4*s + M2*M3*s^2 +
K2^2*fv3*s - K2*fv4^2*s + K2^2*fv4*s + M2*fv3*s^3 + M2*fv4*s^3 +
fv2*fv3*s^2 + fv2*fv4*s^2 + fv3*fv4*s^2 + K2*M2*M3*s^2 +
K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + K2*fv2*fv3*s^2 + K2*fv2*fv4*s^2
+ K2*fv3*fv4*s^2 + K2*M3*fv2*s + K2*M3*fv4*s)/(K2*M3 + K2*fv3*s
+ K2*fv4*s + fv3*fv4*s),                                   -K2,
-(K2*fv3 + K2*fv4 + K2^2*fv3 + K2^2*fv4 + fv2*fv3*s + fv3*fv4*s
+ M2*fv3*s^2 + K2*M2*fv3*s^2 + K2*fv2*fv3*s +
K2*fv3*fv4*s)/(K2*M3 + K2*fv3*s + K2*fv4*s + fv3*fv4*s)]
[
-(K2*fv3^2*s^2 + M2*fv3^2*s^4 + fv2*fv3^2*s^3 - fv3*fv4^2*s^2 +
fv3*fv4^2*s^3 + fv3^2*fv4*s^3 + M2*M3*fv3*s^3 + K2*fv3*fv4*s^2 +
M3*fv2*fv3*s^2 + M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 +
fv2*fv3*fv4*s^3 + K2*M3*fv3*s)/(K2*M3 + K2*fv3*s + K2*fv4*s +
fv3*fv4*s),                            -fv3*s,
-(M2*fv3^2*s^3 - K2*M3 + fv2*fv3^2*s^2 + fv3^2*fv4*s^2 -
K2*fv3*s - K2*fv4*s - fv3*fv4*s + K2*fv3^2*s +
K2*fv3*fv4*s)/(K2*M3 + K2*fv3*s + K2*fv4*s + fv3*fv4*s)]


Ginv2 =

[
(K1*M3 - K2^2*M3 + K2*M3 + K1*fv3*s + K1*fv4*s + K2*fv3*s +
K2*fv4*s + M3*fv1*s + M3*fv3*s + M1*M3*s^2 - K2^2*fv3*s -
K2^2*fv4*s + M1*fv3*s^3 + M1*fv4*s^3 + fv1*fv3*s^2 + fv1*fv4*s^2
+ fv3*fv4*s^2 - K2*fv3*fv4*s)/(K1*M3 + K2*M3 + K1*fv3*s +
K1*fv4*s + K2*fv3*s + K2*fv4*s + M3*fv1*s + M3*fv3*s + M1*M3*s^2
+ M1*fv3*s^3 + M1*fv4*s^3 + fv1*fv3*s^2 + fv1*fv4*s^2 +
fv3*fv4*s^2),                          -K2,
-(K2^2*fv4 + K1*K2*fv4 + K2^2*fv3*s + K2*M1*fv4*s^2 +
K2*fv1*fv4*s + K2*fv3*fv4*s)/(K1*M3 + K2*M3 + K1*fv3*s +
K1*fv4*s + K2*fv3*s + K2*fv4*s + M3*fv1*s + M3*fv3*s + M1*M3*s^2
+ M1*fv3*s^3 + M1*fv4*s^3 + fv1*fv3*s^2 + fv1*fv4*s^2 +
fv3*fv4*s^2)]
[ (K2^2*M3 - K2*M3 + K2*fv4^2*s^2 + fv3*fv4^2*s^2 - K2*fv3*s -
K2*fv4*s - fv3*fv4*s + K2^2*fv3*s + K2^2*fv4*s + K2*M2*M3*s^2 +
K2*M2*fv3*s^3 + K2*M2*fv4*s^3 + K2*fv2*fv3*s^2 + K2*fv2*fv4*s^2
+ K2*fv3*fv4*s^2 + M2*fv3*fv4*s^3 + fv2*fv3*fv4*s^2 +
K2*M3*fv2*s + K2*M3*fv4*s + K2*fv3*fv4*s)/(K1*M3 + K2*M3 +
K1*fv3*s + K1*fv4*s + K2*fv3*s + K2*fv4*s + M3*fv1*s + M3*fv3*s
+ M1*M3*s^2 + M1*fv3*s^3 + M1*fv4*s^3 + fv1*fv3*s^2 +
fv1*fv4*s^2 + fv3*fv4*s^2), K2 + fv2*s + fv4*s + M2*s^2,
(K2^2*fv4 - K2*fv4 - K1*fv4 + M1*fv4^2*s^3 + K1*K2*fv4 +
fv1*fv4^2*s^2 + fv3*fv4^2*s^2 - K2*fv3*s - fv1*fv4*s - fv3*fv4*s
+ K1*fv4^2*s + K2^2*fv3*s + K2*fv4^2*s - M1*fv4*s^2 +
K1*M2*fv4*s^2 + K2*M1*fv4*s^2 + K2*M2*fv3*s^3 + K2*M2*fv4*s^2 +
M1*M2*fv4*s^4 + K2*fv2*fv3*s^2 + K2*fv3*fv4*s^2 + M1*fv2*fv4*s^3
+ M2*fv1*fv4*s^3 + M2*fv3*fv4*s^3 + fv1*fv2*fv4*s^2 +
fv2*fv3*fv4*s^2 + K1*fv2*fv4*s + K2*fv1*fv4*s + K2*fv2*fv4*s +
K2*fv3*fv4*s)/(K1*M3 + K2*M3 + K1*fv3*s + K1*fv4*s + K2*fv3*s +
K2*fv4*s + M3*fv1*s + M3*fv3*s + M1*M3*s^2 + M1*fv3*s^3 +
M1*fv4*s^3 + fv1*fv3*s^2 + fv1*fv4*s^2 + fv3*fv4*s^2)]
```

```
[
-(K2*fv4^2*s^2 + fv3*fv4^2*s^2 + K2*fv3*fv4*s^2 +
K2*M3*fv4*s)/(K1*M3 + K2*M3 + K1*fv3*s + K1*fv4*s + K2*fv3*s +
K2*fv4*s + M3*fv1*s + M3*fv3*s + M1*M3*s^2 + M1*fv3*s^3 +
M1*fv4*s^3 + fv1*fv3*s^2 + fv1*fv4*s^2 + fv3*fv4*s^2),
-fv4*s,
(K1*M3 + K2*M3 - M1*fv4^2*s^3 - fv1*fv4^2*s^2 - fv3*fv4^2*s^2 +
K1*fv3*s + K1*fv4*s + K2*fv3*s + K2*fv4*s + M3*fv1*s + M3*fv3*s
+ M1*M3*s^2 - K1*fv4^2*s - K2*fv4^2*s + M1*fv3*s^3 + M1*fv4*s^3
+ fv1*fv3*s^2 + fv1*fv4*s^2 + fv3*fv4*s^2 -
K2*fv3*fv4*s^2)/(K1*M3 + K2*M3 + K1*fv3*s + K1*fv4*s + K2*fv3*s
+ K2*fv4*s + M3*fv1*s + M3*fv3*s + M1*M3*s^2 + M1*fv3*s^3 +
M1*fv4*s^3 + fv1*fv3*s^2 + fv1*fv4*s^2 + fv3*fv4*s^2)]


Ginv3 =

[
(K1*fv4 + K2*fv3 + K2*fv4 - M2*fv3^2*s^3 - fv2*fv3^2*s^2 -
fv3^2*fv4*s^2 + fv1*fv4*s + fv3*fv4*s - K2*fv3^2*s + M1*fv4*s^2
- K2*fv3*fv4*s)/(K1*fv4 + K2*fv3 + K2*fv4 + fv1*fv4*s +
fv3*fv4*s + M1*fv4*s^2),          -fv3*s,
-(M2*fv3^2*s^3 + K1*K2*fv3 + fv2*fv3^2*s^2 + fv3^2*fv4*s^2 +
K2*fv3^2*s + K1*M2*fv3*s^2 + K2*M1*fv3*s^2 + K2*M2*fv3*s^2 +
M1*M2*fv3*s^4 + M1*fv2*fv3*s^3 + M2*fv1*fv3*s^3 + M1*fv3*fv4*s^3
+ fv1*fv2*fv3*s^2 + fv1*fv3*fv4*s^2 + K1*fv2*fv3*s +
K2*fv1*fv3*s + K2*fv2*fv3*s + K1*fv3*fv4*s +
K2*fv3*fv4*s)/(K1*fv4 + K2*fv3 + K2*fv4 + fv1*fv4*s + fv3*fv4*s
+ M1*fv4*s^2)]
[
-(K2*fv3 + K2*fv4 + K2*fv4^2 + K2*fv3*fv4 + fv2*fv3*s +
fv3*fv4*s + M2*fv3*s^2 + fv3*fv4^2*s + fv2*fv3*fv4*s +
M2*fv3*fv4*s^2)/(K1*fv4 + K2*fv3 + K2*fv4 + fv1*fv4*s +
fv3*fv4*s + M1*fv4*s^2),            -fv4,
-(K1*K2 + M1*fv4^2*s^3 + K1*K2*fv4 + fv1*fv4^2*s^2 +
fv3*fv4^2*s^2 + K1*fv2*s + K2*fv1*s + K2*fv2*s + K1*fv4*s +
K2*fv3*s + K2*fv4*s + K1*M2*s^2 + K2*M1*s^2 + K2*M2*s^2 +
M1*M2*s^4 + K1*fv4^2*s + K2*fv4^2*s + M1*fv2*s^3 + M2*fv1*s^3 +
M1*fv4*s^3 + M2*fv3*s^3 + fv1*fv2*s^2 + fv1*fv4*s^2 +
fv2*fv3*s^2 + fv3*fv4*s^2 + K1*M2*fv4*s^2 + K2*M1*fv4*s^2 +
K2*M2*fv4*s^2 + M1*M2*fv4*s^4 + M1*fv2*fv4*s^3 + M2*fv1*fv4*s^3
+ M2*fv3*fv4*s^3 + fv1*fv2*fv4*s^2 + fv2*fv3*fv4*s^2 +
K1*fv2*fv4*s + K2*fv1*fv4*s + K2*fv2*fv4*s +
K2*fv3*fv4*s)/(K1*fv4*s + K2*fv3*s + K2*fv4*s + M1*fv4*s^3 +
fv1*fv4*s^2 + fv3*fv4*s^2)]
[ (M2*fv3^2*s^3 + K2*M3*fv3 + K2*M3*fv4 + fv2*fv3^2*s^2 +
fv3*fv4^2*s^2 + fv3^2*fv4*s^2 + K2*fv3^2*s + K2*fv4^2*s +
M2*M3*fv3*s^2 + M2*fv3*fv4*s^3 + fv2*fv3*fv4*s^2 +
2*K2*fv3*fv4*s + M3*fv2*fv3*s + M3*fv3*fv4*s)/(K1*fv4 + K2*fv3 +
K2*fv4 + fv1*fv4*s + fv3*fv4*s + M1*fv4*s^2), M3 + fv3*s +
fv4*s, (K1*K2*M3 + K1*fv4^2*s^2 + K2*fv3^2*s^2 + K2*fv4^2*s^2 +
M1*fv4^2*s^4 + M2*fv3^2*s^4 + fv1*fv4^2*s^3 + fv2*fv3^2*s^3 +
fv3*fv4^2*s^3 + fv3^2*fv4*s^3 + K1*fv4*s + K2*fv3*s + K2*fv4*s +
M1*fv4*s^3 + fv1*fv4*s^2 + fv3*fv4*s^2 + K1*M2*M3*s^2 +
K2*M1*M3*s^2 + K2*M2*M3*s^2 + M1*M2*M3*s^4 + K1*M2*fv3*s^3 +
K2*M1*fv3*s^3 + K1*M2*fv4*s^3 + K2*M1*fv4*s^3 + K2*M2*fv3*s^3 +
K2*M2*fv4*s^3 + M1*M3*fv2*s^3 + M2*M3*fv1*s^3 + M1*M2*fv3*s^5 +
M1*M3*fv4*s^3 + M2*M3*fv3*s^3 + M1*M2*fv4*s^5 + K1*fv2*fv3*s^2 +
K2*fv1*fv3*s^2 + K1*fv2*fv4*s^2 + K2*fv1*fv4*s^2 +
K2*fv2*fv3*s^2 + K1*fv3*fv4*s^2 + K2*fv2*fv4*s^2 +
2*K2*fv3*fv4*s^2 + M3*fv1*fv2*s^2 + M1*fv2*fv3*s^4 +
M2*fv1*fv3*s^4 + M3*fv1*fv4*s^2 + M3*fv2*fv3*s^2 +
M1*fv2*fv4*s^4 + M2*fv1*fv4*s^4 + M1*fv3*fv4*s^4 +
M3*fv3*fv4*s^2 + M2*fv3*fv4*s^4 + fv1*fv2*fv3*s^3 +
fv1*fv2*fv4*s^3 + fv1*fv3*fv4*s^3 + fv2*fv3*fv4*s^3 +
K1*K2*fv3*s + K1*K2*fv4*s + K1*M3*fv2*s + K2*M3*fv1*s +
K2*M3*fv2*s + K1*M3*fv4*s + K2*M3*fv3*s + K2*M3*fv4*s)/(K1*fv4*s
+ K2*fv3*s + K2*fv4*s + M1*fv4*s^3 + fv1*fv4*s^2 + fv3*fv4*s^2)]
```