

Symbolic Algebra in Dynamic Systems and Controls Classes

Joey K. Parker
The University of Alabama

Abstract

Large sets of symbolic simultaneous linear equations occur frequently in the types of problems found in system dynamics and control courses. Students often have difficulty with algebraic manipulation of several symbolic equations. Three example problems (finding state variable equations for an electric circuit, developing transfer functions from sets of state variable equations, and block diagram reduction) show how symbolic algebra can be used to reduce tedious algebraic manipulation in system dynamics and control courses.

Introduction

Many mechanical engineering undergraduate programs include courses in dynamic system modeling and control, either separately or within a single course. Mechanical, electrical, and thermal systems with significant dynamic components are frequently modeled. Combinations of electrical and mechanical or electrical and thermal systems are also common. Modeling of these systems results in a small number of differential equations (typically linear, first-order, constant coefficients), with another larger set of related algebraic equations. These systems of equations must be reduced by algebraic manipulation to the proper form, typically state variable equations or transfer functions. Students often have difficulty with symbolic manipulation at this level of complexity - up to five or so "knowns" and five to ten "unknowns." The ability to manipulate these symbols algebraically is not indicative of student understanding of the fundamental concepts of dynamic systems and controls. In fact, many students spend most of their effort on such problems doing algebraic manipulations and are never able to move beyond this stage of solving a systems dynamics problem.

Modern computational systems, such as Matlab and to a lesser degree MathCad and Maple, are often used to solve problems in dynamic system/control system modeling and analysis. A number of textbook supplements^{1,2,3,4,5} have been written that use Matlab for numerical solutions in control system design and analysis. An excellent set of tutorials with extensive examples using Matlab in control system design has been widely available for a few years⁶. Matlab's Simulink⁷ and Visual Solutions/Mathsoft's Micro-VisSim⁸ have graphical interfaces that allow users to build control system block diagrams and perform numerical solutions on the modeled system. Gerber⁹ reports the use of Maple to solve numerical problems in electric circuit analysis. A set of examples showing Maple used in both analog and linear control system analysis including inverse Laplace transforms is available on the Adept Scientific website¹⁰. A set of Maple routines useful for classical frequency domain (Nyquist and Bode plots, root locus) analysis and linear regulator and stochastic optimal controller designs is also available¹¹.

All of the work reported above details the use of modern computational tools to solve numerical problems in dynamic/control system analysis. Maple (and to some degree Mathcad and Matlab) provides extensive symbolic manipulation capabilities at relatively low cost. Use of these symbolic manipulation tools allows valuable class time (and student's out-of-class time) to be focused on important concepts, while low-level tasks are delegated to the computer. Specific examples of the use of symbolic algebra (Maple) to simplify tedious tasks commonly encountered in both dynamic system and control system modeling include

- solving simultaneous linear equations (from electric circuits problems in particular),
- developing transfer functions (from sets of equations and from state variables), and
- block diagram reduction.

Solving Simultaneous Linear Equations

System models often generate a large number of simultaneous linear equations. Students typically attempt to solve these equations using an unorganized back-substitution method. A large amount of unproductive time is wasted doing something easily handled by Maple or other symbolic algebra packages. An example electrical network consisting of two resistors, a capacitor, and an inductor is shown in Figure 1.

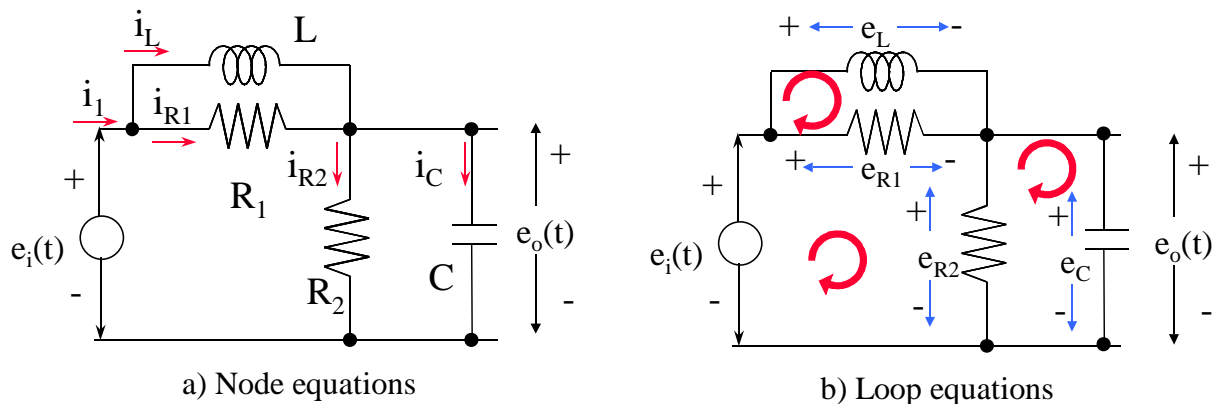


Figure 1 – Circuit for Example #1

Node equations can be written from Figure 1a as

$$i_1 - i_L - i_{R1} = 0 \quad (1)$$

$$i_L + i_{R1} = i_{R2} + i_C \quad (2)$$

Many students would recognize that the two equations above could be combined to eliminate i_1 as a variable. Unfortunately, this step would not occur to all students and leads to some confusion as to how to proceed. By using Maple to solve the simultaneous equations, students can focus on writing correct equations, and not worry about tedious symbolic algebra manipulations. Continuing with the problem, three loop equations can be written from Figure 1b,

$$e_L - e_{R1} = 0 \quad (3)$$

$$e_i(t) - e_L - e_{R2} = 0 \quad (4)$$

$$e_{R2} - e_C = 0 \quad (5)$$

Again, many students would recognize that equation #5 offers the opportunity to eliminate an unknown. Some students would correctly use this equation to eliminate e_{R2} , but others would mistakenly eliminate e_C , which will be needed at a later stage. Four element laws can now be written, one for each element in the system,

$$e_{R1} = R_1 i_{R1} \quad (6)$$

$$e_{R2} = R_2 i_{R2} \quad (7)$$

$$e_C = \frac{1}{C} \int_0^t i_C dt + e_C(0) \rightarrow \frac{de_C}{dt} = \frac{1}{C} i_C \quad (8)$$

$$i_L = \frac{1}{L} \int_0^t e_L dt + i_L(0) \rightarrow \frac{di_L}{dt} = \frac{1}{L} e_L \quad (9)$$

My experience has been that virtually all students that have had a course in electric circuits can learn to write these nine equations correctly in a relatively short time. What many of them have difficulty with is deciding what to do next. A very systematic approach can be taken at this point. State variables must be identified first. There should be a state variable and a state variable equation associated with each independent energy storing element in the system. In this case, the most logical choice of state variables would be

- the voltage across the capacitor, e_C , and
- the current through the inductor, i_L .

One of the requirements of the state variable approach is that a first-order differential equation must be written for each of these state variables. This differential equation must be written in terms of the state variables and the inputs - no other variables (or derivatives of variables) are permitted. For the example problem, equations #8 and #9 are almost in state variable form already. However, two variables on the right-hand side of the equations (i_C and e_L) are neither state variables nor inputs. The other seven equations must be used to express these variables in terms of state variables e_C and i_L and the input $e_i(t)$.

Maple can easily solve this problem of seven simultaneous linear equations. The first seven equations above (#1 - #7) are entered and appear as shown in Figure 2. These equations are easily checked since they appear in exactly the same form that a student would write them.

Maple is a case-sensitive language, so students must be careful to enter all of the symbols consistently. For example, the letter "C" is always written as a capital in this example. These seven equations can be solved by Maple with the "solve" command as shown in Figure 3. The seven equations to be solved are grouped in the first set of curly braces. The second set of curly braces contains the seven dependent variables to be found. The "assign" and "expand" commands are not required in the solution process, but are included to show additional capabilities of Maple. The "assign" command explicitly associates the seven solutions with the seven dependent variables. The "expand" command can then be used with any of the dependent variables to re-format the solution by clearing common terms from numerator and denominator.

```

> eq1 := i[1] - i[L] - i[R1] = 0;
                                eq1 := i1 - iL - iR1 = 0
> eq2 := i[L] + i[R1] = i[R2] + i[C];
                                eq2 := iL + iR1 = iR2 + iC
> eq3 := e[L] - e[R1] = 0;
                                eq3 := eL - eR1 = 0
> eq4 := e[i] - e[L] - e[R2] = 0;
                                eq4 := ei - eL - eR2 = 0
> eq5 := e[R2] - e[C] = 0;
                                eq5 := eR2 - eC = 0
> eq6 := e[R1] = R[1]*i[R1];
                                eq6 := eR1 = R1 iR1
> eq7 := e[R2] = R[2]*i[R2];
                                eq7 := eR2 = R2 iR2

```

Figure 2 – Maple input for Example #1

```

> sol := solve({eq1,eq2,eq3,eq4,eq5,eq6,eq7},
  {i[C],e[L],i[1],i[R1],i[R2],e[R1],e[R2]});
sol := {eR2 = eC, eL = ei - eC, eR1 = ei - eC, iC =
                                iL R1 R2 + R2 ei - R2 eC - eC R1
                                R1 R2,
  iR2 = eC/R2, iR1 = (ei - eC)/R1, i1 = (iL R1 + ei - eC)/R1}
> assign(sol);
> expand(i[C]);
                                iL + ei/R1 - eC/R1 - eC/R2
> expand(e[L]);
                                ei - eC

```

Figure 3 – Maple output for Example #1

Students sometimes have difficulty determining which variables to solve for, i.e., the terms in the second set of curly braces in the "solve" command. The rule for this is actually quite simple - solve for all variables (currents and voltages) that are not state variables or inputs. The results from Maple can be substituted directly into equations #8 and #9 to get the state variable equations.

$$\frac{de_c}{dt} = \frac{1}{C} i_C = \frac{1}{C} i_L - \left(\frac{1}{R_1 C} + \frac{1}{R_2 C} \right) e_C + \frac{1}{R_1 C} e_i(t) \quad \frac{di_L}{dt} = \frac{1}{L} e_L = \frac{1}{L} e_i(t) - \frac{1}{L} e_C \quad (10)$$

Transfer Functions from State Variable Equations

The first example showed how to solve a system of modeling equations to find state variable equations. In the second example, a set of state variable equations will be converted to an input-output transfer function representation. The electro-mechanical system shown in Figure 4 is made up of a permanent magnet DC motor (modeled as an inertia J_m) coupled through a flexible shaft (with stiffness K) to a load modeled as an inertia (J_L) and viscous friction (B_L). The electrical aspects of the DC motor are modeled as an armature resistance (R_a) in series with the armature inductance (L_a). The electro-mechanical equations coupling the two parts of the system are given in Figure 4. There are four independent energy storing elements in this system, thus there are four state variables,

- the motor's armature current, i_a ,
- the angular velocity of the motor, ω_m ,
- the twist in the shaft, $\Delta\theta$, and
- the angular velocity of the load, ω_L ,

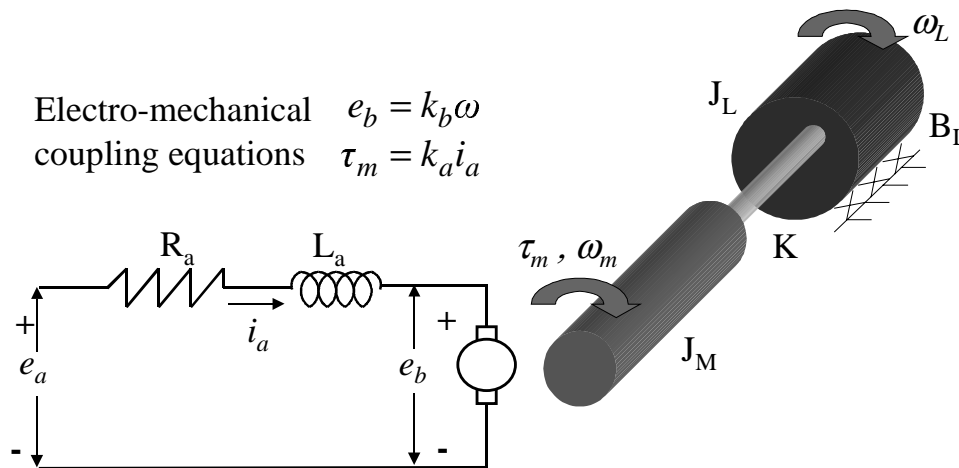


Figure 4 – Electro-mechanical system - Example #2

The state variable equations for this system are given in Figure 5. In preparation for finding the transfer function, the state variable equations have been Laplace transformed and the initial conditions on the four state variables have been set to zero.

The colon used at the end of the "solve" command above suppresses the printed output. Note that all four possible transfer functions are solved with this single command. The "assign" command is again used to associate the solution with the variable ω_L . The first term in the "collect" command in Figure 5 divides the symbolic solution for ω_L by the input e_a . The "collect" command is used to group all of the terms with the same power of s in the solution for the transfer function ω_L/e_a . Unfortunately the denominator of this transfer function is not in the best possible form. Figure 6 illustrates some of the Maple commands that are available to reformat the denominator of the transfer function for better understanding. The "denom" command separates the denominator from the remainder of the solution for ω_L . All terms are also divided by $L_a J_m J_L$, which was the coefficient of the highest power of s in the original denominator. The "expand" and "collect" commands are used as before for formatting.

```

> eq1 := s*i[a] = (1/L[a]) * (e[a] - R[a]*i[a] - k[b]*omega[m]);
      eq1 := s i_a = \frac{e_a - R_a i_a - k_b \omega_m}{L_a}
> eq2 := s*Delta = omega[m] - omega[L];
      eq2 := s \Delta = \omega_m - \omega_L
> eq3 := s*omega[m] = (1/J[m]) * (k[a]*i[a] - K*Delta);
      eq3 := s \omega_m = \frac{k_a i_a - K \Delta}{J_m}
> eq4 := s*omega[L] = (1/J[L]) * (-B[L]*omega[L] + K*Delta);
      eq4 := s \omega_L = \frac{-B_L \omega_L + K \Delta}{J_L}
> sol := solve({eq1, eq2, eq3, eq4}, {i[a], omega[m], omega[L], Delta});
> assign(sol);
> collect(omega[L]/e[a], s);
      K k_a / (s^4 L_a J_m J_L + (L_a J_m B_L + R_a J_m J_L) s^3 + (L_a J_m K + R_a J_m B_L + k_b J_L k_a + L_a K J_L) s^2
      + (R_a J_m K + L_a K B_L + k_b B_L k_a + R_a K J_L) s + k_b K k_a + R_a K B_L)

```

Figure 5 – Electro-mechanical system - Maple input and results

```

> den := denom( omega[L] ) / (L[a]*J[m]*J[L]):
> expand(den);
      s^4 + \frac{s^3 B_L}{J_L} + \frac{s^2 K}{J_L} + \frac{s^2 K}{J_m} + \frac{s K B_L}{J_m J_L} + \frac{R_a s^3}{L_a} + \frac{R_a s^2 B_L}{L_a J_L} + \frac{R_a s K}{L_a J_L} + \frac{R_a K s}{L_a J_m} + \frac{R_a K B_L}{L_a J_m J_L} + \frac{k_b s^2 k_a}{L_a J_m}
      + \frac{k_b s B_L k_a}{L_a J_m J_L} + \frac{k_b K k_a}{L_a J_m J_L}
> collect(% , s);
      s^4 + \left( \frac{B_L}{J_L} + \frac{R_a}{L_a} \right) s^3 + \left( \frac{K}{J_L} + \frac{R_a B_L}{L_a J_L} + \frac{k_b k_a}{L_a J_m} + \frac{K}{J_m} \right) s^2 + \left( \frac{R_a K}{L_a J_L} + \frac{K B_L}{J_m J_L} + \frac{k_b B_L k_a}{L_a J_m J_L} + \frac{R_a K}{L_a J_m} \right) s
      + \frac{k_b K k_a}{L_a J_m J_L} + \frac{R_a K B_L}{L_a J_m J_L}

```

Figure 6 – Electro-mechanical system - reformatted denominator

The advantage of this form for the denominator of the transfer function ω_L/e_a occurs when specific values are substituted for the system constants. Each of the two terms $(B_L/J_L$ and $R_a/L_a)$ multiplying the s^3 term must have the units of sec^{-1} . Similarly, each of the four terms multiplying the s^2 term $(K/J, k_a k_b/L_a J_m, \text{etc.})$ must have the units of sec^{-2} . Units for each of the other six terms in the denominator are found by multiplying two of the six terms that have already been found. Each of these six terms had units of either sec^{-1} or sec^{-2} . Students that struggle with unit conversions find this form of the complicated transfer function particularly useful.

Block Diagram Reduction

A conventional block diagram representation for a control system is shown in Figure 7. This system has two inputs: $R(s)$ is the reference input and $N(s)$ is a noise or disturbance signal. The output of the control system is $C(s)$. The blocks labeled G_i and H_i represent transfer functions for various parts of the system. The goal of block diagram reduction is to combine the elements of the diagram (blocks, summing junctions, "pickoff" points, etc.) to find the two input/output transfer functions

$$\left. \frac{C(s)}{R(s)} \right|_{N(s)=0} \quad \text{and} \quad \left. \frac{C(s)}{N(s)} \right|_{R(s)=0} \quad (11)$$

Many of the rules for block diagram reduction are quite straightforward. For example, two blocks directly in series (no summing junction or pickoff point between them) can be combined by simply multiplying the transfer function in the two blocks. Other rules are not as simple, i.e. the rule for reducing a feedback loop is not intuitively obvious. Once students have applied the rules for block diagram reduction, they need a way to check for correctness. The symbolic capabilities of Maple can be used to provide this check on their work.

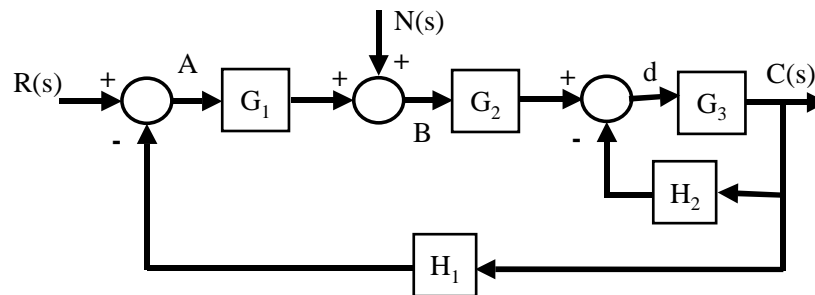


Figure 7 – Control system block diagram #1

The symbolic algebra approach for finding the transfer functions for Figure 7 is shown in Figure 8. The output of the first summing block is defined as a new variable, A, and the resulting equation is assigned to the variable eq1. Similarly, the output of the second summing block is defined as a new variable, B, and the resulting equation is assigned to the variable eq2. Since $C(s)$ is the output, we define the output of the third summing block as variable d (the symbol "D" is reserved in Maple) in eq3. The last equation (eq4) defines the output C in terms of the block G_3 and the "variable" d. The same simultaneous linear equation approach is used to solve for each of the unknowns (A, B, C, and d) in terms of the transfer functions (G_1 , G_2 , G_3 , H_1 , and H_2) and the inputs (R and N).

One of the advantages to the Maple approach is that both transfer functions are found with no additional effort. The solution form shown above also helps students understand the meaning of transfer functions, i.e., that the total output $C(s)$ is formed by the combination of contributions from both $R(s)$ and $N(s)$ inputs.

```

> eq1 := A = R-H[1]*C;
                                eq1 := A = R - H1 C
> eq2:= B = A*G[1] + N;
                                eq2 := B = A G1 + N
> eq3 := d = B*G[2]-C*H[2];
                                eq3 := d = B G2 - C H2
> eq4 := C = G[3]*d;
                                eq4 := C = G3 d
> sol := solve( {eq1,eq2,eq3,eq4} , {A,B,C,d} ) :
> assign(sol) ;
> collect(C, R) ;

```

$$\frac{G_2 G_1 G_3 R}{1 + G_2 G_1 H_1 G_3 + G_3 H_2} + \frac{G_2 G_3 N}{1 + G_2 G_1 H_1 G_3 + G_3 H_2}$$

Figure 8 – Block diagram reduction #1 - Maple solution

Conclusions

Explicit sets of simultaneous linear equations occur frequently in system dynamics and control system classes. Other common problems, such as finding transfer functions from state variable equations and block diagram reduction, can also be cast as sets of simultaneous linear equations. Many students find solving three or more simultaneous equations difficult, particularly when they are written in symbolic form. Three examples drawn from an undergraduate systems dynamics class are shown to demonstrate the process of solving simultaneous linear equations with Maple. These examples can be used as templates for solving similar problems. Many students initially resist using Maple, since not all of them have had previous exposure. However, most students quickly learn that this modern engineering tool frees them from tedious algebraic manipulations and allows them to accurately solve much larger and more significant problems.

References

1. Kuo, B.C. and D.C. Hanselman. (1994) *Matlab[®] Tools for Control System Analysis and Design*, Prentice-Hall.
2. Leonard, N.E. and W.S. Levine. (1992) *Using Matlab[™] to Analyze and Design Control Systems*, Benjamin-Cummings.
3. Ogata, K. (1994) *Solving Control Engineering Problems with Matlab[®]*, Prentice-Hall.
4. Ogata, K. (1994) *Designing Linear Control Systems with Matlab[®]*, Prentice-Hall.
5. Shahian, B. and M. Hassul. (1993) *Control System Design using Matlab[®]*, Prentice-Hall.
6. Messner, W.C. and D. M. Tilbury. *Control Tutorials for Matlab[®] and Simulink*, Addison-Wesley Longman. URL: <http://www.engin.umich.edu/group/ctm/>.
7. URL: <http://www.mathworks.com/products/simulink/>; The Mathworks - Simulink.
8. URL: <http://www.mathsoft.com/mathcad/vissim/microvissim.asp>; Visual Solutions VisSim - Modeling and Simulation Software for Mathcad.
9. Gerber, E.L. (1998) "Maple for Circuits and Systems," Proceedings of the 1998 ASEE Annual Conference and Exhibition, Session 2266, Seattle, WA, June. URL: <http://www.asee.org/conferences/98conf/00369.pdf>.
10. URL: <http://www.adeptsience.com/products/mathsim/maple5/apps/#control>; Adept Scientific Maple Application Center.
11. URL: <http://jhd486.mast.queensu.ca/~jon/>; Homepage for Jon H. Davis.

Joey K. Parker

Joey K. Parker is currently an Associate Professor of Mechanical Engineering at The University of Alabama, where his teaching responsibilities include control systems, instrumentation, and both freshmen and senior capstone design. He has been involved with the Foundation Coalition effort at Alabama since 1993, and recently served as the freshman TIDE (Teaming, Integration, and Design in Engineering) program coordinator. He received his B.S.M.E. degree from Tennessee Technological University in 1978, and his Master's and Ph.D. in Mechanical Engineering from Clemson University in 1981 and 1985, respectively. His research interests include electro-mechanical actuators, microcomputer applications, and industrial automation.