# Math Machines Readme (2014)

## Learning with Math Machines, Inc.

For HD video introductions to Math Machines, link to www.mathmachines.net/Video

# INTRODUCTION

“Math Machines” are simple devices which give an immediate, physical and dynamic expression to “abstract” mathematical equations. Users can enter a virtually unlimited variety of free-form mathematical functions which control motions and colors. Math Machines do not replace paper-and-pencil activities, but instead provide a physical application and immediate, dynamic feedback for those activities. Learners design, test, compare and modify mathematical functions as dynamic relationships. The mathematical level can range from simple proportions and linear functions, through polynomials, exponential and logarithmic functions and trigonometry and to the introduction of differential calculus. Functions can be expressed either explicitly or in recursive form.

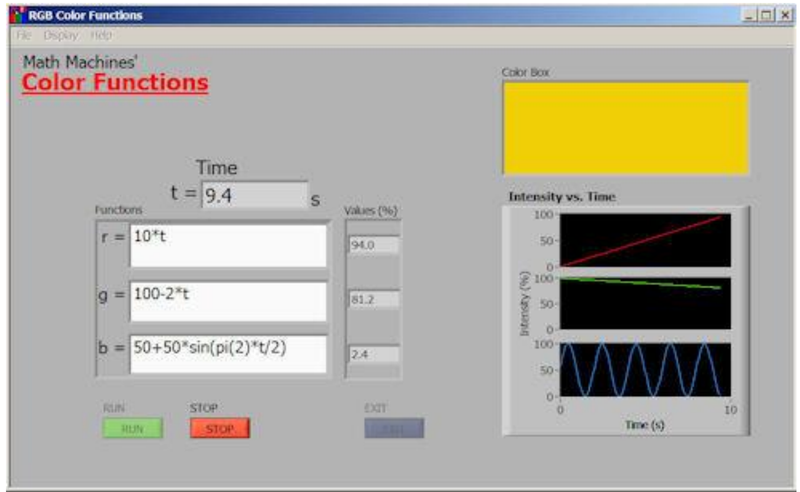
This package installs all drivers required for both the software listed below and the associated hardware. The software can be used without external hardware (this is particularly valuable for instructors or students developing or testing functions at home), but full functionality requires either a SensorDAQ interface from Vernier Software And Technology or a myDAQ from National Instruments, plus the Math Machines “RGB Color Mixer,” the “Function Plane” or the “LACI Smart Cart.” The “Color Functions” program is most appropriate for users who are not using external hardware, since the computer screen itself can provide the color output.

COMPUTER REQUIREMENTS: Windows computer with XP SP3, Vista, Windows 7 or higher. A separate version of Color Functions is available for Macs

INDEX:

# SOFTWARE

## Color Functions




The Color Functions program allows users to enter 3 simultaneous functions, one for each of the 3 primary colors (red, green and blue) used by computer monitors, TVs and other equipment to produce a virtually unlimited range of colors, shades and tints. The independent variable for each of the functions is normally "t", the elapsed time in seconds. The output of each function is the percentage (from 0 to 100) of its maximum possible intensity. Outputs are displayed onscreen as numeric values, graphs and a color box. If the RGB Color Mixer is attached, the color output is also displayed in a ping pong ball “pixel.”


Optionally, one or two constants, c1 and c2, can be adjusted through on-screen sliders. For recursive functions, r, g and b can also be used as variables, always representing the value calculated during the previous calculation. Allowed constants also include r0, g0 and b0 (initial values for each color component) and dt (the calculation interval which is normally 0.05 seconds).

A sample task for learners might be, “Make the red increase at a steady rate from 0 to 100 in 20 seconds while the green decreases from 100 to 0 at the same time.”

Solutions include:

1. “r = 5*t”, and
   “g = 100 – 5*t”
2. “r = 5*t”, and
   “g = 100-r”
3. “r = r+0.5” (with r0 = 0), and
   “g = g-0.5” (with g0 = 100)

A sample task involving polynomials and all three RGB colors might be, “Make the color change from blue to yellow and back to blue in 10 seconds.”

Solutions include:

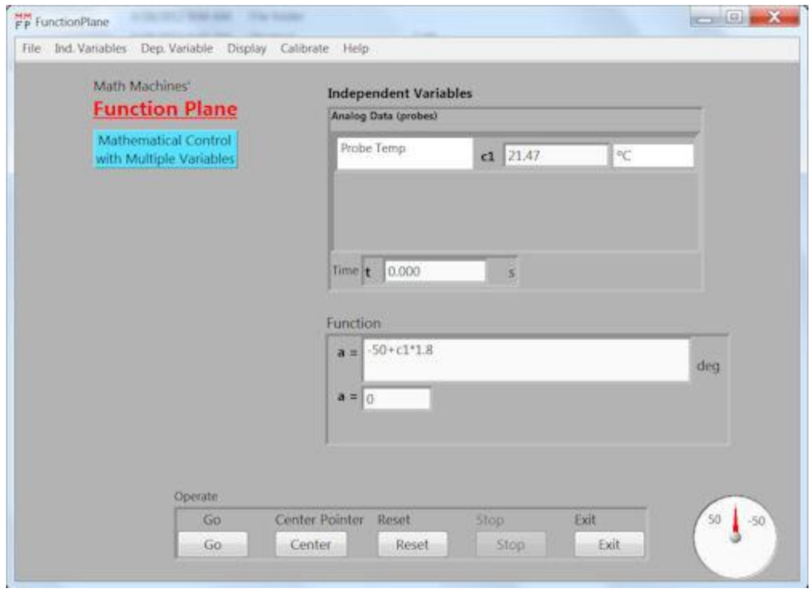
“r = -4*t^2+40*t”,
“g = r”, and
“b = 100-r”

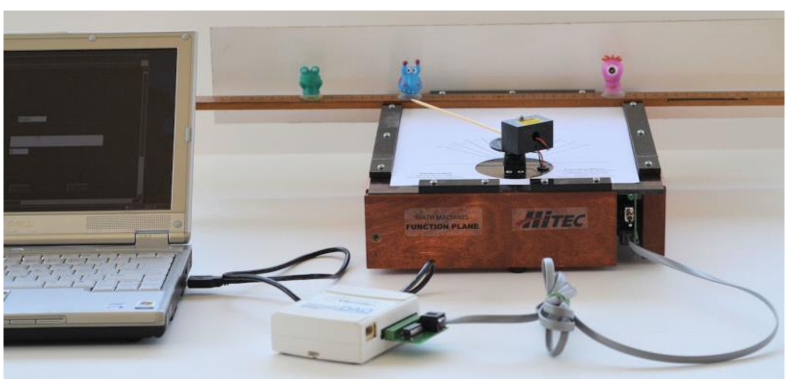
Depending on the learners' ability and the educational goals, teachers might give students a solution similar to the one above so students can observe the color change it produces and then ask learners to reproduce the same outcome entering both g and b as explicit functions of time.

For more details, see the video "Color Functions" at [www.mathmachines.net/Video](http://www.mathmachines.net/Video).

## Function Plane

The Function Plane program allows users to enter a function which controls EITHER the angular position of a servo pointer OR the linear position of a laser dot on a screen or classroom white board. Optionally, 2 inequalities can be used to control the on-off condition of the laser and a "Stop If" criterion for terminating the run. Independent variables can include elapsed time, up to 3 analog probes, a motion detector, or up to 3 on-screen sliders.




Although designed for use with the Math Machines Function Plane hardware, the program can also be used alone with an on-screen dial providing the output.

A sample task for learners might be, "Make the laser dot move at constant speed from the frog to the pink target in 20 seconds."

Assuming learners (or the teacher) have selected millimeters as the units, that the coordinate system follows the markings on a meter stick, and that the objects are located at 300 and 600 mm, solutions could include:

1. "x = 300 + 15*t, or
2. "x = x + 0.75" (with x0 = 300 and dt = 0.05 s)

A sample task involving polynomials or accelerated motion might be, “Make the dot move from the frog to the pink target in 10 seconds and then return to the blue target in one continuous motion.”

Solutions could include:

“x = 300 + 60*t – 3*t^2

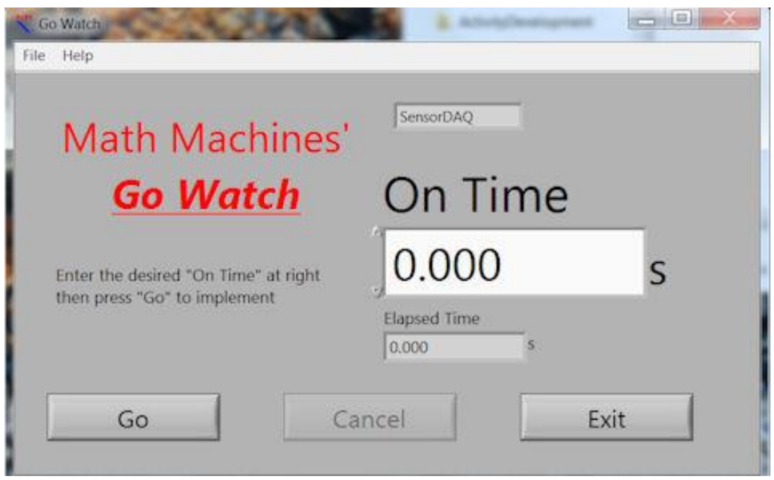
Learners of differential calculus could find more advanced solutions, such as

“x = x + (60 – 6*t)* dt (with x0 = 300)

Depending on the learners’ ability and the educational goals, teachers might give students a solution like that above so students can observe the color change it produces and then ask learners to reproduce the same outcome entering both g and b as explicit functions of time.

For more details, see the video “Controlling a Laser” at www.mathmachines.net/Video.

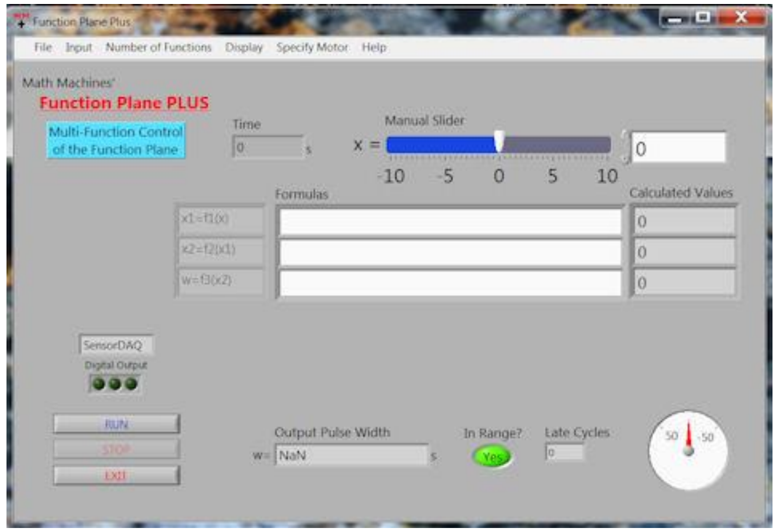
## Go Watch




The Go Watch program is simply an inverted “Stop Watch.” It is used to activate a DC device (such as a Pasco Motorized Cart) via the relay on the RGB Color Mixer.

In a sample activity, students activate a cart for 1.000 seconds and measure how far it travels, then predict how far it will travel if activated for 2 seconds. A typical motion includes initial acceleration, motion at constant speed, and a deceleration after the motor stops. Students can produce graphs of distance versus time and learn to find the time which will make the cart travel any desired distance.

## Function Plane Plus



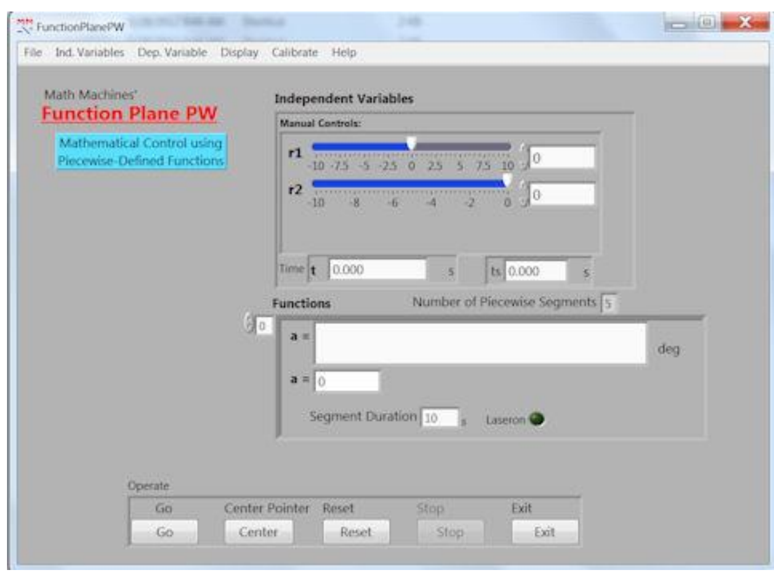

"Function Plane Plus" is a more advanced version of the basic "Function Plane" program. It requires learners to input a sequence of functions, each using the output of the previous function as its input.

It is particularly useful for activities such as ""Earthquake Simulation" which require several steps. Learners can then combine the steps to produce a more efficient calculation. Learners with the ability to write programs in LabVIEW can create their own "sub-VI" to replace the algebraic functions.

Inputs can be either time, an on-screen slider, or an analog probe. The final output must always be "pulse width"—the signal which actually controls the Function Plane's servo motor.

For more details, see the video "Earthquake Simulation?" at www.mathmachines.net/Video.

## Function Plane PW (Piecewise)



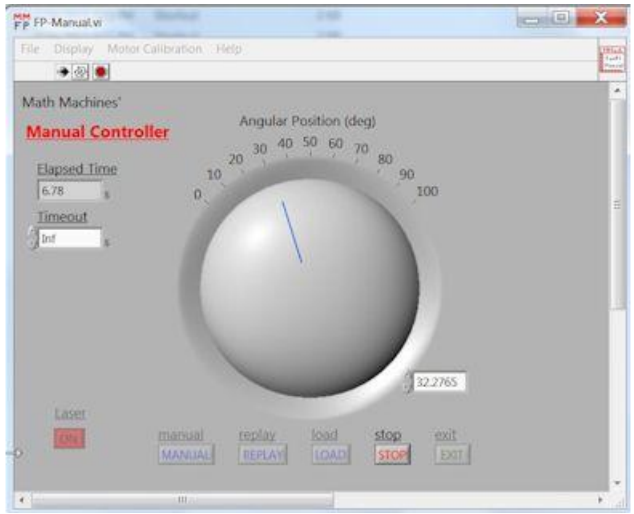

"Function Plane PW" is a specialized version of the "Function Plane" program, allowing users to x= f(t) or a = f(t) in piecewise form.

The program is particularly useful in activities such as "Graph Match," in which learners reproduce non-uniform motions, such as a constant velocity followed by a constant acceleration.

The input is normally time. Optionally, the program also allows inputs from up to 3 on-screen sliders. There can be up to 5 segments, each with arbitrary time duration.

For more details, see the video "Pig or Bacon?" at www.mathmachines.net/Video.

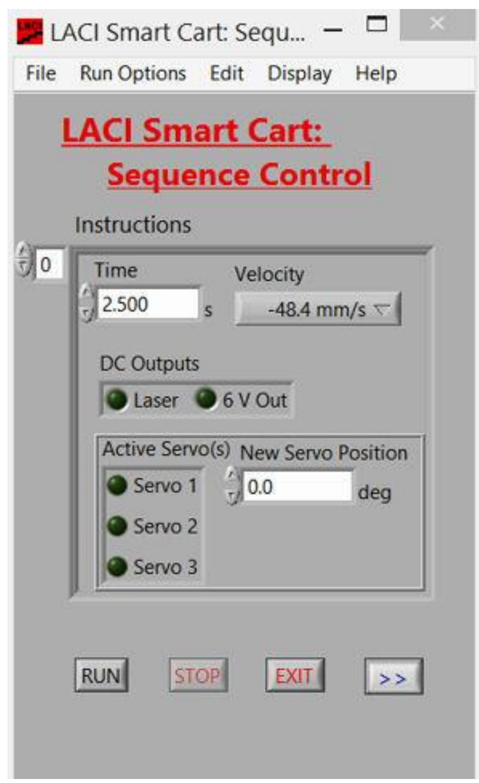
## Function Plane Manual




"Function Plane Manual" control program is designed to demonstrate both the physical capabilities of the Function Plane and the basic concepts of manual versus automated (CNC) control. The position of the Function Plane's motor is controlled by moving an on-screen dial and a button can be used to turn the laser on and off. Manual sequences can be stored as tab-delimited text files which can be read or modified using Excel. The program can also read text files which specify a sequence of desired angular positions.

## LACI Sequence

"LACI Sequence" program is designed to provide basic control of the LACI Smart Cart as it performs a sequence of actions. The program uses numerical instructions, similar to those used in CNC (Computer Numerical Control) machining or robotics.

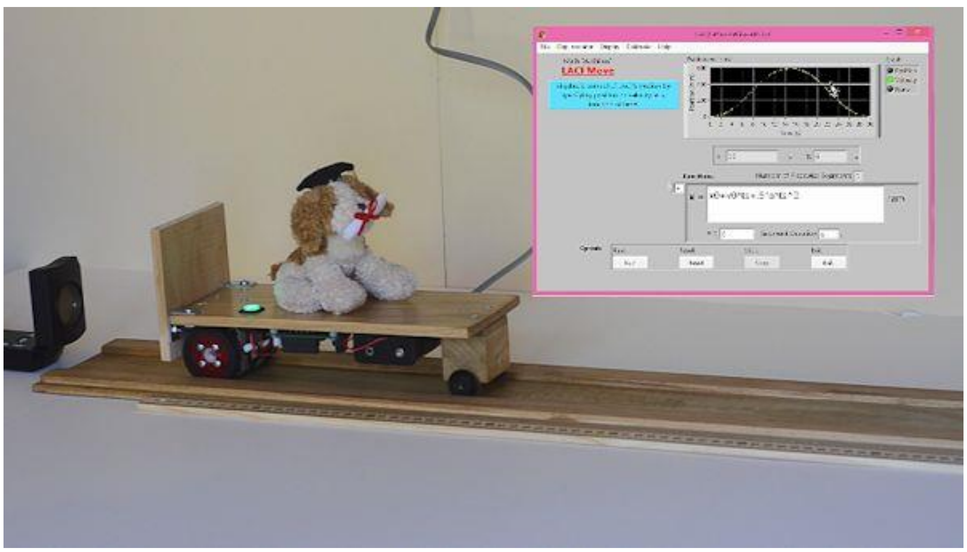


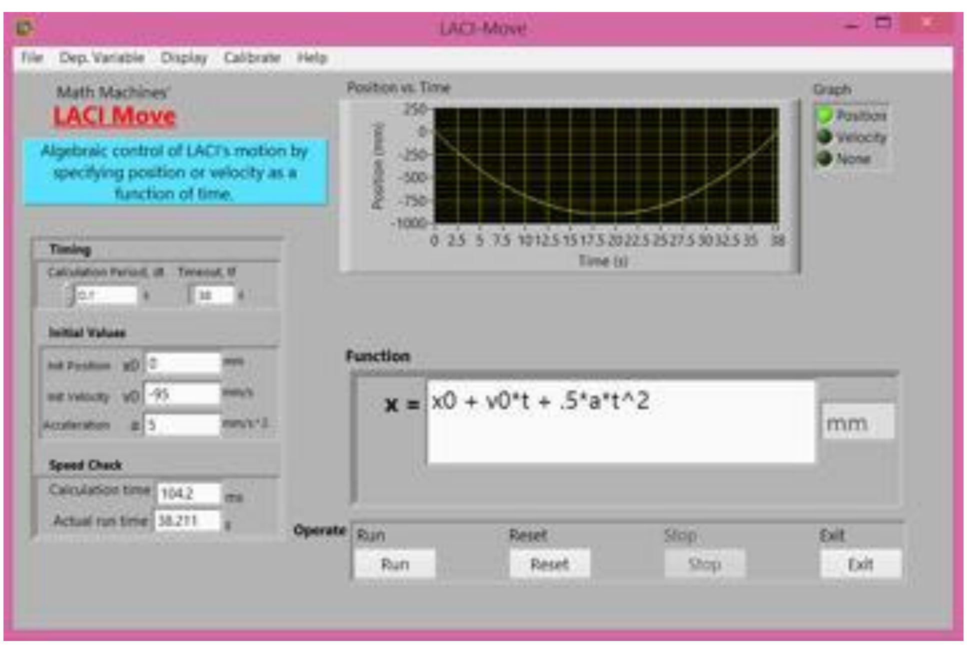
## LACI Move

The “LACI Move” program allows teachers and students to control LACI’s motion using a nearly unlimited variety of algebraic functions. The functions can specify either LACI’s position as a function of time (e.g., “x = 80*t” or “x = x0 + v0*t + .5*a*t^2”) or LACI’s velocity as a function of time (e.g., “v = 120-7*t” or “v = 80*sin(pi(2)*t/10)”).

In addition, the software allows recursive functions (e.g., “x = x +2”), piecewise-defined functions (e.g., one formula for the first 10 seconds and another the next 20 seconds) and a variety of other techniques relevant to all levels of mathematics from pre-algebra through introductory calculus.

The program is especially appropriate as a follow-up to traditional Graph Match activities in which students move in front of a motion detector to reproduce graphs of positon or velocity versus time. After gaining a conceptual understanding of these graphs, students can go on to program LACI (using Excel-like functions) to make LACI move in the way described by the graph.





# FUNCTIONS FOR CONTROLLING MATH MACHINES

The functions listed below are available for use in the LabVIEW Math Script windows used by Math Machines. Other functions may also be available.

## MOST COMMON FUNCTIONS AND OPERATORS

| | |
|---|---|
| +, -, *, / | Add, subtract, multiply or divide. Note that **multiplication must be explicit**. For example “2*t” is valid, but “2t” is not. |
| ^ | Power. For example, “5^3” is five cubed. |
| abs | Absolute value. For example “abs(-7) = 7 |
| pi(1), pi(2), pi(1/2), etc. | Note that “pi” is only allowed when it is followed by a multiplier as shown. |

## TRIGONOMETRIC FUNCTIONS

| | |
|---|---|
| sin, cos, tan, etc. | Trig functions using radians. |
| asin, acos, etc. | Inverse trig functions using radians |
| sinh, acosh, etc. | Hyperbolic trig functions |

## OTHER FUNCTIONS

| | |
|---|---|
| sign | Returns -1, 0 or 1. For example, “sign(-8.4)” = -1. |
| rand | Returns a random number between 0 and 1. Requires a “seed” such as rand(5) or rand(t). |
| sqrt | Square root function. For example “sqrt(4)” = 2 |
| log | Base 10 logarithm |
| log2 | Base 2 logarithm |
| ln | Natural logarithm |
| exp | Exponential function. For example, “exp(3)” = e^3 where e = 2.71828…. (Inverse of ln) |
| ceil | Rounds towards + infinity. For example, “ceil(2.3)” = 3 |
| floor | Rounds towards – infinity. For example, “int(4.6)’ = 4 and “int(-5.2)’ = -6. |
| int | Same as “floor” |
| step | For example, “step(t-5)” has the value 0 when t<5 and 1 when t>5. |
| spike | For example, “spike(t-5)” has the value 1 when 5<t<6 and zero at all other times. |
| square | For example, “square(t)” alternates each second between 0 and 1 |

# ALLOWED VARIABLES AND CONSTANTS IN MATH MACHINE PROGRAMS

## GoWatch, Function Plane Manual and Function Plane Circular

No algebraic variables or constants.

## Color Functions

t The independent variable is normally "t", the elapsed time since the run began.

r, g and b in addition to being the dependent (output) variables, the intensity of each color can be used as an independent variable. For example, the intensity of g might be calculated as "r/2" to make green half as intense as red. These can also be used in recursive calculations, such as r = r + 1 to make the intensity of red increase by 1 percentage point during each calculation cycle.

r0, g0 and b0 can be used as initial value constants, particularly in recursive calculations. For example, one method to decrease the intensity of blue light is to set b0 = 100, and use the formula, "b = b -1", which will begin at 100% intensity and decrease to zero after 100 calculation cycles.

c, c1 and c2 are constants which can be displayed and set using on-screen sliders. They are particularly useful in investigations of the impact of constants (such as slope and intercept) in the behavior of functions.

## Function Plane

Depending upon the input mode selected, the independent variables can be:

t Elapsed time since the run began.

r1. r2 and r3 values set using manual, on-screen slider

c1. c2 and c3 values read from analog sensors in the SensorDAQs channels 1, 2 and 3

m Distance determined by a motion detector attached to the SensorDAQ

a or x These are normally dependent variables, but they can also be used as independent variables in recursive calculation.

dt, dr1, dc1, etc. Adding "d" before any of the variables above gives the change in that value since the previous calculation

a0 or x0 These represent the initial value for either angle, a, or position, x and can be used as constants in calculations. They are especially important for recursive calculations.

tf This is the constant set as "time out," or the time when the run will stop automatically. Note that tf can be set to "Inf" to make the program run continuously until stopped by the stop button.

## Function Plane Plus

Only one independent variable is allowed in each sequential formula.

x The selected input value, whether this is the manual slider, elapsed time, or data from a sensor is the independent variable for the first formula, x1 = f(x).

x1 The result of the calculation specified by the first formula can be used in the second formula, x2 = f(x1)

x2 The result of the calculation specified by the second formula can be used in the third formula. X3 = f(x2), etc.

## Function Plane Piecewise

Depending upon the input mode selected, the independent variables can be:

t Elapsed time since the run began.

ts Elapsed time since the beginning of the current time segment.

r1. r2 and r3 values set using manual, on-screen slider

a or x These are normally dependent variables, but they can also be used as independent variables in recursive calculation.

dt, dr1, da, etc. Adding “d” before any of the variables above gives the change in that value since the previous calculation

a0 or x0 These represent the initial value for either angle, a, or position, x and can be used as constants in calculations. They are especially important for recursive calculations.

## LACI Move

t Elapsed time since the run began.

ts Elapsed time since the beginning of the current time segment.

tf Time at which the motion is scheduled to end.

x or v These are normally dependent variables, but they can also be used as independent variables in recursive calculation.

dt, dv, dx, etc. Adding “d” gives the change in t, v or x since the previous calculation.

x0 or v0 These represent the initial value for position and velocity and can be used as constants in calculations. They are especially important for recursive calculations. In piecewise-defined function, these values “reset” to the final value of each segment for use in the next segment.

a Although intended to represent acceleration, “a” can also be used as any general constant.

# UNITS ALLOWED IN FUNCTION PLANE SOFTWARE

## Time Units

| Symbol | Meaning |
|---|---|
| s | second |
| min | minute |
| h | hour |
| d | day |
| a | year |

## Linear Units (partial list)

| Symbol | Meaning |
|---|---|
| m | meter |
| km | kilometer |
| cm | centimeter |
| mm | millimeter |
| um | micrometer |
| ft | feet |
| in | inch |
| yd | yard |
| mi | mile |
| ly | lightyear |
| Angstrom | Angstrom |
| AU | astronomical unit |
| rod | rod |
| chain | chain |
| cubit | cubit |
| fathom | fathom |
| furlong | furlong |

## Angular Measures

| Symbol | Meaning |
|---|---|
| deg | degrees |
| rad | radians |
| ‘ | arc minutes |
| “ | arc seconds |

# ADDITIONAL INFORMATION



Learning with Math Machines, Inc. is a non-profit, 501(c)(3) organization. If you find this software useful, please consider contributing $25 (or whatever you can afford) through our website, www.mathmachines.net.

This material is based in part upon work supported by the National Science Foundation' Advanced Technological Education Program under Grants No. DUE-0202202 and DUE-1003381. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Please email fred.thomas@mathmachines.net with feedback, to learn about opportunities for collaboration or to inquire about the possibility of a Math Machines workshop in your area.