SANDIA REPORT
SAND2014-1343
Unlimited Release
February 2014

Converting DYNAMO simulations to Powersim Studio simulations

La Tonya Nicole Walker, Leonard Anthony Malczynski

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd.
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2014-1343
Unlimited Release
Printed February 2014

Converting DYNAMO simulations to Powersim Studio simulations

La Tonya Walker and Leonard Malczynski
Earth Systems and Systems & Supply Chain Surety
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1137

Abstract

DYNAMO is a computer program for building and running 'continuous' simulation models. It was developed by the Industrial Dynamics Group at the Massachusetts Institute of Technology for simulating dynamic feedback models of business, economic, and social systems. The history of the system dynamics method since 1957 includes many classic models built in DYANMO. It was not until the late 1980s that software was built to take advantage of the rise of personal computers and graphical user interfaces that DYNAMO was supplanted[1].

There is much learning and insight to be gained from examining the DYANMO models and their accompanying research papers. We believe that it is a worthwhile exercise to convert DYNAMO models to more recent software packages. We have made an attempt to make it easier to turn these models into a more current system dynamics software language, Powersim © Studio produced by Powersim AS[2] of Bergen, Norway. This guide shows how to convert DYNAMO syntax into Studio syntax.

[1] Historically commercial versions were released as follows: Stella - 1985, Vensim - 1991 and SimTek – 1988 (Powersim Studio's original product name).

[2] Please see www.powersim.com

CONTENTS

1. Introduction 7

2. DYNAMO to Powersim Studio conversions 9
    2.1. Basic code 9
        COS example 9
        Division example 9
        Exponentiation example 9
        LOGN example 10
        Multiplication example 10
        SIN example 10
        SQRT example 10
    2.2. Conditional logic 11
        CLIP example 11
        MAX example 11
        MIN example 11
    2.3. Lookup functions 12
        TABHL example 12
        TABLE example 12
        TABPL example 13
        TABXT example 13
    2.4. Model control 14
        SWITCH example 14
    2.5. Delays 14
        DELAY1 example 14
        DELAY3 example 14
        DELAYP example 15
        DLINF3 example 15
        SMOOTH example 15
    2.6. Statistical functions 16
        NOISE example 16
        NORMRN example 16
        SAMPLE example 16
    2.7. Test functions 17
        PULSE example 17
        RAMP example 17
        STEP example 17

3. References 19

Distribution 21

1. INTRODUCTION

"DYNAMO is a computer program for building and running 'continuous' simulation models (models which can be described by a set of differential equations). It was developed by the Industrial Dynamics Group at the Massachusetts Institute of Technology for simulating dynamic feedback models of business, economic, and social systems, but nothing in its design precludes its use for any continuous system."[6]

The history of the system dynamics method since 1957 includes many classic models built in DYANMO. It was not until the late 1980s that software was built to take advantage of the rise of personal computers that DYNAMO was supplanted.[3] In fact there existed a personal computer version of DYNAMO in that era.

There is much learning and insight to be gained from examining the DYANMO models and their accompanying research papers. We believe that it is a worthwhile exercise to convert DYNAMO models to more recent software packages. Classic system dynamics models such as WORLD3 and Urban Dynamics, among many others, have been converted to Studio by other organizations. This is our attempt to make it easier to translate models like those into a more current system dynamics software language, Powersim Studio. Studio is produced by Powersim AS of Bergen, Norway.

The Studio examples use the format specified in Best Practices for System Dynamics Model Design and Construction with Powersim Studio, SAND2011-4108 [4]. DYNAMO required short variable names of six characters or less. In these examples we have kept the original DYNAMO variable name and followed it with an expansion of the name for the Studio conversion since Studio has no practical limit on variable name length.

The pieces of code below are drawn from some of the references cited at the end of this text.

[3] Historically commercial versions were released as follows: Stella - 1985, Vensim - 1991 and SimTek – 1988 (Powersim Studio's original product name).

2. DYNAMO TO POWERSIM STUDIO CONVERSIONS

2.1. Basic code

COS example

Reference: based on Introduction to System Dynamics Modeling with DYNAMO[4]

DYNAMO
TEST.K = AMP * COS(6.283 * TIME.K/PER)

Studio translation
'AMP amplitude of oscillation' * COS(6.283 * TIME / 'PER period of oscillation')

Division example

Model reference: Simulating Violators

DYNAMO
RISK.K = PUNISH.K / TOTVIO.K

Studio translation
'PUNISH percent of violators punished' / 'TOTVIO total percent of violators'

Exponentiation example

Model reference: The Discovery Life Cycle of a Finite Resource

DYNAMO
URP1.K = URI * EXP(GC1 * (TIME.K -1900))

Studio translation
'URI USAGE RATE INITIAL' * EXP('GC1 GROWTH CONSTANT 1' *
NUMBER(TIME - STARTTIME))
STARTTIME = 1900

Note: In some versions of DYNAMO, the notation of B**A would represent B raised to the A power, where both A and B are variables. Also in DYNAMO, to compute powers of numbers other than with *e* base use notation of EXP(B * LOGN(A)). The Studio translation is B^A.[5]

---

[4] Base on means a modification of the original code to provide an unreferenced example.

[5] Note that Studio has a set of functions similar to these that apply to arrays, the exponentiation example is B#^A.

LOGN example

Model reference: A System Dynamics Model of National Energy Usage

DYNAMO
GR.K = EXP((TIME.K - 1950) * LOGN(1.05))

Studio translation
EXP((YEAR(TIME) - 1950) * LN(1.05))

Multiplication example

Model reference: The Discovery Life Cycle of a Finite Resource

DYNAMO
PU.K = (STC.K)(PM.K)

Studio translation
'STC smoothed total cost' * 'PM price multiplier'

SIN example

Reference: Introduction to System Dynamics Modeling with DYNAMO

DYNAMO
TEST.K = AMP * SIN(6.283 * TIME.K/PER)

Studio translation
'AMP amplitude of oscillation' * SIN(6.283 * TIME / 'PER period of oscillation')

SQRT example

Reference: Introduction to System Dynamics Modeling with DYNAMO

DYNAMO
TEST.K = SQRT(A)

Studio translation
SQRT(A)

2.2. Conditional logic

CLIP example

Model reference: Simulating Violators

DYNAMO

```
FORCON.K = CLIP(FGROW.K, FDROP.K, EXPNCH.K, 1)
```

Studio translation

```
IF('EXPNCH expected punishment change' >= 1,
   'FGROW formal control growth',
   'FDROP formal control drop due to reduced punishment'
   )
```

MAX example

Model reference: Simulating Violators

DYNAMO

```
TRYEVA.KL = (1/DELCTE) * MAX(DIFFER.K, 0) * AFRACE.K
```

Studio translation

```
(1 / 'DELCTE average delay time for complier to become an evader') *
MAX('DIFFER difference between expect and actual violators', 0<<%>>) *
'AFRACE actual fraction of violators moving from complier to evaders'
```

MIN example

Model reference: Simulating Violators

DYNAMO

```
PERFCT.K = MIN(TOTVIO.K, ENFORC.K)
```

Studio translation

```
MIN('TOTVIO total percent of violators',
      'ENFORC enforcement percent of population supervised by formal control
       each year')
```

2.3. Lookup functions

TABHL example

Model reference: Simulating Violators

DYNAMO

RFVOI.K=TARHL(TRFVOI,VIOFAC.K,0,10,1)
TRFVOI=11.95/.85/.70/.S0/.30/.15/.07/.03/.02/.01

Studio translation

RFVOI Reduction of Informal Control by Total Violators

GRAPH('VIOFAC ratio of total number of violators to initial number of violators', 0, 1, 'TRFVOI table effect of violators on informal control')

Note: The TABHL function computes the respective dependent value for the independent variable within the specified range and it is linearly interpolated. When an independent variable value is outside that specified range, the function uses the dependent values' respective horizontal asymptotes.

TABLE example

Model reference: Simulating Violators

DYNAMO

RFPOA.K=TABLE(TRFPOA, PERMIS.K, 0, 100, 10)
TRFPOA=1/.95/.85/.65/.55/.5/.5/.5/.5/.5/.5

Studio translation

GRAPH('PERMIS permissiveness', 0, 10, 'TRFPOA table for effect of permissiveness on autonomous control')

TRFPOA table for effect of permissiveness on autonomous control =
{1, 0.95, 0.85, 0.65, 0.55, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5}

Note: There is no exact conversion from DYNAMO to Studio for a TABLE function. In DYNAMO, the compiler would inform the user that the value for input variable was outside the specified range, but DYNAMO would still compute the respective dependent value using horizontal asymptotes. Within the specified independent range, the dependent value is linearly interpolated.

TABPL example

Model reference: based on Simulating Violators

DYNAMO
RFPOA.K=TABPL(TRFPOA, PERMIS.K, 0, 100, 10)
TRFPOA=1/.95/.85/.65/.55/.5/.5/.5/.5/.5/.5/0/0/0/0/0/0/0/0/0/0/0

Studio translation
GRAPHCURVE('PERMIS permissiveness', 0, 10, 'TRFPOA table for effect of permissiveness on autonomous control')

TRFPOA table for effect of permissiveness on autonomous control = {1, 0.95, 0.85, 0.65, 0.55, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5}

Note: Use TABPL to avoid discontinuities between points, it uses a third-order polynomial, a cubic spline, through the given points. With the TABPL, there is no exact conversion since 1) independent values outside the specified range uses horizontal asymptote; 2) an error message is provided when the range is exceeded; and 3) table has dependent values and equal number of zeroes following it that is replaced by the coefficients of the polynomial. Therefore the DYNAMO extra zeroes have to be replaced by the user. In Studio, the GRAPHCURVE function is similar, but it does inform the user of coefficients used in the table and therefore extra zeroes should be removed.

TABXT example

Model reference: based on Simulating Violators

DYNAMO
RFPOA.K=TABXT(TRFPOA, PERMIS.K, 0, 100, 10)
TRFPOA=1/.95/.85/.65/.55/.5/.5/.5/.5/.5/.5

Studio translation
GRAPHLINAS('PERMIS permissiveness', 0, 10, 'TRFPOA table for effect of permissiveness on autonomous control')

TRFPOA table for effect of permissiveness on autonomous control = {1, 0.95, 0.85, 0.65, 0.55, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5}

Note: With the TABXT, the table function interpolates linearly within the specified range. If the input value is outside the specified range, then the dependent value is determined using linear asymptotes.

2.4. Model control

SWITCH example

Model reference: The Discovery Life Cycle of a Finite Resource

DYNAMO

```
P.K = SWITCH(PU.K, PREG.K, SW1)
SW1 = 0
```

Studio translation

```
IF('SW1 REGULATION SWITCH' = 0,
        'PU price unregulated',
        'PREG price regulated'
    )
```

2.5. Delays

DELAY1 example

Model reference: Simulating Violators

DYNAMO

```
GOPUB.KL = DELAY1(PGOPUB.K, DELETO)
```

Studio translation

```
DELAYMTR('PGOPUB predicted rate from evaders into overt violators',
'DELETO average delay time for evader to become complier', 1)
```

Note: DELAY1 is first order material delay.

DELAY3 example

Model reference: Simulating Violators

DYNAMO

```
DCHFUR.K = DELAY3(CHFORC.K, DELINE)
```

Studio translation

```
DELAYMTR('CHFORC change in force expected yearly', 'DELINE average delay
        in increasing enforcement', 3)
```

Note: DELAY3 is third order material delay.

DELAYP example

Model reference: A System Dynamics Model of National Energy Usage

DYNAMO
NUT.KL = DELAYP(CRUT.JK, TNUT, UTUC.K)
UTUC is a variable to track and graph the quantity being delayed.

Studio translation
DELAYPPL('CRUT construction rate of utilities', 'TNUT TIME DELAY FOR NEW UTILITY CONSTRUCTION')

Note: Studio allows a user to automatically track and graph delayed quantities; therefore the variable UTUC (UTUC Utility Under Construction) is not used in Studio when using the pipeline delay function.

DLINF3 example

Model reference: based on The Discovery Life Cycle of a Finite Resource

DYNAMO
AUR.K = DLINF(UR.JK, AURAD)
AUR = AURI

Studio translation
DELAYINF('UR usage rate', 'AURAD AVERAGE USAGE RATE ADJUSTMENT DELAY', 3, 'AURI AVERAGE USAGE RATE INITIAL')

Note: DLINF3 is third order information delay.

SMOOTH example

Model reference: The Discovery Life Cycle of a Finite Resource

DYNAMO
AUR.K = SMOOTH(UR.JK, AURAD)
AUR = AURI

Studio translation
DELAYINF('UR usage rate', 'AURAD AVERAGE USAGE RATE ADJUSTMENT DELAY', 1, 'AURI AVERAGE USAGE RATE INITIAL')

Note: SMOOTH function is first order information delay.

2.6. Statistical functions

NOISE example

Reference: Introduction to System Dynamics Modeling with DYNAMO

DYNAMO
TEST.K = RANGE * NOISE()

Studio translation
RANGE * RANDOM(-0.5,0.5)

NORMRN example

Model reference: The Discovery Life Cycle of a Finite Resource

DYNAMO
DN = NORMRN(1, 0.1)

Studio translation
NORMAL(1, 0.1)

SAMPLE example

Model reference: The Discovery Life Cycle of a Finite Resource

DYNAMO
DNVAL.K = SAMPLE(DN, 1, 1)

Studio translation
SAMPLE('DN random', DATE(1901), 1, 1)

2.7. Test functions

PULSE example

Reference: Introduction to System Dynamics Modeling with DYNAMO

DYNAMO
TEST.K = PULSE(HGHT, STRT, INTVL)

Studio translation
PULSE('HGHT height of the pulse' * TIMESTEP, 'STRT time of the first pulse', 'INTVL time interval between successive pulses')

Note: Pulse function in Studio assumes the volume of pulse is the input, and then calculates height from volume divided by the time step used in the run.

RAMP example

Reference: Introduction to System Dynamics Modeling with DYNAMO

DYNAMO
TEST.K = RAMP(SLP, STRT)

Studio translation
RAMP('SLP slope of linear function', 'STRT starting time for the ramp')

STEP example

Reference: Introduction to System Dynamics Modeling with DYNAMO

DYNAMO
TEST.K = STEP(HGHT, STRT)

Studio translation
STEP('HGHT height of the pulse', 'STRT time of the first pulse')

3. REFERENCES

1. Dabiri, Homayoon E., Principles of Dynamic Systems II, MIT course 15.852/15.872, *Suggested Solution to Modeling Assignment #4, Dynamics of Illegal Mexican Migration*, D-2843. MIT Alfred P. Sloan School of Management, 1977.
2. Jacobsen, Chanoch and Richard Bronson, *Simulating Violators*, Operations Research Society of America, 1985.
3. Low, Gilbert and Homayoon E. Dabiri, Principles of Dynamic Systems II, MIT course 15.852/15.872, *Modeling Assignment #4, Dynamics of Illegal Mexican Migration*, D-2824. MIT Alfred P. Sloan School of Management, 1977.
4. Malczynski, Leonard A., *Best Practices for System Dynamics Model Design and Construction with Powersim Studio*, SAND2011-4108, Sandia National Laboratories, Albuquerque, NM, June 2011.
5. Meadows, Dennis L., *Dynamics of Commodity Production Cycles*, Wright-Allen Press, 1970.
6. Naill, Roger F., *The Discovery Life Cycle of a Finite Resource: A Case Study of U.S. Natural Gas*, pp. 213-256 in Toward Global Equilibrium: Collected Papers, Dennis L. Meadows and Donella H. Meadows editors. Wright-Allen Press, 1973.
7. Powersim AS www.powersim.com accessed 8 January 2014
8. Pugh, Alexander L. III., *DYNAMO User's Manual*, Sixth Edition, The MIT Press. 1983
9. Richardson, George P. and Alexander L. Pugh, III. *Introduction to System Dynamics Modeling with DYNAMO*, Productivity Press, Portland, Oregon. 1981

DISTRIBUTION

External Distribution

1 Powersim AS, Bergen, Norway

Sandia Internal Distribution

1	MS 0735	Jesse Roach
1	MS 0735	Geoffrey Klise
1	MS 0828	Asmeret Bier
1	MS 1137	Thomas Drennen
1	MS 1137	Peter Kobos
1	MS 1137	Stephanie Kuzio
2	MS 1137	Leonard Malczynski
1	MS 1137	Howard Passell
1	MS 1137	William Peplinski
1	MS 1137	Thomas Lowry
1	MS 1137	Marissa Reno
1	MS 1137	Vince Tidwell
2	MS 1137	La Tonya Walker
1	MS 9406	Jesse Westbrook
1	MS 0899	Technical Library

Sandia National Laboratories