Multi-Objective Decision Models in System Dynamics

Dr. Raimo Keloharju
The Helsinki School of Economics

Part 1

Abstract

System dynamics models can be heuristically optimized but the process of optimization usually implies that an objective function has already been selected.

This paper demonstrates the use of a new tool, called SDRDYN, which allows a model builder to experiment with various objective functions from a time-sharing terminal. A solved application example with artificial decision parameters indicated that both a control approach and an economic approach were needed for the best objective function of those attempted.

1. Introduction

System dynamics has until recently been a simulation based approach. Nelson and Krisbergh (5) and Krallmann(4) have demonstrated, however, that system dynamics models can be heuristically optimized when a razor search algorithm is attached to the simulation language Dynamo.

The search decision rule (SDR) pattern search algorithm is older and less effective than the razor algorithm, but it is generally available and very well documented(1). In the Helsinki School of Economics a modification has now been developed that combines SDR with Dynamo (3). This new version is called SDRDYN and it has been specially designed for multi-objective decision making purposes. The flow-chart of Fig. 1 shows how SDRDYN works.

A model builder defines up to three potential objective functions and then selects one function from this set for either maximizing or minimizing. The SDR-algorithm is able to treat any equation of the Dynamo-model, either of cumulative or non-cumulative type, as an objective function. Any Dynamo-parameters can be defined as SDR-variables, which means that those parameter values are automatically changed accoridng to a criterium, defined in the selected objective function. When these 'hybrid' parameter-variables are given

their upper and lower bounds the model builder specifies certain run-details and indicates what kind of information he wishes to get. A flow chart for the operation of SDR DYNAMO is given in Fig. 1.

Notes (a) - (h) describe its operation in more detail.

(a) in the model testing phase he may need information that helps calibrate the search-routine. This information is given in the SDR-table.

(b) development histories of parameter values will be obtained if so desired.

(c) the real run-length of simulation is independent of the run-length specified in the model.

(d) the number of iterations must be specified before the run. After having obtained final results the model builder can obtain any number of new iterations and he can thus continue from where he left off.

(e) the step-size multiplier is a part of the SDR-algorithm and defines the magnitude of individual parameter changes.

(f) 'Number of iteration rows' refers to listed rows counted from the end. When '2' is given to the computer, for example, information will be produced from the two last iterations. Because of the structure of the algorithm the number of output rows is usually somewhat larger than asked for.

(g) results of the optimizing process are summarized in the 'final solution', which gives the final parameter values as well as the initial and final value of the objective function. Now the model experimenter has three choices: he can remain in the optimizing mode either by obtaining more iterations or by changing the objective function or he can move to the simulation mode of conventional Dynamo.

(h) It is possible to let Dynamo run the 'optimized' model and/or change parameters by giving the usual 'rerun'-command. An 'SDR'-command will change the control back to the SDR-mode again.

Fig. 2 demonstrates the use of SDRDYN from a technical point of view. A sample problem will then be solved to show how this new tool could be used for model building purposes.

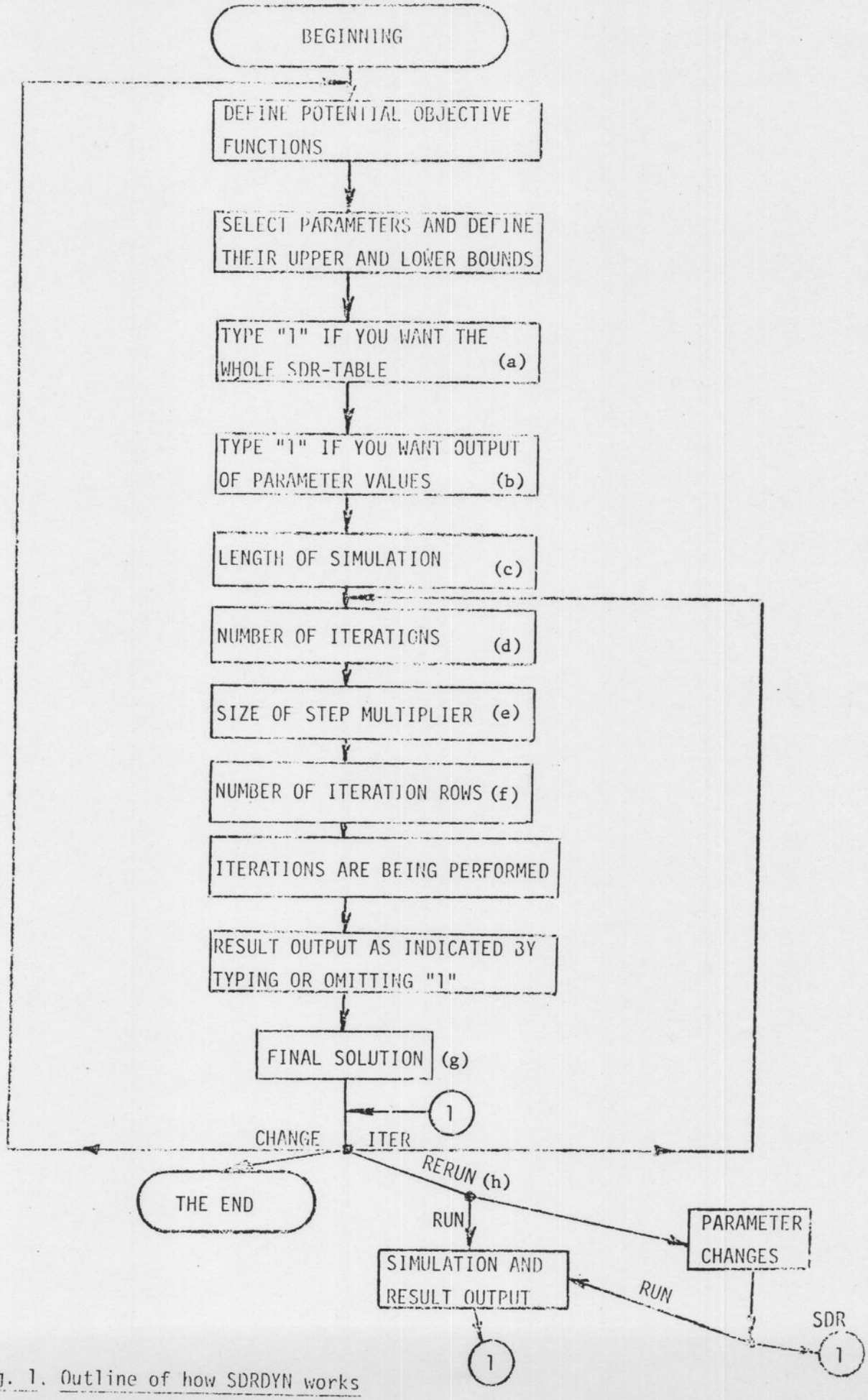

Fig. 1. Outline of how SDRDYN works

```
*** DYNAMO II F VERS. SDR  ***
RUN-TIME PACKAGE     MOD KKK SJ

   NAME OF OBJECTIVE FUNCTION?
>objf
   DO YOU WANT MAX OR MIN?
>min


   NAME OF PARAMETER  1 ?
>a1
   MIN VALUE OF A1     ?
>0
   MAX VALUE OF A1     ?
>1


   NAME OF PARAMETER  2 ?
>end

   END OF PARAMETER INPUT


   TYPE 1 IF YOU WANT THE WHOLE SDR-TABLE
>

   TYPE 1 IF OUTPUT OF PARAMETER VALUES

>
   LENGTH OF SIMULATION?
>50
   NUMBER OF ITERATIONS?
>150
   SIZE OF STEP MULTIPLIER?
>0.1
   NO OF OUTPUT ROWS?
>1


   ITER   VALUE OF OBJ FCT

    150     .696+05
    151     .696+05
    152     .696+05


   FINAL SOLUTION
   A1      =          .02

   NO OF OBJ FCT EVALUATIONS     152
   INITIAL VALUE OF OBJ FCT     .60563+09
   FINAL   VALUE OF OBJ FCT     .69641+05


>rerun
   RERUN-MODE

>run


    PAGE  1, PRODUCTION-DISTRIBUTION SYSTEM                          RUN-
   TIME       RS       RI    CRICH     OBJF      FOR       PA      LFP
              RO    CFPCO    CRICO     OBJ2      DDR      ARS     OBJP
   E+00     E+00     E+00     E+00     E+00     E+00     E+00     E+00
            E+00

*OUTPUT INTERRUPT*
@@x o
            E+00     E+00     E+00     E+00     E+00     E+00     E+00
>rerun
   RERUN-MODE

>c a1=0.8
>sdr


   BACK IN SDR-MODE
>change
   NAME OF OBJECTIVE FUNCTION?
>objp
   DO YOU WANT MAX OR MIN?
>max


   NAME OF PARAMETER  1 ?
...
```

Fig. 2. An example of SDRDYN in actual use

2. The Model

A well known inventory case study from Jarmain's "Problems in Industrial Dynamics" was selected, where a retailer had to define an ordering rule. (2) The original case developed four policy alternatives which will all be combined now into the following equation:

```
RO.KL=ARS.K+A1*(B1*RIDC+B2*RID.K-RI.K)/TAI+
      A2*(C1*DRO+C2*DDR.K)*ARS.K-FOB.K)/TAPL   ,  where
```

RO= retail orders

ARS= average retail sales

RIDC= retail inventory desired (if a constant)

RID= " " " (if a variable)

DRO= delay in receiving orders

DDR= delivery delay recognized

FOB= factory order backlog

TAI, TAPL= parameters

Retail ordering was based on known future demand. Average retail sales (ARS) was needed in the ordering rule (RO) to decouple two rate-variables from each other.

Parameters A1, A2, B1, C1 and C2 might be called decision parameters, and have been added to identify various policy alternatives. They are artificial constructs that allow the optimizing process to select both used information sources and their weights. Figure 3 below defines cases (a),...,(d), treated in Jarmain's book in terms of fixed decision parameter values:

	c a s e s			
	(a)	(b)	(c)	(d)
A1	0	1	1	1
A2	0	0	1	1
B1	1	1	0	0
B2	0	0	1	1
C1	1	1	1	0
C2	0	0	0	1

Fig. 3. Decision parameter values for some ordering policies

In the work under discussion all decision parameters were SDR-variables having upper and lower bounds of 1 and 0. Ordinary parameters were given their published values although they might also be included in the search process as long as the pre-defined limit of 20 parameters is not exceeded.

Hypothetical quadratic cost functions were added to the model in order to test the significance of cost assumptions for the study. Total cost was assume to be an additive combination of the following cost components:

(a) $300*(FP)^2$

(b) $3*(RI)^2$

(c) $0.03*(WAS*ARS-RI)^2$, where

FP= factory production
RI= retail inventory
WAS= a parameter (weeks of average sales)
ARS= average retail sales

Three independent and two derived objective functions were formulated in order to judge the usefulness of SDRDYN for multi-objective decision making:

(1) CFPCO = cumulative factory production change cost
(2) CRICH = cumulative retail inventory change cost
(3) CRICO = cumulative retail inventory cost
(4) OBJ2 = CFPCO + CRICH
(5) OBJF = CFPCO + CRICH + CRICO

Objective functions (1) and (2) try to minimize squared deviations during a solution interval. Therefore, they are based on control approach. Objective function (3) tries to minimize target inventory related costs and is thus based on an economic approach. Both OBJ2 and OBJF are derived functions as they combine objective functions (1), (2), and (3).

3. Results

The model was solved for each objective function alternative using a step input and 150 iterations of a simulation length of 50 periods. The decision parameters had initial values of case (b) from Fig. 3.

The optimization strategies discovered in terms of decision parameter values were as follows:

Used objective function	Decision parameters and their values					
	A1	A2	B1	B2	C1	C2
CFPCO	0	0.34	0.88	0	0.38	0
CRICH	0	1.0	1.0	0	1.0	0.01
CRICO	0.99	1.0	1.0	0.12	1.0	0.08
OBJ2	0.1	1.0	1.0	0.47	0.74	0
OBJF	0.05	1.0	1.0	0.5	1.0	0

Fig. 4 'Optimum' decision parameter values

Figure 4 provokes some comments:

(a) Only the objective functions OBJF and CRICO include an inventory term. For this reason other model variations have A1-0. As CRICO is only part of OBJF the weighting of inventory information in OBJF was of minor importance (A1=0.05).

(b) parameter A2 measures importance of feedback information from order backlog. Only production rate-change based rule CFPCO does not utilize inventory information directly and this can be seen from the relatively low value of A2 (=0.34).

(c) C2 was a destabilizing parameter as it brought a positive feedback loop to the model. In all examined cases this parameter had a very small value.

(d) all parameter values were obtained after a heuristic search process of a fixed length. Therefore, the final values are not to be judged as accurate.

Figure 5 summarizes total costs incurred, their ranking in magnitude (in left-hand corners), and total rankings. The costs have been given in exponential form to save space. For instance, .230+04 = $.230*10^4$ = 2300 money units (m.u.). The rows indicate the objective function that has been used for optimization purposes. The columns list all potential objective functions of the study. A small example will clarify the figure: When OBJF was given as the objective function to be minimized, the total costs incurred were 45900 m.u. This, of course,was the lowest value received for OBJF. When the model was

optimized for CRICH, OBJF had a value of 56700. This was the second lowest OBJF-value obtained and, therefore, the corresponding small box has a ranking number of '2'. Total rankings were obtained by adding rankings within each row.

Objective function used for optimization	CFPCO	CRICH	CRICO	OBJ2	OBJF	Total ranking
CFPCO	^{1}.230+04	^{5}.542+05	^{5}.591+06	^{4}.565+05	^{5}.647+06	20
CRICH	^{3}.216+05	^{1}.349+04	^{3}.316+05	^{2}.251+05	^{2}.567+05	11
CRICO	^{5}.217+06	^{3}.163+05	^{1}.274+04	^{5}.233+06	^{4}.236+06	18
OBJ2	^{2}.923+04	^{2}.813+04	^{4}.126+06	^{1}.174+05	^{3}.143+06	12
OBJF	^{4}.295+05	^{4}.454+05	^{2}.118+05	^{3}.341+05	^{1}.459+05	14

Fig. 5. Objective function costs & rankings

Total rankings indicate that objective functions CFPCO and CRICO cannot be recommended for the problem under study. Retail inventory time series in Fig. 6 confirm this preliminary finding and also provide a guide for selecting the best one from the remaining objective function candidates.

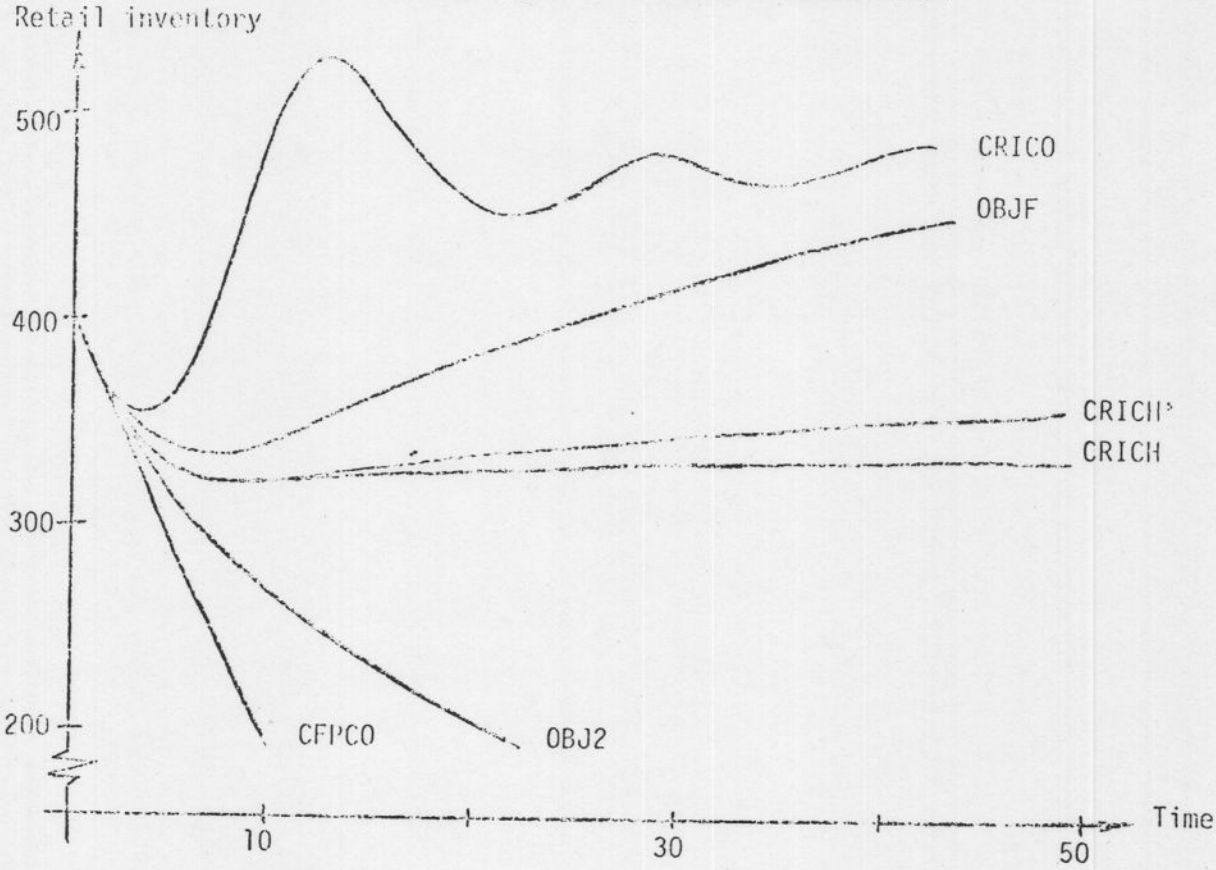

Fig. 6. Retail inventory time series for various objective function alternatives

CFPCO was the last in the total ranking and it unquestionably also led to the most unsatisfactory behaviour. CRICO was the second from the bottom of the ranking list and besides this caused self-generated fluctuations in the system. OBJ2 is unsatisfactory as it implies inventory shortages in the future. From those still left the total objective function OBJF is better than CRICH because retail inventory should respond in a proper way to a step increase in incoming orders. At least this simple example demonstrated that an optimization process should thus be based both on a control and an economic approach. The final value of OBJF (.459+05) was only 18.7% of the value of OBJF in the best simulation run (case c) when original parameter values were used. This means that the heuristic procedure used gave a solution that produced considerably smaller costs than the best of the conventional system dynamics type solutions in Jarmain's book. In this specific situation the conclusion was based on assumed hypothetical cost functions.

Let us return to Fig. 6 again. Most of the time-series proved to be unsatisfactory as they led to insufficient amounts of inventory. The control-approach based objective function CFPCO, for example, even led to negative inventory values. In a situation like this it might be suggested that a terminal inventory constraint would remove the problem symptoms. However, because CFPCO does not include an inventory term, a terminal inventory constraint will not help either. When the target inventory was given a value of 450 units in period 48, the time series of CFPCO and OBJ2 remained unchanged. The objective function CRICH was based on inventory change cost and, therefore, retail inventory reacted to some extent to the given terminal constraint. A new time-series, marked CRICH, shows how CRICH was changed. These experiments indicate clearly that a terminal constraint is not able to compensate for mistakes in objective function selection.

It is also interesting to note that the best objective function OBJF produced inventory time-series data that are even 'sounder' than those found in the reference (2). In that solution inventory approached its final value after fluctuations, but in the optimizing run, based on OBJF, it approached gradually after the initial depletion phase.

4. Discussion

The approach outlined above leads to many questions. To get this rethinking process started and to stimulate it further some differences from the old way of doing things will be selected for a closer examination.

In the model building phase, defining and selecting bybrid parameters is going to be an important step in the future. The work under discussion indicated one approach that was based on artificially created decision parameters. There might be many other ways of linking a Dynamo-model to a higher order optimizing algorithm, however. This subject is likely to be a fruitful area for future research. The optimizing process has been made completely automatic in SDRDYN and, therefore, no extra programming work in addition to the ordinary simulation model is needed.

The full power of SDRDYN shows up in a multi-objective decision-making situation, and this implies an interactive framework through a time-sharing terminal. At least in a multi-objective situation a new mode of co-operation is possible and perhaps even necessary between management scientist and manager. The manager has now to take an active part in the model exploration phase through an SDR search process because he, after all, is the real decision maker and the only one who can judge and weight variables that are not strictly comparable. The important job of model exploration requires as such no programming or programming knowledge at all. The manager needs only to be able to read Dynamo-programs at most in order to make sure that he agrees with the model.

When the decision maker has accepted the model, new policy design will be a process of co-operation: The designer adds new alternatives when necessary and the decision maker explores the model, either alone or with the designer.

Besides control engineering type test series, historical time-series from real life are useful if available. By this means model-produced cost or profit figures can be compared with historical data available and an extra source for model validation secured.

The current study leads to a tentative conclusion that the chosen objective function should include both control variables and economic variables. To choose the exact form of this function and the parameter values used is just the important job a manager is needed for.

References

(1) Buffa, E.S. and Taubert, W.H. (1972) Production-Inventory Systems, Richard D. Irwin.

(2) Jarmain, W.E. (Editor) (1964) Problems in Industrial Dynamics (pp. 119-124), The M.I.T. Press

(3) Juutilainen, S. (1976) Unpublished SDR-Dynamo Manual (in Finnish)

(4) Krallmann, H. (1976) The Intergration of alternative modelling with Dynamo, A paper given in the 5th International System Dynamics Conference, Norway.

(5) Nelson, C.W. and Krisbergh, H.M. (1974) A Search Procedure for Policy Oriented Simulations: Application to Urban Dynamics, Management Science, Vol. 20, No. 8.