THE UNIVERSITY OF WESTERN ONTARIO
DEPARTMENT OF CIVIL AND
ENVIRONMENTAL ENGINEERING

Water Resources Research Report

Generic Framework for Computation of Spatial Dynamic Resilience

By:

Roshan K. Srivastav
and
Slobodan P. Simonovic

Report No: 085
Date: January 2014

ISSN: (print) 1913-3200; (online) 1913-3219;
ISBN: (print) 978-0-7714-3067-1; (online) 978-0-7714-3068-8;

GENERIC FRAMEWORK FOR COMPUTATION OF SPATIAL DYNAMIC RESILIENCE

By

Roshan K. Srivastav

and

Slobodon P. Simonovic

Department of Civil and Environmental Engineering

Western University, Canada

January 2014

Executive Summary

Increasing number of catastrophic disaster events across the world due to natural hazards (floods, droughts, hurricanes, and tornadoes) has led to loss of thousands of human lives and shattered the global economic growth. The current scientific consensus is that this could be due to the effects of climate change. According to the Intergovernmental Panel for Climate Change fifth assessment report (IPCC, 2013) the climate characteristics and patterns are changing in time and space. Further it is reported that the increase in greenhouse gases would lead to more catastrophic events in the near future with varying degree of exposures. Therefore, there is a need to (a) increase our understanding of mechanisms causing natural disasters; (b) investigate various options for mitigation of their impacts; and (c) develop various adaptation options in order to minimize their future impacts Natural disaster management is embedded in the study of complex systems (natural and constructed) that are vulnerable to multiple failures.

Recent research focus of many disaster management studies has been related to the concept of resilience – measure that integrates vulnerability and adaptive capacity of a complex system (Park et al., 2013). Resilience of complex systems is linking their behavior to system structure through feedback (in time and space). The concept of resilience has been adopted in various fields such as ecology, economics, risk management, and others (Holling, 1973; Arthur, 1999; Folke et al. 2002; Starr et al. 2003; Fiksel 2006; Park et al, 2013). Recently this concept is been adopted to quantify the impacts of climate change on urban environments (Chang et al., 2013; Simonovic and Peck, 2013).

The work presented in this report is part of the project "Simple Proxies for Risk Analysis and Natural Hazard Estimation" supported by MITACS and Property and Casualty Insurance Compensation Corporation. This report focuses on the concept of resilience for understanding the complex behavior of a system exposed to various impacts caused by natural disasters. The proposed framework quantifies the dynamic behavior of the system which can serve for the improvement in understanding of the impacts and inform decision making processes. The objectives of the study are to (i) use the system dynamics modeling approach for integration of complex system behavior as a consequence of system structure; and (ii) develop a generic computerized framework for quantification of spatial dynamic resilience.

The remainder of the report is organized as follows. Section 1 introduces the background of the research work. In Section 2, the concept of resilience is outlined. Following this, the details of the generic framework for spatial dynamic resilience using system dynamics modeling is presented in Section 3. The illustration of the proposed framework through a hypothetical case example is presented in Section 4. Finally Section 5 outlines the summary and conclusion of the present study and the scope for future research work. The details of C# programming code for linking dynamic models, and MATLAB programming code to produce dynamic resilience map are provided in Appendix B and C, respectively. List of previous reports published in this series is provided in Appendix D.

TABLE OF CONTENTS

List of Tables

List of Figures

1.0 Introduction

Past disaster management work focused on management, planning, resource allocation, structural/non-structural issues, risk, rehabilitation, sustainability and system resilience (Simonovic, 2011). Recently, the research focus is shifted to addressing the gaps between the resource allocation, sustainability and resilience. This report presents the work in progress that is focused on the development of a generic spatial dynamic resilience using system dynamics simulation modeling to assist in natural disasters management. The research presented in the report is a collaborative effort between Western University and Property and Casualty Insurance Compensation Corporation (PACICC). The main objective of this research is to quantify an innovative concept of resilience and propose its implementation in natural disasters management as a replacement for a traditional risk management. The research program is to developing simple proxies for exposure based upon the most important factors associated with each hazard: earthquake, flood and wind. These proxies will be evaluated for various population densities, infrastructure properties and governmental policies.

Simonovic and Peck (2013) proposed a framework to quantify spatial dynamic resilience through system dynamics simulation to assess the impacts of climate change on coastal megacities. Later, Peck and Simonovic (2013) presented the design and implementation of generic system dynamics simulation model (GSDSM) for use in coastal city resilience quantification and assessment. In their work, the authors' show the implementation of the dynamic resilience (which is addressing temporal change in resilience measure) but not the spatial dynamic resilience. This report summarizes the extension of the work by Peck and Simonovic (2013) and presents a generic framework for spatial dynamic resilience. In this report we are presenting the integration of spatial and dynamic characteristics of resilience to assess the physical, economic, social, health and organizational impacts of sudden hazard event in a hypothetical case study area. The implementation of the proposed framework is not restricted to the quantification of only above mentioned impacts. It is flexible and can be use for creating dynamic resilience maps for a wide variety of impacts. Further, in this study, we integrate the system dynamics modeling and the spatial mapping. We have adopted VENSIM simulation software (Ventana Systems, 1995) for system dynamics modeling.

The system when subjected to any external shock such as flood or an earthquake, has a limited capacity to resist,   absorb and recover from this shock depending upon the the shock intensity, system vulnerability and adaptive capacity. The external shocks are usually termed as impacts. For example in case of natural hazards like flooding in the city, Simonovic and Peck (2013) has identified five major possible impact areas which include physical, economic, social, health and organizational changes of city systems. The effect of each of these impacts and their inter-dependencies are captured in a casual loop diagram shown in Figure 1. For detail description of each of these impacts the readers are directed to read Peck and Simonovic (2013).

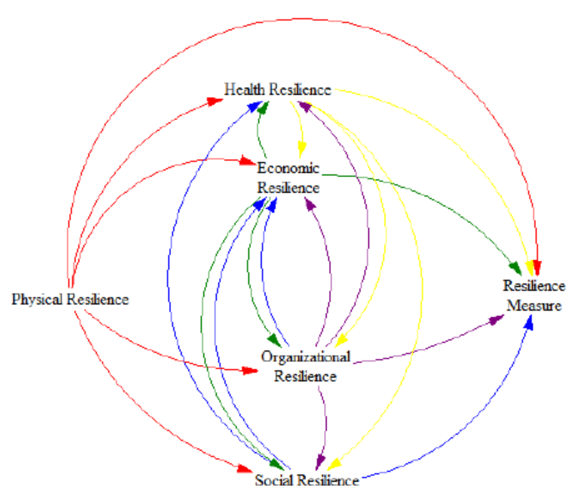

Fig. 1: Casual loop diagram of various impacts on a city
(after Peck and Simonovic, 2013)

The capacity to handle the shocks or impacts is termed as adaptive capacity of the system. The adaptive capacity (AC) of the system is an integrated behavior of the various components within the system that varies with time and space. It can be measured using four performance indicators: robustness, redundancy, resourcefulness and rapidity (four R's) first introduced by Bruneau et al. (2003, 2007). These performance indicators are defined as

(i) Robustness (R_1): ability of a system to resists external shocks without suffering any damages
(ii) Redundancy (R_2): the readiness of the system in which it resists external shocks beyond its natural capacity
(iii) Resourcefulness (R_3): the ability of the system to disseminate the resources during external shocks
(iv) Rapidity (R_4): the ability of the system to recover (in terms of time) from damages caused by external shocks

The adaptive capacity of a system is a function of both, time and space. It is mathematically expressed as a function of four R's as:

$$AC(t,s) = f\left(R_j(t,s)\right) \quad j = 1,2,3,4 \tag{1}$$

where AC is the adaptive capacity of the system; f() is the mathematical function combining the effects of four R's; *j* is the index for each of the R; *t* represents the time period; *s* represents the spatial location. In the following section we present a detail explanation on the calculation of spatial dynamic resilience using the adaptive capacity function (1).

2.0 Resilience

According to the United Nations (UNISDR, 2012) report it is expected that there will be significant increase of world's population and the majority of economic capital will the concentrated in cities which are mostly situated in disaster prone areas (Ayyub, 2013; UNISDR, 2012). It is also reported that in 2011 alone there were 29,782 deaths and a loss of USD $366 billion due to natural disasters. It is expected that these numbers will increase with future increase infrequency and magnitude of natural hazards due to climate change.

In recent years the concept of system resilience is adopted to study the impacts of natural hazards to reduce future impacts of and provide more efficient response to unseen future events. Earlier most of the disaster management systems were based on minimizing the risks of failure. Two definitions of risk can be located in the literature: (a) hazard based risk definition; and (b) impact based definition (Simonovic, 2012). According to the first definition the risk is defined as the chance of a particular event occurring. According to the second one, the risk is a combination of the chance of a particular event with the impacts that the event would cause if it occurred. The main characteristic of these definitions is a static view of disaster risk – no change in risk with time.

The concept of resilience is much broader. It describes the ability of a complex system to respond and recover from disasters and includes those conditions that allow the system to absorb impacts and cope with an event, as well as post-event, adaptive processes that facilitate the ability of the system to re-organize, change, and learn in response to a threat (Simonovic and Peck, 2013). There are many definitions of resilience, from general:

(i) The ability to recover quickly from illness, change or misfortune.

(ii) Buoyancy.

(iii) The property of material to assume its original shape after deformation.

(iv) Elasticity.

to ecology–based:

(i) The ability of a system to withstand stresses of 'environmental loading'.

to hazard–based:

(i) Capacity for collective action in response to extreme events.

(ii) The capacity of a system, community, or society potentially exposed to hazards to adapt, by resisting or changing, in order to reach and maintain an acceptable level of functioning and structure.

(iii) The capacity to absorb shocks while maintaining function.

(iv) The capacity to adapt existing resources and skills to new situations and operating conditions.

The common elements of these definitions include: (i) minimization of losses, damages and community disruption; (ii) maximization of the ability and capacity to adapt and adjust when there are shocks to systems; (iii) returning systems to a functioning state as quickly as possible; (iv) recognition that resilient systems are dynamic in time and space; and (v) acknowledgements that post-shock functioning levels may not be the same as pre-shock levels. Resilience is a dynamic process, but for measurement purposes is often viewed as static. It is important to understand the difference between concept of risk management and resilience. They both offer different perspectives for handling impacts. Table 1 gives one potential comparison between the risk and resilience (Park et al., 2013).

Table 1. Comparison of concepts of resilience and risk (Park et al., 2013)

	Risk Management	**Resilience**
Design principles	minimize risk of failures	Adaptation to changing conditions without permanent loss of function
Design objectives	Minimization of probability of failure	Minimization of consequences of failure
Design strategies	Armoring, strengthening, oversizing, resistance, redundancy, isolation	Diversity, adaptability, cohesion, flexibility, renewability, regrowth, innovation, transformation
Relation to sustainability	Security, longevity	Recovery, renewal, innovation
Mechanisms of coordinating response	Centralized, hierarchical decision structures coordinate efforts according to response plans	Decentralized, autonomous agents respond to local conditions
Modes of analysis	Quantitative (probability-based) and semi-quantitative (scenario-based) analysis	Possible consequence analysis of involving scenarios with unidentified causes

In this report we extend the concept of resilience proposed by Simonovic and Peck (2013) to capture the process of dynamic disaster resilience simulation in both, time and space. In Peck and Simonovic (2013) a theoretical concept for space-time resilience index is presented. However, the framework did not incorporate any spatial calculation aspects. In this section we present (i) mathematical framework for the concept of resilience based on Simonovic and Peck (2013) (ii) extend the computation framework to spatial dynamic resilience calculation and (iii) integration of various impacts in the computational framework.

A typical system performance under shock and recovery is presented in Fig 2. Before the shock or impact, the system performance level is denoted by $P_o(t,s)$. The system experiences a shock and change of performance between time t_o, which denotes the beginning of the disaster event, and time t1, which indicates the end of the event. The systems performance change continues beyond time t_1, till t_r the time when system fully recovers from the shock. . It is evident from Fig 2. that ending system performance level may (i) be the same as the initial level (solid line) (ii) not reach over the initial performance level (dashed line); or (iii) improve the performance beyond the initial level (dotted line).

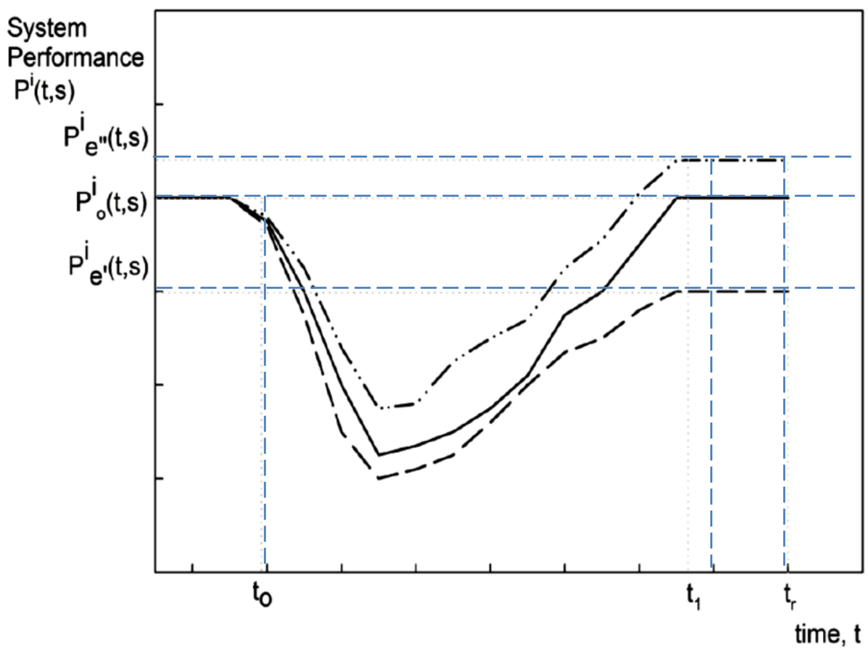

Fig. 2: Performance of a system when subject to a shock (after Simonovic and Peck, 2013)

Mathematically the system performance measure can be represented as the area under the system performance curve between the start of the impact (t_o) and the recovery time period (t_r). It can be represented as

$$\rho^i(t,s) = \int_{t_o}^{t} \left(P_o^i - P_o^i(\tau,s)\right) d\tau \quad where\, t \in [t_o, t_r] \tag{2}$$

The system resilience is calculated from the change in system performance as follows

$$r^i(t,s) = 1 - \left[\frac{\rho^i(t,s)}{P_o^i - (t - t_o)} \right] \quad (3)$$

where r^i now takes value between 0 and 1 (0 denoting no resilience and 1 denoting maximum resilience).

Figure 3 and 4 illustrate the concept of resilience calculation when a systems performance is subject to shock and recovery. As it can be observed from Fig 3, the shaded area between t_o and t_1 shows the cumulative loss/gain of system performance after the system has been subject to a shock. The time of recovery t_r indicates the moment when the resilience value reaches one.

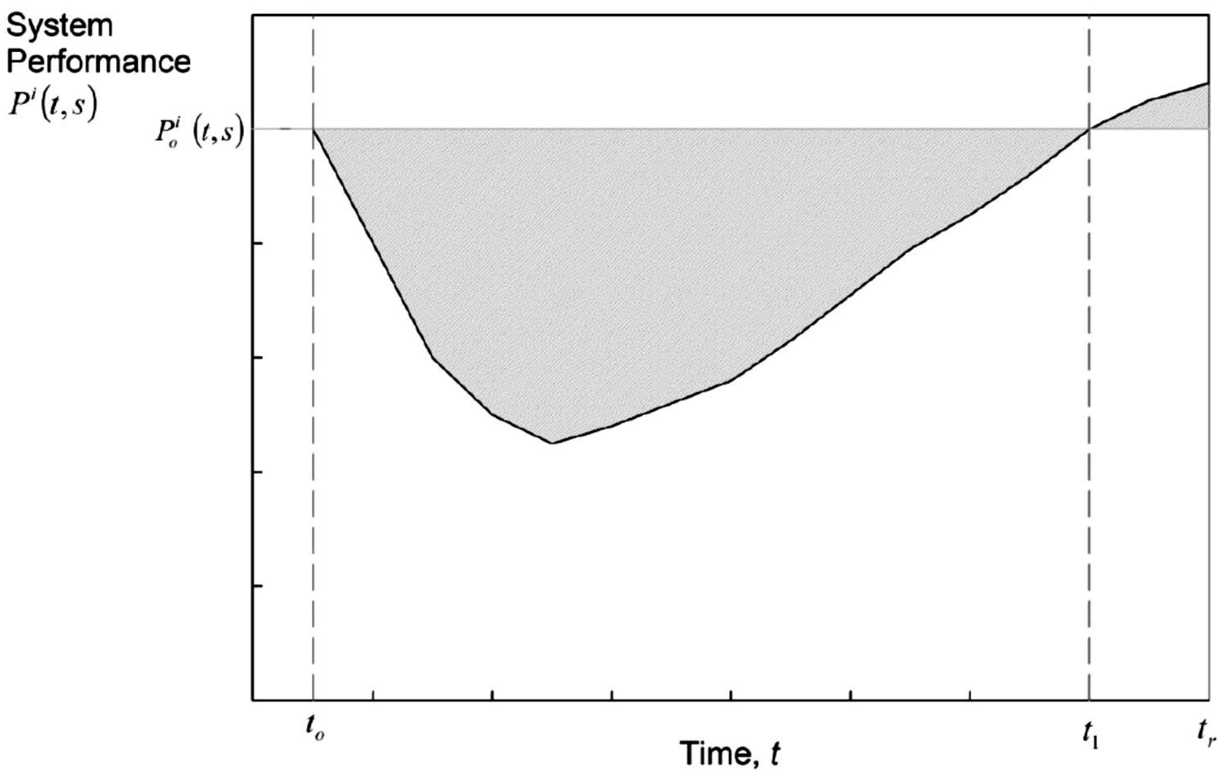

Fig. 3: Variation of system performance after a shock (after Simonovic and Peck, 2013)

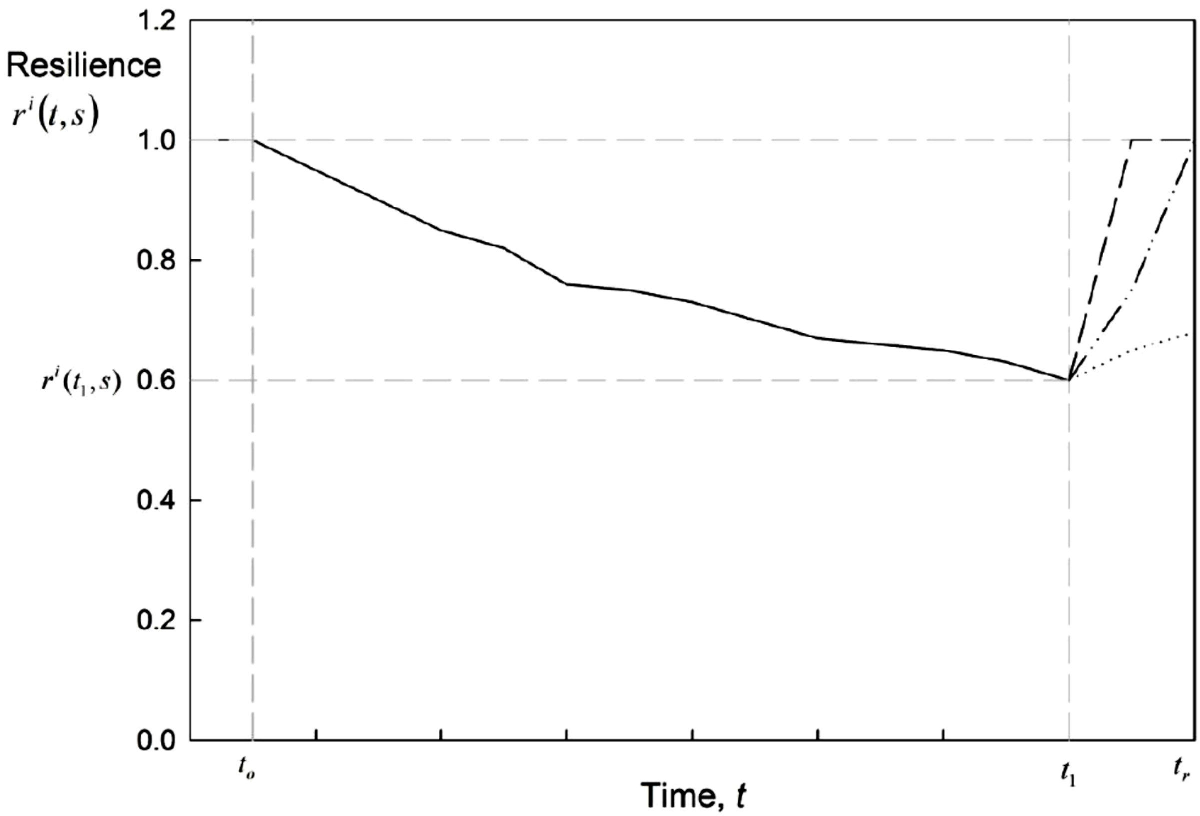

Fig. 4: Variation of system resilience after a shock

The three possible outcomes of system performance illustrated in the resilience space are shown in Fig 4. A shock to the system may create various impacts. The same computational principles can be used for each impact and the system resilience can be obtained by their integration using

$$R(t,s) = \left(\prod_{i=1}^{M} r^i (t,s) \right)^{\frac{1}{M}} \tag{4}$$

where M is the total number of impacts.

It is to be noted that the integrated system resilience will vary with both time and space. In the next section we present a detail computational framework for implementation of this spatial dynamic resilience calculation under various impacts and its use in the development of dynamic spatial maps that can assist in better understanding of systems behavior under shock.

3.0 Generic Framework for Spatial Dynamic Resilience Computation

In this section, we present a generic computational framework for the development of spatial dynamic resilience maps. The details of the framework are presented in Figure 5. The framework consists of three main components: (i) System dynamics models (ii) C# modules and (iii) Mapping modules using MATLAB/GIS.

3.1 System Dynamics Modeling

We have adopted VENSIM simulation software (Ventana Systems, 1995) for system dynamics modeling. This software has been used in wide range of applications which include aerospace, engineering, city management, economics, energy, environment, financial services, hospitality, housing, integrated policy, pharmaceuticals and retail surveys. The VENSIM software provides an interface for the development of system structure (input) and presentation of the system dynamic behavior (output). One of the advantages of using VENSIM software is that it provides facility to link with external programs VENSIM also provides the flexibility during the simulation process which allows monitoring the progress at any time during the simulation.

In this study we use the dynamic models to describe each impact used in system resilience calculation. The details on how to develop a system dynamics model are provided in Simonovic (2009) and Peck and Simonovic (2013). In the hypothetical example presented in the following section offers some details of system dynamics modeling. The only drawback of using VENSIM software is that the model structure cannot be altered externally. Therefore, it remains static during the simulation period. The impacts of change in system structure can be examined through multiple simulation runs created with different system structure.

3.2 C# Modules

In building the spatial dynamic resilience map, a link is required between the VENSIM software and the external mapping tool. Further, to describe system resilience we need a different system dynamics model for each impact considered. These models have to be linked externally. In this study we use C# modules that enable connection between the mapping tool for presenting a spatial variability of resilience and system dynamics models that are presenting temporal variability of resilience. C# programming language is a pure object-oriented language which is very flexible and easy to learn and adopt. The main advantages of C# over the existing other computer languages are (i) it has access to all the .NET Framework class libraries (ii) ease-of-development and (iii) support of distributed system structures. Since the proposed resilience framework is dealing with many system dynamics models and their components, C# is found to be the best choice for their integration. The programming steps and details on linking VENSIM software with C# interface is provided in Appendix B.

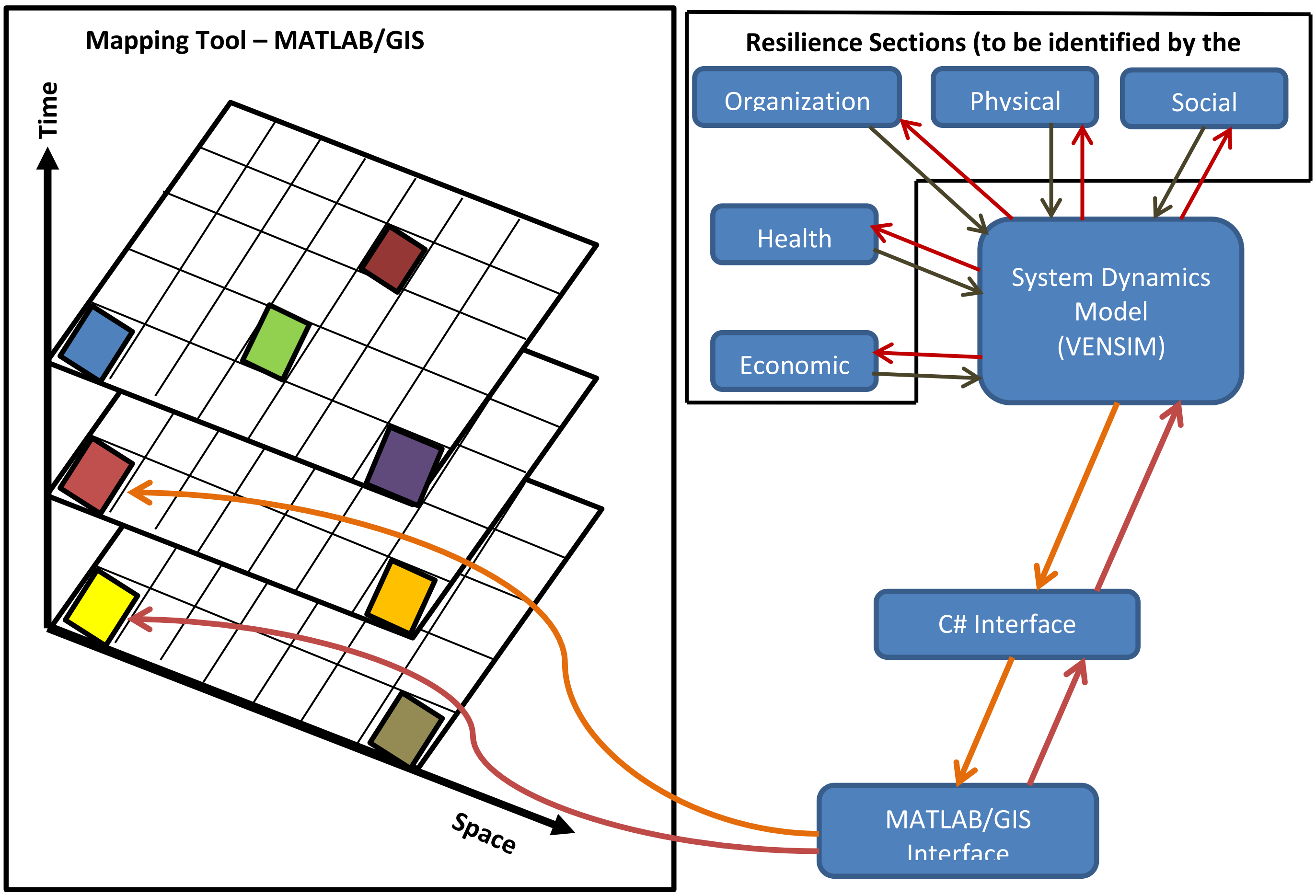

Fig. 5: Generic framework for mapping spatial dynamic resilience

3.3 Mapping Tool

Dynamic maps are developed to represent the spatial and temporal change of the resilience. The mapping tools can provide the decision makers a quick overview of the existing conditions in space and their changes over time. In this study we provide a flexible framework to adopt any mapping tool software which can be used to link externally with the VENSIM system dynamics software and C#. MATLAB is used as an external programming language which has a mapping toolbox. There are many other mapping tools which can be adopted instead of MATLAB. One of the popular family of tools for spatial analyses is Geographical Information System (GIS). However, external linking with GIS requires knowledge of PYTHON programming. The computational framework for the development of dynamic maps presented in this report is only one way of doing this task and other computational arrangements are equally feasible.

MATLAB (matrix laboratory) is proprietary software based on fourth-generation programming language. It has an interface with wide variety of programming languages including Fortran, Java and C. It consists of many inbuilt toolboxes and functions used for data computations, modelling and plotting. MATLAB is used in science and engineering and has been very popular with academia, research institutions and industry.

In this study we use the MATLAB's C# module and mapping toolbox. With the help of mapping toolbox we can create dynamic maps of resilience. The programming steps and details of dynamic resilience mapping are provided in Appendix C.

3.4 Steps Involved in Generic Framework Implementation

The following steps are involved in the development of dynamic resilience maps by using VENSIM system dynamics software, C# modules and MATLAB mapping tool:

1. Identify all impacts associated with the system disturbance. For example, in Fig.5 five impacts of system disturbance are considered, including economic, health, organizational, physical and social.
2. Build a system dynamics model to capture system performance in the domain of each considered impact.
3. Build a C# VENSIM wrapper module to link the VENSIM dynamic link library (dll) and VENSIM external commands (Appendix B.1)
4. Build a C# main module to run the VENSIM software using the C# VENSIM wrapper module and system dynamics models build in step 2 (Appendix B.2). The main module can include forms or run independently using standalone executable files.

5. Build a MATLAB main file to link the C# main module and mapping toolbox.

The roles of different computational modules are as follows:

1. MATLAB as the main driving tool executes the C# main module.
2. C# main module modifies the resilience sectors (representing considered set of impacts) and sends the information through C# VENSIM wrapper.
3. C# VENSIM wrapper is an external function which can send the information in an appropriate syntax to VENSIM dynamic linked library. The functions in this wrapper are used to run the simulations of the system dynamics models and return the calculated resilience value.
4. C# main module receives the resilience value from the C# VENSIM command wrapper and send it to the MATLAB.
5. MATLAB uses the resilience value for each cell of the spatial grid and presents it with a color code.
6. These color coded values are displayed as a map.

4.0 A Hypothetical Example

A hypothetical example is developed to illustrate the mapping of dynamic resilience for a small study area consisting of 10×10 spatial units. In this example five impacts (presented as resilience sectors) are considered as a consequence of system disturbance following Simonov and Peck (2013). In their work an urban flooding event is considered as a system disturbance. This event creates five impacts (social, economic, health, organizational, and physical) that are used in calculation of spatial dynamic resilience. The system dynamics simulation models for each of the impacts/resilience sectors are presented in Figs. 6 to 10. It is to be noted that the system dynamics model structure is very similar for each of the impacts. For more details on how to develop system dynamics models for each of the resilience sectors considered in this example the readers are directed to Peck and Simonovic (2013).

There are four variables involved in calculation of the resilience value for a given impact. They are identified as performance measure indicators which are used in the quantification of impacts and adaptive capacity. These four variables are **robustness (R1), rapidity (R2), resourcefulness (R3) and redundancy (R4)**. The values of four R’s vary spatially. Each spatial unit have different value for each of the R’s for each time interval and they are not mutually exclusive. In such a way, each R is represented as a time dependent function for each location. A typical time dependent function of each R is presented in Fig. 11.. The change of these functions with time is the consequence of interdependence of impacts and their influence on each other. The resilience value for a given spatial location for all impacts considered in this study is calculated using equation (4). The final resilience value is shown as a color coded map for each time step.

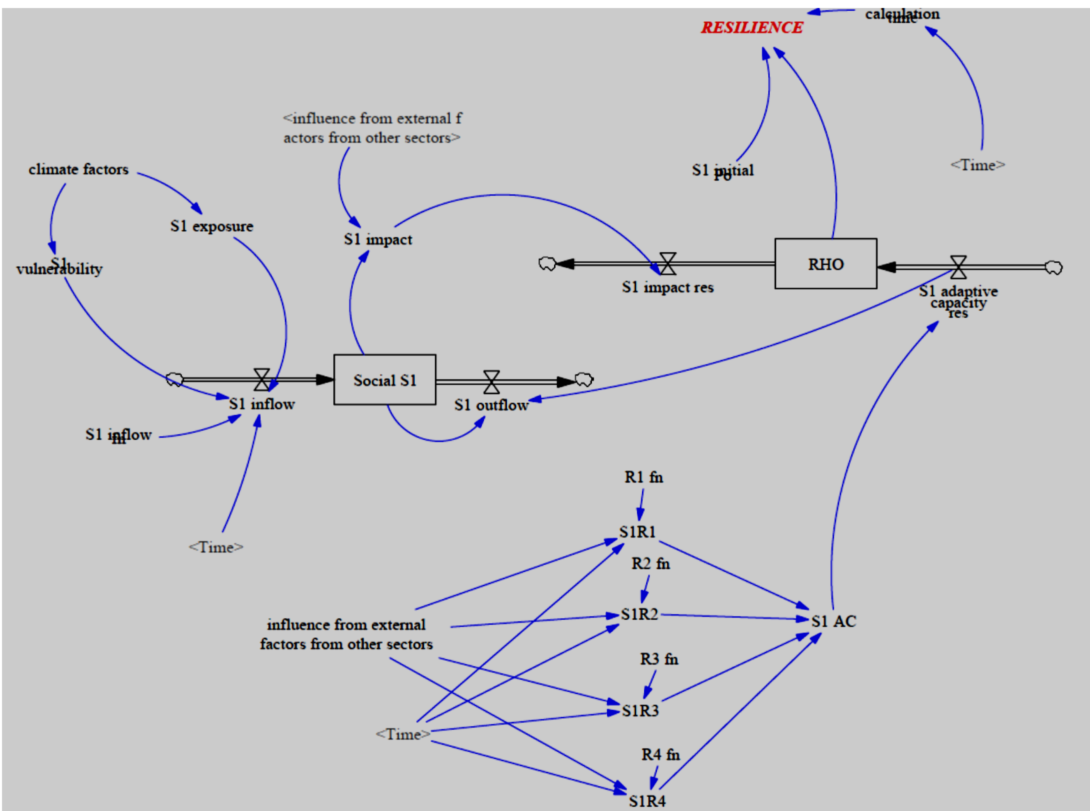

Fig 6: System dynamics simulation model structure for Social Resilience Sector

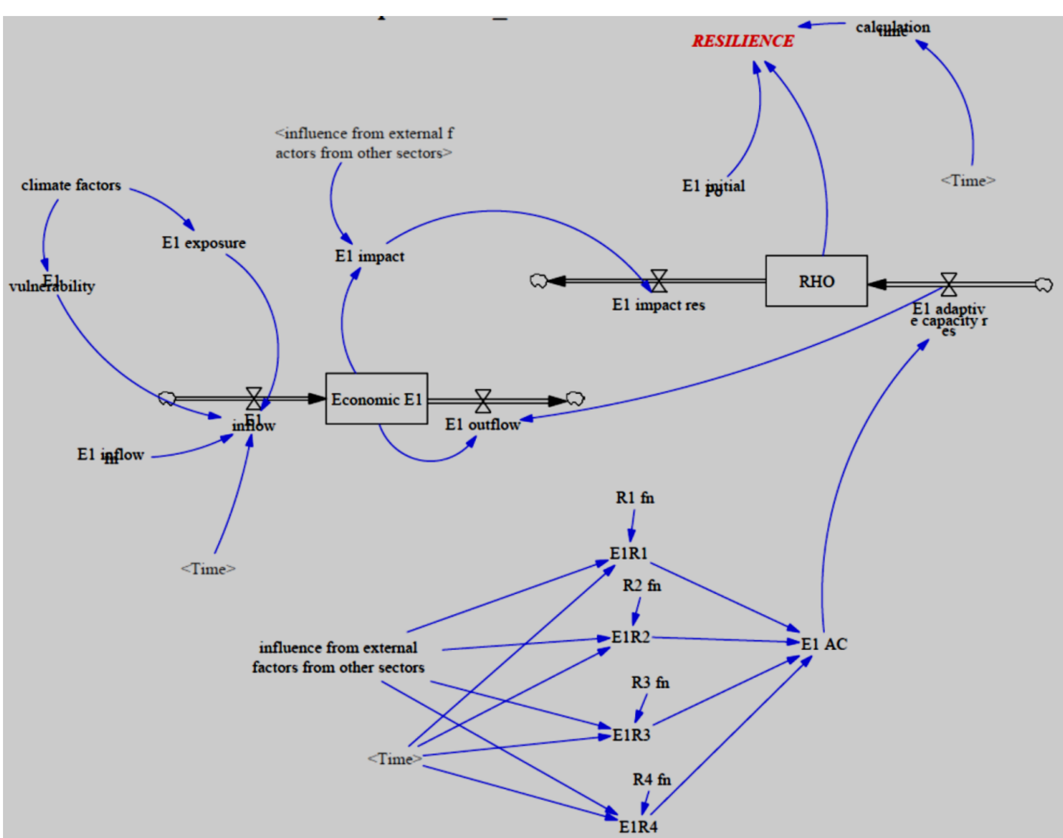

Fig 7: System dynamics simulation model structure for Economic Resilience Sector

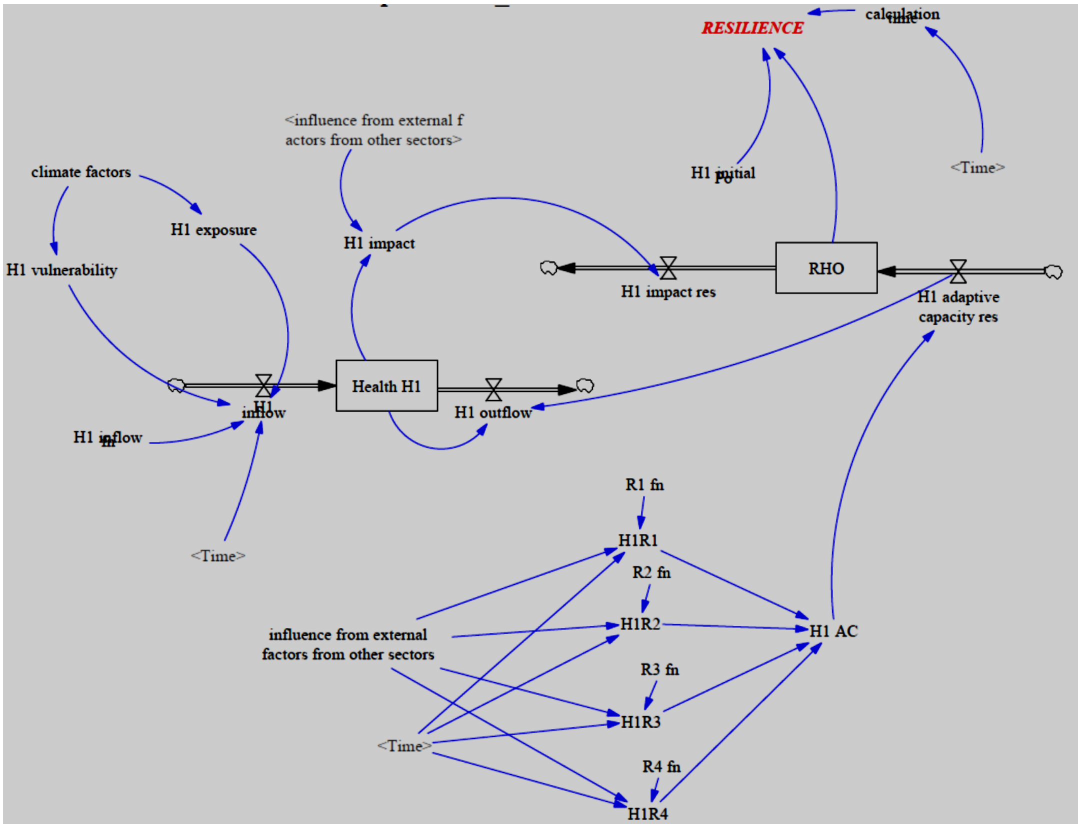

Fig 8: System dynamics simulation model structure for Health Resilience Sector

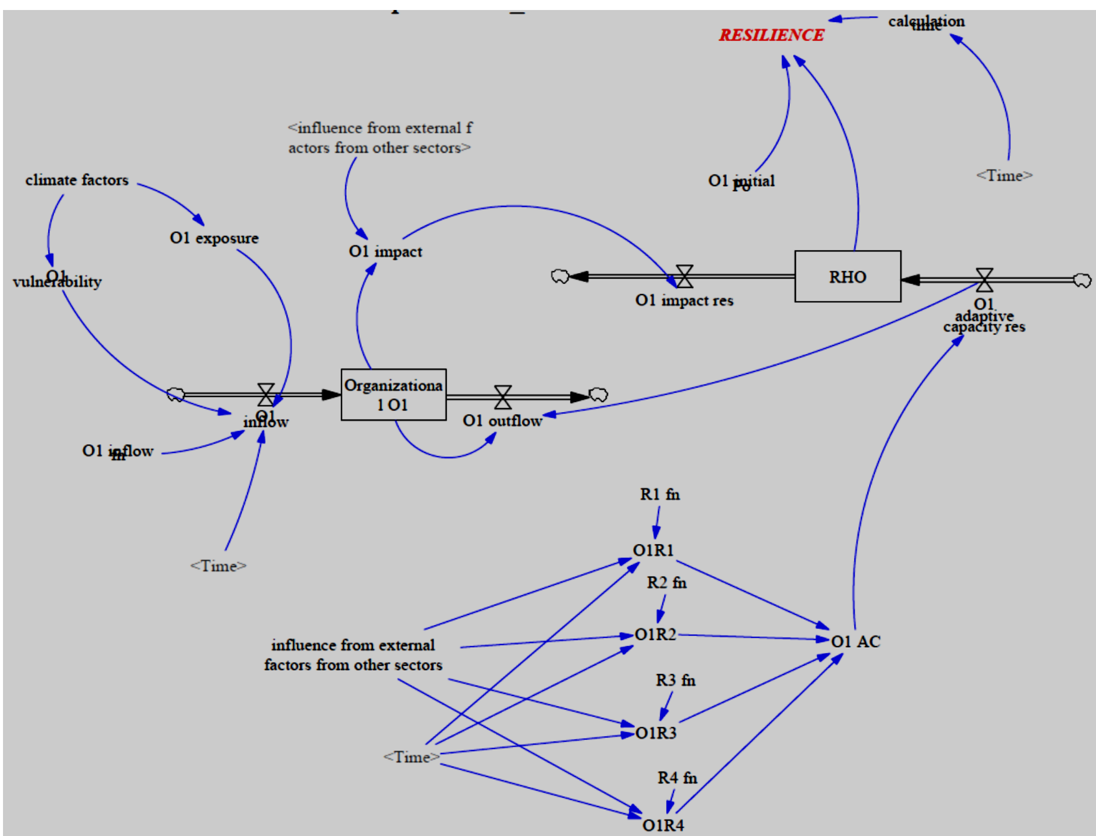

Fig 9: System dynamics simulation model structure for Organizational Resilience Sector

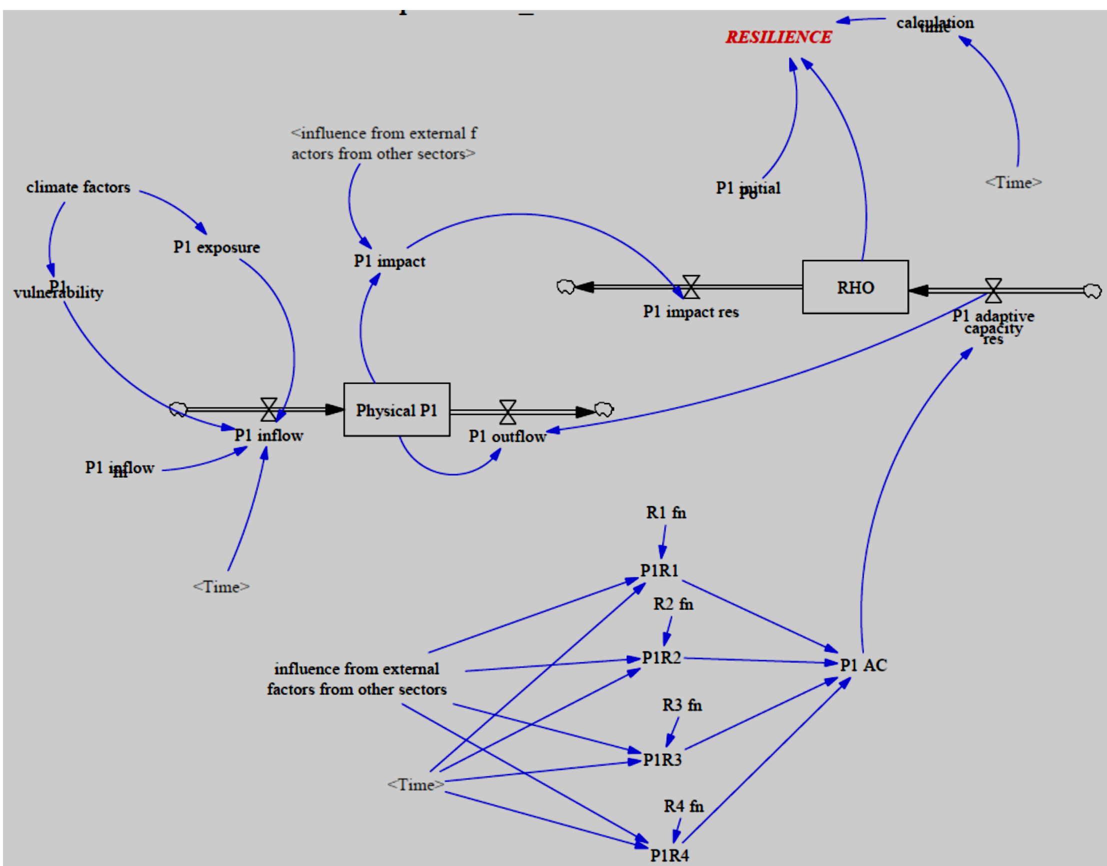

Fig 10: System dynamics simulation model structure for Physical Resilience Sector

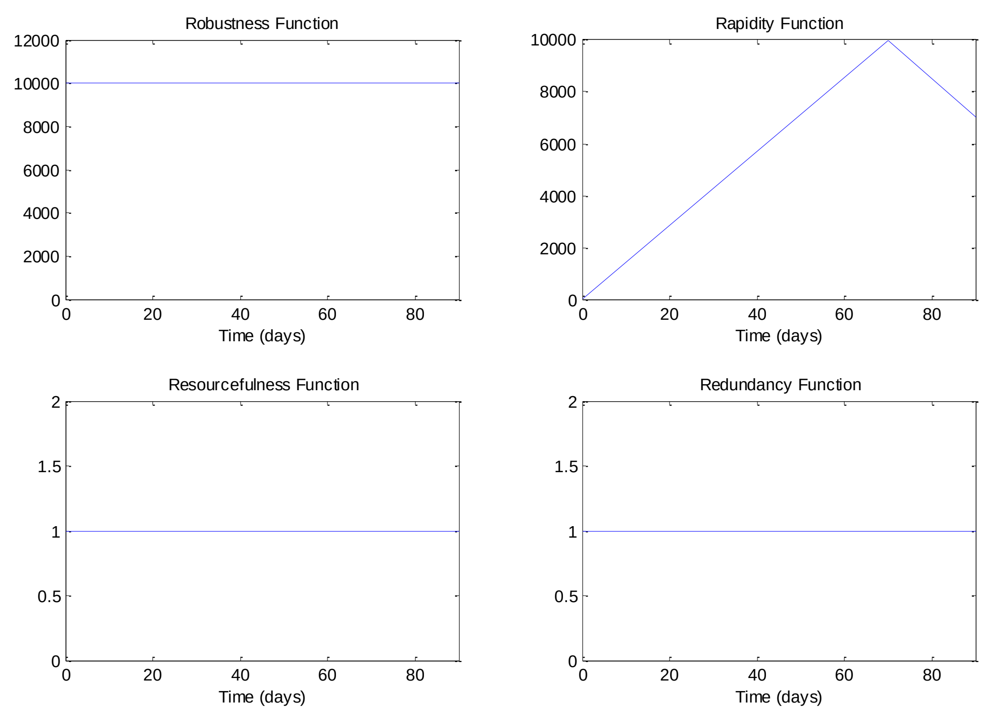

Fig 11: A typical example of four R's functions at a given location

In this example 10×10 spatial units of area is considered for illustrative purpose only. Assume that this is a city map and each spatial unit would be affected differently based on the system disturbance level at a particular location. For example, the functionality of a hospital depends on the available roots that are connecting a hospital and potential users of its services. However, in case of extreme events such as floods or earthquakes the transportation roots may be disrupted, limiting the functionality of a hospital. Therefore, the resilience of population at disconnected locations in space becomes severely affected and the overall resilience of the area decreases. The spatial dynamic resilience can be used for very effective representation of impacts of various adaptation scenarios that can serve emergency management actions (on a short time scale) or long term emergency planning.

The area map with 10×10 spatial units of area is shown in Fig 12 when there is no impact (i.e., at time t= 0). The upper panel of Fig 12 shows the resilience of each of the spatial units (value of 1 for all the locations). Once the area is subjected to an impact, the resilience on the area map changes according to the location and its interactions with the other locations. The final spatial resilience map for the hypothetical area at the end of 90 days is shown in Fig 13. It can be observed from the Fig 13 that most of the locations within the area under consideration did not recover after 90 days and their resilience values remains below one. However, two locations show the value of resilience above one. It is evident from the dynamic simulation of area map that the resilience value changes with both time and location. The developed MATLAB program generates an animated map which shows varying value of resilience over time. The animated map can be obtained from the project website (http://www.eng.uwo.ca/research/iclr/fids/publications/products/relpixel.gif).

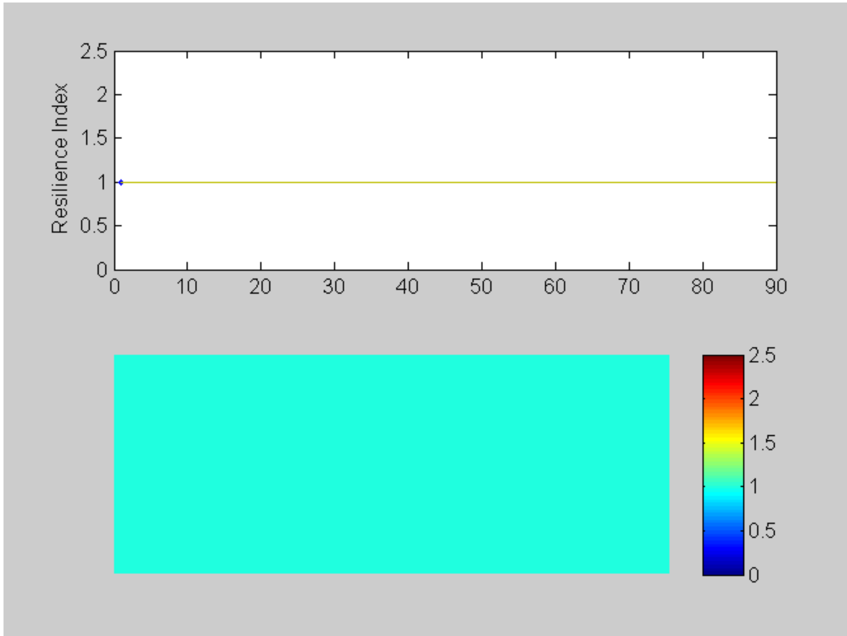

Fig 12: Resilience map at the initial time period (t = 0)

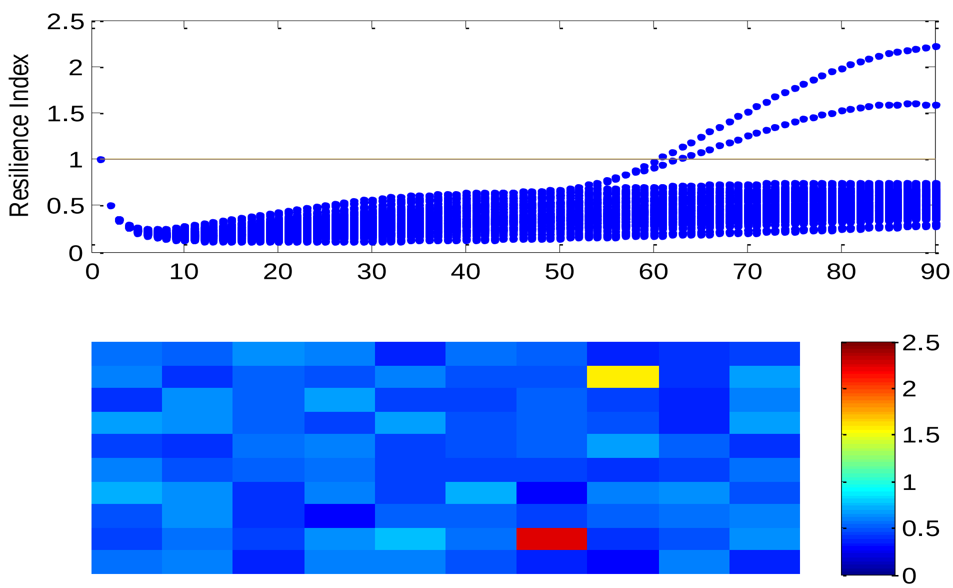

Fig 13: Resilience map at the final time period (t=90 days)

5.0 CONCLUSIONS

This report presents a generic computational framework for implementation of spatial dynamic resilience. . The framework consists of VENSIM software for system dynamics modeling, C# modules for linking VENSIM with external programs and MATLAB mapping tools for development of dynamic resilience maps. The implementation of the procedure on a hypothetical example shows that the developed framework can be used for effective development of spatial dynamic resilience maps. The hypothetical example is developed with five impacts on a 10×10 spatial units map. Number of impacts and size of the spatial domain (area) are not limited in any way. Further, based on the guidelines provided in this report, the necessary computational modules can be developed using other programming languages. The framework can also incorporate other mapping tools.

The future work includes, implementation of the framework using two real case studies selected by the Montpelier Re and the Property and Casualty Insurance Compensation Corporation (PACICC). Further this framework will be modified to include simple proxies for risk analysis and natural earthquake, flood and wind hazards estimation.. These proxies will be evaluated for various population densities, infrastructure systems and governmental policies.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the financial support for the research provided by the MITACS Accelerate Fellowship awarded to the first author and Montpelier Re grant awarded to the second author through Property and Casualty Insurance Compensation Corporation.

REFERENCES

Arthur, W., 1999. Complexity and the economy. *Science* 284(5411):107–109.

Ayyub, B. M. 2013. Systems Resilience for Multihazard Environments: Definition, Metrics, and Valuation for Decision Making. *Risk Analysis* doi: 10.1111/risa.12093

Bruneau, M., Chang, S.E., Eguchi, R.T., Lee, G.C., O'Rourke, T.D., Reinhorn, A.M., Shinozuka, M., Tierney, K., Wallace, W.A., and von Winterfeldt, D., 2003. A framework to quantitatively assess and enhance the seismic resilience of communities. *Earthquake Spectra*, 19(4):733- 752.

Bruneau M, Reinhorn A., 2007. Exploring the concept of seismic resilience for acute care facilities. *Earthquake Spectra*, 23(1):41–62.

Chang, S. E., McDaniels, T., Fox, J., Dhariwal, R. and Longstaff, H., 2013. Toward Disaster-Resilient Cities: Characterizing Resilience of Infrastructure Systems with Expert Judgments. *Risk Analysis*. doi: 10.1111/risa.12133

Folke, C., Carpenter, S., Elmquist, T., Gunderson, L., Holling, C., and Walker, B., 2002. Resilience and sustainable development: building adaptive capacity in a world of transformations. *Ambio*, 31(5):437–440.

Fiksel, J., 2006. Sustainability and resilience: toward a systems approach. *Sustainability: Science, Practice, & Policy*, 2(2):14–21

Holling, C.S., 1973. Resilience and stability of ecological systems. *Annual Review of Ecology and Systematics*, 4:1–23.

IPCC, 2013. Working Group I Contribution to the IPCC Fifth Assessment Report (AR5), Climate Change 2013: The Physical Science Basis http://www.ipcc.ch/report/ar5/wg1/#.Utgpk7SqRsw

Park, J., Seager, T. P., Rao, P. S. C., Convertino, M. and Linkov, I., 2013. Integrating Risk and Resilience Approaches to Catastrophe Management in Engineering Systems. *Risk Analysis*, 33: 356–367.

Peck, A. and S. P. Simonovic, 2013. Coastal Cities at Risk (CCaR): Generic System Dynamics Simulation Models for Use with City Resilience Simulator. *Water Resources Research Report no. 083*, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 55 pages. ISBN: (print) 978-0-7714-3024-4; (online) 978-0-7714-3025-1.

Simonovic, S.P., 2009. *Managing Water Resources: Methods and Tools for a Systems Approach*, Routledge Taylor & Francis Group, London, UK, 640 pages.

Simonovic, S.P., 2011. *Systems Approach to Management of Disasters: Methods and Applications,* A John Wiley & Sons, Inc., Hoboken, NJ, USA, 308 pages.

Simonovic, S.P., 2012. *Floods in a Changing Climate- Risk Management*, Cambridge University Press, Cambridge, UK, 179 pages.

Simonovic, S.P. and A. Peck, 2013. Dynamic Resilience to Climate Change Caused Natural Disasters in Coastal Megacities Quantification Framework, *British Journal of Environment and Climate Change,* 3(3):2231–4784.

Starr, R., Newfrock, J., and Delurey, M., 2003. Enterprise resilience: managing risk in the networked economy. *Strategy and Business* 30(1):1–150

UNISDR, 2012. Making cities resilient: My city is getting ready! A global snapshot of how local governments reduce disaster risk, *United Nations Office for Disaster Risk Reduction Report*, Geneva, Switzerland. Available at: www.unisdr.org/campaign (last accessed January 16, 2014).

Ventana Systems, 1995. *Vensim User's Guide*. Ventana Systems, Inc., Belmont, MA.

APPENDIX A:

Simonovic and Peck (2013). Dynamic Resilience to Climate Change Caused Natural Disasters in Coastal Megacities Quantification Framework, *British Journal of Environment and Climate Change,* 3(3):2231–4784.

APPENDIX B:

C# interface – linking VENSIM FILES

In this appendix detail C# coding on (i) how to read the VENSIM commands from the dynamic link library (dll) and (ii) run the dynamics models

B.1: File to read VENSIM commands

```
using System;
using System.Text;
using System.Runtime.InteropServices;
/// <summary>
/// Summary description for Class1
/// </summary>
///
namespace VensimDLLAPI
{
    public class VensimDLL
    {
        const string strVensimDLL = "vendll32.dll";

        /* information queries for vensim_get_info */
        public enum VensimInfoWanted
        {
            INFO_DLL = 1,          /* returns Minimal, Silent, Full or
Redist */
            INFO_VERSION,       /* version info for Vensim */
            INFO_USER,          /* user name \0 user company \0 */
            INFO_DIRECTORY,     /* the currently active directory */
            INFO_MODELNAME,       /* the name of the currently loaded
model (no directory) */
            INFO_TOOLSET,       /* the name of the toolset */
            INFO_TOOLLIST,         /* the names of the tools in the
loaded toolset */
            INFO_GRAPHSET,      /* name of the loaded graph set */
            INFO_GRAPHLIST,     /* list of available graphs */
```

```
            INFO_RUNLIST,       /* the list of loaded runs */
            INFO_RUNNAME,          /* the name of the run to be maded
changed with SIMULATE>RUNNAME */
            INFO_CINFILES,      /* cin files set with SIMULATE>CHGFILE
*/
            INFO_DATAFILES,     /* data file set with SIMULATE>DATA */
            INFO_BASED,            /* name of run to base on set with
SIMULATE>BASED */
            INFO_OPTMARM,             /* optimization parameter files
SIMULATE>OPTPARM */
            INFO_PAYOFF,              /* name of payoff control files
SIMULATE>PAYOFF */
            INFO_RESUME,                   /* resume satus (0 or 1)
SIMULATE>RESUME */
            INFO_SAVELIST,                  /* name of savelist file
SIMULATE>SAVELIST */
            INFO_SENSSAVELIST,  /* name of sensitivity save list files
SIMULATE>SENSSAVELIST */
            INFO_SENSITIVITY,      /* name of sensitity contol file
SIMULATE>SENSITIVITY */
            INFO_BENCHVAR,         /* workbench var name as it would
appear in title bar */
            INFO_VIEWLIST,      /* list of views in the model */
            INFO_TIMEAXIS      /* min\0max\0special */
        };

        /* return values for vensim_check_status */
        public enum VensimStatus
        {
            STATUS_IDLE = 0,
            STATUS_SIMULATING,
```

```
            STATUS_SIMHANG,
            STATUS_BLOCKACTION,
            STATUS_MEMLOCK,
            STATUS_INGAME,
            STATUS_NEEDFREE
        };
        /* attribute types for vensim_get_varattrib */
        public enum VensimVarAttrib
        {
            ATTRIB_UNITS = 1,
            ATTRIB_COMMENT,
            ATTRIB_EQUATIONS,
            ATTRIB_CAUSES,
            ATTRIB_USES,
            ATTRIB_INITCAUSES,      /* outputs only initial causes */
            ATTRIB_ACTIVECAUSES,      /* outputs only active causes - not initial */
            ATTRIB_SUBFAMILY,            /* list the subscript ranges associated with the variable */
            ATTRIB_SUBALL,             /* lists the expanded subscript list for the variable */
            ATTRIB_SUBWORK,              /* lists the expanded set of subscipts that would be used on tool invocation */
            ATTRIB_MIN,
            ATTRIB_MAX,
            ATTRIB_INCREMENT,
            ATTRIB_VARTYPE,
            ATTRIB_GROUP
        };
        /* variable types  for vensim_get_varnames */
        public enum VensimVarNames
```

```
{
    VARTYPE_WORKBENCH = -1,
    VARTYPE_ALL,
    VARTYPE_LEVEL,
    VARTYPE_AUXILIARY,
    VARTYPE_DATA,
    VARTYPE_INIITAL,
    VARTYPE_CONSTANT,
    VARTYPE_LOOKUP,
    VARTYPE_GROUP,
    VARTYPE_SUBSCRIPT,
    VARTYPE_CONSTRAINT,
    VARTYPE_TEST_INPUT,
    VARTYPE_TIME_BASE,
    VARTYPE_GAME,
    VARTYPE_SUBSCRIPT_CONSTANT
};
[DllImport(strVensimDLL)]
public static extern int vensim_be_quiet(int quietflag);
[DllImport(strVensimDLL)]
public static extern VensimStatus vensim_check_status();
[DllImport(strVensimDLL)]
public static extern int vensim_command(string command);
[DllImport(strVensimDLL)]
public static extern int vensim_continue_simulation(int
   number_time_step);
[DllImport(strVensimDLL)]
 public static extern int vensim_get_data(string filename, string
varname, string  timename, float[] varvals, float[] timevals, int
maxpoints);
[DllImport(strVensimDLL)]
```

```
    public static extern int vensim_get_dpval(string varname,
       double[] varval);
    [DllImport(strVensimDLL)]
    public static extern int vensim_get_dpvecvals(int[] vecoff,
       double[] varvals, int veclen);
    [DllImport(strVensimDLL)]
    public static extern int vensim_get_info(VensimInfoWanted
       infowanted, byte[] buf, int maxbuflen);
    [DllImport(strVensimDLL)]
    public static extern int vensim_get_sens_at_time(string
       filename, string varname, string timename, float[] attime,
       float[] vals, int maxpoint);
    [DllImport(strVensimDLL)]
    public static extern int vensim_get_substring(byte[]
fullstring, int frompos, StringBuilder buf, int maxbuflen);
    [DllImport(strVensimDLL)]
    public static extern int vensim_get_val(string varname,
       float[] varval);
    [DllImport(strVensimDLL)]
    public static extern int vensim_get_varattrib(string varname,
   VensimVarAttrib attrib, byte[] buf, int maxbuflen);
    [DllImport(strVensimDLL)]
    public static extern int vensim_get_varnames(StringBuilder
  filter, VensimVarNames vartype, byte[] buf, int maxbuflen);
    [DllImport(strVensimDLL)]
    public static extern int vensim_get_varoff(string varname);
    [DllImport(strVensimDLL)]
    public static extern int vensim_get_vecvals(int[] vecoff,
       float[] varvals, int nelm);
    [DllImport(strVensimDLL)]
    public static extern int vensim_set_parent_window(long
```

```
            vwidnow, long r1, long r2);
        [DllImport(strVensimDLL)]
        public static extern int vensim_show_sketch(int viewnum, int
            wantscroll, int zoompercent, long Vwindow);
        [DllImport(strVensimDLL)]
        public static extern int vensim_start_simulation(int
            loadfirst, int game, int overwrite);
        [DllImport(strVensimDLL)]
        public static extern int vensim_tool_command(string Vcommand,
            long Vwindow, int iswip);
    }
}
```

B.2: C# Program to Run VENSIM Dynamic Model

```
using System;
using System.IO;
namespace ReadVensim
{
    class Vensim
    {
        string strResultString;
        string run_model_name;
        bool check_print = false;
        public void initven()
        {
            string strVensimCommand;
            string path = Directory.GetCurrentDirectory();
            if (check_print == true) Console.WriteLine("Current
           Directory is {0}", path);
          string strModelPath = "SPS_Ex1.vpm";
            strVensimCommand = String.Concat("SPECIAL>LOADMODEL|",
           strModelPath);
            int nResult =
           VensimDLL.vensim_command(strVensimCommand);
          if  (check_print  ==  true)  Console.WriteLine("The  value  of
          nResult {0}", nResult);
            if (nResult == 0)
            {
                Console.WriteLine("Error loading model! Aborting.");
                return;
            }
```

```
//List of Vensim Variables.
StringBuilder sFilter = new StringBuilder("*");
                  VensimDLL.VensimVarNames nVarNamesWanted =
VensimDLL.VensimVarNames.VARTYPE_ALL;

 int nMaxStringLen = VensimDLL.vensim_get_varnames(sFilter,
nVarNamesWanted, null, 0);
 if (check_print == true) Console.WriteLine("The value of
nMaxStringLen = {0}", nMaxStringLen);

 byte[] sBuffer = new byte[nMaxStringLen];
 nResult = VensimDLL.vensim_get_varnames(sFilter,
nVarNamesWanted, sBuffer, nMaxStringLen);
 bool bExitLoop = false;
 int nFromPosition = 0;
 string[] vname = new string[50];
 int num = 0;
 while (bExitLoop == false)
 {
     StringBuilder strVensimVarName = new
     StringBuilder(100);
     nResult = VensimDLL.vensim_get_substring(sBuffer,
     nFromPosition, strVensimVarName, 100);
     if (nResult != 0)
     {
         nFromPosition = nFromPosition + nResult;
         if (check_print == true)
         {
             Console.WriteLine("sBuffer {0}", sBuffer);
             Console.WriteLine("nFromPosition {0}",
```

```
                        nFromPosition);
                    Console.WriteLine("StrVensimVarName = {0}",
                  strVensimVarName);
                }
                vname[num] = Convert.ToString(strVensimVarName);
                num++;
            }
            else
            {
                bExitLoop = true;
            }
        }
    } //END OF OBJECT INITVEN


    public void run_name()
    {
        //now set the runname

        string strVensimCommand =
       string.Concat("SIMULATE>RUNNAME|", run_model_name, "|o");
        int nResult = VensimDLL.vensim_command(strVensimCommand);
        if (check_print == true) Console.WriteLine("nResult for
            run name = {0}", nResult);
        if (nResult == 0)
        {
            Console.WriteLine("Error setting runname! Aborting.");
            return;
        }
    }//END OF RUN NAME
```

```
public void SetInitialPopulation()
{

    VensimDLL.VensimStatus nVensimStatus =
   VensimDLL.vensim_check_status();
    string strVensimCommand;
    if (nVensimStatus == VensimDLL.VensimStatus.STATUS_INGAME)
    {
        strVensimCommand =
        string.Concat("SIMULATE>SETVAL|Infection Fraction = ",
        "0.005");
        int nResult =
        VensimDLL.vensim_command(strVensimCommand);
    }
    else
    {
        //set the initial population
        strVensimCommand =
        string.Concat("SIMULATE>SETVAL|initial affected
        population = ", "5000");
        int nResult =
        VensimDLL.vensim_command(strVensimCommand);
        strVensimCommand =
        string.Concat("SIMULATE>SETVAL|adaptive capacity fn",
        lookup);
        nResult = VensimDLL.vensim_command(strVensimCommand);
        strVensimCommand =
        string.Concat("SIMULATE>SETVAL|injured patients
        threshold = ", "2000");
        nResult = VensimDLL.vensim_command(strVensimCommand);
        strVensimCommand =
```

```
                string.Concat("SIMULATE>SETVAL|INITIAL TIME = ",
                "0");
               nResult = VensimDLL.vensim_command(strVensimCommand);
               strVensimCommand =
                string.Concat("SIMULATE>SETVAL|FINAL TIME = ",
                "90");
               nResult = VensimDLL.vensim_command(strVensimCommand);
               strVensimCommand = string.Concat("SIMULATE>SETVAL|TIME
                STEP = ", "1");
               nResult = VensimDLL.vensim_command(strVensimCommand);
               if (check_print == true) Console.WriteLine("I am here
                \n nResult for  SetInitialPopulation = {0}", nResult);
               if (nResult == 0)
               {
                   Console.WriteLine("Error setting value!
                     Aborting.");
                   return;
               }
           }
       } //END OF OBJECT SetInitialPopulation

       public void checkstatus()
       {
           VensimDLL.vensim_command(""); /* clears memory - comment
out to see needfree status */
           VensimDLL.VensimStatus nStatus =
                VensimDLL.vensim_check_status();
           strResultString = string.Concat("vensim_check_status
          returns ", nStatus);
           Console.WriteLine("I am in Loop Vensim Check Status {0}",
          strResultString);
```

```
        } // END OF OBJECT  CHECKSTATUS

        public void vensim_quite()
        {
            int nResult = VensimDLL.vensim_be_quiet(1);
            strResultString = string.Concat("vensim_be_quiet returns
            ", nResult);
        } // END OF OBJECT vensim_quite
        public void vensim_simulate()
        {
            SetInitialPopulation();
            float[] varreq = new float[100];
            int[] nVaroffs = new int[3];
            int nResult = VensimDLL.vensim_start_simulation(1, 0, 1);
            strResultString   =   string.Concat("vensim_start_simulation
returns ", nResult);
            if             (check_print             ==             true)
Console.WriteLine(strResultString);

            /* vensim_get_varoff */
            nVaroffs[0] = VensimDLL.vensim_get_varoff("H1RES");
            nVaroffs[1]  = VensimDLL.vensim_get_varoff("Injured  Patient
impacts");
            nVaroffs[2] = VensimDLL.vensim_get_varoff("time");
            float[] varvals = new float[3];
            nResult = VensimDLL.vensim_get_vecvals(nVaroffs, varvals)
            int count = 0;
            varreq[count] = varvals[0];
            while (VensimDLL.vensim_continue_simulation(1) != 0)
            {
                nResult = VensimDLL.vensim_get_vecvals(nVaroffs,
```

```
        varvals, 3);
        if (check_print == true)
        Console.WriteLine(strResultString);
        varreq[count] = varvals[0];
        count++;
    }
    nResult = VensimDLL.vensim_get_vecvals(nVaroffs, varvals)
    varreq[count] = varvals[0];
    nResult = VensimDLL.vensim_finish_simulation();
    strResultString = string.Concat("vensim_finish_simulation
    returns ", nResult);
} //END OF OBJECT vensim_simulate

public void vensim_run()
{
    initven();
    vensim_quite();
    SetInitialPopulation();

    int[] nVaroffs = new int[3];
    /* vensim_get_varoff */
    nVaroffs[0] = VensimDLL.vensim_get_varoff("Infected
    People");
    nVaroffs[1] = VensimDLL.vensim_get_varoff("Susceptible
    People");
    nVaroffs[2] = VensimDLL.vensim_get_varoff("TIME");
    Console.WriteLine("{0} {1} {2}", nVaroffs[0],
    nVaroffs[1], nVaroffs[2]);
    int nResult = VensimDLL.vensim_command("MENU>GAME");
    nResult =
    VensimDLL.vensim_command("MENU>GAME>GAMEINTERVAL|TIME
```

```
STEP");
Console.WriteLine("GAME  Initialization  is  Successful  if
nResult is one = {0}", nResult);
 if (nResult == 0)
 {
     Console.WriteLine("Error Simulating! Aborting.");
     return;
 }
 float[] varvals = new float[3];
 nResult = VensimDLL.vensim_get_vecvals(nVaroffs, varvals);
 strResultString = string.Concat("At time ", varvals[2], "
 Infected People is ", varvals[0], " Susceptible People is
 ", varvals[1]);
 Console.WriteLine(strResultString);
 nResult = VensimDLL.vensim_start_simulation(1, 2, 1);
 strResultString = string.Concat("vensim_start_simulation
 returns ", nResult);
 Console.WriteLine(strResultString);
 //float[] varvals = new float[3];
 nResult = VensimDLL.vensim_get_vecvals(nVaroffs, varvals);
 strResultString = string.Concat("At time ", varvals[2], "
 Infected People is ", varvals[0], " Susceptible People is
 ", varvals[1]);
 Console.WriteLine(strResultString);
 VensimDLL.vensim_continue_simulation(1);
 nResult = VensimDLL.vensim_get_vecvals(nVaroffs, varvals);
 strResultString = string.Concat("At time ", varvals[2], "
 Infected People is ", varvals[0], " Susceptible People is
 ", varvals[1]);
 Console.WriteLine(strResultString);
 //nResult = VensimDLL.vensim_command("MENU>RUN1|o");
```

```
            string strVensimCommand =
            string.Concat("SIMULATE>SETVAL|Infection Fraction = ",
            "0.01");//,"x=", "10");
            nResult = VensimDLL.vensim_command(strVensimCommand);
            Console.WriteLine("nResult for  Infection Fraction = {0}",
            nResult);
            //nResult = VensimDLL.vensim_command("MENU>GAME>GAMEON");
            VensimDLL.vensim_continue_simulation(1);
            nResult = VensimDLL.vensim_get_vecvals(nVaroffs, varvals,
            3);
            strResultString = string.Concat("At time ", varvals[2], "
            Infected People is ", varvals[0], " Susceptible People is
            ", varvals[1]);
            Console.WriteLine(strResultString);
            nResult = VensimDLL.vensim_command("MENU>GAME>ENDGAME");
        } //END OF OBJECT vensim_run


        public void vensim_data_retrive(string strVensimParameter)
        {
            float[] rVal;
            float[] tVal;
            rVal = new float[500];
            tVal = new float[500];
            run_model_name = "Current.vdf";
            int   nResult   =   VensimDLL.vensim_get_data(run_model_name,
strVensimParameter, "time", rVal, tVal, 500);

            Console.WriteLine("nResult {0}", nResult);
```

```
            for (int i = 0; i < nResult; i++)
            {
                strResultString = string.Concat(strVensimParameter, "
: Time = ", tVal[i], ", Value = ", rVal[i]);
                Console.WriteLine(strResultString);

            }

        }
    class Program
    {
        static void Main(string[] args)
        {
            Vensim run = new Vensim();
            run.initven();
            run.vensim_simulate();
        }
    }
}
```

B.3: Output file generated from VENSIM

Time	Resilience Index
1	1.000000
2	0.960176
3	0.920383
4	0.880668
5	0.841081
6	0.801679
7	0.762464
8	0.724115
9	0.688239
10	0.654907
11	0.62412
12	0.595896
13	0.570303
14	0.547396
15	0.527212
16	0.509771
17	0.495088
18	0.483172
19	0.474028
20	0.46766
21	0.464071
22	0.463291
23	0.465453
24	0.470662
25	0.478278
26	0.486585
27	0.495559
28	0.505217
29	0.515586
30	0.526811
31	0.539033
32	0.552344
33	0.566801
34	0.582438
35	0.599279
36	0.617339

37	0.636625
38	0.657143
39	0.678764
40	0.701209
41	0.724288
42	0.747878
43	0.771903
44	0.796316
45	0.821085
46	0.846191
47	0.871591
48	0.896979
49	0.922031
50	0.946544
51	0.970391
52	0.993489
53	1.015788
54	1.037255
55	1.05787
56	1.07762
57	1.096481
58	1.114375
59	1.131236
60	1.147022
61	1.161708
62	1.175276
63	1.187716
64	1.199022
65	1.209189
66	1.218214
67	1.226097
68	1.232835
69	1.238429
70	1.242878
71	1.246181
72	1.248348
73	1.249474
74	1.249674
75	1.249021
76	1.247559
77	1.245319
78	1.242318
79	1.238569
80	1.234078
81	1.228849

82	1.222887
83	1.216192
84	1.208771
85	1.200667
86	1.191928
87	1.182583
88	1.172651
89	1.162143
90	1.151068
91	1.139457

APPENDIX C:

MATLAB Code for MAPPING

In this appendix a detail MATLAB code for generating animated spatial maps of resilience index is presented.

```
% MATLAB CODE TO GENERATE SPATIAL MAPS of RESILIENCE INDEX
% AUTHOR: Roshan Srivastav, Slobodon P. Simonovic
% LAB: Facility for Intelligent Decision Support, UWO, CANADA

% Number of Pixels
numpixel = 100;
% SETTING UP THE INPUT FOR THE VENSIM
R1 = xlsread('adaptive_cap_fn.xlsx','R1');
R2 = xlsread('adaptive_cap_fn.xlsx','R2');
R3 = xlsread('adaptive_cap_fn.xlsx','R3');
R4 = xlsread('adaptive_cap_fn.xlsx','R4');
tlim = [20 70];
caplim_R1 = [5000 10000];
caplim_R2 = [5000 10000];
caplim_R3 = [0.5 1];
caplim_R4 = [0.5 1];
%%
for i = 1:100
    %Create Input FILES
    fid = fopen('Input.txt','wt');
    trand = tlim(1,1) + diff(tlim)*rand(1,1);
    caprand = caplim_R1(1,1) + diff(caplim_R1)*rand(1,1);
    data_text_R1 =
strcat('((',num2str(R1(1,1)),',',num2str(R1(1,2)),')',',','(',num2str(
trand),',',num2str(caprand),')',',',...
        '(',num2str(R1(3,1)),',',num2str(R1(3,2)),'))');
    trand = tlim(1,1) + diff(tlim)*rand(1,1);
    caprand = caplim_R2(1,1) + diff(caplim_R2)*rand(1,1);
    data_text_R2 =
strcat('((',num2str(R2(1,1)),',',num2str(R2(1,2)),')',',','(',num2str(
trand),',',num2str(caprand),')',',',...
```

```
        '(',num2str(R2(3,1)),',',num2str(R2(3,2)),'))');
    trand = tlim(1,1) + diff(tlim)*rand(1,1);
    caprand = caplim_R3(1,1) + diff(caplim_R3)*rand(1,1);
    data_text_R3 =
strcat('((',num2str(R3(1,1)),',',num2str(R3(1,2)),')',',','(',num2str(
trand),',',num2str(caprand),')',',',...
        '(',num2str(R3(3,1)),',',num2str(R3(3,2)),'))');
    trand = tlim(1,1) + diff(tlim)*rand(1,1);
    caprand = caplim_R4(1,1) + diff(caplim_R4)*rand(1,1);
    data_text_R4 =
strcat('((',num2str(R4(1,1)),',',num2str(R4(1,2)),')',',','(',num2str(
trand),',',num2str(caprand),')',',',...
        '(',num2str(R4(3,1)),',',num2str(R4(3,2)),'))');
    fprintf(fid,'%s\n',data_text_R1);
    fprintf(fid,'%s\n',data_text_R2);
    fprintf(fid,'%s\n',data_text_R3);
    fprintf(fid,'%s',data_text_R4);
    fclose(fid);
    output =[];
    % Excecute the VENSIM C# FILES
    run('Vensim.c#');
    rel(:,i) = power(prod(output,2),(1/5));
    fprintf('%d\n',i);
end
% Create Dynamic Resilience Index Map
h2 = figure('Renderer','zbuffer');
set(gca,'NextPlot','replaceChildren');
num =10;

for i = 1: 91
    a = reshape(rel(i,:),10,10);
```

```
    subplot(2,1,1), plot(i*ones(1,numpixel),rel(i,:),'.');hold on
    plot(1:90,ones(90))
    axis([0 90 0 2.5]);

    subplot(2,1,2)
    ix = imagesc(a,[0 2.5]);
    alpha(1);
    colormap(jet);
    colorbar
    M(i) = getframe(gcf);
    M(i) = getframe(h2);

end
movie(M)
movie2avi(M,'ResiliencePixel.avi','compression', 'None','fps',1,
'quality',50);
```

APPENDIX D:

List of Previous Reports in the Series

ISSN: (Print) 1913-3200; (online) 1913-3219
In addition to 53 previous reports (No. 01 – No. 53) prior to 2007

Predrag Prodanovic and Slobodan P. Simonovic (2007). Dynamic Feedback Coupling of Continuous Hydrologic and Socio-Economic Model Components of the Upper Thames River Basin. Water Resources Research Report no. 054, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 437 pages. ISBN: (print) 978-0-7714-2638-4; (online) 978-0-7714-2639-1.

Subhankar Karmakar and Slobodan P. Simonovic (2007). Flood frequency analysis using copula with mixed marginal distributions. Water Resources Research Report no. 055, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 144 pages. ISBN: (print) 978-0-7714-2658-2; (online) 978-0-7714-2659-9.

Jordan Black, Subhankar Karmakar and Slobodan P. Simonovic (2007). A web-based flood information system. Water Resources Research Report no. 056, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 133 pages. ISBN: (print) 978-0-7714-2660-5; (online) 978-0-7714-2661-2.

Angela Peck, Subhankar Karmakar and Slobodan P. Simonovic (2007). Physical, economical, infrastructural and social flood risk - vulnerability analyses in GIS. Water Resources Research Report no. 057, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 79 pages. ISBN: (print) 978-0-7714-2662-9; (online) 978-0-7714-2663-6.

Predrag Prodanovic and Slobodan P. Simonovic (2007). Development of rainfall intensity duration frequency curves for the City of London under the changing climate. Water Resources Research Report no. 058, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 51 pages. ISBN: (print) 978-0-7714-2667-4; (online) 978-0-7714-2668-1.

Evan G. R. Davies and Slobodan P. Simonovic (2008). An integrated system dynamics model for analyzing behaviour of the social-economic-climatic system: Model description and model use guide. Water Resources Research Report no. 059, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 233 pages. ISBN: (print) 978-0-7714-2679-7; (online) 978-0-7714-2680-3.

Vasan Arunachalam (2008). Optimization Using Differential Evolution. Water Resources Research Report no. 060, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 38 pages. ISBN: (print) 978-0-7714-2689-6; (online) 978-0-7714-2690-2.

Rajesh R. Shrestha and Slobodan P. Simonovic (2009). A Fuzzy Set Theory Based Methodology for Analysis of Uncertainties in Stage-Discharge Measurements and Rating Curve. Water Resources Research Report no. 061, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 104 pages. ISBN: (print) 978-0-7714-2707-7; (online) 978-0-7714-2708-4.

Hyung-Il Eum, Vasan Arunachalam and Slobodan P. Simonovic (2009). Integrated Reservoir Management System for Adaptation to Climate Change Impacts in the Upper Thames River Basin. Water Resources Research Report no. 062, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 81 pages. ISBN: (print) 978-0-7714-2710-7; (online) 978-0-7714-2711-4.

Evan G. R. Davies and Slobodan P. Simonovic (2009). Energy Sector for the Integrated System Dynamics Model for Analyzing Behaviour of the Social-Economic-Climatic Model. Water Resources Research Report no. 063, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 191 pages. ISBN: (print) 978-0-7714-2712-1; (online) 978-0-7714-2713-8.

Leanna King, Tarana Solaiman and Slobodan P. Simonovic (2009). Assessment of Climatic Vulnerability in the Upper Thames River Basin. Water Resources Research Report no. 064, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 62 pages. ISBN: (print) 978-0-7714-2816-6; (online) 978-0-7714-2817-3.

Slobodan P. Simonovic and Angela Peck (2009). Updated rainfall intensity duration frequency curves for the City of London under the changing climate. Water Resources Research Report no. 065, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 64 pages. ISBN: (print) 978-0-7714-2819-7; (online) 978-0-7714-2820-3.

Leanna King, Tarana Solaiman and Slobodan P. Simonovic (2010). Assessment of Climatic Vulnerability in the Upper Thames River Basin: Part 2. Water Resources Research Report no. 066, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 72 pages. ISBN: (print) 978-0-7714-2834-0; (online) 978-0-7714-2835-7.

Christopher J. Popovich, Slobodan P. Simonovic and Gordon A. McBean (2010). Use of an Integrated System Dynamics Model for Analyzing Behaviour of the Social-Economic-Climatic System in Policy Development. Water Resources Research Report no. 067, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 37 pages. ISBN: (print) 978-0-7714-2838-8; (online) 978-0-7714-2839-5.

Hyung-Il Eum and Slobodan P. Simonovic (2009). City of London: Vulnerability of Infrastructure to Climate Change, Background Report #1 - Climate and Hydrologic Modeling. Water Resources Research Report no. 068, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 103 pages. ISBN: (print) 978-0-7714-2844-9; (online) 978-0-7714-2845-6.

Dragan Sredojevic and Slobodan P. Simonovic (2009). City of London: Vulnerability of Infrastructure to Climate Change, Background Report #2 - Hydraulic Modeling and Floodplain Mapping. Water Resources Research Report no. 069, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 147 pages. ISBN: (print) 978-0-7714-2846-3; (online) 978-0-7714-2847-0.

Tarana A. Solaiman and Slobodan P. Simonovic (2011). Quantifying Uncertainties in the Modelled Estimates of Extreme Precipitation Events at Upper Thames River Basin. Water Resources Research Report no. 070, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 167 pages. ISBN: (print) 978-0-7714-2878-4; (online) 978-0-7714-2880-7.

Tarana A. Solaiman and Slobodan P. Simonovic (2011). Assessment of Global and Regional Reanalyses Data for Hydro-Climatic Impact Studies in the Upper Thames River Basin. Water Resources Research Report no. 071, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 74 pages. ISBN: (print) 978-0-7714-2892-0; (online) 978-0-7714-2899-9.

Tarana A. Solaiman and Slobodan P. Simonovic (2011). Development of Probability Based Intensity-Duration-Frequency Curves under Climate Change. Water Resources Research Report no. 072, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 89 pages. ISBN: (print) 978-0-7714-2893-7; (online) 978-0-7714-2900-2.

Dejan Vucetic and Slobodan P. Simonovic (2011). Water Resources Decision Making Under Uncertainty. Water Resources Research Report no. 073, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 143 pages. ISBN: (print) 978-0-7714-2894-4; (online) 978-0-7714-2901-9.

Angela Peck, Elisabeth Bowering, and Slobodan P. Simonovic (2011). City of London: Vulnerability of Infrastructure to Climate Change, Final Report. Water Resources Research Report no. 074, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 66 pages. ISBN: (print) 978-0-7714-2895-1; (online) 978-0-7714-2902-6.

M. Khaled Akhtar, Slobodan P. Simonovic, Jacob Wibe, Jim MacGee and Jim Davies (2011). An Integrated System Dynamics Model for Analyzing Behaviour of the Social-Energy-Economy-Climate System: Model Description. Water Resources Research Report no. 075, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 211 pages. ISBN: (print) 978-0-7714-2896-8; (online) 978-0-7714-2903-3.

M. Khaled Akhtar, Slobodan P. Simonovic, Jacob Wibe, Jim MacGee and Jim Davies (2011). An Integrated System Dynamics Model for Analyzing Behaviour of the Social-Energy-Economy-Climate System: User's Manual. Water Resources Research Report no. 076, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 161 pages. ISBN: (print) 978-0-7714-2897-5; (online) 978-0-7714-2904-0.

Nick Millington, Samiran Das and Slobodan P. Simonovic (2011). The Comparison of GEV, Log-Pearson Type 3 and Gumbel Distributions in the Upper Thames River Watershed under Global Climate Models. Water Resources Research Report no. 077, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 53 pages. ISBN: (print) 978-0-7714-2898-2; (online) 978-0-7714-2905-7.

Andre Schardong and Slobodan P. Simonovic (2011). Multi-objective Evolutionary Algorithms for Water Resources Management. Water Resources Research Report no. 078, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 167 pages. ISBN: (print) 978-0-7714-2907-1; (online) 978-0-7714-2908-8.

Samiran Das and Slobodan P. Simonovic (2012). Assessment of Uncertainty in Flood Flows under Climate Change. Water Resources Research Report no. 079, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 67 pages. ISBN: (print) 978-0-7714-2960-6; (online) 978-0-7714-2961-3.

Rubaiya Sarwar, Sarah E. Irwin, Leanna King and Slobodan P. Simonovic (2012). Assessment of Climatic Vulnerability in the Upper Thames River basin: Downscaling with SDSM. Water Resources Research Report no. 080, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 65 pages. ISBN: (print) 978-0-7714-2962-0; (online) 978-0-7714-2963-7.

Sarah E. Irwin, Rubaiya Sarwar, Leanna King and Slobodan P. Simonovic (2012). Assessment of Climatic Vulnerability in the Upper Thames River basin: Downscaling with LARS-WG. Water Resources Research Report no. 081, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 80 pages. ISBN: (print) 978-0-7714-2964-4; (online) 978-0-7714-2965-1.

Samiran Das and Slobodan P. Simonovic (2012). Guidelines for Flood Frequency Estimation under Climate Change. Water Resources Research Report no. 082, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 44 pages. ISBN: (print) 978-0-7714-2973-6; (online) 978-0-7714-2974-3.

Angela Peck and Slobodan P. Simonovic (2013). Coastal Cities at Risk (CCaR): Generic System Dynamics Simulation Models for Use with City Resilience Simulator. Water Resources Research Report no. 083, Facility for Intelligent Decision Support, Department of Civil and Environmental Engineering, London, Ontario, Canada, 55 pages. ISBN: (print) 978-0-7714-3024-4; (online) 978-0-7714-3025-1.