Extending the CO2 model to 3D with random walk

|  |
| --- |
| Amir Hoseein Ghorab |
| December 2020 |
|  |

ABSTRACT

In the length of history of technology development, scientists model the real-world phenomena to get insight into the real system dynamics. Modeling techniques have been developed remarkably from using a purely mathematic differential equation to using a powerful digital computer to model complex systems such as the spread of the virus in indoor places. During the past months, as the COVID-19 pandemic spreads around the world, it has become a hot topic among researchers in different fields especially in the field of modeling and simulation. In this paper, we investigate the spread of the COVID-19 virus using CO2 particles. First, we model the realistic movement of the CO2 particles by using the random walk algorithm in 3D space. Then, we conduct simulations for different indoor floor plans using the Cadmium Cell-DEVS library. The simulation results perfectly demonstrate the validity of our model.

# INTRODUCTION

Modeling is a powerful tool for studying complex natural or artificial systems. There are mainly four types of modeling technics based on the system dynamics as follows [1]:

* Differential equation systems specification (DSS)
* Discrete-time system specification (DTSS)
* Discrete event systems specification (DEVS)
* Discrete Dynamic Systems (DDS)

Recently, DEVS as a formalism for modeling discrete-event dynamic systems has gained a lot of attention. The DEVS atomic model is a single component of the model. These atomic models may be hierarchically connected together and form a DEVS coupled model. Moreover, these atomic models may communicate with each other inside the coupled model. Similarly, The DEVS formalism coupled can be combined together to form a bigger and more complex model. Although DEVS can be used for modeling many complex systems, researchers found it challenging to model cell space systems. Therefore, CELL-DEVS formalism was introduced to cope with new challenges.

Nowadays, with the rapid spread of COVID‐19 across the world, researchers try to model different aspect of this virus such as indoor space soreading [4, 5], the effectiveness of face masks in preventing airborne transmission [3], and the outbreak pattern of the virus in the cities due to the different factors.

In this paper, we use CELL-DEVS formalism to model the indoor spread of viruses like COVID-19 using the CO2 particles. In addition, the random walk algorithm is used to apply realistic movements to the CO2 particles.

The remainder of this paper is arranged as follows. Section II introduces some backgrounds relating to the CELL\_DEVS formalism and random walk algorithm. In Section III, we present our model. In Section IV, we present the simulation results of the model. We summarize the paper in Section V.

# Background

DEVS is an advanced formalism to model discrete event systems. In this type of system, the state of the model changes upon certain events such as receiving new input or the termination internal time delay. The formalism of the DEVS atomic model is as follows [1].

|  |  |
| --- | --- |
|  | (1) |

X, Y, and S are set of input, output, and stats, respectively. the and are internal and external transmission functions in turn. The *ta* is the time advance function defining the duration of the state and is the output function that is triggered when the *ta* expires. After the expiration, the state of the model changes according to the .

Diagram

Description automatically generated

Figure 1: A simple DEVS coupled model

Figure 1 shows a simple DEVS coupled model which includes atomic and the coupled model. These subsystems can communicate with each other through internal ports.

The DEVS formalism can be extended to the CELL-DEVS by adding cellular models specification such as neighboring cell parameters, border cells parameter, and the size of the cell. The CELL-DEVS model is as follows [1]:

|  |  |
| --- | --- |
|  | (2) |

In statement 2, the term S is a set of state, n is the size of the cell (i.e., height and width), set of {t} is the number of cells on each of the dimensions (X, Y, and Z), T is a global transition function, is local computing function which compute the next state. This function can be activated upon receiving input. The B is a parameter to define the border shape such as *wrapped* or *nowrapped*, finally N is a neighborhood set. In Fig. 2, several neighborhoods are shown.

Diagram

Description automatically generated

Figure 2: Examples of neighborhood sets [1]

There are several novel software and library to simulate the CELL-DEVS model such as CD++ and CADMIUM. These tools can be run on any platform as they are written in C++ programming language. The CADMIUM is more mature than CD++ so in this project, we use CADMIUM for simulation. In order to run the model in the CADMIUM CELL-DEVS, two functions need to be defined as follows.

* Local\_computation: This function is the local computing function of the cellular model.
* output\_delay: It is a delay time to change the state of the cell.

To define the behavior of the cell the local transition function need to be modified.

In the case of the CO2 modeling, first we need to define this function according to the random walk algorithm. In a random walk, each object can move freely in all directions. In the 2D version of a random walk, as shown in figure 3, the objects can move in four directions up, down, left, and right. Similarly in the 3D version of it (figure 4), the object can also move upward and downward. The probability of choosing the direction is varied and can be changed in each step. For example, the probability of moving to the right can be 70% and 10% in time t and t+1, respectively.

A picture containing shoji, vector graphics

Description automatically generated

Figure 3: Random walk in 2D [2]

Chart, scatter chart

Description automatically generated

Figure 4: Random walk in 3D [2]

Due to the randomness of the movements, the random walk is considered a perfect solution to model the movement of the air particles in 2D and 3D space.

Second, the specific parameters of each cell need to be defined. This can be done by adding these parameters to the config JSON file. In the CO2 model, there are several types of objects such as doors, walls, windows, vent, and CO2 sources. Moreover, these objects have specific parameters such as *concentration*, or *conc\_increase* that need to be defined in the JSON file. Figure 5 shows a simple JSON file for the CO2 model.

Timeline

Description automatically generated

Figure 5: A simple CO2 model’s configuration file

In Section III, more details about the CADMIUM coding will be presented.

# Defining the CO2 model with the random walk

The main purpose of this project to model the realistic CO2 movement in 3D space using the random walk algorithm. We first present a CO2 model with a random walk in 2D space and extend it to the 3D space. In this way, we can show the differences in simulation in both 2D and 3D.

The formal specification of the model is shown as follows:

< X, Y, I, S, θ, N, d, δint, δext, τ, λ, D >

X = Ø

Y = Ø

N = {(-1,0,0), (0, -1,0), (0, 0,0), (0, 1,0), (1, 0,0), (0, 0,1), (0,0,-1)} <=> B = No-Wrapped

d = 100 Ms

τ: N -> S

S type= {

* Air: -100
* CO2\_Sources: -200
* Walls: -300
* Door: -400
* Window: -500
* Vent: -600
* Workstation: -700

}

R (row) = 10 (can be changed)

C (column) = 10 (can be changed)

In the CO2 model, CO2 sources increase the concentration of CO2, and the Vent, doors, and windows decrease the concentration of the CO2. This can be achieved by adding the concentration of the neighbor cells to the concentration of the cell. Increasing the concentration means there are more CO2 particles that have been generated or moved to this cell. In the previous model of CO2, CO2 particles move to the neighbor cells without any randomness as shown in Fig 6-10. The dark red in the middle of the figure 6 is a CO2 source that increase the concentration of the cell. In the next iteration, the CO2 particles move to the neighbor cells and increase their CO2 concentration.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Figure 6: Spreading CO2 particles iteration #1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Figure 7: Spreading CO2 particles iteration #2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Figure 8: Spreading CO2 particles iteration #3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Figure 9: Spreading CO2 particles iteration #4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Figure 10: Spreading CO2 particles iteration #5

Figure 11-13 show the spread of CO2 combined with randomness.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Figure 11: Spreading CO2 particles with random walk iteration #1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Figure 12: Spreading CO2 particles with random walk iteration #2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Figure 13: Spreading CO2 particles with random walk iteration #3

To add randomness to the air particle (including the CO2) in 2D space, we need to take the following steps:

* Adding a random number generator: This random function is used to generate random numbers. A random number generation shown in figure 14, generates a random number from -70 to 65. This range can be tuned based on the model requirements.

Text

Description automatically generated

Figure 14: A random number generator

* Adding randomness to the received neighbors' concentration: To add randomness, first, the position of the neighbor should be fetched. This is shown in figure 15.

A screenshot of a computer

Description automatically generated with medium confidence

Figure 15: Collecting the neighbors' location

Notice that the term *currentLocation* is the location of cell. Moreover, *neighbor.first* contains the the location parameter (i.e., X, Y, Z) of the neighbor.

* Adding a random number to the concentration: Figure 16 shows how to add a random number to the concentration.

Text

Description automatically generated with medium confidence

Figure 16: Adding randomness to the concentration

To add another dimension to the model, two more neighbors need to be defined in the model. These neighbors have the same X and Y location as the cell but they have different Z. Figure 17 shows such neighbors.

Graphical user interface, text

Description automatically generated

Figure 17: Neighbors for 3D space

It is worth noting that the *neighbors.first[2]* and the *std::get<2>(currentLocation)* are the Z location of the neighbor and the cell, respectively.

# Simulation and performance analysis

In this section, we simulate and analyze our model with different scenarios. To create different scenarios for simulation we designed a 3D web application as shown in figure 18.

Graphical user interface, application

Description automatically generated

Figure 18: A web application to design scenarios

Figure 19-20 show the simulation results for the 2D CO2 spreading with the random walk and without the random walk.

Diagram

Description automatically generated

Figure 19: 2D with the random walk

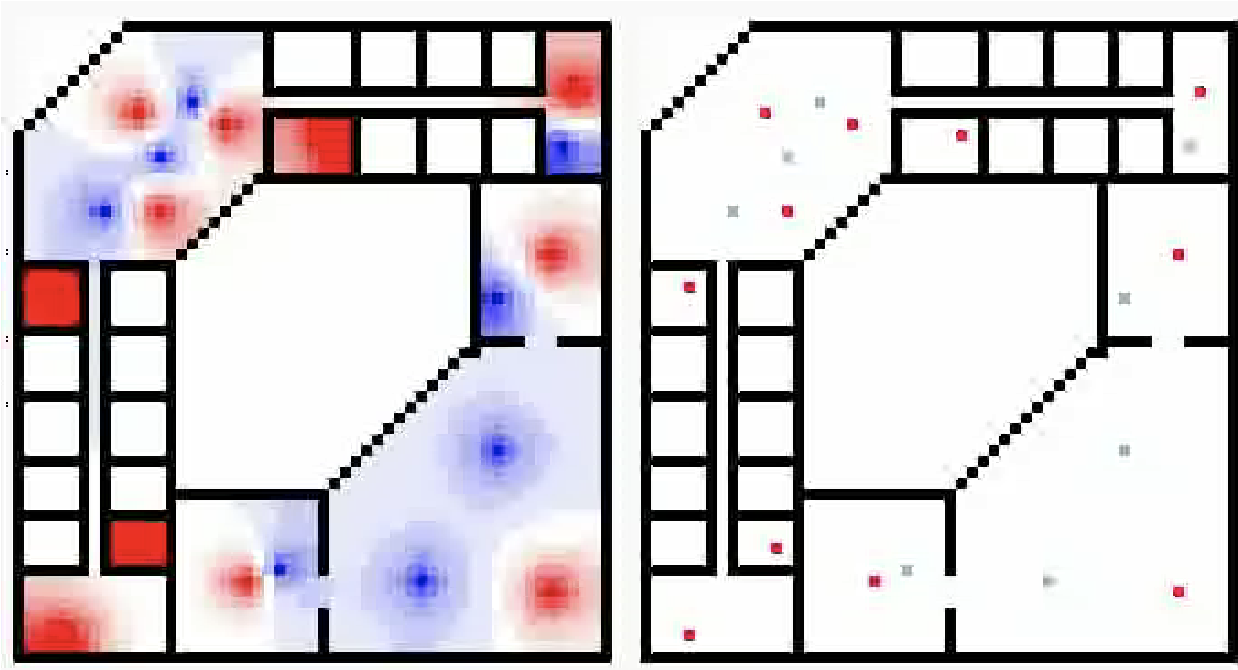


Figure 20: 2D without the random walk

Graphical user interface

Description automatically generated

Figure 21: Designed scenario

As can be seen in the simulation result, spreading with the random walk is much faster than the previous version of the CO2 model. Additionally, with the random walk, it is possible that in some locations on the cellular space the concentration of the CO2 becomes less that than other locations even without having a vent, window, or door in that location. Such situations are shown in figure 22 where there are some points near the CO2 source that their concentration became less than 500 (default value of the concentration for white color for visualization). These happen due to the fact that in the random walk each particle can move freely and at some points, the concentration of the cell becomes less than its neighbors.

Diagram

Description automatically generated

Figure 22: Cells with less CO2 concentration compare to their neighbors

Figure 23-24 shows the default scenario for 3D building occupations simulation results with and without the random walk.

A picture containing qr code

Description automatically generated

Figure 23: 3D CO2 model with random walk

A picture containing text

Description automatically generated

Figure 24: 3D CO2 model without random walk

# CONCLUSION

In this paper, we extended the CO2 model in 3D space with the random walk. This extension is critical to make the movement of CO2 particles and simulation more realistic. In the previous CO2 model, the concentration of each was the average of the neighboring cells' concentration. However, in this extended model, the concentration of the cell is a random amount of the concentration of the neighboring cells (the CO2 particles). In this way, we added the random walk to the model without putting an extra load on simulation tools to add randomness to every single particle.

We compare the simulation results of our model in 2D and 3D with different scenarios designed in a developed web application. The results show that our results are more realistic compare to the previous model.

# REFERENCES

[1] Wainer, Gabriel A. *Discrete-event modeling and simulation: a practitioner's approach*. CRC press, 2017.

[2] "Random walk", En.wikipedia.org, 2020. [Online]. Available: <https://en.wikipedia.org/wiki/Random_walk>.

[3] H. Ueki, Y. Furusawa, K. Iwatsuki-Horimoto, M. Imai, H. Kabata, H. Nishimura and Y. Kawaoka, "Effectiveness of Face Masks in Preventing Airborne Transmission of SARS-CoV-2," mSphere, vol. 5, 2020

[4] L. Marr, S. Miller, K. Prather, C. Hass, W. bahnfleth, R. Corsi, J. Tang, H. Herrmann and k. Pollitt, "FAQs on Protecting Yourself from COVID-19 Aerosol Transmission," 1 October 2020. [Online]. Available: <https://tinyurl.com/FAQ-aerosols>.

[5] S. L. Miller, W. W. Nazaroff, J. L. Jimenez, A. Boerstra, G. Buonanno, S. J. Dancer, J. Kurnitski, L. C. Marr, L. Morawska and C. Noakes, "Transmission of SARS‐CoV‐2 by inhalation of respiratory aerosol in the Skagit Valley Chorale superspreading event," Indoor Air, 2020.