



Formal verification of robotic cell injection systems

Adnan Rashid, Osman Hasan and Iram Tariq Bhatti

School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan



1 Introduction

Biological cell injection is a method used for the insertion of small amount of substances, such as biomolecules, sperms, genes, and proteins, into the suspended or adherent cells. It is widely used in gene injection (Kuncova and Kallio, 2004), drug development (Nakayama et al., 1998), intracytoplasmic sperm injection (ICSI) (Yanagida et al., 1999), and in vitro fertilization (IVF) (Sun and Nelson, 2002). For example, in IVF, the sperm is injected into matured eggs for the treatment of infertility. Similarly, drug development involves the injection of drugs into a cell and the observation of its implication at the cellular level.

Robotic cell injection systems can automatically perform the task of cell injection as opposed to the conventionally used manual and semiautomated injection procedures, which require trained operators and involve time-consuming processes and also have low success rates. The most important parameters of a robotic cell injection system are coordinate frames, capturing the orientation and movement of its various components such as injection manipulator, digital cameras, sensors and microscope, and the force controlling the injection pipette (Huang et al., 2009a). A slight error in the orientation and movement of these components may result in injection into an undesired part of the cell. Similarly, a slight excess of force may damage the membrane of the cell (Huang et al., 2006) or an insufficient force may not be able to pierce the cell (Faroque and Nizam, 2016). Thus the accuracy of the orientation and movement of these fundamental components and the injection force are vital for the reliability of the overall system. Therefore, the robotic cell injection system designs need to be analyzed and verified quite carefully to ensure that these requirements are exhibited by the final systems.

A robotic cell injection system is generally categorized into three types, namely 2-degree of freedom (DOF), 3-DOF, and 4-DOF, based on the DOF of the cell injection manipulator that is mounted on the motion stage and controls the motion of the injection pipette. The first step in the analysis of a robotic cell injection system is to model the coordinate frames corresponding to the orientations of its various components, such as the injection manipulator, cameras, and images. This model allows us to capture the movement and thus the positions of these components during the process of cell injection. Moreover, the relationship between these coordinates provides the relative positions of these components, which is an essential part of a successful cell injection procedure. Next, in order to perform the process of cell injection, the motion planning of the injection pipette is modeled using some force control algorithms, such as the contact-space-impedance force (Sun and Liu, 1997; Huang et al., 2009a) and the image-based torque controllers (Huang et al., 2006). These controllers capture the overall dynamics of the system and are mainly responsible for successful cell injection procedure and smooth functionality of the overall system.

Conventionally, the robotic cell injection systems have been analyzed using paper-and-pencil methods. However, these manual analysis techniques are prone to human error and also are not scalable for analyzing complex models like the robotic cell injection systems. Moreover, in some cases, all the required assumptions are not documented in the mathematical analysis, which may lead to erroneous designs and analysis. Similarly, the computer-based simulations and numerical techniques have been used for analyzing these systems. However, due to the involvement of the continuous-time (differential equation-based) models of the system in the analysis and the limited amount of computer memory and computational resources, the system is analyzed for a certain number of test cases only and thus the absolute accuracy cannot be achieved. Computer algebra systems, such as Mathematica (Mathematica, 2019), have also been used for analyzing these systems (Nethery and Spong, 1994). However, the symbolic algorithms residing in the core of these systems are unverified (Durán et al., 2013), which compromises the accuracy of these analyses. Due to the safety-critical nature of robotic cell injection systems, the above-mentioned conventional methods cannot be trusted as they are either prone to error or incomplete, which may lead to an undetected error in the analysis that may in turn lead to disastrous consequences.

Formal methods (Hasan and Tahar, 2015) are computer-based mathematical analysis techniques that can overcome the above-mentioned

inaccuracies. Primarily, these techniques involve the development of a mathematical model of a system and verification of its properties using computer-based mathematical reasoning. There are mainly two types of formal methods, that is, probabilistic model checking (Clarke et al., 1999) and higher-order-logic theorem proving (Harrison, 2009), that have been used in this context. Probabilistic model checking involves the development of a state-space-based probabilistic model of the underlying system and the formal verification of its intended properties that are specified in temporal logic. It has been used by Sardar and Hasan (2017) for analyzing the robotic cell injection systems. However, the formal model involves the discretization of the differential equations modeling the dynamics of these systems, which compromises the accuracy of the corresponding analysis. Higher-order-logic theorem proving (Harrison, 2009) is an interactive verification method that can overcome these limitations. It primarily involves developing a mathematical model of the system based on higher-order logic and verifying its properties based on deductive reasoning. Given the high expressiveness of higher-order logic, it can truly capture the behavior of differential equations, which is not possible in the probabilistic model checking-based analysis. Rashid and Hasan (2018) have recently used this technique for the formal verification of the same robotic cell injection system. This chapter presents an overview of these efforts, which includes the formal verification of these systems using probabilistic model checking and higher-order-logic theorem proving. Finally, it highlights the strengths and weaknesses of these techniques for analyzing robotic cell injection systems.

The rest of the chapter is organized as follows: Section 2 provides some of the related work that has been done for analyzing the robotic cell injection systems. We provide a brief overview of the formal methods, that is, model checking and theorem proving in Section 3. Section 4 presents the formal verification of the robotic cell injection system using probabilistic model checking. We present the formal analysis of the robotic cell injection system using higher-order-logic theorem proving in Section 5. We provide some discussion about the analyses, presented in Sections 4 and 5, while highlighting their strengths and weaknesses in Section 6. Finally, Section 7 concludes the chapter.



2 Related work

Huang et al. (2006) have proposed a vision-based force control framework for the robotic cell injection systems. The authors have used

biomembrane point-load model for the measurement of the injection force using visual cell deformation feedback. They have also developed a two-dimensional impedance force control strategy for the process of the robotic cell injection. Similarly, [Huang et al. \(2007\)](#) have presented a vision-based impedance force control algorithm for analyzing the three-dimensional cell injection systems. The authors have used xy coordinate frame deformation of the cell and microscope for the measurement of the total cell deformation during the process of cell injection. Later, the same authors have proposed a prototype cell injection system for automatic batch injection of suspended cells ([Huang et al., 2009b](#)). A force sensor is used to measure the real-time injection force applied to the cells during the process of cell injection. Moreover, a force control algorithm is developed and also used for controlling the motion of the injection pipette during an out-of-plane cell injection. However, all contributions presented earlier are based on conventional techniques and cannot be trusted, considering the safety-critical nature of the robotic cell injection systems.

Formal methods, such as probabilistic model checking and higher-order-logic theorem proving, have also been used for the formal analysis of the robotic cell injection systems. [Sardar and Hasan \(2017\)](#) have used probabilistic model checking ([Clarke et al., 1999](#)), that is, a state-based formal method, to formally analyze the robotic cell injection systems. However, their methodology involves the discretization of the differential equations that model the dynamics of these systems, which compromises the accuracy of the corresponding analysis. Moreover, the analysis also suffers from the inherent state-space explosion problem ([Clarke et al., 2012](#)). Recently, [Rashid and Hasan \(2018\)](#) have proposed to use higher-order-logic theorem proving for formally analyzing the robotic cell injection systems. The authors have formalized various coordinate frames and formally verified their interrelationship. Moreover, they have also formally analyzed the dynamical behavior of these systems and the motion planning of the injection pipette used in the process of cell injection. The main focus of this chapter is to present the efforts that have been made in these formal method-based analyses of the robotic cell injection systems.



3 Formal methods

3.1 Probabilistic model checking and PRISM

Model checking ([Clarke et al., 1999](#)) is one of the major formal techniques that is commonly used for analyzing concurrent systems, such as network and routing protocols. The model checking-based analysis involves the

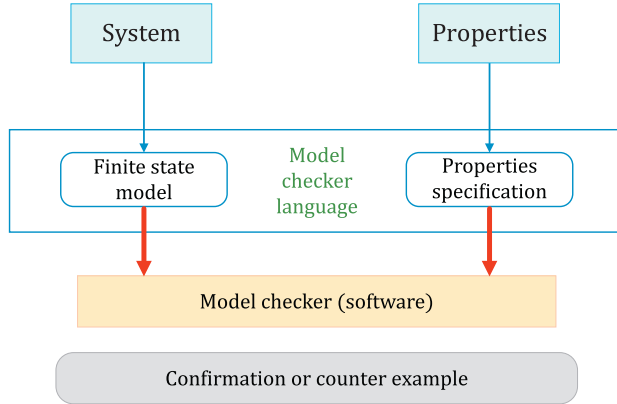


Fig. 1 Model checking.

construction of a state-space model of the given system and specification of the intended properties of the system in temporal logic (Pnueli, 1977), as depicted in Fig. 1. Next, both the model and properties are given to the model checker, which explores the state space exhaustively and automatically verifies the given system based on these properties. In the case of failure of a property, the model checker generates an error trace, which helps a user to identify and rectify the error found in the system’s model. For larger systems, this technique is subject to the problem of state-space explosion (Baier and Katoen, 2008) caused by the limited availability of computer memory and other computational resources. The abstraction of the model or usage of the efficient bounded and symbolic model checking techniques are generally used to overcome this problem.

PRISM (Kwiatkowska et al., 2011) is a probabilistic model checker that is widely adopted for the modeling and analysis of the systems exhibiting probabilistic behaviors. It involves the development of a state-space model of the underlying system by assigning probabilities to its various transitions. Next, the intended properties, captured in temporal logic, are verified for the state-space model, developed in the previous step. Based on the nature of the underlying system, that is, discrete time or continuous time, it supports various types of probabilistic models such as discrete-time Markov chains (DTMCs) (Kulkarni, 2016), continuous-time Markov chains (Kulkarni, 2016), probabilistic timed automata (Puterman, 2014), and Markov decision process (Segala and Lynch, 1995).

The formal model of a system is developed using a state-based language, that is, *PRISM language*, which is based on Alur’s reactive modules formalism. *PRISM language* provides several fundamental components, that is,

modules and *variables*, for the development of a model. A PRISM model generally consists of several modules, which can interact with each other. A module contains a set of local variables and guarded *commands*. The values of these variables at a time instant provide the state of the modules. Moreover, the guarded *commands* capture the behavior of the modules. The global state of the overall system is identified by the local states of all modules. The syntax of a PRISM command is given below:

```
[ ] guard -> prob_1 : update_1 + ... + prob_n : update_n;
```

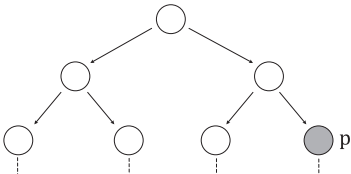
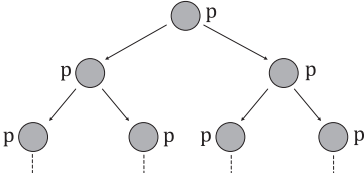
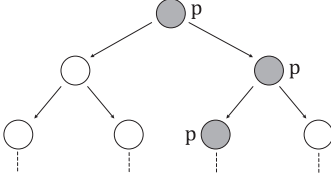
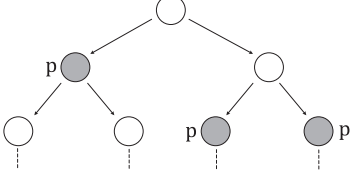
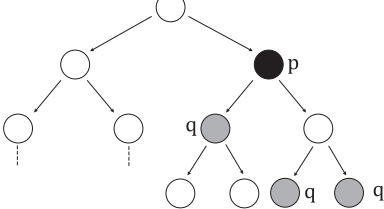
The guard is a predicate over all variables. If it is true then it allows the transition of states and performs the update of states based on the initially assigned probabilities of the transitions.

The PRISM model of a system can be formally verified using the properties specified in probabilistic temporal logics, such as probabilistic computation tree logic (PCTL (Hansson and Jonsson, 1994) and PCTL* (Aziz et al., 1995)), continuous stochastic logic (Aziz et al., 1996), probabilistic linear temporal logic, and nonprobabilistic temporal logic computational tree logic (Orgun and Ma, 1994), which are formulas in temporal logic and support specifications based on temporal, path, logical, probability, and reward operators. The supported temporal operators include next (X), always (G), and eventually (F). Similarly, the path operators are there exists (E) and forall (A), whereas negation (!), disjunction (|), conjunction (&), and implication (\Rightarrow) are the logical operators. The probability operators include probability (P), maximum probability (Pmax), and minimum probability (Pmin). Similarly, the reward operator involves reward (R). Table 1 presents the property specification, the temporal and path operators, and their graphical representation.

3.2 Theorem proving and HOL Light

Theorem proving (Harrison, 2009) involves constructing a mathematical model of the given system and the intended properties of the system based on an appropriate logic as shown in Fig. 2. Next, both the model and properties are presented as theorems to the theorem prover, which uses deductive reasoning to develop their proofs based on well-known axioms and hypothesis and thus verifies the given system based on these properties. Based on the decidability or undecidability of the underlying logic, that is, propositional or higher-order logic, theorem proving can be automatic or interactive,

Table 1 Property specification using the temporal and path operators.

Property	Graphical representation
$E F p$	<div><p>There exists a path where the property p eventually holds</p></div>
$A G p$	<div><p>p always holds for all the paths</p></div>
$E G p$	<div><p>There exists a path where p always holds</p></div>
$A F p$	<div><p>p will eventually hold for all paths</p></div>
$p \Rightarrow q$	<div><p>Whenever the property p holds q will eventually hold</p></div>

respectively. Every theorem prover comes with a set of axioms and inference rules, which, along with the already verified theorems, are the only ways to prove the new theorems. This purely deductive feature ensures soundness, that is, every sentence proved in the system is actually true.

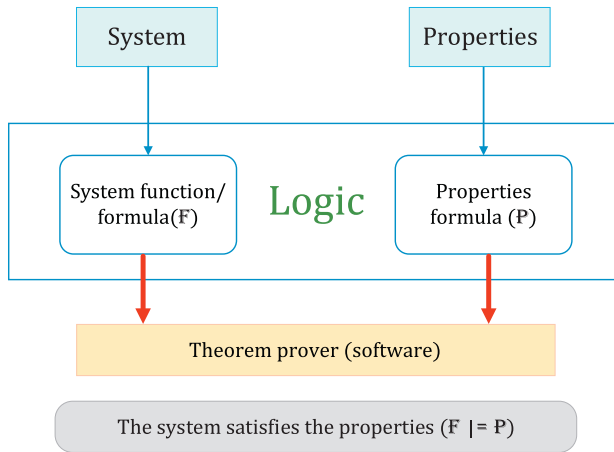


Fig. 2 Theorem proving.

HOL Light (Harrison, 1996a) is a higher-order-logic proof assistant that ensures secure theorem proving using the Objective CAML (OCaml) language, which is a variant of the strongly typed functional programming language ML (Paulson, 1996). HOL Light users can interactively verify theorems by applying the available proof tactics and proof procedures. A HOL Light theory consists of types, constants, definitions, and theorems. HOL Light theories are built in a hierarchical fashion and new theories can inherit the definitions and theorems of their parent theories. HOL Light provides an extensive support for the analysis based on Boolean algebra, real arithmetics, multivariable calculus, and vectors (Harrison, 2013). There are many automatic proof procedures (Harrison, 1996b) available in HOL Light, which help the user in concluding a proof more efficiently.

Table 2 illustrates some symbols, that is, their HOL Light and standard representations, and their meanings, which have been very often used in this chapter.



4 Model checking-based analysis of robotic cell injection systems

4.1 Robotic cell injection systems

A robotic cell injection system is generally composed of three modules, namely executive, sensory, and control modules as depicted in Fig. 3. The executive module comprises positioning table, working plate, and

Table 2 HOL Light symbols.		
HOL Light symbols	Standard symbols	Meanings
\wedge	And	Logical <i>and</i>
\vee	Or	Logical <i>or</i>
\sim	Not	Logical <i>negation</i>
\implies	\rightarrow	Implication
\iff	$=$	Equality in Boolean domain
$\! \lambda x.t$	$\forall x.t$	For all $x:t$
$\?x.t$	$\exists x.t$	There exists $x:t$
$\lambda x.t$	$\lambda x.t$	Function that maps x to $t(x)$
num	$\{0, 1, 2, \dots\}$	Natural numbers data type
real	All real numbers	Real data type
SUC n	$(n + 1)$	Successor of natural number
$\&a$	$\mathbb{N} \rightarrow \mathbb{R}$	Typecasting from natural to real numbers
abs x	$ x $	Absolute function
EL $n\ l$	<i>element</i>	<i>n</i> th element of list l

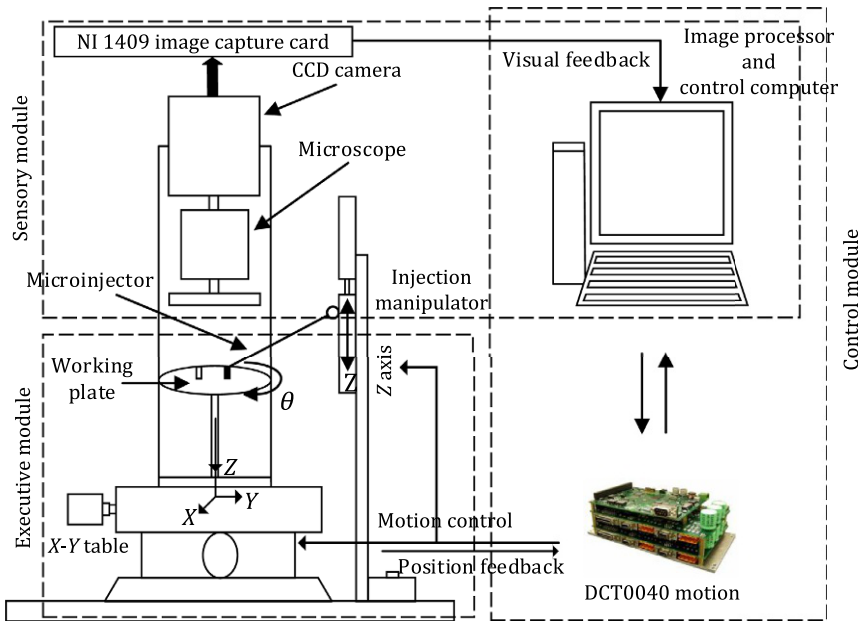


Fig. 3 Robotic cell injection systems.

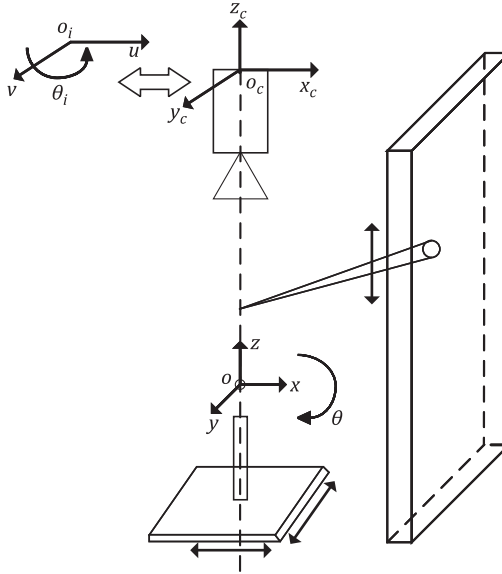


Fig. 4 Configuration of the robotic cell injection systems.

the injection manipulator. The cells that need to be injected are placed on a working plate, which is mounted on a positioning table ($XY \theta$ -axis) and the injection manipulator is mounted on Z -axis as shown in Fig. 3.

The sensory module consists of a vision system that has four parts, namely charge-coupled device (CCD) camera, peripheral component interconnect (PCI) image capture, optical microscope, and a processing card. The CCD camera is used to capture the process of the robotic cell injection using a PCI image capture. The control module comprises a host computer and a DCT0040 motion control system. Fig. 4 depicts the configuration of a robotic cell injection system. The axis $o - xyz$ represents the stage (table and working plate) coordinate frame, where o is the origin of these coordinates representing the center of the working plate and z is along the optical axis of the microscope. Similarly, $o_c - x_c y_c z_c$ represents the camera coordinate frame where o_c is the center of the microscope. The coordinate frame in image plane is represented by $o_i - uv$, where o_i is the origin and the axis uv is perpendicular to the optical axis.

4.2 Proposed formal model

Probabilistic model checking (Baier and Katoen, 2008) was used by Sardar and Hasan (2017) for the formalization of the robotic cell injection systems.

This model incorporates various random factors that are vital for the analysis of these systems due to their safety-critical nature. Probabilistic model checking has been widely utilized for analyzing a variety of systems exhibiting the probabilistic or random behavior that belong to various domains, such as randomized distributed algorithms, security, communication and routing protocols, biological systems, etc. The proposed model supports quantitative analysis while capturing the real-world factors of random nature and thus helps in determining the efficiency of these systems.

4.2.1 Proposed modeling approach and formalization

The force control module of the robotic cell injection systems is modeled as a closed-loop control system since it involves a feedback loop from the output to the input of the underlying system and thus can provide a better performance by minimizing the peak error as compared to the open-loop system. Fig. 5 depicts the model for the control of the injection pipette's trajectory in the X -coordinate. The term plant is used for the robotic cell injection system to distinguish it from the rest of the controller. The variable X_d represents the desired position of the pipette, whereas the position of the pipette, given by the encoder, is denoted by X_n . The difference of both these positions, that is, $e_x = X_d - X_n$ denotes the trajectory error in X -axis. The disturbances (internal and external) are modeled as additive noise. The injection pipette controllers adjust the driving motor's input torque τ_x using the error e_x and the current value of the external force f_{ex} applied to the actuators. The resulting torque τ_n is provided to the plant, which is further used to compute the actual position of the injection pipette X . The measurement noise due to various factors, such as encoder noise, calibration error, fabrication variation, is modeled as additive noise to X , providing X_n , which is fed back to the controller (Sardar and Hasan, 2017).

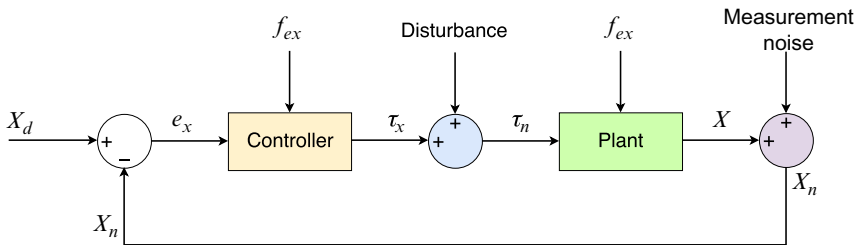


Fig. 5 Closed-loop model of the system (Sardar and Hasan, 2017).

Our proposed formalization of the robotic cell injection systems is based on DTMC and involves the discretization of the implementation of the actual system. In the case of the robotic cell injection systems, there is a specific time interval for capturing the images of the injected cell using CCD camera. Also, it requires time to process the values of radius a and depth w_d of the dimple created. These values are used to estimate the force feX , which is fed to the controller for the computation of the torque Tau_x , as depicted in Fig. 6. After adding disturbances (internal and external), the resulting torque Tau_n is provided to the plant's motor. This noise also corrupts the actual position of the pipette X_cur resulting in encoder measurement X_n . Finally, the error ex in the trajectory of the pipette is computed and fed back to the force controller (Sardar and Hasan, 2017). We consider

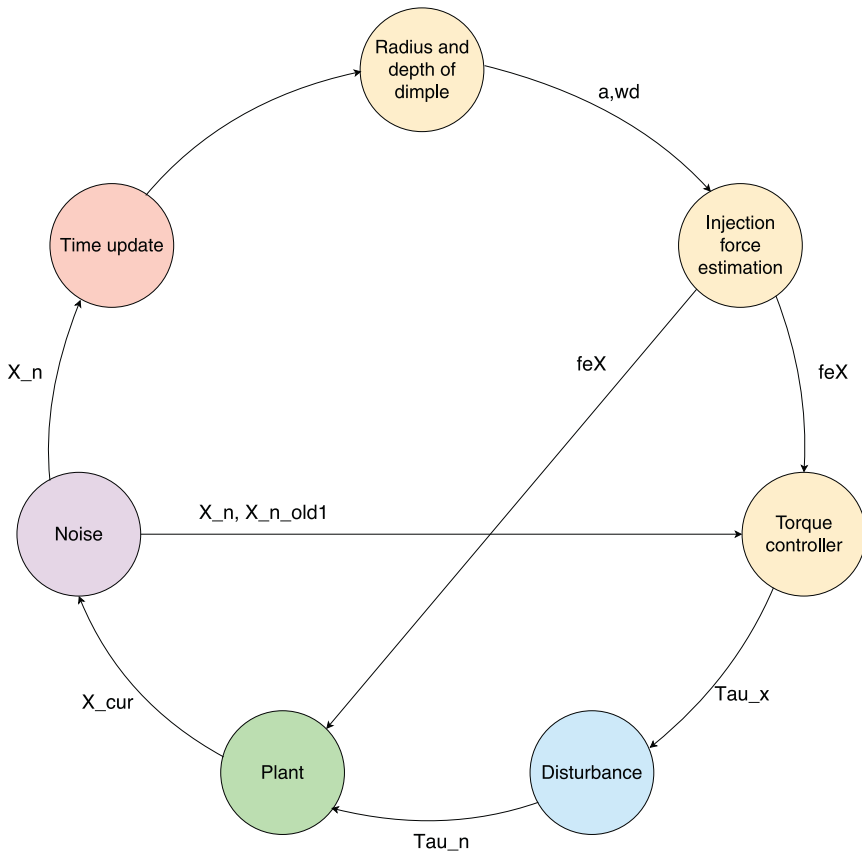


Fig. 6 Finite state-space model of the robotic cell injection system (Sardar and Hasan, 2017).

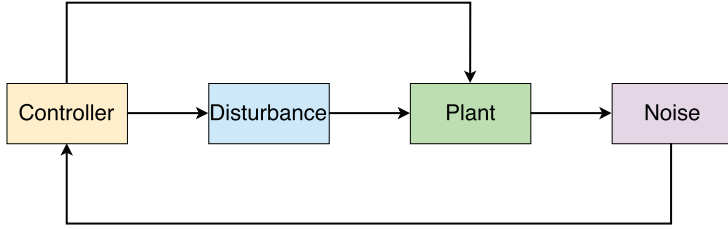


Fig. 7 PRISM modules and their interconnections (Sardar and Hasan, 2017).

one time unit as the total time taken during the process of robotic cell injection. This process is repeated at each time instant for various tasks, such as prepiercing, piercing, injection, and injector pulling out periods. These tasks are repeated for each of the new cells and this process continues in this manner to provide the injection for the whole batch of cells. Our PRISM model comprises four main modules: plant, controller, noise, and disturbance, and their interaction is shown in Fig. 7. The plant module models the dynamics of the plant. The control module implements the image-based force controller capturing the movement of the pipette. The noise and disturbance modules provide our formalization of the internal and external disturbance, as well as the measurement noise (Sardar and Hasan, 2017).

4.3 Formalization of the plant

The main modules and the configuration (orientation of various coordinate frames) of the robotic cell injection systems are depicted in Figs. 3 and 4. The image-stage coordinate frame interrelationship is mathematically expressed as

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f_x \cos \alpha & f_x \sin \alpha \\ -f_y \sin \alpha & f_y \cos \alpha \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + \begin{bmatrix} f_x d_x \\ f_y d_y \end{bmatrix} \quad (1)$$

where $f_x = \lambda/\delta_u$ and $f_y = \lambda/\delta_v$ are the display resolutions of the vision system in x - and y -directions, respectively. Here, λ represents the magnification factor of the microscope objective. Similarly, δ_u and δ_v denote the actual distances between the two adjoining pixels in the CCD sensor u - v coordinate frame. The variables d_x and d_y denote distances between the stage and the camera coordinate frames in x - and y -directions, respectively. Also, α represents the angle between the stage and the camera coordinate frames.

The dynamics of the 2-DOF motion stage, based on Lagrange's equation, is mathematically expressed as

$$\begin{bmatrix} m_x + m_y + m_p & 0 \\ 0 & m_y + m_p \end{bmatrix} \begin{bmatrix} \frac{d^2x}{dt^2} \\ \frac{d^2y}{dt^2} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix} - \begin{bmatrix} f_{ex}^d \\ f_{ey}^d \end{bmatrix} \quad (2)$$

where m_p , m_x , and m_y are the masses of the working plate and the x and y positioning tables, respectively. Similarly, τ_x and τ_y model the x and y components of the input torque to the driving motor, respectively. Similarly, f_{ex}^d and f_{ey}^d present the x and y components of the desired external force applied to the actuators during the process of cell injection.

Using the values obtained from the real-world experimental setup in Eq. (2), we obtain the following second-order differential equation after simplification (Sardar and Hasan, 2017):

$$0.022180\ddot{X} + (2.465e - 0.03)\dot{X}^2 - 0.0479\dot{X} = 1.5 \times 10^{-7}(\tau_x - f_{ex}) - 1.146 \quad (3)$$

The earlier differential equation is discretized to its corresponding differential equation by using finite difference method (LeVeque, 2007) having backward difference operators. We utilize the first-order approximation to avoid the overhead cost generated by *higher-order difference* approach. Representing the value of the variable X at time instant t_i as X_i , the first- and second-order derivatives of X using the backward difference method are mathematically expressed as Sardar and Hasan (2017)

$$\dot{X}_i = \frac{X_i - X_{i-1}}{\Delta t} \quad (4)$$

$$\ddot{X}_i = \frac{X_i - 2X_{i-1} + X_{i-2}}{(\Delta t)^2} \quad (5)$$

where Δt denotes the step size for time. Similarly, X_{i-1} and X_{i-2} represent the value of X at time instants t_{i-1} and t_{i-2} , respectively. Using these values in Eq. (3) and ignoring the higher-order terms, the solution for X_i , for $\Delta t = 1$ after simplification and rounding off the values to six decimal places, is given as (Sardar and Hasan, 2017)

$$\begin{aligned} X_i = & -77.143817X_{i-1} + 39.510075X_{i-2} \\ & - 5.926511 \times 10^{-6}(\tau_x - f_{ex}) + 45.278546 \end{aligned} \quad (6)$$

The earlier equation is formalized in PRISM as follows (Sardar and Hasan, 2017):

```
[ ] guard -> (X_cur' = ceil (p1 * X_old1 + p2 * X_old2 -
                                p3 * (TauX - feX) + p4));
```

where X_{cur} represents the current position of the injection pipette along the X -axis. Similarly, X_{old1} and X_{old2} capture the pipette's position along the X -axis at time instants t_{i-1} and t_{i-2} , respectively. Moreover, $p1$, $p2$, $p3$, and $p4$ are defined as constants in PRISM and their values are obtained from Eq. (6). Moreover, TauX represents the per-unit torque input to the driving motors and feX is the external force applied to the actuators during the process of the robotic cell injection. Since PRISM does not support rational numbers for state variables, we utilized the PRISM function `ceil` to round the final result to the nearest integer value. Moreover, the guard represents the condition for sequencing using a global variable count.

4.3.1 Formalization of the controller

The biomembrane point load model (Sun et al., 2003) is used for the vision-based estimation of the cell injection force F , which is mathematically expressed as

$$F = \frac{2\pi E h w_d^3}{a^2(1-\gamma)(1-\zeta^2)(1-\zeta^2 + \ln \zeta^2)} \frac{3 - 4\zeta^2 + \zeta^4 + 2 \ln \zeta^2}{(1-\zeta^2)(1-\zeta^2 + \ln \zeta^2)^3} \quad (7)$$

where E and h represent the membrane elastic modulus and thickness of the biomembrane, respectively. Similarly, w_d denotes the depth of the dimple resulted due to injection of the pipette and a represents the radius of the dimple after injection of the pipette. γ is the Poisson ratio and $\zeta = c/a$, where c is the radius of the pipette. Eq. (7) is formalized in PRISM as Sardar and Hasan (2017)

```
[ ] guard -> (Force' = min (ceil ((2 * pi * EM * h * pow (wd,3)) *
    (3 - 4 * pow (c/a,2) + pow (c/a,4) + 2 * log (pow (c/a,2),e)) /
    ((pow (a,2) * (1 - gamma)) * (1 - pow (c/a,2)) *
    pow ((1 - pow (c/a,2) + log (pow (c/a,2),e)),3))),
    Force_max));
```

where the variable `Force` captures the injection force and `EM` denotes the membrane elastic modulus. Similarly, `e` and `pi` are defined as constants in PRISM. The PRISM functions `pow (i,j)` and `log (i,j)` compute i to the

power of j and the log of i to the base j , respectively. Similarly, the function $\min(i, j)$ accepts two values i and j and returns the minimum out of those values. It is utilized to restrict the values of the applied force under its upper bound Force_max .

To reproduce the image processing results using our proposed approach, a and w_d are modeled differently in the four different time zones: prepiercing, piercing, injection, and pulling out of the injection pipette. During the prepiercing phase, the values of a and w_d increase by a nondecreasing factor due to the increasing velocity, whereas their values decrease during the piercing phase. Similarly, during the injection phase, they remain constant. Finally, the values decrease initially and then increase during the pulling out period.

4.3.2 Formalization of the random factors

Two main random factors have been incorporated into our PRISM model, which are classified as either disturbance or measurement noise. Disturbance can be further categorized into types: internal and external disturbances (Sardar and Hasan, 2017). The internal disturbances include plant uncertainties, such as electromagnetic effects of the components of the system, variation in parameters of the process, and distortion due to nonlinear elements (Dorf and Bishop, 2011). External disturbances mainly occur due to the environmental effects, such as temperature and electromagnetic effects of components in the surrounding. For example, the high temperature may cause the image degradation (PULNiX, 2017), which may introduce an error in the values of the parameters a and w_d of the biological cell. The measurement noise is caused by the sensor error, which includes calibration error, fabrication variation, and the lifetime of the sensor (Dorf and Bishop, 2011). The amount of the noise is characterized by the noise-to-signal ratio (Levine, 1999). Our PRISM models of the disturbance and measurement noise are depicted in Figs. 8 and 9. Since the disturbance is generally greater than

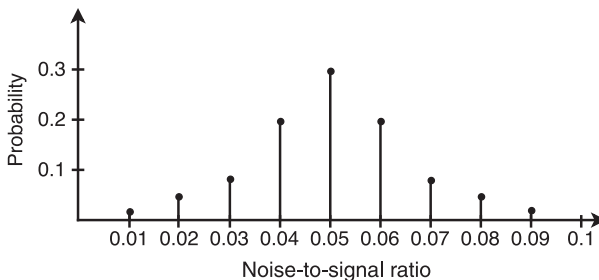


Fig. 8 Model of the disturbance (Sardar and Hasan, 2017).

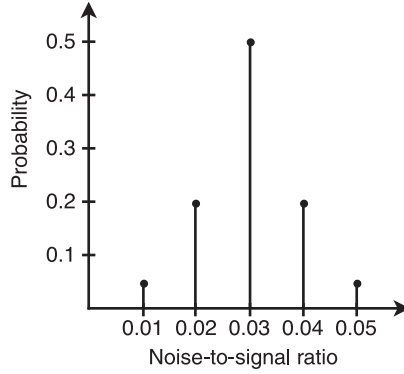


Fig. 9 Model of the measurement noise (Sardar and Hasan, 2017).

the measurement noise, the noise-to-signal ratios of 0.1–0.9 are used for disturbance, with the peak value at 0.05, as shown in Fig. 8. We implemented this in PRISM as follows (Sardar and Hasan, 2017):

```
[] guard -> 0.02:(Tau_n' = ceil (Tau_x + 0.01 * Tau_x)) +
            0.05:(Tau_n' = ceil (Tau_x + 0.02 * Tau_x)) +
            0.08:(Tau_n' = ceil (Tau_x + 0.03 * Tau_x)) +
            0.2:(Tau_n' = ceil (Tau_x + 0.04 * Tau_x)) +
            0.3:(Tau_n' = ceil (Tau_x + 0.05 * Tau_x)) +
            0.2:(Tau_n' = ceil (Tau_x + 0.06 * Tau_x)) +
            0.08:(Tau_n' = ceil (Tau_x + 0.07 * Tau_x)) +
            0.05:(Tau_n' = ceil (Tau_x + 0.08 * Tau_x)) +
            0.02:(Tau_n' = ceil (Tau_x + 0.09 * Tau_x));
```

where Tau_x and Tau_n represent the per-unit torques from the controller and after the addition of distortion.

The noise-to-signal ratios of 0.1–0.5 are used for measurement noise, with peak value at 0.03, as shown in Fig. 9. This is implemented in PRISM as follows (Sardar and Hasan, 2017):

```
[] guard -> 0.05:(X_n' = ceil (X_cur + 0.01 * X_cur)) +
            0.2:(X_n' = ceil (X_cur + 0.02 * X_cur)) +
            0.5:(X_n' = ceil (X_cur + 0.03 * X_cur)) +
            0.2:(X_n' = ceil (X_cur + 0.04 * X_cur)) +
            0.05:(X_n' = ceil (X_cur + 0.05 * X_cur));
```

where X_{cur} denotes the position of the pipette on the X -axis and X_n represents its position after adding the measurement noise of the encoder.



5 Theorem proving-based analysis of robotic cell injection systems

This section provides the theorem proving-based formal analysis of the robotic cell injection systems, which mainly includes the higher-order-logic formalization of various coordinate frames and their interrelationship. It also includes the formal verification of the solutions of the differential equations capturing the continuous dynamics of the 2-DOF robotic cell injection systems using HOL Light theorem prover. Moreover, we also present the formal modeling of the torque and force controllers and the formal verification of their implication relationship.

In order to facilitate the understanding of the chapter, the formal analysis of the robotic cell injection system is presented using a mix Math/HOL Light notation.

5.1 Formalization of the coordinate frames and their interrelationship

The coordinate frames of a robotic cell injection system, such as camera, image, and stage coordinates, are generally modeled as two-dimensional coordinates. These coordinates are modeled in HOL Light as follows ([Rashid and Hasan, 2018](#)):

Definition 5.1. Two-dimensional coordinates

$$\vdash_{def} \forall x \ y \ t. \text{two_dim_coordin } x \ y \ t = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}$$

where $x: \mathbb{R} \rightarrow \mathbb{R}$ and $y: \mathbb{R} \rightarrow \mathbb{R}$ are functions of time modeling the respective axes and t is a variable representing the time.

Next, we model a matrix providing a rotation from the stage coordinate frame ($o - xyz$) to the camera coordinate frame ($o_c - x_c y_c z_c$), and the two-dimensional vector representing the displacement between the origins of both these frames ([Rashid and Hasan, 2018](#)):

Definition 5.2. (Rotation matrix and displacement vector).

$$\vdash_{def} \forall \alpha. \text{rotat_matrix } \alpha = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$$

$$\vdash_{def} \forall dx \ dy. \text{displace_vector } dx \ dy = \begin{bmatrix} dx \\ dy \end{bmatrix}$$

where α denotes the angle between the two frames. Similarly, the variables dx and dy denote distances between the two coordinate frames in x - and y -directions, respectively.

The camera, image, and stage coordinate frames ensure the correct orientation and movement of various components of a robotic cell injection system, like, injection manipulator, microscope, stage frame, etc., and are mainly responsible for a reliable operation of the system. Therefore, the verification of the relationship between these coordinate frames is of utmost importance. The camera-stage coordinate frame interrelationship is formally verified by the following HOL Light theorem (Rashid and Hasan, 2018):

Theorem 5.1. *Camera-stage coordinate frame interrelationship*

$\vdash_{thm} \forall xc \ yc \ x \ y \ \alpha \ dx \ dy \ t.$

$[A1]: 0 < dx \wedge [A2]: 0 < dy$

$\Rightarrow (relat_camera_stage_coordin \ xc \ yc \ x \ y \ \alpha \ dx \ dy \ t \Leftrightarrow$

$$\begin{bmatrix} xc(t) \\ yc(t) \end{bmatrix} = \begin{bmatrix} x(t) * \cos \alpha + y(t) * \sin \alpha + dx \\ -x(t) * \sin \alpha + y(t) * \cos \alpha + dy \end{bmatrix}$$

where the function `relat_camera_stage_coordin` represents the camera-stage coordinate frame interrelationship. Assumptions A1-A2 ensure the nonnegativity of the distances dx and dy , respectively, and are design constraints for the relationship. The verification of the previous theorem is based on the properties of vectors and matrices alongside some real arithmetic reasoning. Now, to verify the image-camera coordinate frame interrelationship, we require modeling the display resolution matrix (Rashid and Hasan, 2018):

Definition 5.3. Display resolution matrix

$\vdash_{def} \forall fx \ fy. display_resol_matrix \ fx \ fy = \begin{bmatrix} fx & 0 \\ 0 & fy \end{bmatrix}$

where fx and fy are the display resolutions of the vision system in x - and y -directions, respectively. Next, the image-camera coordinate frame interrelationship is verified as

Theorem 5.2. *Image-camera coordinates interrelationship*

$\vdash_{thm} \forall xc \ yc \ u \ v \ t \ fx \ fy.$

$[A1]: 0 < fx \wedge [A2]: 0 < fy$

$\Rightarrow (relat_image_camera_coordin \ xc \ yc \ u \ v \ t \ fx \ fy \Leftrightarrow$

$$\begin{pmatrix} u(t) \\ v(t) \end{pmatrix} = \begin{bmatrix} fx * xc(t) \\ fy * yc(t) \end{bmatrix}$$

where the function `relat_image_camera_coordin` provides the image-camera coordinate frame interrelationship. Assumptions A1-A2 of Theorem 5.2 model the design constraints for the relationship, that is, the nonnegativity of fx and fy , respectively.

Next, to verify the image-stage coordinate frame interrelationship, given in Eq. (1), we require modeling the transformation matrix:

Definition 5.4. Transformation matrix

$$\vdash_{def} \forall fx \ fy \ \alpha. \text{transform_matrix } fx \ fy \ \alpha = \begin{bmatrix} fx * \cos \alpha & fx * \sin \alpha \\ -fy * \sin \alpha & fy * \cos \alpha \end{bmatrix}$$

Now, the image-stage coordinate frame interrelationship (Eq. 1) is verified as

Theorem 5.3. Image-stage coordinates interrelationship

$$\begin{aligned} \vdash_{thm} \forall x \ y \ u \ v \ t \ fx \ fy \ dx \ dy \ \alpha \ xc \ yc. \ [A1]: 0 < dx \wedge \ [A2]: 0 < dy \wedge \\ [A3]: 0 < fx \wedge \ [A4]: 0 < fy \wedge \\ [A5]: \text{two_dim_coordin } u \ v \ t = \text{display_resol_matrix } fx \ fy \ ** \\ \text{two_dim_coordin } xc \ yc \ t \wedge \\ [A6]: \text{two_dim_coordin } xc \ yc \ t = \text{rotat_matrix } \alpha \ ** \\ \text{two_dim_coordin } x \ y \ t + \text{displace_vector } dx \ dy \\ \Rightarrow \text{two_dim_coordin } u \ v \ t = \text{transform_matrix } fx \ fy \ \alpha \ ** \\ \text{two_dim_coordin } x \ y \ t + \begin{bmatrix} fx * dx \\ fy * dy \end{bmatrix} \end{aligned}$$

where $**$ represents the operator for the multiplication of a matrix with a vector and vice versa. Assumptions A1-A4 present the design constraints for the image-stage coordinate frame interrelationship. Assumption A5 models the image-camera coordinates interrelationship. Similarly, Assumption A6 provides the camera-stage coordinate frame interrelationship. Finally, the conclusion captures the relationship between the image and stage coordinate frames. The proof process of [Theorem 5.3](#) is mainly based on [Theorems 5.1](#) and [5.2](#) along with some properties of the vectors and matrices. The verification of these relationships ensures the correct orientation and movement of various components of the robotic cell injection system, that is, injection manipulator, working plat, microscope, camera, etc., and is vital considering the safety-critical nature of the underlying system.

Next, we model the dynamics of the robotic cell injection systems, which are generally modeled as a set of differential equations and formally verify the solution of these differential equations. For the sake of simplicity, we consider 2-DOF motion stage of the system, which considers the process of cell injection in the xy plane only. The dynamics of the cell injection system, that is, Eq. (2), is formalized in HOL Light as follows:

Definition 5.5. Dynamics of the 2-DOF motion stage

$$\begin{aligned} \vdash_{def} \forall mx \ my \ mp \ x \ y \ t \ \text{taux} \ \text{tauy} \ \text{fexd} \ \text{feyd}. \\ \text{dynamics_2dof_motion_stage } mx \ my \ mp \ x \ y \ t \ \text{taux} \ \text{tauy} \ \text{fexd} \ \text{feyd} \Leftrightarrow \\ \text{mass_matrix } mx \ my \ mp \ ** \ \text{sec_order_deriv_stage_coordin } x \ y \ t + \\ \text{posit_table_matrix } ** \ \text{fir_order_deriv_stage_coordin } x \ y \ t = \\ \text{torque_vector } \text{taux} \ \text{tauy} - \text{desired_force_vector } \text{fexd} \ \text{feyd} \end{aligned}$$

where `mass_matrix` is the matrix containing the respective masses. The function `sec_order_deriv_stage_coordin` models the first-order derivative of the stage coordinates. Similarly, the function `posit_table_matrix` provides the diagonal matrix. The function `sec_order_deriv_stage_coordin` captures the second-order derivative of the stage coordinates. Similarly, the functions `torque_vector` and `desired_force_vector` present the vectors with their elements representing the x and y components of the applied torque and desired force, respectively.

Under the condition of the applied torque and force vectors equal to zero, the injection pipette does not touch the cells. Thus, the dynamics of the underlying system, that is, Eq. (2), is transformed as follows:

$$\begin{bmatrix} m_x + m_y + m_p & 0 \\ 0 & m_y + m_p \end{bmatrix} \begin{bmatrix} \frac{d^2x}{dt^2} \\ \frac{d^2y}{dt^2} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (8)$$

The solution of the dynamics of the motion stage of the cell injection system, that is, Eq. (8), is verified as

Theorem 5.4. *Verification of solution of dynamics of motion stage*

$\vdash_{thm} \forall x \ y \ m_x \ m_y \ m_p \ \tau_{aux} \ \tau_{auy} \ f_{exd} \ f_{eyd} \ \alpha \ x_0 \ y_0 \ x_{d0} \ y_{d0}.$

$$\begin{aligned} [A1]: & 0 < m_x \wedge & [A2]: & 0 < m_y \wedge & [A3]: & 0 < m_p \wedge \\ [A4]: & x(0) = x_0 \wedge & [A5]: & y(0) = y_0 \wedge & [A6]: & \frac{dx}{dt}(0) = x_{d0} \wedge \\ [A7]: & \frac{dy}{dt}(0) = y_{d0} \wedge & [A8]: & \begin{bmatrix} \tau_{aux} \\ \tau_{auy} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \wedge & [A9]: & \begin{bmatrix} f_{exd} \\ f_{eyd} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \wedge \\ [A10]: & (\forall t. x(t) = (x_0 + x_{d0} * (m_x + m_y + m_p))) \\ & - x_{d0} * (m_x + m_y + m_p) * e^{\frac{-1}{m_x + m_y + m_p}t} \wedge \end{aligned}$$

$$\begin{aligned} [A11]: & (\forall t. y(t) = (y_0 + y_{d0} * (m_y + m_p)) - y_{d0} * (m_y + m_p) * e^{\frac{-1}{m_y + m_p}t}) \\ \Rightarrow & \text{dynamics_2dof_motion_stage } m_x \ m_y \ m_p \ x \ y \ t \ \tau_{aux} \ \tau_{auy} \ f_{exd} \ f_{eyd} \end{aligned}$$

Assumptions A1-A3 present the nonnegativity of the masses m_x , m_y , and m_p , respectively. Assumptions A4-A7 model the initial conditions, that is, the values of the stage coordinates x and y and their first-order derivatives $\frac{dx}{dt}$ and $\frac{dy}{dt}$ at $t = 0$. Assumptions A8-A9 provide the condition that the torque and the force vectors are zero. Assumptions A10-A11 capture the values of the xy stage coordinates at any time t . Finally, the conclusion provides the dynamical behavior of the 2-DOF motion stage of the underlying system. The verification of Theorem 5.4 involves the properties of derivatives of the real-valued functions, transcendental functions, vectors, and matrices along with some arithmetic reasoning. We verify an alternate form of the image-stage

coordinate frame interrelationship, which depends on the dynamical behavior of the motion stage ([Definition 5.5](#)) and is characterized as a vital property for the analyzing the robotic cell injection systems. For this purpose, we need to model the positioning table and inertia matrices:

Definition 5.6. Positioning table and inertia matrices

$$\begin{aligned} &\vdash_{def} \forall fx \ fy \ \alpha. \text{posit_table_matrix_fin } fx \ fy \ \alpha = \\ &\quad \text{posit_table_matrix} ** \text{matrix_inv } (\text{transform_matrix } fx \ fy \ \alpha) \\ &\vdash_{def} \forall mx \ my \ mp \ fx \ fy \ \alpha. \text{inertia_matrix } mx \ my \ mp \ fx \ fy \ \alpha = \\ &\quad \text{mass_matrix } mx \ my \ mp ** \text{matrix_inv } (\text{transform_matrix } fx \ fy \ \alpha) \end{aligned}$$

where the function `matrix_inv` takes a matrix $A: \mathbb{R}^{N^M}$ and returns its inverse. Now, the alternate representation of the relationship of the image and the stage coordinate frames is verified in HOL Light as the following theorem:

Theorem 5.5. Image-stage coordinates interrelationship

$$\begin{aligned} &\vdash_{thm} \forall xc \ yc \ u \ v \ x \ y \ fx \ fy \ dx \ dy \ mx \ my \ mp \ \text{taux} \ \text{tauy} \ \text{fexd} \ \text{feyd} \ \alpha. \\ &[A1]: 0 < dx \wedge [A2]: 0 < dy \wedge [A3]: 0 < fx \wedge [A4]: 0 < fy \wedge \\ &[A5]: \text{invertible } (\text{transform_matrix } fx \ fy \ \alpha) \wedge \\ &[A6]: (\forall t. u \ \text{real_differentiable } \text{atreal } t) \wedge \\ &[A7]: (\forall t. v \ \text{real_differentiable } \text{atreal } t) \wedge \\ &[A8]: (\forall t. \frac{du}{dt} \ \text{real_differentiable } \text{atreal } t) \wedge \\ &[A9]: (\forall t. \frac{dv}{dt} \ \text{real_differentiable } \text{atreal } t) \wedge \\ &[A10]: (\forall t. \text{relat_image_camera_coordin } xc \ yc \ u \ v \ t \ fx \ fy) \wedge \\ &[A11]: (\forall t. \text{relat_camera_stage_coordin } xc \ yc \ x \ y \ \alpha \ dx \ dy \ t) \wedge \\ &[A12]: \text{dynamics_2dof_motion_stage } mx \ my \ mp \ x \ y \ t \ \text{taux} \ \text{tauy} \ \text{fexd} \ \text{feyd} \\ &\quad \Rightarrow \text{inertia_matrix } mx \ my \ mp \ fx \ fy \ \alpha ** \\ &\quad \quad \text{second_order_deriv_image_coordin } u \ v \ t + \\ &\quad \text{posit_table_matrix_fin } fx \ fy \ \alpha ** \text{first_order_deriv_image_coordin } u \ v \ t = \\ &\quad \text{torque_vector } \text{taux} \ \text{tauy} - \text{desired_force_vector } \text{fexd} \ \text{feyd} \end{aligned}$$

Assumptions A1–A4 provide the design constraints for the image-stage coordinate frame interrelationship. Assumption A5 presents the condition about the existence of the inverse of the transformation matrix, that is, $(\text{transform_matrix}, \text{Definition 5.4})$ is invertible. Assumptions A6–A9 describe the differentiability condition for the image coordinates and their first-order derivatives. Assumptions A10–A11 model the image-camera and camera-stage coordinate frame interrelationships, respectively. Assumption A12 presents the dynamical behavior of the 2-DOF motion stage of the system. Finally, the conclusion of [Theorem 5.5](#) provides the alternate form the relationship between the image and stage coordinate frames. The proof process of [Theorem 5.5](#) is mainly based on the properties of the derivatives of the real-valued functions, vectors, and matrices along with some real arithmetic reasoning.

5.2 Formalization of the motion planning of the injection pipette

The motion of the injection pipette is vital for the process of the robotic cell injection as a slight excessive force applied on the pipette may damage the membrane of the cell or an insufficient force may not be able to pierce the cell. The motion of the pipette is generally controlled by the force and the torque controllers, which are mainly responsible for controlling the applied injection force and the torque applied to the deriving motor. We formalize both these controllers and formally verify their implication relationship. The impedance force control for a robotic cell injection system is mathematically expressed as

$$m\ddot{e} + b\dot{e} + ke = f_e \quad (9)$$

where m , b , and k represent the desired impedance parameters. Similarly, f_e is the two-dimensional vector containing the x and y components of the applied force. Moreover, e , \dot{e} , and \ddot{e} are the two-dimensional vectors capturing the position errors of the xy motion stage coordinate frame, its first-order and second-order derivatives, respectively, and are given as follows:

$$e = \begin{bmatrix} x_d \\ y_d \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix}, \quad \dot{e} = \begin{bmatrix} \frac{dx_d}{dt} \\ \frac{dy_d}{dt} \end{bmatrix} - \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix}, \quad \ddot{e} = \begin{bmatrix} \frac{d^2x_d}{dt^2} \\ \frac{d^2y_d}{dt^2} \end{bmatrix} - \begin{bmatrix} \frac{d^2x}{dt^2} \\ \frac{d^2y}{dt^2} \end{bmatrix} \quad (10)$$

where x and y are the actual axes and x_d and y_d are the desired axes of the stage coordinate frame. Now, the image-based torque controller for the xy stage coordinates is mathematically expressed as

$$\begin{aligned} \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix} &= \begin{bmatrix} m_x + m_y + m_p & 0 \\ 0 & m_y + m_p \end{bmatrix} \begin{bmatrix} f_x \cos \alpha & f_x \sin \alpha \\ -f_y \sin \alpha & f_y \cos \alpha \end{bmatrix} \begin{bmatrix} \frac{d^2x_d}{dt^2} \\ \frac{d^2y_d}{dt^2} \end{bmatrix} \\ &+ \begin{bmatrix} m_x + m_y + m_p & 0 \\ 0 & m_y + m_p \end{bmatrix} \begin{bmatrix} f_x \cos \alpha & f_x \sin \alpha \\ -f_y \sin \alpha & f_y \cos \alpha \end{bmatrix} \\ &\times m^{-1}(b\dot{e} + ke - f_e) + \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \cos \alpha & f_x \sin \alpha \\ -f_y \sin \alpha & f_y \cos \alpha \end{bmatrix}^{-1} \right) \\ &\times \begin{bmatrix} f_x \cos \alpha & f_x \sin \alpha \\ -f_y \sin \alpha & f_y \cos \alpha \end{bmatrix} \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} + \begin{bmatrix} f_x x^d \\ f_y y^d \end{bmatrix} \end{aligned} \quad (11)$$

Eq. (11) can be written in a compact form as

$$\vec{\tau} = MT \begin{bmatrix} \frac{d^2 x_d}{dt} \\ \frac{d^2 y_d}{dt} \end{bmatrix} + MTm^{-1}(b\dot{e} + ke - f_e) + NT \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} + \vec{f}_{ed} \quad (12)$$

where $\vec{\tau}$ and \vec{f}_{ed} describe the torque and desired force vectors. Similarly, M , N , and T model the inertia, positioning table, and transformation matrices. Eq. (12) was wrongly presented in simulation-based analysis (Huang et al., 2006) as follows:

$$\vec{\tau} = M \begin{bmatrix} \frac{d^2 x_d}{dt} \\ \frac{d^2 y_d}{dt} \end{bmatrix} + Mm^{-1}(b\dot{e} + ke - f_e) + N \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} + \vec{f}_{ed} \quad (13)$$

In Eq. (13) (used in the simulation-based analysis; Huang et al., 2006), the transformation matrix (T) is missing, which involves the contributions of the applied force and the angle at which the injection pipette is pierced into the cell and its absence can lead to disastrous consequences, that is, excess substance injection, injection of the substance at the wrong location, cell tissues damage, etc. We caught this wrong interpretation of Eq. (12) in the simulation-based analysis during the verification of the implication relationship between the force and torque controllers. We verified the implication relationship between the impedance force control and the image-based torque controller (Eq. 12) in HOL Light as follows:

Theorem 5.6. *Relationship between force and torque controllers*

$\vdash_{thm} \forall x d y d x y t m x m y m p f x f y \alpha \text{taux tauy fex fey fexd feyd m b k.}$

[A1]: $0 < m \wedge$ [A2]: $0 < k \wedge$ [A3]: $0 < b \wedge$

[A4]: `invertible (transform_matrix fx fy α) \wedge`

[A5]: `force_controller x d y d x y t m b k fex fey \wedge`

[A6]: `dynamics_2dof_motion_stage m x m y m p x y t taux tauy fexd feyd`
 \Rightarrow `torque_controller x d y d x y t m x m y m p f x f y`

$\alpha \text{taux tauy fex fey fexd feyd m b k}$

Assumptions A1–A3 model the nonnegativity of the desired impedance parameters. Assumption A4 presents the existence of the inverse of the transformation matrix. Assumption A5 provides the impedance force controller (Eq. 9). Similarly, Assumption A6 describes the dynamical behavior of the 2-DOF motion stage. Finally, the conclusion provides the image-based torque controller (Eq. 11). The proof process of Theorem 5.6 is mainly based on the properties of derivatives of the real-valued

functions, matrices, and vectors. This concludes our formalization of the motion planning of the injection pipette used in the process of the robotic cell injection.



6 Discussions

The probabilistic model checking-based formalization of the robotic cell injection systems using PRISM, presented earlier, is based on DTMC and thus involves the discretization of the continuous dynamics (modeling differential equations) of these systems. Moreover, due to the assignment of probabilities to transitions of the state-based PRISM model, it incorporates various disturbances and measurement noises associated with the underlying system. However, the proposed framework only involves the development of the formal model and thus it lacks the property verification corresponding to this model, which can be done automatically. Moreover, it enables the formalization of 2-DOF cell injection system and cannot be used to reason about 3-DOF and 4-DOF robotic injection system.

In comparison to the model checking-based analysis, the higher-order-logic theorem proving-based approach allows us to model the dynamics of the cell injection systems involving differential and derivative (Eqs. 2, 9, 11) in their true form, whereas, in their model checking-based analysis (Sardar and Hasan, 2017), they are discretized and modeled using a state-transition system. Moreover, all the verified theorems are universally quantified and can thus be specialized to the required values based on the requirement of the analysis of the cell injection systems. However, due to the undecidable nature of the higher-order logic, the verification results involve manual interventions and human guidance. Moreover, it only provides the formalization of 2-DOF cell injection system and cannot be used to reason about 3-DOF and 4-DOF robotic injection system. Table 3 presents a comparison of various analysis techniques, summarizing their strength and weaknesses, for analyzing the robotic cell injection systems. This comparison is performed based on various parameters such as expressiveness, accuracy, and automation. For example, in model checking, we cannot truly model the differential equations, and their discretization results in an abstracted model, which makes it less expressive. Moreover, higher-order-logic theorem proving enables the verification in an interactive manner due to the undecidable nature of the underlying logic.

Table 3 Comparison of techniques for analyzing robotic cell injection systems.

	Paper-and-pencil proof	Simulation	Computer algebra system	Model checking	Theorem proving
Expressiveness	✓	✓	✓		✓
Accuracy	✓ (?)			✓	✓
Automation		✓	✓	✓	

7 Conclusions

Robotic cell injection involves the insertion of biomolecules, sperms, DNA, and proteins into a specific location of suspended or adherent cells and is widely used in drug development, cellular biology research, and trans-genics. This chapter provides our probabilistic model checking and higher-order-logic theorem proving-based formalization of the 2-DOF robotic cell injection systems. Finally, a discussion provides the strengths and weaknesses of these analyses and thus enables a user to select the appropriate analysis technique based on a particular scenario.

References

Aziz, A., Singhal, V., Balarin, F., Brayton, R.K., Sangiovanni-Vincentelli, A.L., 1995. It usually works: the temporal logic of stochastic systems. In: LNCS. Computer Aided Verification, vol. 939. Springer, pp. 155–165.

Aziz, A., Sanwal, K., Singhal, V., Brayton, R., 1996. Verifying continuous time Markov chains. In: LNCS. Computer Aided Verification, vol. 1102. Springer, pp. 269–276.

Baier, C., Katoen, J.P., 2008. Principles of Model Checking. MIT Press.

Clarke, E.M., Grumberg, O., Peled, D., 1999. Model Checking. MIT Press.

Clarke, E.M., Klieber, W., Nováček, M., Zuliani, P., 2012. Model checking and the state explosion problem. In: LNCS. Tools for Practical Software Verification, vol. 7682. Springer, pp. 1–30.

Dorf, R.C., Bishop, R.H., 2011. Modern Control Systems. Pearson.

Durán, A.J., Pérez, M., Varona, J.L., 2013. The misfortunes of a mathematicians’ trio using computer algebra systems: can we trust? CoRR. abs/1312.3270.

Faroque, M., Nizam, S., 2016. Virtual reality training for micro-robotic cell injection. Technical report. Deakin University, Australia.

Hansson, H., Jonsson, B., 1994. A logic for reasoning about time and reliability. Form. Asp. Comput. 6 (5), 512–535.

Harrison, J., 1996a. HOL Light: a tutorial introduction. In: LNCS Formal Methods in Computer-Aided Design. vol. 1166. Springer, pp. 265–269.

- Harrison, J., 2009. *Handbook of Practical Logic and Automated Reasoning*. Cambridge University Press.
- Harrison, J., 2013. The HOL Light theory of Euclidean space. *J. Autom. Reason.* 50 (2), 1–18.
- Harrison, J., 1996b. *Formalized Mathematics*. Turku Centre for Computer Science.
- Hasan, O., Tahar, S., 2015. Formal verification methods. In: *Encyclopedia of Information Science and Technology*. third ed. IGI Global Pub., pp. 7162–7170.
- Huang, H., Sun, D., Mills, J.K., Li, W.J., 2006. A visual impedance force control of a robotic cell injection system. In: *Robotics and Biomimetics*, IEEE, pp. 233–238.
- Huang, H., Sun, D., Mills, J.K., Li, W.J., 2007. Visual-based impedance force control of three-dimensional cell injection system. In: *Robotics and Automation*, IEEE, pp. 4196–4201.
- Huang, H., Sun, D., Mills, J.K., Li, W.J., Cheng, S.H., 2009. Visual-based impedance control of out-of-plane cell injection systems. *Trans. Autom. Sci. Eng.* 6 (3), 565–571.
- Huang, H.B., Sun, D., Mills, J.K., Cheng, S.H., 2009. Robotic cell injection system with position and force control: toward automatic batch biomanipulation. *IEEE Trans. Robot.* 25 (3), 727–737.
- Kulkarni, V.G., 2016. *Modeling and Analysis of Stochastic Systems*. CRC Press.
- Kuncova, J., Kallio, P., 2004. Challenges in capillary pressure microinjection. In: *Engineering in Medicine and Biology Society*, vol. 2. IEEE, pp. 4998–5001.
- Kwiatkowska, M., Norman, G., Parker, D., 2011. PRISM 4.0: verification of probabilistic real-time systems. In: *LNCS. Computer Aided Verification*, vol. 6806. Springer, pp. 585–591.
- LeVeque, R.J., 2007. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. vol. 98. SIAM.
- Levine, W.S., 1999. *Control System Applications*. CRC Press, Boca Raton, FL.
- Mathematica, 2019. <https://www.wolfram.com/mathematica/>. (Accessed January 2019).
- Nakayama, T., Fujiwara, H., Tastumi, K., Fujita, K., Higuchi, T., Mori, T., 1998. A new assisted hatching technique using a piezo-micromanipulator. *Fertil. Steril.* 69 (4), 784–788.
- Nethery, J.F., Spong, M.W., 1994. Robotica: a mathematica package for robot analysis. *IEEE Robot. Autom. Mag.* 1 (1), 13–20.
- Orgun, M.A., Ma, W., 1994. An overview of temporal and modal logic programming. In: *LNCS. Temporal Logic*, vol. 827. Springer, pp. 445–479.
- Paulson, L.C., 1996. *ML for the Working Programmer*. Cambridge University Press.
- Pnueli, A., 1977. The temporal logic of programs. In: *Foundations of Computer Science*, IEEE, pp. 46–57.
- PULNiX, 2017. TM-6701 AN progressive scan full-frame shutter camera. www.jai.com/SiteCollectionDocuments/Camera_Solutions_Datasheets/TM-Datasheets/Datasheet_TM-6701AN.pdf.
- Puterman, M.L., 2014. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons.
- Rashid, A., Hasan, O., 2018. Formal analysis of robotic cell injection systems using theorem proving. In: *LNCS Special Issue on the Theme of Design, Modeling and Evaluation of Cyber Physical Systems*. Springer. http://save.seecs.nust.edu.pk/pubs/2018/CyPhy_2017.pdf.
- Sardar, M.U., Hasan, O., 2017. Towards probabilistic formal modeling of robotic cell injection systems. In: *Models for Formal Analysis of Real Systems*. Open Publishing Association, Uppsala, Sweden, pp. 271–282.
- Segala, R., Lynch, N., 1995. Probabilistic simulations for probabilistic processes. *Nordic J. Comput.* 2 (2), 250–273.
- Sun, D., Liu, Y., 1997. Modeling and impedance control of a two-manipulator system handling a flexible beam. In: *IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, pp. 1787–1792.

- Sun, Y., Nelson, B.J., 2002. Biological cell injection using an autonomous microrobotic system. *Robot. Res.* 21 (10–11), 861–868.
- Sun, Y., Wan, K.T., Roberts, K.P., Bischof, J.C., Nelson, B.J., 2003. Mechanical property characterization of mouse zona pellucida. *IEEE Trans. Nanobiosci.* 2 (4), 279–286.
- Yanagida, K., Katayose, H., Yazawa, H., Kimura, Y., Konnai, K., Sato, A., 1999. The usefulness of a piezo-micromanipulator in intracytoplasmic sperm injection in humans. *Hum. Reprod.* 14 (2), 448–453.