https://doi.org/10.1007/s00165-020-00514-3
BCS © 2020
Formal Aspects of Computing (2020) 32: 229–250

Formal Aspects of Computing

Formal verification of robotic cell injection systems up to 4-DOF using HOL Light

Adnan Rashid[1] and Osman Hasan[1]

[1]School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan

Abstract. Cell injection is an approach used for the delivery of small sample substances into a biological cell and is widely used in drug development, gene injection, intracytoplasmic sperm injection and in-vitro fertilization. Robotic cell injection systems provide the automation of the process as opposed to the manual and semi-automated cell injection systems, which require expert operators and involve time consuming processes and also have lower success rates. The automation of the cell injection process is obtained by controlling the orientation and movement of its various components, like injection manipulator, microscope etc., and planning the motion of the injection pipette by controlling the force of the injection. The conventional techniques to analyze the cell injection process include paper-and-pencil proof and computer simulation methods. However, both these techniques suffer from their inherent limitations, such as, proneness to human error for the former and the approximation of the mathematical expressions involved in the numerical algorithms for the latter. Formal methods have the capability to overcome these limitations and can provide an accurate analysis of these cell injection systems. Model checking, i.e., a state-based formal method, has been recently used for analyzing these systems. However, it involves the discretization of the differential equations capturing the continuous dynamics of the system and thus compromises on the completeness of the analysis of these safety-critical systems. In this paper, we propose a higher-order-logic theorem proving (a deductive-reasoning based formal method) based framework for analyzing the dynamical behavior of the robotic cell injection systems upto 4-DOF. The proposed analysis, based on the HOL Light theorem prover, enabled us to identify some discrepancies in the simulation and model checking based analysis of the same robotic cell injection system.

Keywords: Robotic cell injection systems; Formal verification; Theorem proving; Higher-order logic; HOL Light

Correspondence to: Adnan Rashid, E-mail: adnan.rashid@seecs.edu.pk

1. Introduction

Cell injection is a microbiological procedure that allows delivering small and precise amounts of substances, i.e., proteins, bio-molecules, genes and sperms, into the adherent or suspended cells. It is widely adopted in drug development [NFT+98], gene injection [KK04], in-vitro fertilization (IVF) [SN02] and intracytoplasmic sperm injection (ISCI) [YKY+99]. For example, it is used for the the treatment of infertility in IVF, by injecting the sperm into matured eggs. Similarly, drug development includes injecting drugs into a cell and observing its implications at the cellular level.

Robotic cell injection systems are designed to automatically perform the task of cell injection as opposed to the conventionally used manual and semi-automated injection methods, which involve time-consuming processes and require trained operators, and also have lower success rates. The most vital parameters of a robotic cell injection system are the coordinate frames, capturing the orientation and movement of its various components, such as, the injection manipulator, sensors, digital cameras and microscope, and the force controlling the injection pipette [HSM+09]. 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 excessive force may damage the membrane of the cell [HSML06] or an insufficient force may not be able to pierce the cell [FN16]. Thus the accuracy of the orientation and movement of these fundamental components and the injection force is pivotal for a reliable robotic cell injection system. Therefore, the robotic cell injection system designs need to be analyzed quite carefully to ensure that the final system exhibits all these requirements.

Robotic cell injection systems are generally categorized into three types, namely 2-DOF, 3-DOF and 4-DOF, based on the degree of freedoms (DOF) of the cell injection manipulator that is mounted on the motion stage and controls the motion of the injection pipette. For example, a 2-DOF robotic cell injection system involves the movement of the cell injection manipulator in a single plane (horizontal/vertical), 2-dimensions (x, y), only. Whereas, the 3-DOF systems adds an extra degree of freedom for the motion stage of the robotic cell injection system, allowing the translation of the system's components in the third dimension and thus contributes to the movement of components along z-axis in addition to the x and y axes. Similarly, 4-DOF system adds another degree of freedom for the motion stage by extending the coordinate frames from 3-dimensional (x, y, z) to 4-dimensional (x, y, z, θ) coordinates. Therefore, the fourth dimension provides the rotation of the system's components along θ in addition to their translation along the x, y and z axes.

The first step for analyzing a robotic cell injection system involves modeling the coordinate frames corresponding to the orientations of its various components, i.e, cameras, images and the injection manipulator. This model enables us to capture the movement and thus the positions of these components during the cell injection procedure. Moreover, the relationship between these coordinates provides the relative positions of these components, which is an essential part of a complete and successful cell injection procedure. Next, to carry out the process of the cell injection, the motion planning of the injection pipette is modeled using some force control algorithms, such as the contact-space-impedance force [SL97, HSM+09] and the image-based torque controllers [HSML06]. Contact-space-impedance force is a combination of impedance control and the vision-based injection force estimation and results in a hybrid impedance control method. This impedance control is responsible for the desired dynamics of a system, which are obtained by controlling the corresponding impedance to a desired value. Similarly, the image-based torque controller is mainly based on the input torques of the driving motors. These controllers capture the overall dynamics of the system and are mainly responsible for the successful process of cell injection and smooth functionality of the overall system.

Conventionally, robotic cell injection systems have been analyzed using paper-and-pencil proof methods. However, these manual analysis techniques are prone to human error and do not scale well to complex models. Moreover, in some cases, not all the necessary assumptions are recorded in the mathematical analysis, which may lead to an erroneous design and analysis. Similarly, the computer based numerical and simulations-based techniques have been utilized 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 the computational resources, the analysis is carried out for a certain number of test cases only and thus absolute accuracy cannot be obtained. Computer algebra systems, such as Maple [Map20] and Mathematica [Mat20], have also been used for the analysis of these systems [NS94]. However, the core of these systems contains unverified symbolic algorithms [DPV13], 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 result in an undetected error in the analysis that may in turn lead to disastrous consequences.

Formal methods [HT15] are computer-based mathematical analysis techniques that can overcome the above-mentioned inaccuracies. Primarily, these techniques involve the development of a mathematical model of the given system and verification of its properties using computer-based mathematical reasoning. There are mainly two types of formal methods, i.e., probabilistic model checking [CGP99, BDA95] and higher-order-logic theorem proving [Har09, Gor88] 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 et al. [SH17] for analyzing the robotic cell injection systems. Similarly, Ayub et al. [AH17] performed the probabilistic analysis of a virtual fixture control algorithm for a Al-Zahrawi surgical robot and formally verified the reachability, out-of-boundary problem and deadlock freedom properties using PRISM model checker. However, the formal models used in these model checking based works involve discretization of the differential equations that are used for modeling the dynamics of these systems, which compromises the accuracy of the corresponding analysis. Moreover, due to the inherent state-space explosion problem [CKNZ12], the formal model of the cell injection system cannot be completely analyzed in a model checker. Bresolin et al. [BGM+15] verified the control system properties of the autonomous robotic systems. The authors developed a hybrid automaton model of the system and performed the reachability analysis of the robotic surgery tasks using ARIADNE, which is based on Taylor expansion approximations for an efficient manipulation of functions in the Euclidean space. Kouskoulas et al. [KRPK13] formally analyzed a control algorithm for surgical robots and formally proved its safe operation for all possible inputs using KeYmaeraD, which is a theorem prover for analyzing distributed hybrid systems. The authors considered non-linear damping factors while analyzing the control algorithm. In the dynamical model of the underlying system, the matrix G represents the scaling factor and generally consists of non-linear (exponential) terms. However, the authors have taken it as constant in their proposed framework. Also, the corresponding differential equations are linear differential equations with variables coefficients. However, this analysis is based on the differential dynamic logic, and cannot capture all the continuous aspects of the given model in their true form. Higher-order-logic theorem proving [Har09] 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 verification of its properties using deductive reasoning. Given the high expressiveness of higher-order logic, it can completely capture the behavior of the differential equations, which is not achievable in the models used in probabilistic model checking based analysis. Rashid et al. [RH17, RH18] recently used this technique for the formal verification of the same robotic cell injection system. However, the focus of that work was limited to the 2-DOF robotic cell injection systems only.

In this paper, we propose a higher-order-logic theorem proving framework, depicted in Fig. 1, for formally analyzing the robotic cell injection systems [HSML06] upto 4-DOF using the HOL Light theorem prover [Har96b]. The main motivation for opting for HOL Light is the availability of foundational libraries of multivariate calculus [hol20a], real calculus [hol20b], vectors [hol20d] and matrices [hol20d], which are some of the foremost requirements to formally reason about the robotic cell injection systems. Moreover, in comparison to Kouskoulas et al.'s work [KRPK13], described above, our proposed framework is based on higher-order logic and thus caters for the continuous dynamics, modeled using the linear differential equations with constant coefficients, of the robotic cell injection system.

The major contributions of the paper are:

- Formalization of the robotic cell injection systems, force and torque controllers along with the differential equations based functional models, capturing the dynamical behavior (dynamics of the motion stage upto 4-DOF) of these systems and formal verification of their analytical solutions.
- Formalization of the coordinate frames, which involves the formal modeling of camera, image and stage coordinates and formal verification of their interrelationships.
- Formal verification of the motion planning of the injection pipette, which includes the formal modeling of the two-dimensional (2D) and three-dimensional (3D) contact-space-impedance force and the image-based torque controllers and formal verification of their relationships.
- Identification of the discrepancies in the simulation and model checking based analysis of these systems.

The source code of our HOL Light development is available for download at [Ras20] and thus can be used by other researchers and cell biologists interested in the design and verification of the robotic cell injection systems.

The rest of the paper is organized as follows: Sect. 2 presents an introduction to the HOL Light theorem prover, multivariate calculus theories of HOL Light and the robotic cell injection system.

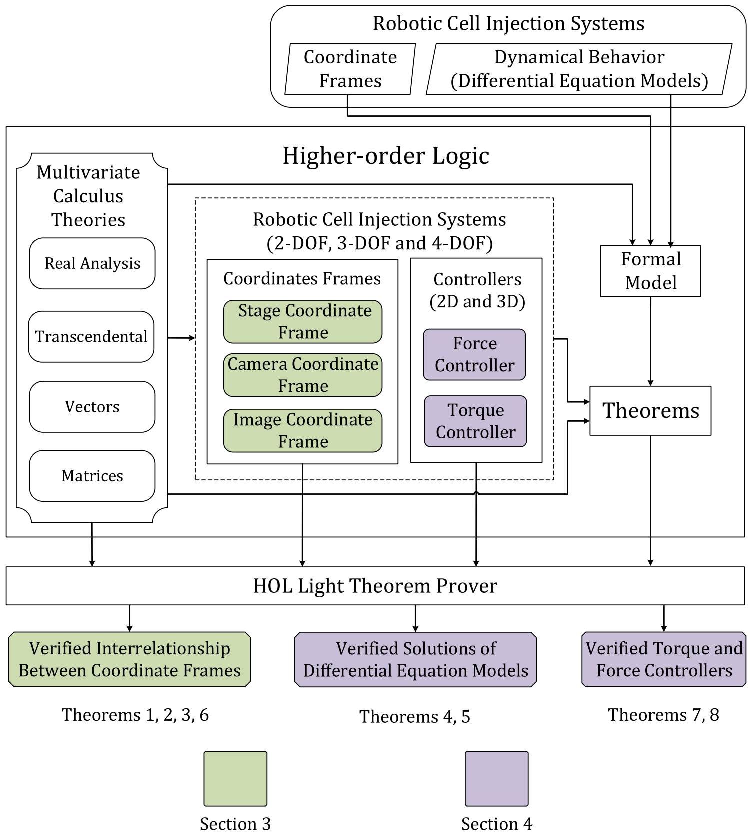

Fig. 1. Proposed framework

We provide the formalization of the robotic cell injection system in Sect. 3. Section 4 presents the formalization of motion planning of the injection pipette. This also includes the identification of the discrepancies in the simulation and model checking based analysis of the same system. Finally, Sect. 5 concludes the paper.

2. Preliminaries

This section presents an introduction to the HOL Light theorem prover, multivariate calculus theories of HOL Light and the robotic cell injection system.

2.1. HOL Light theorem prover

HOL Light [Har96a] is an interactive theorem proving environment for conducting proofs in higher-order logic. The logic in the HOL Light system is represented in the strongly-typed functional programming language ML [Pau96]. A theorem is a formalized statement that may be an axiom or could be deduced from already verified theorems by an inference rule. A theorem consists of a finite set Ω of Boolean terms, called the assumptions, and a Boolean term S, called the conclusion. Soundness is assured as every new theorem must be verified by applying the basic axioms and primitive inference rules or any other previously verified theorems/inference rules. A HOL Light theory is a collection of valid HOL Light types, constants, axioms, definitions, and theorems. Various mathematical foundational concepts have been formalized and saved as HOL Light theories.

Table 1. HOL Light symbols

HOL Light symbols	Standard symbols	Meanings
`/\`	and	Logical *and*
`\/`	or	Logical *or*
`~`	not	Logical *negation*
`==>`	$\longrightarrow$	Implication
`<=>`	=	Equivalence in Boolean domain
`!x.t`	$\forall\, x.t$	For all $x : t$
`?x.t`	$\exists\, x.t$	There exists $x : t$
`λx.t`	$\lambda\, x.t$	Function that maps x to $t(x)$
`num`	$\{0, 1, 2, \ldots\}$	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 Naturals numbers to Reals
`Cx(a)`	$\mathbb{R} \rightarrow \mathbb{C}$	Typecasting from Reals to Complex
`%`	$*$	Scalar multiplication of a vector or matrix
`**`	$*$	Matrix-vector multiplication
`@f`	Hilbert choice operator	Returns f if it exists
`atreal x`	Real Net	At real variable x

The HOL Light theorem prover provides an extensive support of theorems regarding Boolean algebra, arithmetic, real numbers, transcendental functions and multivariate analysis such as differential, integration, vectors and topology, in the form of theories. In fact, one of the primary reasons to choose the HOL Light theorem prover for the proposed formalization was the presence of an extensive support of higher-order-logic theories of multivariable calculus, which are quite dense amongst all higher-order-logic theorem provers and have been extensively used in the formal verification of the robotic cell injection. In this paper, we only incorporate the linear dynamics of the robotic cell injection system. However, one of our future directions is to formally verify the robotic cell injection systems exhibiting the non-linear dynamics and to verify their non-analytical/numerical solutions, which are based on the interval arithmetic, providing the reliable and guaranteed solutions of their differential equations based dynamical models. Therefore, we believe that higher-order logic is the right choice for the formalization of the interval arithmetic and the verification of robotic cell injection systems exhibiting the non-linear dynamics.

Table 1 presents the standard and HOL Light representations and the meanings of some commonly used symbols in this paper.

2.2. Multivariable calculus theories in HOL Light

A N-dimensional vector is represented as an $\mathbb{R}^N$ column matrix with each of its elements as a real number in HOL Light [Har13]. All of the vector operations are thus performed using matrix manipulations. A complex number is defined as a 2-dimensional vector, i.e., an $\mathbb{R}^2$ column matrix or the data-type $\mathbb{C}$, in HOL Light. Similarly, all of the multivariable calculus theorems are verified in HOL Light for functions with an arbitrary data-type $\mathbb{R}^N \rightarrow \mathbb{R}^M$.

Some of the frequently used HOL Light functions in the reported formalization are explained below:

Definition 2.1 *Vector (HOL Light Library, vectors.ml [hol20d])*
$\vdash_{def}$ ∀l. vector l = (lambda i. EL (i - 1) l)

The function vector accepts an arbitrary list l : α `list` and returns a vector having each component of data-type α [hol20d]. It uses the function EL i l, which return the i^{th} element of a list l. Here, the lambda operator in HOL Light is used for constructing a vector from its components [Har13].

Definition 2.2 *Real Cosine and Sine Functions (HOL Light Library, transcendentals.ml [hol20c])*
$\vdash_{def}$ ∀x. sin x = Re (csin (Cx x))
$\vdash_{def}$ ∀x. cos x = Re (ccos (Cx x))

The HOL Light functions cos : $\mathbb{R} \rightarrow \mathbb{R}$ and sin : $\mathbb{R} \rightarrow \mathbb{R}$ represent the real cosine and real sine [hol20a], respectively. These functions are formally defined using the complex cosine ccos : $\mathbb{C} \rightarrow \mathbb{C}$ and complex sine csin : $\mathbb{C} \rightarrow \mathbb{C}$ functions, respectively [hol20c].

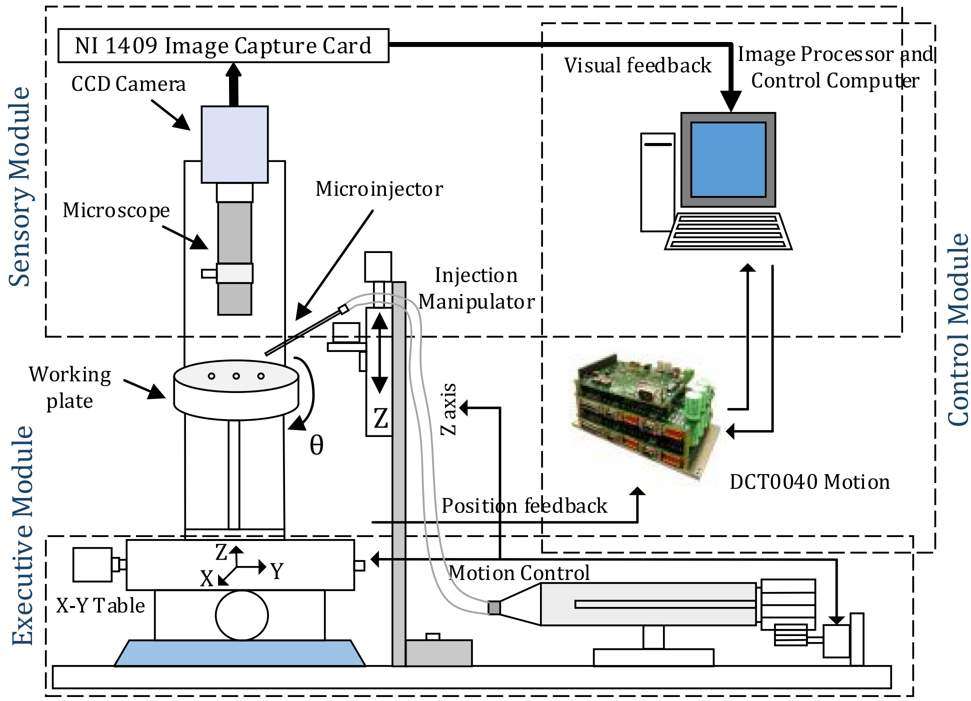

Fig. 2. Robotic cell injection systems

Definition 2.3 *Real Derivative (HOL Light Library, realanalysis.ml [hol20b])*
$\vdash_{def}$ ∀f x. real_derivative f x = (@f′. (f has_real_derivative f′) (atreal x))

The function real_derivative accepts a function f : $\mathbb{R} \rightarrow \mathbb{R}$ and a real number x, which is the point at which f has to be differentiated, and returns a variable of data-type $\mathbb{R}$, providing the differential of f at x. The function has_real_derivative defines the same relationship in the relational form [hol20b].

We utilize the above-mentioned fundamental functions of multivariable calculus for formally analyzing the robotic cell injection system in Sects. 3 and 4 of the paper.

2.3. Robotic cell injection systems

A robotic cell injection system mainly consists of three modules, namely executive, sensory and control modules as depicted in Fig. 2. The executive module consists of positioning table, working plate and 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. 2.

The sensory module consists of a vision system that has four parts, namely optical microscope, charged coupled device (CCD) camera, peripheral component interconnect (PCI) image capture and a processing card. The CCD camera is used to capture the cell injection process using a PCI image capture. The control module contains a host computer and a DCT0040 motion control system. Figure 3 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, $o_c - x_c y_c z_c$ is the camera coordinate frame with o_c representing the center of the microscope. The coordinate frame in image plane is represented as $o_i - uv$, where o_i is the origin and the axis uv is perpendicular to the optical axis.

The correct orientation, relative position and movement of various components, i.e., injection manipulator, camera and microscope, and the contact-space-impedance force and the image-based torque controllers, providing the motion planning of the injection pipette, are the foremost requirements for an efficient design and a reliable functionality of a robotic cell injection.

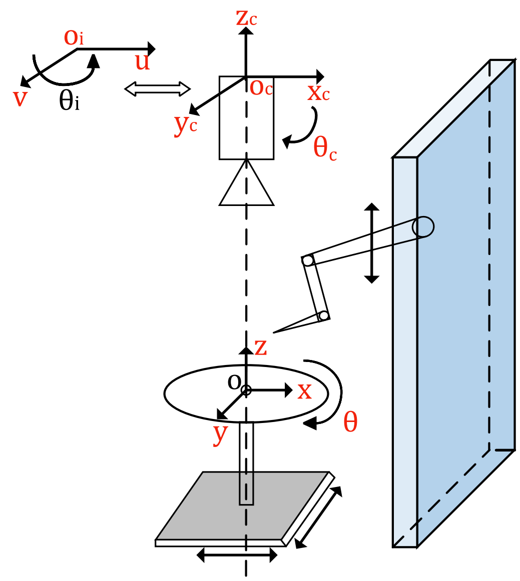

Fig. 3. Configuration of the robotic cell injection systems

We provide the formalization of coordinate frames, capturing the orientation and relative position of its various components (Theorems 3.1, 3.2 and 3.3) and motion planning of the injection pipette (Theorems 4.1, 4.3, 4.4 and 4.5) in Sects. 3 and 4 of the paper, respectively.

3. Formalization of robotic cell injection system

We provide the higher-order-logic based formalization of the robotic cell injection system using standard mathematical notations rather than the pure HOL Light notations to facilitate the understanding of the paper for a non-HOL user. The readers interested in viewing the exact HOL Light formalization can obtain the source code for our formalization from [Ras20].

In our formalization, we model robotic cell injection system based on different degree of freedoms, i.e., 2-DOF, 3-DOF and 4-DOF as enumerated type definition in HOL Light as follows:

Definition 3.1 *Robotic Cell Injection System*

```
define_type "robotic_cis = TwoDOF |
                           ThreeDOF |
                           FourDOF"
```

Next, in order to capture the orientation and movement of various components of the robotic cell injection system, we need to model the coordinate systems. In this regard, we require modeling a point, which captures the position and orientation of a component in a coordinate system. Therefore, we use the type abbreviation in HOL Light to define new types for various points as follows:

Definition 3.2 *Points of a Coordinate System*

```
new_type_abbrev ("one_dim_point",‘:(ℝ → ℝ)‘)
new_type_abbrev ("timed_one_dim_point",‘:(one_dim_point × ℝ)‘)
new_type_abbrev ("two_dim_point",‘:(one_dim_point × one_dim_point)‘)
new_type_abbrev ("timed_two_dim_point",‘:(two_dim_point × ℝ)‘)
new_type_abbrev ("three_dim_point",‘:(one_dim_point × two_dim_point)‘)
new_type_abbrev ("timed_three_dim_point",‘:(three_dim_point × ℝ)‘)
new_type_abbrev ("four_dim_point",‘:(one_dim_point × three_dim_point)‘)
new_type_abbrev ("timed_four_dim_point",‘:(four_dim_point × ℝ)‘)
```

Table 2. Data types for 2-DOF system parameters

Parameter description	Standard symbol	HOL Light Symbol:Type
Angle between two frames	α	α:$\mathbb{R}$
Distance between two frames in x direction	d_x	dx:$\mathbb{R}$
Distance between two frames in y direction	d_y	dy:$\mathbb{R}$
Display resolution of the vision system in x direction	f_x	fx:$\mathbb{R}$
Display resolution of the vision system in y direction	f_y	fy:$\mathbb{R}$
Mass of x positioning table	m_x	mx:$\mathbb{R}$
Mass of y positioning table	m_y	my:$\mathbb{R}$
Mass of the working plate	m_p	mp:$\mathbb{R}$

The type one_dim_point provides a point in one-dimensional coordinate system, represented by its location as a function of time, and thus captures the orientation and position of various components of the system. Similarly, the type timed_one_dim_point is a pair, with its first element as one_dim_point providing the location of a component of the underlying system that changes with time, whereas its second element models the time. The camera, stage and image coordinates for a 2-DOF robotic cell injection system are two-dimensional coordinates, which are modeled in HOL Light as:

Definition 3.3 *Two-dimensional Coordinates*

$\vdash_{def}$ ∀x y t. two_dim_coord (((x,y),t):timed_two_dim_point) = $\begin{bmatrix} \texttt{x(t)} \\ \texttt{y(t)} \end{bmatrix}$

The function two_dim_coord accepts a variable of data-type *timed_two_dim_point* and returns a two-dimensional vector describing the corresponding coordinates. Now, the two-dimensional motion stage coordinates are modeled in HOL Light as follows:

Definition 3.4 *Two-dimensional Motion Stage Coordinates*
$\vdash_{def}$ ∀x y t.

two_dim_motion_stage_coord TwoDOF ((x,y),t) = two_dim_coord ((x,y),t)

We model various parameters of the 2-DOF robotic cell injection systems as a tuple (α,*dx*,*dy*,*fx*,*fy*,*mx*,*my*,*mp*), where the description and the type of each parameter is given in Table 2. These parameters characterize various physical aspects of the robotic cell injection systems (e.g., the angle between two frames α and the mass of the working plate *mp*). We formalize the parameters tuple as type abbreviations:

Definition 3.5 *2-DOF System Parameters*
new_type_abbrev ("twodof_sys_par",‘:(α × dx × dy × fx × fy × mx × my × mp)‘)

The verification of the relationship between camera, image and stage coordinates provides key information for the reliable operation of the robotic cell injection system and thus, ensures the accuracy of the orientation and position of its various components, i.e., stage frame, injection manipulator, camera and microscope, by providing the accurate translation and rotation of the corresponding coordinate frames. Firstly, to verify the relationship between the stage and camera coordinates, we model the rotation matrix from the stage coordinate frame (o-xyz) to the camera coordinate frame ($o_c - x_c y_c z_c$), and the two-dimensional displacement vector between the origins of both these frames:

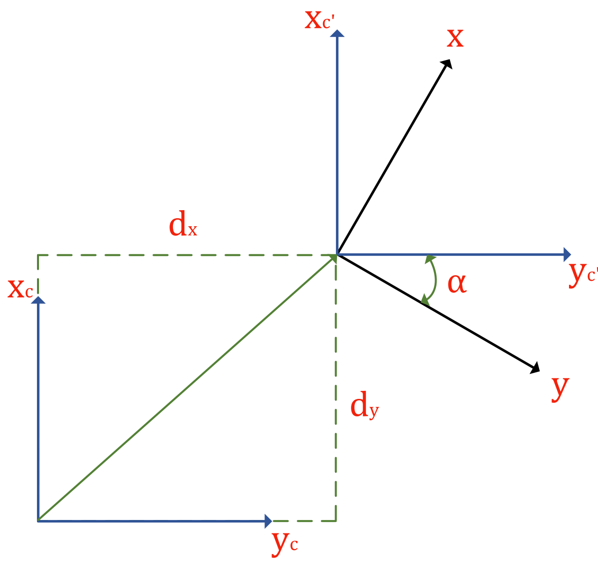

Fig. 4. Transformation from camera to stage coordinate frames

Definition 3.6 *Rotation Matrix and Displacement Vector*

$\vdash_{def}$ ∀dx dy fx fy mx my mp α.

rotation_matrix_2dim (α,dx,dy,fx,fy,mx,my,mp) = $\begin{bmatrix} \cos\ \alpha & \sin\ \alpha \\ -\sin\ \alpha & \cos\ \alpha \end{bmatrix}$

$\vdash_{def}$ ∀α fx fy mx my mp dx dy. disp_vec_2dim (α,dx,dy,fx,fy,mx,my,mp) = $\begin{bmatrix} \mathrm{dx} \\ \mathrm{dy} \end{bmatrix}$

Now, the relationship between the camera and stage coordinate frames, as depicted in Fig. 4, is formalized in HOL Light as:

Definition 3.7 *Relationship Between Camera and Stage Coordinates*

$\vdash_{def}$ ∀xc yc x y t α dx dy fx fy mx my mp.

relat_camera_stage_coord_frame_2dim (xc,yc) (x,y) (α,dx,dy,fx,fy,mx,my,mp) t ⇔
two_dim_coord ((xc,yc),t) = rotation_matrix_2dim (α,dx,dy,fx,fy,mx,my,mp) $**$
two_dim_motion_stage TwoDOF ((x,y),t) + disp_vec_2dim (α,dx,dy,fx,fy,mx,my,mp)

where $**$ models the matrix-vector multiplication in HOL Light. Next, in order to ensure that the given parameters for the camera-stage interrelationship indeed represent a valid relationship, we need to formalize the associated constraints on the distances between two frames in the x and y directions, i.e., dx and dy. Since the distance is always a non-negative quantity. Therefore, these distances should be greater than or equal to zero, i.e., dx ≥ 0, dy ≥ 0. However, the frames would always be displaced during the process of the robotic cell injection, providing dx $\neq 0$ and dy $\neq 0$. Moreover, they appear in the denominator of some of the expressions (intermediate steps) during the verification of the relationships between the coordinate frames and would result in an indeterminate form, if they are equal to zero. We formalize these constraints as a predicate `is_valid_camera_stage_relat_2dim` (Definition 3.8) asserting the positivity of distances between these two frames:

Definition 3.8 *Valid Camera and Stage Coordinates Interrelationship*

$\vdash_{def}$ ∀α fx fy mx my mp dx dy.

is_valid_camera_stage_relat_2dim (α,dx,dy,fx,fy,mx,my,mp) ⇔ &0 $<$ dx $\wedge$ &0 $<$ dy

Now, we verify the relationship between camera and stage coordinate frames, which ensures the correct orientation and the relative position of the injection manipulator and cameras:

Theorem 3.1 *Relationship Between Camera and Stage Coordinates*
$\vdash_{thm}$ ∀xc yc x y α dx dy fx fy mx my mp t.
let sp = ((α,dx,dy,fx,fy,mx,my,mp):twodof_sys_par) in
cc = ((xc,yc):two_dim_point) and
sc = ((x,y):two_dim_point) in
is_valid_camera_stage_relat_2dim sp
⇒ (relat_camera_stage_coord_frame_2dim cc sc sp t ⇔

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

The above theorem is verified using the properties of vectors (*vectors.ml*, [hol20d]) and matrices (*vectors.ml*, [hol20d]) along with some real arithmetic reasoning. Next, to verify the relationship between image and camera coordinate frames, we first model the display resolution matrix as follows:

Definition 3.9 *Display Resolution Matrix*

$\vdash_{def}$ ∀α dx dy mx my mp fx fy. display_resol_matrix_2dim (α,dx,dy,fx,fy,mx,my,mp) = $\begin{bmatrix} \texttt{fx} & 0 \\ 0 & \texttt{fy} \end{bmatrix}$

The positivity of the display resolutions of the vision system provides a valid relationship between image and camera coordinate frames and is formalized in HOL Light as:

Definition 3.10 *Valid Image and Camera Coordinates Interrelationship*
$\vdash_{def}$ ∀α dx dy mx my mp fx fy.
is_valid_image_camera_relat_2dim (α,dx,dy,fx,fy,mx,my,mp) ⇔ &0 < fx ∧ &0 < fy

Now, the image-camera coordinate frames interrelationship for the 2-DOF robotic cell injection system, providing the correct orientation and the relative position of cameras and images, is verified as:

Theorem 3.2 *Relationship Between Image and Camera Coordinates*
$\vdash_{thm}$ ∀xc yc u v t dx dy fx fy mx my mp α.
let sp = ((α,dx,dy,fx,fy,mx,my,mp):twodof_sys_par) in
cc = ((xc,yc):two_dim_point) and
ic = ((u,v):two_dim_point) in
is_valid_image_camera_relat_2dim sp
⇒ (relat_image_camera_coord_frame_2dim cc ic sp t ⇔

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

where the HOL Light function relat_image_camera_coord_frame_2dim is modeled in a similar way to relat_camera _stage_coord_frame_2dim and provides the relationship between the image and the camera coordinates. Next, we model the transformation matrix between image and stage coordinate frames, which is used in the verification of their interrelationship and is given as follows:

Definition 3.11 *Transformation Matrix*
$\vdash_{def}$ ∀dx dy mx my mp fx fy α.

transformation_matrix_2dim (α,dx,dy,fx,fy,mx,my,mp) = $\begin{bmatrix} \texttt{fx} * \cos\alpha & \texttt{fx} * \sin\alpha \\ -\texttt{fy} * \sin\alpha & \texttt{fy} * \cos\alpha \end{bmatrix}$

We model constraints for the validity of the image-stage coordinate frames, i.e., positivity of its various parameters as:

Definition 3.12 *Valid Image and Stage Coordinate Interrelationship*
$\vdash_{def}$ ∀α mx my mp dx dy fx fy.
is_valid_image_stage_relat_2dim (α,dx,dy,fx,fy,mx,my,mp) ⇔
&0 < dx ∧ &0 < dy ∧ &0 < fx ∧ &0 < fy

Now, we verify an important relationship between the image and stage coordinates, asserting the correct orientation and the relative position of the injection manipulator and images, as the following HOL Light theorem:

Theorem 3.3 *Relationship Between Image and Stage Coordinates*

$\vdash_{thm}$ ∀x y u v t fx fy dx dy mx my mp α xc yc.
  let sp = ((α,dx,dy,fx,fy,mx,my,mp):twodof_sys_par) and
    cc = ((xc,yc):two_dim_point) and
    sc = ((x,y):two_dim_point) and
    ic = ((u,v):two_dim_point) in
    [A1:] is_valid_image_stage_relat_2dim sp ∧
    [A2:] relat_image_camera_coord_frame_2dim cc ic sp t ∧
    [A3:] relat_camera_stage_coord_frame_2dim cc sc sp t
      ⇒ twod_coord (ic,t) = transformation_matrix_2dim sp $**$

$$\text{two\_dim\_motion\_stage TwoDOF (sc,t)} + \begin{bmatrix} \texttt{fx} * \texttt{dx} \\ \texttt{fy} * \texttt{dy} \end{bmatrix}$$

Assumption A1 provides the validity of the relationship between stage and image coordinate frames (Definition 3.12). Similarly, Assumptions A2–A3 present the relationships between the image and camera, and camera and stage (Definition 3.7) coordinate frames, respectively. The verification of Theorem 3.3 is mainly based on Theorems 3.1 and 3.2, and some classical properties of the vectors (*vectors.ml* [hol20d]) and matrices (*vectors.ml* [hol20d]). The verification of these relationships raises our confidence level about the orientation and position of the vital components of a 2-DOF robotic cell injection system, i.e., working plate, camera, microscope and injection manipulator. Similarly, we formally verify the relationships between the camera, stage and image coordinate frames for 3-DOF and 4-DOF robotic cell injection systems and the verified theorems regarding these relationships for 3-DOF and 4-DOF systems are presented in Tables 3 and 4, respectively. These theorems have been included in the paper to make it self-contained especially for the non-users of HOL. More experienced readers may skim through these details.

The transformation from 2-DOF to 3-DOF extends the corresponding coordinate frames from 2-dimensional (x, y) to 3-dimensional (x, y, z) coordinates and thus adds an extra degree of freedom for the motion stage of the robotic cell injection system. This allows the translation of the system's components in the third dimension and thus contributes to the movement of components along z-axis in addition to the x and y axes. Similarly, moving to 4-DOF system adds another degree of freedom for the motion stage by extending the coordinate frames from 3-dimensional (x, y, z) to 4-dimensional (x, y, z, θ) coordinates and thus provides the rotation of the system's components along θ in addition to their translation along the x, y and z axes. This completes our verification of various coordinate frames interrelationships, capturing the position and movement of various components, for the robotic cell injection systems up to 4-DOF. More details about their verification can be found at [Ras20].

Table 3: Coordinate frames interrelationships for 3-DOF system

Relationship Between Camera and Stage Coordinates
$\vdash_{thm}$ ∀xc yc zc x y z α dx dy dz fx fy mx my mz mp t. let sp = ((α,dx,dy,dz,fx,fy,mx,my,mz,mp):threedof_sys_par) and sc = ((x,y,z):three_dim_point) and cc = ((xc,yc,zc):three_dim_point) in is_valid_camera_stage_relat_3dim sp $\Rightarrow \Big($ relat_camera_stage_coord_frame_3dim cc sc sp t $\Leftrightarrow$ $\begin{bmatrix} \texttt{xc(t)} \\ \texttt{yc(t)} \\ \texttt{zc(t)} \end{bmatrix} = \begin{bmatrix} \texttt{x(t) * cos } \alpha \texttt{ + y(t) * sin } \alpha \texttt{ + dx} \\ \texttt{- x(t) * sin } \alpha \texttt{ + y(t) * cos } \alpha \texttt{ + dy} \\ \texttt{z(t) + dz} \end{bmatrix} \Big)$
Relationship Between Image and Camera Coordinates
$\vdash_{thm}$ ∀xc yc zc u v z t dx dy dz fx fy mx my mz mp α. let sp = ((α,dx,dy,dz,fx,fy,mx,my,mz,mp):threedof_sys_par) and cc = ((xc,yc,zc):three_dim_point) and ic = ((u,v,z):three_dim_point) in is_valid_image_camera_relat_3dim sp $\Rightarrow \Big($ relat_image_camera_coord_frame_3dim cc ic sp t $\Leftrightarrow$ $\begin{bmatrix} \texttt{u(t)} \\ \texttt{v(t)} \\ \texttt{z(t)} \end{bmatrix} = \begin{bmatrix} \texttt{fx * xc(t)} \\ \texttt{fy * yc(t)} \\ \texttt{zc(t) - dz} \end{bmatrix} \Big)$
Relationship Between Image and Stage Coordinates
$\vdash_{thm}$ ∀x y u v z t fx fy dx dy dz mx my mz mp α xc yc zc. let sp = ((α,dx,dy,dz,fx,fy,mx,my,mz,mp):threedof_sys_par) and cc = ((xc,yc,zc):three_dim_point) and sc = ((x,y,z):three_dim_point) and ic = ((u,v,z):three_dim_point) in [A1:] is_valid_image_stage_relat_3dim sp ∧ [A2:] relat_image_camera_coord_frame_3dim cc ic sp t ∧ [A3:] relat_camera_stage_coord_frame_3dim cc sc sp t $\Rightarrow \Big($ three_dim_coord (ic,t) = transformation_matrix_3dim sp ∗∗ three_dim_motion_stage ThreeDOF (sc,t) + $\begin{bmatrix} \texttt{fx * dx} \\ \texttt{fy * dy} \\ \texttt{\&0} \end{bmatrix} \Big)$

Table 4: Coordinate frames interrelationships for 4-DOF system

Relationship Between Camera and Stage Coordinates
$\vdash_{thm}$ ∀xc yc zc thetac x y z theta α dx dy dz fx fy mx my mz mp lp t. let sp = ((α,dx,dy,dz,fx,fy,mx,my,mz,mp,lp):fourdof_sys_par) and cc = ((xc,yc,zc,thetac):four_dim_point) and sc = ((x,y,z,theta):four_dim_point) in is_valid_camera_stage_relat_4dim sp ⇒ (relat_camera_stage_coord_frame_4dim cc sc sp t ⇔ $\begin{bmatrix} \texttt{xc(t)} \\ \texttt{yc(t)} \\ \texttt{zc(t)} \\ \texttt{thetac(t)} \end{bmatrix} = \begin{bmatrix} \texttt{x(t)} * \texttt{cos}\ \alpha + \texttt{y(t)} * \texttt{sin}\ \alpha + \texttt{dx} \\ - \texttt{x(t)} * \texttt{sin}\ \alpha + \texttt{y(t)} * \texttt{cos}\ \alpha + \texttt{dy} \\ \texttt{z(t) + dz} \\ \texttt{theta(t)} \end{bmatrix}$)
Relationship Between Image and Camera Coordinates
$\vdash_{thm}$ ∀xc yc zc thetac u v z thetai t dx dy dz fx fy mx my mz mp lp α. let sp = ((α,dx,dy,dz,fx,fy,mx,my,mz,mp,lp):fourdof_sys_par) and cc = ((xc,yc,zc,thetac):four_dim_point) and ic = ((u,v,z,thetai):four_dim_point) in is_valid_image_camera_relat_4dim sp ⇒ (relat_image_camera_coord_frame_4dim cc ic sp t ⇔ $\begin{bmatrix} \texttt{u(t)} \\ \texttt{v(t)} \\ \texttt{z(t)} \\ \texttt{thetai(t)} \end{bmatrix} = \begin{bmatrix} \texttt{fx} * \texttt{xc(t)} \\ \texttt{fy} * \texttt{yc(t)} \\ \texttt{zc(t) - dz} \\ \texttt{thetac(t)} \end{bmatrix}$)
Relationship Between Image and Stage Coordinates
$\vdash_{thm}$ ∀x y z theta u v thetai t fx fy dx dy dz mx my mz mp lp α xc yc zc thetac. let sp = ((α,dx,dy,dz,fx,fy,mx,my,mz,mp,lp):fourdof_sys_par) and cc = ((xc,yc,zc,thetac):four_dim_point) and sc = ((x,y,z,theta):four_dim_point) and ic = ((u,v,z,thetai):four_dim_point) in [A1:] is_valid_image_stage_relat_4dim sp ∧ [A2:] relat_image_camera_coord_frame_4dim cc ic sp t ∧ [A3:] relat_camera_stage_coord_frame_4dim cc sc sp t ⇒ (four_dim_coord (ic,t) = transformation_matrix_4dim sp ** four_dim_motion_stage FourDOF (sc,t) + $\begin{bmatrix} \texttt{fx} * \texttt{dx} \\ \texttt{fy} * \texttt{dy} \\ \texttt{\&0} \\ \texttt{\&0} \end{bmatrix}$)

Table 5. Data types for two-dimensional controllers parameters

Parameter description	Standard Symbol	HOL Light Symbol:Type
x component of the torque input to the driving motors	τ_x	taux:$\mathbb{R} \to \mathbb{R}$
y component of the torque input to the driving motors	τ_y	tauy:$\mathbb{R} \to \mathbb{R}$
x component of the desired force applied to actuators	${f_{ex}}^d$	fexd:$\mathbb{R} \to \mathbb{R}$
y component of the desired force applied to actuators	${f_{ey}}^d$	feyd:$\mathbb{R} \to \mathbb{R}$
x component of the actual force applied to actuators	f_{ex}	fex:$\mathbb{R} \to \mathbb{R}$
y component of the actual force applied to actuators	f_{ey}	fey:$\mathbb{R} \to \mathbb{R}$
Mass coefficient of impedance control	m	m:$\mathbb{R}$
Damping coefficient of impedance control	b	b:$\mathbb{R}$
Damping coefficient of impedance control	k	k:$\mathbb{R}$

4. Formalization of the motion planning of the injection pipette

In this section, we present the formalization of the dynamical behavior of the robotic cell injection systems upto 4-DOF, based on differential equations and the verification of their solutions. We also formalize the contact-space-impedance force and image based torque controllers (2D and 3D controllers) and formally verify the relationship between these controllers. First, we model various parameters of these controllers as a tuple (*taux,tauy,fexd, feyd,fex,fey,m,b,k*), where the description and the type of each parameter is given in Table 5. These parameters characterize various physical aspects of these controllers (e.g., x component of the torque input to the driving motors *taux* etc.). We formalize the parameters tuple in HOL Light as type abbreviations:

Definition 4.1 *Two-dimensional Controller Parameters*
new_type_abbrev ("two_dim_cont_par",':(taux × tauy × fexd × feyd × fex × fey × m × b × k)')

Now, the dynamics of the 2-DOF motion stage based robotic cell injection system is mathematically expressed as [HSML06]:

$$\begin{bmatrix} m_x + m_y + m_p & 0 \\ 0 & m_y + m_p \end{bmatrix} \begin{bmatrix} \dfrac{d^2x}{dt} \\ \dfrac{d^2y}{dt} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dfrac{dx}{dt} \\ \dfrac{dy}{dt} \end{bmatrix} = \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix} - \begin{bmatrix} {fex}^d \\ {fey}^d \end{bmatrix} \tag{1}$$

We formalize Eq. (1) as the following HOL Light function:

Definition 4.2 *Dynamics of the 2-DOF Motion Stage*
$\vdash_{def}$ ∀α dx dy fx fy mx my mp x y t taux tauy fexd feyd fex fey m b k.
dynamics_motion_stage TwoDOF ((x,y),t) (α,dx,dy,fx,fy,mx,my,mp) (taux,tauy,fexd,feyd,fex,fey,m,b,k) ⇔
mass_mat_2dim (α,dx,dy,fx,fy,mx,my,mp) ** second_order_deriv_stage_coord_2dim ((x,y),t) +
posit_table_mat_2dim ** first_order_deriv_stage_coord_2dim ((x,y),t) =
torque_vec_2dim (taux,tauy,fexd,feyd,fex,fey,m,b,k) -
desired_force_vec_2dim (taux,tauy,fexd,feyd,fex,fey,m,b,k)

where mass_mat_2dim is the matrix containing the respective masses. Similarly, posit_ table_mat_2dim is the diagonal matrix. It is generally taken as a matrix with arbitrary diagonal entries, providing the dynamics of the 2-DOF motion stage as a set of second-order non-linear differential equations and thus requires the point load model for the corresponding analysis [HSML06]. However, we have simplified the model by taking the diagonal matrix as an identity matrix, i.e., we take the diagonal entries as 1 rather than taking them as arbitrary values. Therefore, it results into a second-order linear homogeneous differential equation. Similarly, torque_vec_2dim and desired_ force_vec_2dim are the two-dimensional vectors with their elements representing the components of the applied torque and desired force. The HOL Light functions first_order_deriv_stage_coord_2dim and second_order_deriv_stage_coord_2dim model the two-dimensional vectors, which are first-order and second-order derivatives of the stage coordinates, respectively.

Definition 4.3 *First and Second-order Derivative Vectors*
$\vdash_{def}$ ∀x y t. first_order_deriv_stage_coord ((x,y),t) = deriv_vec_first [x; y] t
$\vdash_{def}$ ∀x y t. second_order_deriv_stage_coord ((x,y),t) = deriv_vec_second [x; y] t

where deriv_vec_first and deriv_vec_second accept a list containing the functions of data-type $\mathbb{R} \rightarrow \mathbb{R}$ and return the corresponding first and second-order derivative vectors [Ras20].

If the applied torque and force vectors are zero, then the injection pipette does not touch the cells. This describes a scenario, when the process of the robotic cell injection has not been started. Thus, Eq. (1) can be transformed for this particular scenario as follows:

$$\begin{bmatrix} m_x + m_y + m_p & 0 \\ 0 & m_y + m_p \end{bmatrix} \begin{bmatrix} \dfrac{d^2x}{dt} \\ \dfrac{d^2y}{dt} \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \dfrac{dx}{dt} \\ \dfrac{dy}{dt} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2}$$

To verify the solution of the above equation (will be verified as Theorem 4.1), we first model the constraint on masses, i.e., positivity of m_x, m_y and m_p, as the following predicate:

Definition 4.4 *Valid Masses*
$\vdash_{def}$ ∀dx dy fx fy α mx my mp.
valid_masses_2dim (α,dx,dy,fx,fy,mx,my,mp) = &0 < mx ∧ &0 < my ∧ &0 < mp

Similarly, the initial conditions for the x and y components of the stage coordinate frames and their derivatives in HOL Light are modeled as follows:

Definition 4.5 *Initial Conditions*
$\vdash_{def}$ ∀x y t. initial_conditions_2dim ((x,y),t) = x(0) = x0 ∧ y(0) = y0 ∧ $\frac{\texttt{dx}}{\texttt{dt}}$(0)= xd0 ∧ $\frac{\texttt{dy}}{\texttt{dt}}$(0)= yd0

Now, we verify the solution of Eq. (2), providing the dynamics of 2-DOF motion stage, as the following HOL Light theorem:

Theorem 4.1 *Solution of Dynamical Behavior of 2-DOF Motion Stage*
$\vdash_{thm}$ ∀x y mx my mp dx dy fx fy taux tauy fex fey fexd feyd α b k m t x0 y0 xd0 yd0.
let sp = ((α,dx,dy,fx,fy,mx,my,mp):twodof_sys_par) and
cp = ((taux,tauy,fexd,feyd,fex,fey,m,b,k):two_dim_cont_par) and
sc = ((x,y):two_dim_point) in
[A1:] valid_masses_2dim sp ∧
[A2:] initial_conditions_2dim (sc,t) x0 y0 xd0 yd0 ∧
[A3:] $\begin{bmatrix} \texttt{taux t} \\ \texttt{tauy t} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ ∧ [A4:] $\begin{bmatrix} \texttt{fexd t} \\ \texttt{feyd t} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ ∧
[A5:] (∀t. 0 ≤ t ⇒ x(t) = (x0 + xd0 ∗ (mx + my + mp)) - xd0 ∗ (mx + my + mp) ∗ $e^{\frac{-1}{\texttt{mx+my+mp}}\texttt{t}}$) ∧
[A6:] (∀t. 0 ≤ t ⇒ y(t) = (y0 + yd0 ∗ (my + mp)) - yd0 ∗ (my + mp) ∗ $e^{\frac{-1}{\texttt{my+mp}}\texttt{t}}$)
⇒ dynamics_motion_stage TwoDOF (sc,t) sp cp

Assumption A1 provides the validity of the masses mx, my and mp (Definition 4.4). Similarly, Assumption A2 presents the initial conditions for x and y components of the stage coordinate frame (Definition 4.5). Assumptions A3-A4 model the condition that the two-dimensional torque and force vectors are zero. Assumptions A5-A6 provide the values of xy coordinates at any time t. Finally, the conclusion represents the dynamics of the robotic cell injection system having 2-DOF motion stage, which is composed of the working plate holding the cells, placed on the positioning table, as shown in Figure 2. The verification of Theorem 4.1 is based on the properties of real derivatives (*realanalysis.ml* [hol20b]), transcendental functions (*transcendentals.ml* [hol20c]), vectors (*vectors.ml* [hol20d]) and matrices (*vectors.ml* [hol20d]) along with some real arithmetic reasoning. Similarly, we formally verify the solutions of the dynamics of the 3-DOF and 4-DOF robotic cell injection systems and the details about their verification can be found at [Ras20]. Next, we verify the solution of Eq. (1), by considering the applied torque and the desired force vectors exhibiting the behavior of the exponential function, as the following HOL Light theorem:

Theorem 4.2 *Solution of Dynamical Behavior of* 2*-DOF Motion Stage*

$\vdash_{thm}$ $\forall$x y mx my mp dx dy fx fy taux tauy fex fey fexd feyd α b k m t x0 y0 xd0 yd0 ax ay bx by.
let sp = ((α,dx,dy,fx,fy,mx,my,mp):twodof_sys_par) and
cp = ((taux,tauy,fexd,feyd,fex,fey,m,b,k):two_dim_cont_par) and
sc = ((x,y):two_dim_point) in
[A1:] physical_constraints_2dim sp ax ay bx by $\wedge$
[A2:] initial_conditions_2dim (sc,t) x0 y0 xd0 yd0 $\wedge$

[A3:] $\begin{bmatrix} \text{taux t} \\ \text{tauy t} \end{bmatrix} = \begin{bmatrix} e^{(ax)t} \\ e^{(ay)t} \end{bmatrix} \wedge$ [A4:] $\begin{bmatrix} \text{fexd t} \\ \text{feyd t} \end{bmatrix} = \begin{bmatrix} e^{(bx)t} \\ e^{(by)t} \end{bmatrix} \wedge$

[A5:] $(\forall t.\ 0 \leq t \Rightarrow x(t) = (x0 + xd0 * (mx + my + mp)) - xd0 * (mx + my + mp) * e^{\frac{-1}{mx+my+mp}t} + \frac{1}{(mx+my+mp) * (ax)^2+ax} e^{(ax)t} - \frac{1}{(mx+my+mp) * (bx)^2+bx} e^{(bx)t}) \wedge$

[A6:] $(\forall t.\ 0 \leq t \Rightarrow y(t) = (y0 + yd0 * (my + mp)) - yd0 * (my + mp) * e^{\frac{-1}{my+mp}t} + \frac{1}{(my+mp) * (ay)^2+ay} e^{(ay)t} - \frac{1}{(my+mp) * (by)^2+by} e^{(by)t})$

$\Rightarrow$ dynamics_motion_stage TwoDOF (sc,t) sp cp

Assumption A1 provides physical constraints associated with the system. Assumption A2 is the same as that of Theorem 4.1. Assumptions A3-A4 assert the two-dimensional torque and force vectors as exponential functions. Assumptions A5-A6 present the values of xy coordinates at any time t. Finally, the conclusion provides the dynamics of the robotic cell injection system during the process of injection. The verification of Theorem 4.2 is based on the properties of transcendental functions (*transcendentals.ml* [hol20c]), real derivatives (*realanalysis.ml* [hol20b]), vectors (*vectors.ml* [hol20d]) and matrices (*vectors.ml* [hol20d]) along with some real arithmetic reasoning. Similarly, we formally verify the solutions of the dynamics of the 3-DOF and 4-DOF robotic cell injection systems during injection process and the details about their verification can be found at [Ras20]. Next, we verify an alternate representation of the image-stage coordinate frames interrelationship, which depends on the dynamics of the 2-DOF motion stage (Definition 4.2) and is a vital property for analyzing the robotic cell injection systems. For this purpose, we first model the positioning table and inertia matrices in HOL Light as:

Definition 4.6 *Positioning Table and Inertia Matrices*

$\vdash_{def}$ $\forall\alpha$ dx dy fx fy mx my mp.
posit_table_mat_fin_2dim (α,dx,dy,fx,fy,mx,my,mp) =
posit_table_mat_2dim $**$ matrix_inv (transformation_matrix_2dim (α,dx,dy,fx,fy,mx,my,mp))
$\vdash_{def}$ $\forall\alpha$ dx dy fx fy mx my mp.
inertia_mat_2dim (α,dx,dy,fx,fy,mx,my,mp) =
mass_mat_2dim (α,dx,dy,fx,fy,mx,my,mp) $**$ matrix_inv
(transformation_matrix_2dim (α,dx,dy,fx,fy,mx,my,mp))

where the HOL Light function matrix_inv accepts a matrix A:$\mathbb{R}^{M\ N}$ and returns its inverse. Now, the alternate representation of the image-stage coordinate frames interrelationship, providing the relationship between the dynamics of the 2-DOF motion stage and the image-stage coordinate frames interrelationship, is verified as the following HOL Light theorem:

Theorem 4.3 *Image-stage Coordinate Frames Interrelationship*
$\vdash_{thm}$ ∀xc yc u v x y fx fy dx dy mx my mp taux tauy fexd feyd fex fey α b m k t.
let sp = ((α,dx,dy,fx,fy,mx,my,mp):twodof_sys_par) and
cp = ((taux,tauy,fexd,feyd,fex,fey,m,b,k):two_dim_cont_par) and
cc = ((xc,yc):two_dim_point) and
sc = ((x,y):two_dim_point) and
ic = ((u,v):two_dim_point) in
[A1:] is_valid_image_stage_relat_2dim sp ∧
[A2:] invertible (transformation_matrix_2dim sp) ∧
[A3:] (∀t. u real_differentiable atreal t) ∧
[A4:] (∀t. v real_differentiable atreal t) ∧
[A5:] (∀t. $\frac{\mathtt{du}}{\mathtt{dt}}$ real_differentiable atreal t) ∧
[A6:] (∀t. $\frac{\mathtt{dv}}{\mathtt{dt}}$ real_differentiable atreal t) ∧
[A7:] (∀t. relat_image_camera_coord_frame_2dim cc ic sp t) ∧
[A8:] (∀t. relat_camera_stage_coord_frame_2dim cc sc sp t) ∧
[A9:] dynamics_motion_stage TwoDOF (sc,t) sp cp
⇒ inertia_mat_2dim sp ∗∗ sec_ord_der_gen_coord_2dim (ic,t) +
posit_table_mat_fin_2dim sp ∗∗ fir_ord_der_gen_coord_2dim (ic,t) =
torque_vec_2dim cp - desired_force_vec_2dim cp

Assumption A1 provides a valid relationship between the image and stage coordinate frames (Definition 3.12). Similarly, Assumption A2 ensures that the transformation matrix (transformation_matrix_2dim, Definition 3.11) is invertible, i.e., its inverse exists. Assumptions A3-A6 model the differentiability condition for the image coordinate frames and their first-order derivatives. Assumptions A7-A8 provide the image-camera and camera-stage (Definition 3.7) coordinate frames interrelationships. Assumption A9 models the dynamics of the robotic cell injection system having 2-DOF motion stage (Definition 4.2). Finally, the conclusion of Theorem 4.3 presents the relationship between the dynamics of the 2-DOF motion stage and the image-stage coordinate frames interrelationship. The verification of Theorem 4.3 is based on the properties of the real derivative (*realanalysis.ml* [hol20b]), matrices (*vectors.ml* [hol20d]) and vectors (*vectors.ml* [hol20d]) along with some real arithmetic reasoning. For a particular scenario, if we are given with the dynamics of the 2-DOF motion stage of a robotic cell injection system, we can obtain the relationship between the image and stage coordinate frames using Theorem 4.3. Similarly, we formally verify the relationship of the dynamics of motion stage and the image-stage coordinate frames interrelationship of the 3-DOF and 4-DOF robotic cell injection systems and the details about their verification can be found at [Ras20].

The injection motion controller is another vital part of the cell injection systems and its verification is necessary for a reliable system. It mainly includes the control of the applied injection force and the torque applied to the driving motor. In order to formalize these force and torque controllers, we first model the types of these controllers using the enumerated type feature of HOL Light as:

Definition 4.7 *Force Controller*
define_type "force_control = TwoDimF |
ThreeDimF"

Definition 4.8 *Torque Controller*
define_type "torque_control = TwoDimT |
ThreeDimT"

where `TwoDimF` and `ThreeDimF` model the two and three-dimensional force controllers, respectively. Similarly, `TwoDimT` and `ThreeDimT` present the two and three-dimensional torque controllers, respectively.

Now, we first formalize the force and torque controls for a 2-DOF robotic cell injection system and formally verify the relationship between both of these controllers. The impedance force control for a 2-DOF robotic cell injection system (will be formalized as Definition 4.9) is represented as follows:

$$m\ddot{e} + b\dot{e} + ke = f_e \tag{3}$$

where f_e is the two-dimensional vector having f_{ex} and f_{ey} as its elements. Moreover, e, $\dot{e}$ and $\ddot{e}$ are the two-dimensional vectors representing the position errors of the xy stage coordinates, their first-order and second-order derivatives, respectively, and are mathematically expressed as:

$$e = \begin{bmatrix} x_d \\ y_d \end{bmatrix} - \begin{bmatrix} x \\ y \end{bmatrix}, \dot{e} = \begin{bmatrix} \frac{dx_d}{dt} \\ \frac{dy_d}{dt} \end{bmatrix} - \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix}, \ddot{e} = \begin{bmatrix} \frac{d^2x_d}{dt} \\ \frac{d^2y_d}{dt} \end{bmatrix} - \begin{bmatrix} \frac{d^2x}{dt} \\ \frac{d^2y}{dt} \end{bmatrix} \tag{4}$$

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 (will be formalized as Definition 4.10) is mathematically expressed as:

$$\begin{aligned} \begin{bmatrix} \tau_x \\ \tau_y \end{bmatrix} &= \begin{bmatrix} m_x + m_y + m_p & 0 \\ 0 & m_y + m_p \end{bmatrix} \begin{bmatrix} f_x \cos\alpha & f_x \sin\alpha \\ -f_y \sin\alpha & f_y \cos\alpha \end{bmatrix} \begin{bmatrix} \frac{d^2x_d}{dt} \\ \frac{d^2y_d}{dt} \end{bmatrix} + \\ &\begin{bmatrix} m_x + m_y + m_p & 0 \\ 0 & m_y + m_p \end{bmatrix} \begin{bmatrix} f_x \cos\alpha & f_x \sin\alpha \\ -f_y \sin\alpha & f_y \cos\alpha \end{bmatrix} m^{-1}(b\dot{e} + ke - f_e) + \\ &\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_x \cos\alpha & f_x \sin\alpha \\ -f_y \sin\alpha & f_y \cos\alpha \end{bmatrix}^{-1} \right) \begin{bmatrix} f_x \cos\alpha & f_x \sin\alpha \\ -f_y \sin\alpha & f_y \cos\alpha \end{bmatrix} \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} + \begin{bmatrix} fex^d \\ fey^d \end{bmatrix} \end{aligned} \tag{5}$$

Equation (5) can be alternatively written as:

$$\overrightarrow{\tau} = MT \begin{bmatrix} \frac{d^2x_d}{dt} \\ \frac{d^2y_d}{dt} \end{bmatrix} + MTm^{-1}(b\dot{e} + ke - f_e) + NT \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} + \overrightarrow{f_{ed}} \tag{6}$$

where M, N and T in the above equation denote the inertia, positioning table and transformation matrices. For example, the inertia matrix M contains the respective masses, i.e., m_x, m_y and m_p. Similarly, the positioning table represents a multiplication of a diagonal matrix with the inverse of the transformation matrix, i.e., T^{-1}. The above equation was wrongly presented in simulations [HSML06] and model checking [SH17] based analysis as follows:

$$\overrightarrow{\tau} = M \begin{bmatrix} \frac{d^2x_d}{dt} \\ \frac{d^2y_d}{dt} \end{bmatrix} + Mm^{-1}(b\dot{e} + ke - f_e) + N \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} + \overrightarrow{f_{ed}} \tag{7}$$

$$\overrightarrow{\tau} = MT \begin{bmatrix} \frac{d^2x_d}{dt} \\ \frac{d^2y_d}{dt} \end{bmatrix} + MTm^{-1}(b\dot{e} + ke - f_e) + NT \begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \end{bmatrix} + \overrightarrow{f_e} \tag{8}$$

In Eq. (7) (used in the simulations based analysis [HSML06]), the transformation matrix (T) is missing, which involves the amount of applied force and the angles at which the pipette is injected into the cell: its absence can lead to disastrous consequences, i.e., damaging cell tissues, excess substance injection etc. Similarly, in Eq. (8) (used in the model checking based analysis [SH17]), f_{ed} is wrongly interpreted as f_e, i.e., the desired force, is taken equal to the applied force, which can never happen in a real-world system. We obtained these incorrect interpretations of Eq. (6) in the simulations and model checking based analyses during the verification of the relationship between force and torque controllers. We first started the verification of this relationship using Eq. (7) and ended up with the identification of this issue. We used the backward (goal directed) proof method to verify the relationship, modeled as the HOL Light theorem of the form A1, A2, ..., An ⇒ C, where A1, A2, ..., An capture the set of assumptions and C models the conclusion and is the main goal of the HOL Light theorem. It involves the concept of a tactic (an ML function), which divides the main goal into subgoals. These tactics are repeatedly used to reduce or simplify the main goal (required theorem) into intermediate subgoals until they match with the assumptions of the HOL Light theorem. However, the transformation matrix (T) was missing in one of the subgoals, resulting in a mismatch with the corresponding assumption, which ended up with the identification of the issue. Next, we took Eq. (8) and again, during its verification, identified its wrong interpretation, which enabled us to obtain its right interpretation as given in Eq. (6). To formally verify the relationship between the two-dimensional force and torque controllers, we first formalize these controllers in HOL Light as follows:

Definition 4.9 *Two Dimensional Force Controller*
$\vdash_{def}$ ∀xd yd x y t taux tauy fexd feyd fex fey m b k.
contact_space_imped_control_2d TwoDimF (xd,yd) (x,y) (taux,tauy,fexd,feyd,fex,fey,m,b,k) t ⇔
m % posit_errors_second_order_deriv_2dim (xd,yd) (x,y) t +
b % posit_errors_first_order_deriv_2dim (xd,yd) (x,y) t +
k % posit_errors_vector_2dim (xd,yd) (x,y) t = external_force_2dim (taux,tauy,fexd,feyd,fex,fey,m,b,k)

Definition 4.10 *Two Dimensional Torque Controller*
$\vdash_{def}$ ∀xd yd α dx dy fx fy mx my mp x y t taux tauy fexd feyd fex fey m b k.
image_based_torque_control_2d TwoDimT (x,y) (xd,yd)
(α,dx,dy,fx,fy,mx,my,mp) (taux,tauy,fexd,feyd,fex,fey,m,b,k) t ⇔
torque_vec_2dim (taux,tauy,fexd,feyd,fex,fey,m,b,k) =
(inertia_mat_2dim (α,dx,dy,fx,fy,mx,my,mp) ** transformation_matrix_2dim (α,dx,dy,fx,fy,mx,my,mp)) **
posit_second_order_deriv_2dim (xd,yd) t +
(inertia_mat_2dim (α,dx,dy,fx,fy,mx,my,mp) ** transformation_matrix_2dim (α,dx,dy,fx,fy,mx,my,mp)) **
inv m % (b % posit_errors_first_order_deriv_2dim (xd,yd) (x,y) t +
k % posit_errors_vector_2dim (xd,yd) (x,y) t - external_force_2dim (taux,tauy,fexd,feyd,fex,fey,m,b,k)) +
(posit_table_mat_fin_2dim (α,dx,dy,fx,fy,mx,my,mp) **
transformation_matrix_2dim (α,dx,dy,fx,fy,mx,my,mp)) ** posit_first_order_deriv_2dim (x,y) t +
desired_force_vec_2dim (taux,tauy,fexd,feyd,fex,fey,m,b,k)

Next, we model various conditions on the impedance parameters, i.e., their positivity, as constraint in HOL Light:

Definition 4.11 *Constraint on Impedance Parameters*
$\vdash_{def}$ ∀taux tauy fexd feyd fex fey m b k.
constraints_imped_parameters (taux,tauy,fexd,feyd,fex,fey,m,b,k) ⇔ &0 < m ∧ &0 < k ∧ &0 < b

Now, we formally verify the relationship between the contact-space-impedance force [Eq. (3)] and the image-based torque controller [Eq. (5)], providing the smooth functionality of the robotic cell injection process, as the following HOL Light theorem:

Theorem 4.4 *Relationship Between Force and Torque Controllers*
$\vdash_{thm}$ ∀xd yd x y t mx my mp fx fy dx dy α taux tauy fex fey fexd feyd m b k.
let sp = ((α,dx,dy,fx,fy,mx,my,mp):twodof_sys_par) and
cp = ((taux,tauy,fexd,feyd,fex,fey,m,b,k):two_dim_cont_par) and
dc = ((xd,yd):two_dim_point) and
sc = ((x,y):two_dim_point) in

[A1:] constraints_imped_parameters cp ∧
[A2:] invertible (transformation_matrix_2dim sp) ∧
[A3:] contact_space_imped_control_2d TwoDimF dc sc cp t ∧
[A4:] dynamics_motion_stage TwoDOF (sc,t) sp cp
⇒ image_based_torque_control_2d TwoDimT sc dc sp cp t

Assumption A1 captures the conditions on the desired impedance parameters (Definition 4.11). Similarly, Assumption A2 provides the condition that the transformation matrix (transformation_matrix_2dim) is invertible. Assumption A3 models the impedance force control [Eq. (3)]. Assumption A4 presents the dynamics of the robotic cell injection system having 2-DOF motion stage (Definition 4.2). Finally, the conclusion represents the image-based torque controller [Eq. (5)], controlling the process of the robotic cell injection. The verification of Theorem 4.4 is mainly based on the properties of real derivative (*realanalysis.ml* [hol20b]), vectors (*vectors.ml* [hol20d]) and matrices (*vectors.ml* [hol20d]). Similarly, we formally verify the relationship between the 3D force and torque controllers and the details about their verification can be found at [Ras20]. Finally, we package both (2D and 3D) force and torque controllers in inductive predicates `gen_force_control` and `gen_torque_control`, which take the *type* and *parameters* of the controllers, and coordinate frames, and return the predicate describing the corresponding 2D or 3D controller. For example, for the 2D torque controller `TwoDimT`, the inductive predicate `gen_torque_control` returns `image_based_ torque_control_2d`[1].

We verify a general theorem, which describes the relationship between generalized force controller fc and torque controller tc, as follows:

Theorem 4.5 *Relationship Between Generalized Force and Torque Controllers*
$\vdash_{thm}$ ∀(fc:force_control) (tc:torque_control) (sys:robotic_cis) xd yd zd x y z t.
let sp = ((α,dx,dy,dz,fx,fy,mx,my,mz,mp):threedof_sys_par) and
cp = ((taux,tauy,tauz,fexd,feyd,fezd,fex,fey,fez,m,b,k,g):three_dim_cont_par) and
sc = ((x,y,z):three_dim_point) and
dc = ((xd,yd,zd):three_dim_point) in
[A1]: gen_phy_imped_constraints sp cp ∧
[A2]: gen_force_control fc dc sc cp t ∧
[A3]: dynam_mot_stage sys (sc,t) sp cp
⇒ gen_torque_control tc sc dc sp cp t

where the predicate `gen_phy_imped_constraints` encapsulates the physical and impedance parameters constraints of all types of controllers (2D and 3D) in our formalization. Similarly, the predicate `dynam_mot_stage` captures the dynamics of the robotic cell injection systems upto 4-DOF. The proof process of Theorem 4.5 is based on induction on `fc:force_control`, `tc:torque_control` and `sys:robotic_cis` along with the verified theorems, providing the relationship of the force and torque controllers, for 2D and 3D controllers (e.g., Theorem 4.4 for 2D controllers).

This concludes our formalization of the robotic cell injection systems in HOL Light. Due to the undecidable nature of the higher-order logic, the verification results, presented in Sects. 3 and 4, involved manual interventions and human guidance. However, we developed some tactics to automate the verification process. For example, we developed a tactic VEC_MAT_SIMP_TAC, which simplifies the vector and matrix arithmetics involved in the formal analysis of the robotic cell injection system. The details about these tactics and rest of the formalization can be found in our proof script [Ras20].

The distinguishing feature of our formal analysis is that all of the verified theorems are of generic nature, i.e., all of the functions and variables are universally quantified and thus can be specialized based on the requirement of the analysis of the cell injection systems. Whereas, in the case of computer based simulations, we need to model each case individually. Moreover, the inherent soundness of the theorem proving technique ensures that all the required assumptions are explicitly present along with the theorem. Similarly, due to the high expressiveness of the higher-order logic, our approach allows us to model the dynamics of the robotic cell injection systems involving differential and derivative [Eqs. (1), (3), (5)] in their true form, whereas, in their model checking based analysis [SH17] they are discretized and modeled using a state-transition system, which may compromise the accuracy and completeness of the corresponding analysis.

[1] We have omitted the formal definitions of gen_force_control and gen_torque_control for the sake of conciseness, however, interested readers can find the formal definitions and HOL Light code on the project's webpage [Ras20].

Table 6. Verification details for each theorem

Formalized theorems	Proof lines	Man-hours
Theorem 3.1 (Relationship Between Camera and Stage Coordinates, 2-DOF)	19	1
Theorem 3.2 (Relationship Between Image and Camera Coordinates, 2-DOF)	7	0.5
Theorem 3.3 (Relationship Between Image and Stage Coordinates, 2-DOF)	34	2
Table 3: Relationship Between Camera and Stage Coordinates, 3-DOF	21	1
Table 3: Relationship Between Image and Camera Coordinates, 3-DOF	7	0.5
Table 3: Relationship Between Image and Stage Coordinates, 3-DOF	29	3
Table 4: Relationship Between Camera and Stage Coordinates, 4-DOF	9	0.5
Table 4: Relationship Between Image and Camera Coordinates, 4-DOF	7	0.5
Table 4: Relationship Between Image and Stage Coordinates, 4-DOF	32	3
Theorem 4.1 (Solution of Dynamical Behavior of 2-DOF Motion Stage)	270	16
Theorem 4.2 (Solution of Dynamical Behavior of 2-DOF Motion Stage)	350	20
Theorem 4.3 (Image-stage Coordinate Frames Interrelationship)	125	12
Theorem 4.4 (Relationship Between Force and Torque Controllers)	135	10
Theorem 4.5 (Relationship Between Generalized Force and Torque Controllers)	7	1

The effort involved in the verification of the individual theorem in the form of proof lines and the man-hours is presented in Table 6. The proof process for the formalization of coordinate frames for the 2-DOF, 3-DOF and 4-DOF robotic cell injection system took 165 lines and only 12 man-hours. Similarly, the verification of Theorem 4.5 involved only 7 lines of HOL Light code and an hour, which clearly illustrates the benefit of Theorem 4.4 and the corresponding theorems for 3 and 4-DOF robotic cell injection systems. Moreover, the development of the formal verification of the robotic cell injection system (formal definitions, and corresponding theorems and their proofs) on paper, before their actual implementation in HOL Light, took around 90 man-hours. It is important to note that the man-hours are based on the number of lines of code as well as the complexity of the proof. So the number of lines of the proof script do not have a direct relationship with the man-hours. For example, the proof lines for the verification of relationship between the image and stage coordinates for 2- DOF system are greater than that for 3-DOF system, whereas the man-hours for the former are less than that for the latter.

Our formal analysis of the dynamical behavior of the robotic cell injection presented above is quite vital for the correct design and functionality of the robotic cell injection systems. For example, the verified relationships between various coordinate frames enable the correct orientation, relative position and movement of its various components, i.e., injection manipulator, camera and microscope etc. Similarly, the verification of the contact-space-impedance force and the image-based torque controllers can help in planning the motion of the injection pipette by developing the efficient force and torque control algorithms and thus to regulate the process of the robotic cell injection.

5. Conclusions

Robotic cell injection involves the insertion of bimolecular, sperm, DNA or protein into a specific location of suspended or adherent cells and is widely used in drug development, cellular biology research and transgenics. In this paper, we proposed a higher-order-logic theorem proving based framework for analyzing the dynamical behavior of the robotic cell injection system upto 4-DOF. We first formalized various coordinate frames, i.e., stage, camera and image coordinate, which are the main components of a robotic cell injection system, and formally verified their interrelationship using the HOL Light theorem prover. We also formalized the dynamics of the robotic cell injection systems upto 4-DOF, based on differential equations and verified their solutions in HOL Light. Finally, we formalized the contact-space-impedance force and image-based torque controllers (2D and 3D) and verified their relationship. Our formalization helped us to identify some key discrepancies in the simulation-based and model checking based analysis of these systems, which shows the usefulness of using higher-order-logic theorem proving in the formal analysis of safety-critical systems.

In future, we plan to formally verify the robotic cell injection system exhibiting the non-linear dynamics [HSML06], i.e., the dynamics of the motion stage are modeled as a set of second-order non-linear differential equations. This requires the verification of the non-analytical/numerical solutions of their dynamics. We need the formalization of the interval arithmetic [Daw11], providing the reliable and guaranteed solutions of their differential equations based dynamical models.

References

[AH17] Ayub MS, Hasan O (2017) Formal probabilistic analysis of a virtual fixture control algorithm for a surgical robot. In: Verification and evaluation of computer and communication systems, volume 10466 of LNCS, pp 1–16. Springer

[BDA95] Bianco A, De Alfaro L (1995) Model checking of probabilistic and nondeterministic systems. I:n Foundations of software technology and theoretical computer science, volume 1026 of LNCS, pp 499–513. Springer

[BGM+15] Bresolin D, Geretti L, Muradore R, Fiorini P, Villa T (2015) Formal verification of robotic surgery tasks by reachability analysis. Microprocess Microsyst 39(8):836–842

[CGP99] Clarke EM, Grumberg O, Peled D (1999) Model checking. MIT Press

[CKNZ12] Clarke EM, Klieber W, Nováček M, Zuliani P (2012) Model checking and the state explosion problem. In: Tools for practical software verification, volume 7682 of LNCS, pp 1–30. Springer

[Daw11] Dawood H (2011) Theories of interval arithmetic: mathematical foundations and applications. LAP Lambert Academic Publishing

[DPV13] Durán AJ, Pérez M, Varona JL (2013) The misfortunes of a mathematicians' Trio using computer algebra systems: can we trust? CoRR. arXiv:1312.3270

[FN16] Faroque M, Nizam S (2016) Virtual reality training for micro-robotic cell injection. Technical report, Deakin University, Australia

[Gor88] Gordon Michael JC (1988) HOL: a proof generating system for higher-order logic. In: VLSI specification, verification and synthesis, volume 35 of SECS, pp 73–128. Springer

[Har96a] Harrison J (1996) HOL light: a tutorial introduction. In: Srivas M, Camilleri A (eds) Proceedings of the first international conference on formal methods in computer-aided design (FMCAD'96), volume 1166 of LNCS, pp 265–269. Springer-Verlag

[Har96b] Harrison J (1996) HOL light: a tutorial introduction. In: Formal methods in computer-aided design, volume 1166 of LNCS, pp 265–269. Springer

[Har09] Harrison J (2009) Handbook of practical logic and automated reasoning. Cambridge University Press

[Har13] Harrison J(2013) The HOL light theory of euclidean space. J Autom Reason 1–18

[hol20a] HOL Light Multivariate Calculus (2020) https://github.com/jrh13/hol-light/blob/master/Multivariate

[hol20b] HOL Light Real Calculus (2020) https://github.com/jrh13/hol-light/blob/master/Multivariate/realanalysis.ml

[hol20c] HOL Light Transcendentals (2020) https://github.com/jrh13/hol-light/blob/master/Multivariate/transcendentals.ml

[hol20d] HOL Light Vectors and Matrices (2020) https://github.com/jrh13/hol-light/blob/master/Multivariate/vectors.ml

[HSM+09] Huang H, Sun D, Mills JK, Li WJ, Cheng SH (2009) Visual-based impedance control of out-of-plane cell injection systems. Trans Autom Sci Eng 6(3):565–571

[HSML06] Huang H, Sun D, Mills JK, Li Wen J (2006) A visual impedance force control of a robotic cell injection system. In: Robotics and biomimetics, pp 233–238. IEEE

[HT15] Hasan O, Tahar S (2015) Formal verification methods. In: Encyclopedia of information science and technology, pp 7162–7170. IGI Global Pub

[KK04] Kuncova J, Kallio Pasi (2004) Challenges in capillary pressure microinjection. In: Engineering in Medicine and Biology Society, volume 2, pp 4998–5001. IEEE

[KRPK13] Kouskoulas Y, Renshaw D, Platzer A, Kazanzides P (2013) Certifying the safe design of a virtual fixture control algorithm for a surgical robot. In: Hybrid systems: computation and control, pp 263–272. ACM

[Map20] Maple (2020) https://www.maplesoft.com/

[Mat20] Mathematica (2020) https://www.wolfram.com/mathematica/

[NFT+98] 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

[NS94] Nethery JF, Spong MW (1994) Robotica: a mathematica package for robot analysis. IEEE Robot Autom Mag 1(1):13–20

[Pau96] Paulson LC (1996) ML for the working programmer. Cambridge University Press

[Ras20] Rashid A (2020) Formal verification of robotic cell injection systems upto 4-DOF using HOL Light. http://save.seecs.nust.edu.pk/fvrcis/

[RH17] Rashid A, Hasan O (2017) Formal analysis of robotic cell injection systems using theorem proving. In: Design, modeling, and evaluation of cyber physical systems, volume 11267 of LNCS, pp 127–141. Springer

[RH18] Rashid A, Hasan O (2018) Formal modeling of robotic cell injection systems in higher-order logic. In: Formal verification of physical systems, volume 2307, pp 1–9. CEUR-WS

[SH17] Sardar MU, Hasan O (2017) Towards probabilistic formal modeling of robotic cell injection systems. In: Models for formal analysis of real systems, pp 271–282

[SL97] Sun D, Liu Y (1997) Modeling and impedance control of a two-manipulator system handling a flexible beam. In: Proceedings of 1997 IEEE International Conference on Robotics and automation, volume 2, pp 1787–1792. IEEE

[SN02] Sun Y, Nelson BJ (2002) Biological cell injection using an autonomous microrobotic system. Robot Res 21(10-11):861–868

[YKY+99] 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

Received 29 April 2019

Accepted in revised form 5 May 2020 by Ana Cavalcanti and Pedro Ribeiro

Published online 22 June 2020