

**CZECH TECHNICAL UNIVERSITY IN PRAGUE**  
**FACULTY OF MECHANICAL ENGINEERING**

**Department of mechanics, biomechanics and mechatronics**

**BACHELOR THESIS**

**Kinematic analysis of cinema dolly**

**Suren Ali-Ogly**

**Prague 2015**



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ v Praze

Fakulta strojní, Ústav mechaniky, biomechaniky a mechatroniky

Technická 4, 16607 Praha 6

Akademický rok: 2014/2015

---

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

pro:

**Suren Ali-Ogly**

obor:

Teoretický základ strojího inženýrství

Název tématu: **Kinematický model kamerového jeřábu**

Zásady pro vypracování:

1. Seznamte se s problematikou technologií pro kamerové jeřáby
2. Vytvořte kinematický 3D model zvolené struktury kamerového jeřábu
3. Vytvořte 3D virtuální model kamerového jeřábu
4. Implementujte v Simulinku řešení kinematického modelu z bodu 2.
5. Propojte kinematické řešení z bodu 4. s virtuálním modelem z bodu 3.

## Annotation List

Authors Name: Suren Ali-Ogly

Name of Bachelor Thesis (cz): Kinematický model kamerového jeřábu

Name of Bachelor Thesis (eng): Kinematic model of cinema dolly.

Year: 2015

Field of study: Theoretical fundamentals of mechanical engineering

Department: Department of mechanics, biomechanics and mechatronics

Supervisor: Ing. Martin Nečas, MSc. Ph.D.

Consultant: Ing. Martin Nečas, MSc. Ph.D.

Bibliographical data:	Number of pages	63
	Number of figures	53
	Number of tables	2
	Number of diagrams	3
	Number of attachments	1

Keywords: kinematics, movie industry, intermittent mechanisms, camera motion control, robotics, camera dolly, numerical solution.

Abstract:

This Bachelor thesis deals with development of mechanisms in movie industry. First an overview of current mechanisms is presented, afterwards one is chosen to be analyzed using the kinematic analysis. And last, kinematics of camera dolly mechanism is solved using numerical iteration method.

## **Prohlášení:**

Prohlašuji, že jsem svou bakalářskou práci vypracoval samostatně a použil jsem pouze podklady uvedené v příloženém seznamu.

V Praze dne .....

.....

podpis

## **Acknowledgements:**

I would like to thank supervisor of this thesis Ing. Martin Necas, MSc., Ph.D., for professional guidance, valuable advice and time he devoted to me.

# Contents

Introduction .....	10
Chapter 1 Historical Revision of Mechanisms in Movie Industry .....	11
1.1 Intermittent mechanism.....	11
1.1.1 Evolution of Intermittent mechanism.....	10
1.3 Camera motion control.....	15
Chapter 2 Mechanisms in Movie Industry.....	17
2.2 Mechanisms inside of movie artefact.....	17
2.3 Mechanisms in movie production.....	19
2.4 Camera motion controllers.....	21
Chapter 3 Technodolly .....	22
3.1 TECHNODOLLY® dimensions.....	22
3.2 Technodolly® ‘s crane features.....	24
3.2.1 Teach in.....	24
3.2.2 Memory.....	24
3.2.3 Camera holder mechanism.....	25
Chapter 4 Fundamentals of kinematic theory .....	26
4.1 Introduction.....	26
4.2 Kinematics.....	30
4.2.1 Position.....	30
4.2.2 Orientation.....	30
4.2.3 Homogeneous transformation matrices.....	32
4.3 Matrix method for solving Position Problem.....	35
4.4 Numerical solution (Theory) .....	37
Chapter 5 Numerical solution .....	39
5.1 Detailed description of the Newton-Raphson method.....	50
5.2 Animation of Camera Dolly.....	53
5.3 Colors of MATLAB® drawings.....	56
5.4 Figures from MATLAB® .....	56
Chapter 6 Conclusion .....	59
Appendix .....	60

Bibliography ..... 62

## List of Figures

Figure 1-1 Maltese cross, intermitten mechanism [3].....	11
Figure 1-2 Contre-griffe .....	12
Figure 1-3 Kindelman's Claw and registration pin mechanism [6] .....	13
Figure 1-4 Photo-electric control devices for slit-scan mechanism. [7] .....	15
Figure 1-5 Dykstraflex on the shoot of "Star Wars".....	15
Figure 2-1 Shark, from movie "Jaws", 1975.....	17
Figure 2-2 Kraken from "20.000 Leagues under the sea", 1954.....	18
Figure 2-3 Dinosaur from "Jurassic Park" 1993 .....	18
Figure 2-4 Robotic deer from "Santa-Claus" 1994.....	18
Figure 2-5 Submarine engine. " Vynález zkázy " 1958.....	19
Figure 2-6 Jet's parallel manipulator "True Lies", James Cameron, 1994.....	19
Figure 2-7 Rotating hall "Inception", 2010 .....	20
Figure 2-8 Moving on the rolled road "Inception", 2010.....	20
Figure 2-9 Centrifugal machine "2001: A space Odyssey" .....	20
Figure 2-10 "Milo" motion controller .....	21
Figure 2-11 Technodolly.....	21
Figure 3-1 Technodolly dimensions [18].....	22
Figure 3-2 Dolly view.....	23
Figure 3-3 18 meter track, consist of 6 parts, 3 m each .....	23
Figures 3-4 Teach in. [26].....	24
Figure 3-5 Camera holder .....	25
Figure 4-1 Revolute Joint .....	26
Figure 4-2 Prismatic joint [13, p. 87].....	27
Figure 4-3 Cylindrical joint .....	27
Figure 4-4 Open kinematic chain .....	29
Figure 4-5 Closed loop chain.....	29
Figure 4-6 Mixed kinematic chain.....	29
Figure 4-7 Point P in space with reference A.....	30
Figure 4-8 rotation about the moving coordinate axes. z-x-z.....	31
Figure 4-9 Basic motion along X.....	33
Figure 4-10 Basic motion along Y.....	33
Figure 4-11 Basic motion along Z.....	33
Figure 4-12 Revolution around X .....	34
Figure 4-13 Revolute around Y .....	34
Figure 4-14 Revolute around Z.....	34
Figure 4-16 $f(q,z)=0$ construction.....	36
Figure 5-1 Base with jacks and locking rod .....	39
Figure 5-2 Track wheels an track [11, p. 20].....	39
Figure 5-3 Technodooly on track .....	40
Figure 5-4 Dolly on track.....	41
Figure 5-5 Loop of dolly on the track.....	45
Figure 5-6 XY view of certain position .....	48
Figure 5-7 XZ view for certain position .....	48
Figure 5-8 Camera dolly in $q_f$ position.....	53

Figure 5-9 Camera dolly in $q_i$ position.....	53
Figure 5-11 Camera dolly in $q_f$ position MATLAB® <sup>4</sup> .....	54
Figure 5-10 Camera Dolly in $q_f$ position. MATLAB® .....	54
Figure 5-12 Position at 25th second .....	56
Figure 5-13 Mechanism at 195th seconds .....	57
Figure 5-14 Mechanism at 391st second .....	57
Figure 5-15 Mechanism at 552st second .....	58
Figure 5-16 Mechanism at 779th second .....	58

## List of Tables

Table 1 Specifications and characteristics of Various Types of Intermittent Motion mechanism [4] .....	14
Table 2 Types of joints [15, p. 23] .....	28

## List of Diagrams

Diagram 1 Newton-Raphson method (short form).....	49
Diagram 2 Newton-Raphson's structure for MATLAB® .....	51
Diagram 3 Kinematic solution .....	55

## Introduction

One hundred and twenty years have already passed since the very first public movie was shown in France. It was a few seconds video of a train coming to the railway station. It was a historical event after which Lumieres brothers have made the history as inventors of camera and cinematography. Nowadays, movie industry has developed into sophisticated industry, from movies of a few seconds length it has grown into a four hours dramas about adventures in space etc. Cinematography has evolved into the nowadays level by the big investment of time, ideas of engineers and major investment from producers.

If we take a closer look inside movie industry, we will see, that one of the biggest contributions to the appearance of first movie camera came from the field of mechanics and further development of industry was always closely related with mechanical engineering. As the result, almost all movies that people are watching today were produced using advanced technologies like robotics, high resolution cameras, wireless technologies etc.

The first goal of this work is to review mechanisms used in movie industry and explain their functions. The second goal is to produce 3D virtual model of a chosen camera motion controller and to perform its kinematic analysis.

## Chapter 1 Historical Revision of Mechanisms in Movie Industry

Mechanics and cinematography form an inseparable whole. The reason for it is simple if we look in cinematography history.

Films are widely spread nowadays, almost as widely available as art pictures or music. But how do they get made in the first place? Even the simplest home video camera is based on fiendishly complex technology. [1]

### Movie camera

The movie camera, film camera or cinema-camera is a type of photographic camera which takes a rapid sequence of photographs on an image sensor or on a film. In contrast to a still camera, which captures a single snapshot at a time, the movie camera takes a series of images; each image constitutes a "frame". This is accomplished through an intermittent mechanism. The frames are later played back in a movie projector at a specific speed, called the frame rate (number of frames per second). While viewing at a particular frame rate, a person's eyes and brain merge the separate pictures together to create the illusion of motion. [2]

The intermittent mechanism is in fact a „heart“, a basic thing that makes camera work, which means a recording a moving image on a photosensitive film strip.

### 1.1 Intermittent mechanism

The intermittent mechanism produces interrupted movement of film on the step frame. Usually this mechanism consists of a continuously rotating body which translates rotary motion to another body in such a way that it stops for some moment at some certain position and then continues to move and stops again, and this procedure repeats again. You can see an example of this mechanism on the Figure 1-1

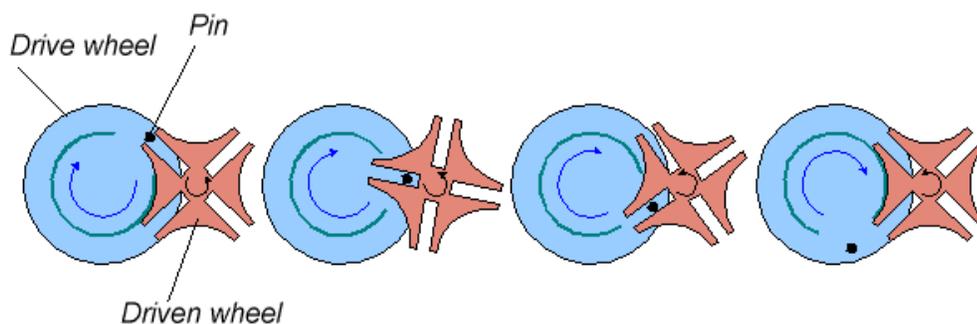


Figure 1-1 Maltese cross, intermittent mechanism [3]

On the Figure 1-1 is the Maltese cross or Geneva mechanism. The rotating drive wheel has a pin that reaches into a slot of the driven wheel advancing it by one step. The drive wheel also has a raised circular blocking disc that locks the driven wheel in position between steps. [3]

A wide variety of mechanisms has been used in photographic equipment. Such things as epicyclical gears, mutilated gears, and Geneva can be found in cameras and projectors built in the late 1800's [4]. Intermittent mechanisms vary in design. All have a pull down claw and pressure plate. Some have a registration pin as well. The pull down claw engages the film perforation and moves the film down one frame. It then disengages and goes back up to pull down the next frame. While the claw is disengages, the pressure plates holds the film steady for the period of exposure. [5]

One of the most well-known mechanisms is the registration pin or also called Contre-griffe. See Figure 1-2. It is the most popular mechanism in cameras.

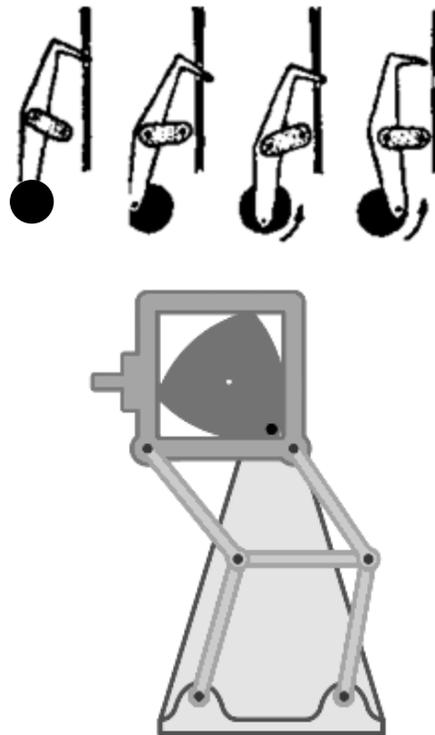


Figure 1-2 Contre-griffe

Generally considered, the invention provides a simple claw mechanism which will operate to move forward and engage the holes in a film and then move it downwardly. As it moves forward to engage the film an element associated with the claw mechanism will in each and every case engage part of a stop pin mechanism to withdraw a stop pin from engagement with the film. As the claw then moves downwardly with the film it will proceed a certain distance in this direction when it will release cooperative connection with the pin mechanism, whereupon a spring mechanism connected to the pin mechanism will

automatically advance the pin into pressing engagement with the surface of the film so that as soon as a hole is presented opposite the pin it will snap there into. In some cases the pin level is even with the top of the stroke of the claw and in other cases the pin level is even with the bottom of the claw stroke. In the latter case the actuation of the pin with the holes is made absolutely independent of the possibility as to whether or not the vertical distance between films holes engaged by the pin are variable, due to expansion of the film or shrinkage thereof. See Figure 1-3. [6]

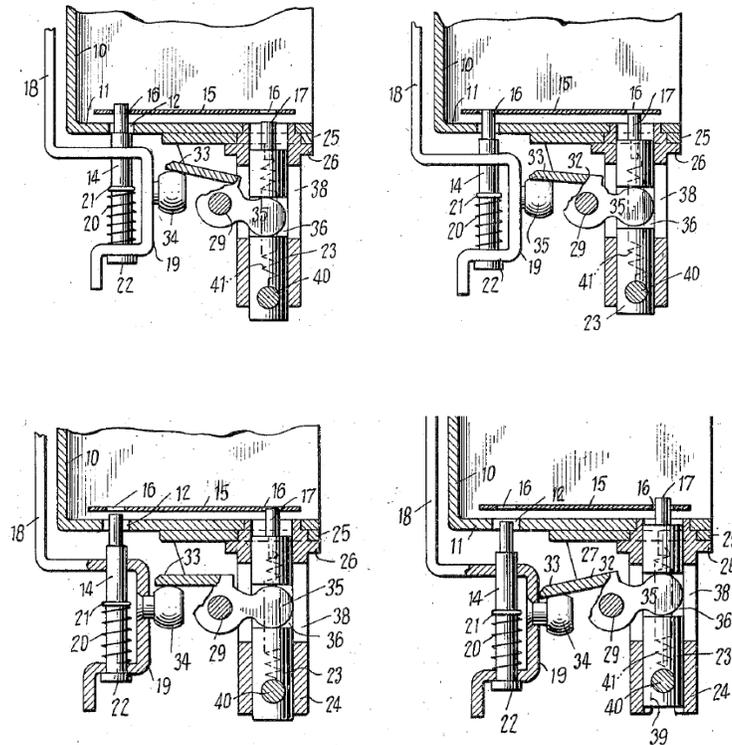


Figure 1-3 Kindelman's Claw and registration pin mechanism [6]

### 1.1.1 Evolution of Intermittent mechanism

The evolution of intermittent mechanism was fast, as the movie industry became very popular after mass realization of cameras by Bell's factories. Modernizations were provided to things like size, weight, speed capabilities, design, but the principle stayed the same.

In the (Table 1) are shown some intermittent mechanisms and their technical characteristics.

Type of Input to Mechanism	Input Displacement		Output Displacement per Motion	Number of Output Dwells		Designed to Run at Fixed (Synchronous) or Variable (Asynchronous) Rate	Input-Output Parallel or Right Angle
	During Output Dwell	During Output Motion		Per Output Revolution	Per Input Revolution		
Electrical pulses or cam loaded spring	5° to 30°	5° to 30°	1° to 45°	8 to 360	D.A.	Either	D.A.
Rotating cam	90° to 300°	60° to 270°	0.1° to 90°	4 to hundreds	1 to 2	Fixed	Parallel
Rotating shaft	90° to 300°	90° to 270°	90° to 270°	1 to 4	1 to 5	Fixed	Either
Rotating shaft	200° to 270°	160° to 90°	20° to 90°	4 to 8	1 to 2	Fixed	Parallel
Rotating shaft	200° to 270°	160° to 90°	20° to 90°	4 to 8	1 to 2	Fixed	Parallel
Rotating shaft	10° to 350°	10° to 350°	10° to many turns	1 to 36	1 to 5	Fixed	Parallel
Rotating shaft	Few degrees (theoretically 0°)	30° to 360°	30° to nearly 360°	1 to 12	1 to 12	Fixed	Parallel
Two rotating shafts	0° to many turns	Fractions of a degree to many turns	Fraction of a degree to hundreds of degrees	One to hundreds	One to hundreds	Fixed	Either
Stalled rotating shaft	0°	10° to 30°	10° to 30°	12 to 36	12 to 36	Fixed	Parallel
Rotating shaft	Few degrees to many turns	Few degrees to many turns	Few degrees to many turns	1 to 100	Fraction of one to 100	Variable	Parallel
Rotating shaft or electrical pulses	5° to 300°	5° to 180°	4° to 60°	6 to 90	D.A.	Variable	Parallel
Rotating shaft	Partial revolution to many revolutions	Partial Revolution to many revolutions	A few degrees to many revolutions	One to hundreds	One to hundreds	Variable	Mostly parallel
Electrical pulses	D.A.	D.A.	0.9° to 180°	2 to 400	D.A.	Either	D.A.
Rotating shaft	30° to 300°	60° to 320°	60° to 360°	1 to 8	1 to 3	Fixed	Parallel
Rotating shaft	180° to many turns	30° to many turns	30° to 360°	1 to 12	1 to 5	Variable	Parallel

Table 1 Specifications and characteristics of Various Types of Intermittent Motion mechanism [4]<sup>1</sup>

<sup>1</sup> Note that all mechanism listed in the table are used not only in cameras. Also: in machine tools, business machines, vending machines, productions machines, watch industry etc.

### 1.3 Camera motion control

Motion-control photography is a technology used for combined shooting of several expositions, based on the multiple exact repetition of some certain camera movement. Exact multiple repetitions, as a goal, can be reached by using technologies of robotic panoramic heads, robotic crane shots (camera cranes), and robotic camera dolly.

The main reason to use motion-control technology is to make special effects.

Early camera motion control mechanism was created by designers with the participation of Stanley Kubrick in 1968 year, during shooting the movie called "2001: A Space Odyssey". Mechanism that was holding camera, consisting of several big mechanical rigs that was conducting camera in the desired path and as actuator of motion of camera was electric motor installed to the mechanism. See Figure 1-4.

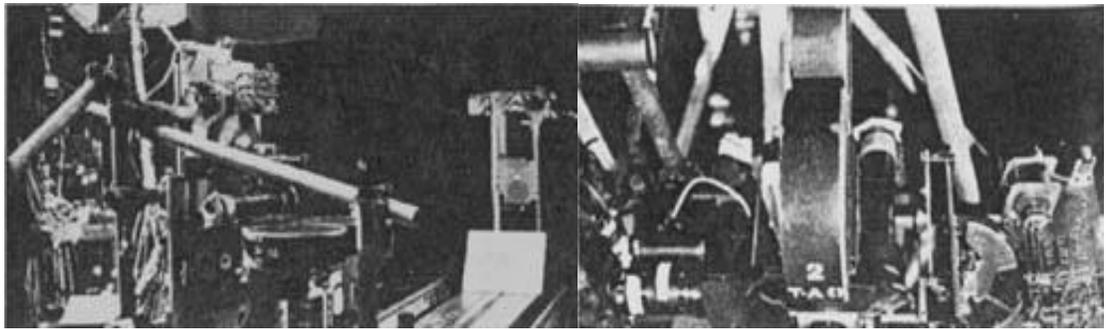


Figure 1-4 Photo-electric control devices for slit-scan mechanism. [7]

The first large usage of camera motion control technique was in the production on the movie "Star Wars: Episode 4" (1977). In this motion picture for large-scale application was used camera motion control system called "Dykstraflex" (see Figure 1-5). Dykstraflex performed precise repeatable motion in automatic way, enabled complex motion by better coordinated motion that greatly reduced error, compared with old-fashion technologies.



Figure 1-5 Dykstraflex on the shoot of "Star Wars"

Development of 3D animation, computer-generated imagery (CGI) and robotics inspired engineers to develop technologies for camera motion control and nowadays we have controllers that can make a variety of complex camera motions.

Another reason of evolution and development of camera motion controllers is the cost of movie production and profits from movies realization. Standard movies using these technologies are action, fantastic, fantasy, epic movies etc. Production budget of these kind of movies usually estimates for several millions of dollars. Main part of costs consist of technical needs. In order to receive regular income, quality of products has to increase. One of the factors of movie quality improvement is to increase amount of possible movements of cameras and decrease time demand for realization of assigned tasks.

## Chapter 2 Mechanisms in Movie Industry

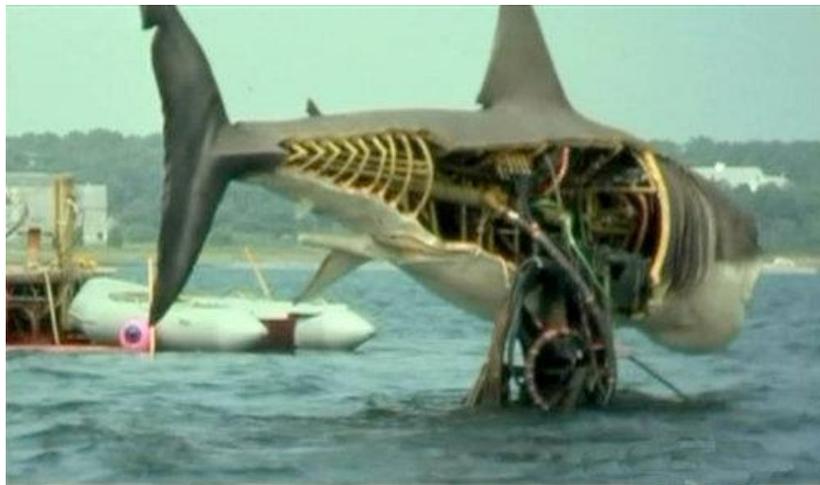
As previously mentioned in the Chapter 1, mechanics and mechanism are very important for cinema industry, especially for so-called “blockbuster” movies. For more comprehensible introduction, mechanisms used in industry can be divided into 4 groups.

- 1) Mechanisms inside camera (described in Chapter 1)
- 2) Mechanisms inside of movie artefacts
- 3) Mechanisms in movie production
- 4) Camera motion controllers

### 2.2 Mechanisms inside of movie artefacts

Mechanisms in movies were very useful until computer-generated imagery (CGI) technology rendered them as obsolete. Usually they performed decorative and visual functions

One of the well-known examples can be seen in the movie “Jaws”, 1975.



*Figure 2-1 Shark, from movie "Jaws", 1975*

This shark mechanism was created by visual effect master Robert “BOB’ Matthey (1910-1993) [8].

Also famous monster in cinema made by engineers were a dinosaurs from movie franchise “Jurassic Park” (1993, 1997, 2001). They were motioned controlled robots, first of that kind. (see Figure 2-3).

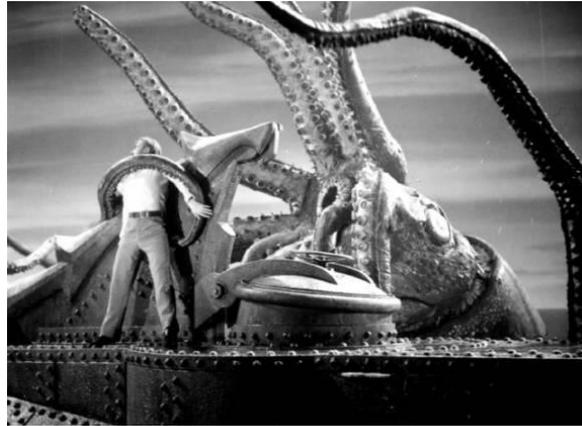


Figure 2-2 Kraken from "20,000 Leagues under the sea", 1954



Figure 2-4 Robotic deer from "Santa-Claus" 1994

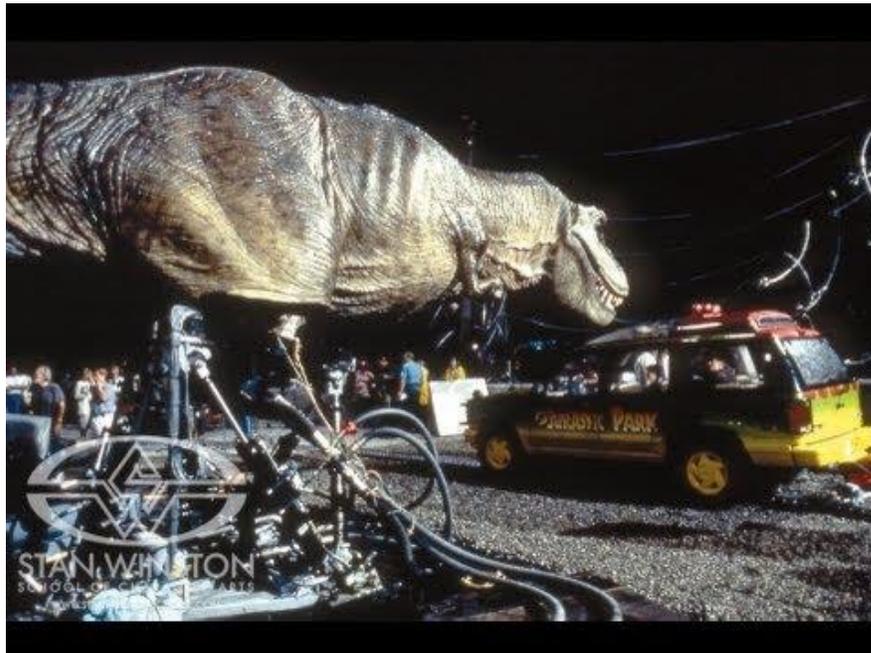


Figure 2-3 Dinosaur from "Jurassic Park" 1993



Figure 2-5 Submarine engine. "Vynález zkázy " 1958

### 2.3 Mechanisms in movie production

In this category mechanisms (typically mechanical manipulators-robots) are usually not visible in final movie product. These mechanisms help to make visual effects, like moving or supporting elements of decorations in movies, and during post-production they "disappear".

Very popular robots for that usage are parallel manipulators. This kind of mechanisms helps movie technicians to reach desired motion of subjects.



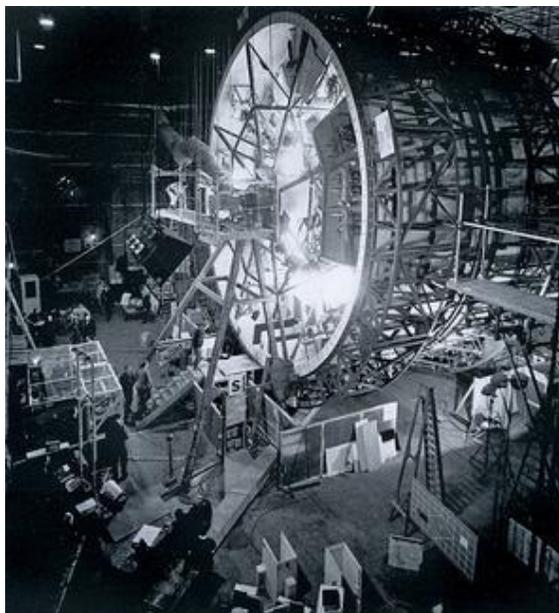
Figure 2-6 Jet's parallel manipulator "True Lies", James Cameron, 1994.



*Figure 2-7 Rotating hall "Inception", 2010*



*Figure 2-8 Moving on the rolled road "Inception", 2010*



*Figure 2-9 Centrifugal machine "2001: A space Odyssey"*

## 2.4 Camera motion controllers

In Chapter 1, Section 1.3 the presence of this kind of mechanisms in movie industry was described. Motion of camera can be controlled by this mechanisms automatically or manually.

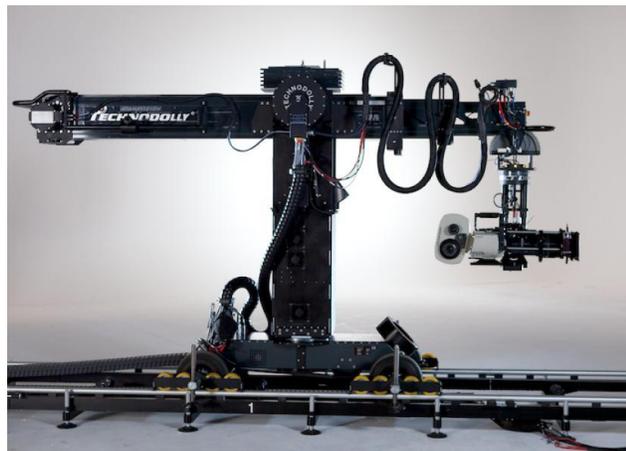
Old-fashion motion camera controllers are mostly manually controlled, but with the development of robotics, camera motion controllers gradually became automatically controlled. Modern controllers nowadays are often 6 axis robots. Their use enables more complicated camera motion control.

Basically all new camera motion controllers are the same. Difference is in sizes, weight and operations.

In this work we decided to choose one exact model and make kinematic analysis of it. For this purpose TECHNODOLLY® camera motion controller developed by Technocrane s.r.o., was chosen.



*Figure 2-10 "Milo" motion controller*



*Figure 2-11 TECHNODOLLY®*

## Chapter 3 Technodolly

TECHNODOLLY® is camera motion control robot, with possibility of manual usage. It has a variety of features that can help to create required complex camera motion. It was produced by Technocrane s.r.o. This company was founded about 20 years ago by Horst Burbulla and colleagues from city of Pilsen, Czech Republic. Horst Burbulla as representative of Technocrane has also won an Academy Award Oscar® in 2005 (77<sup>th</sup> Oscar ceremony). He holds several patents for camera crane technologies. [9] [10].

### 3.1 TECHNODOLLY® dimensions

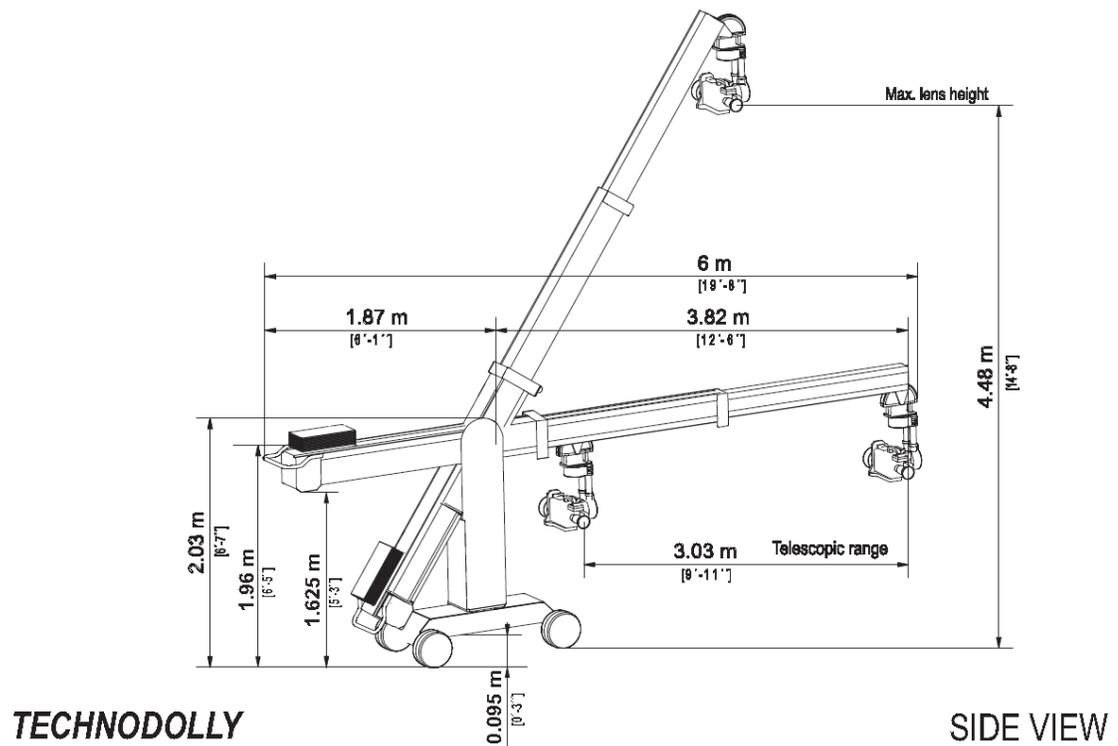


Figure 3-1 Technodolly dimensions [18]

All important dimensions of Technodolly camera crane are shown in Figure 3-1.

Track length can be extended from 6 meters up to 18 meters. Railway track width is 0.88 meters.

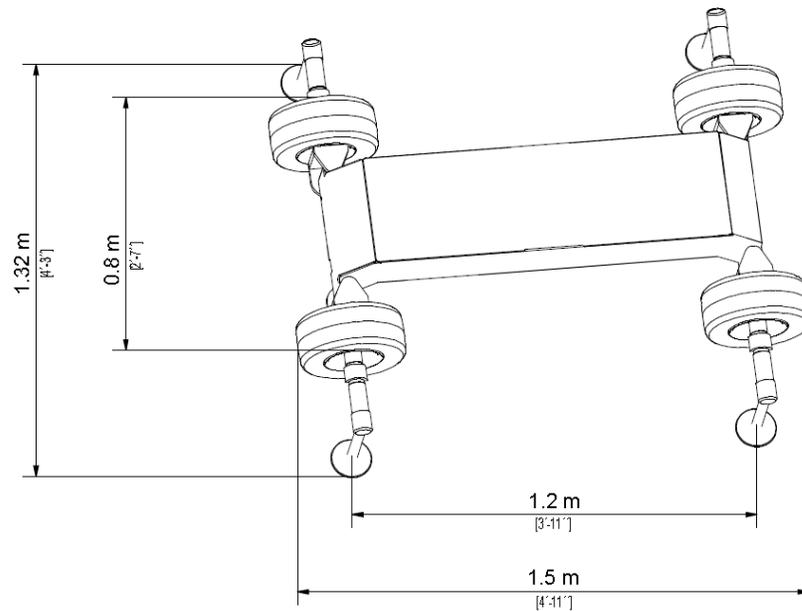


Figure 3-2 Dolly view

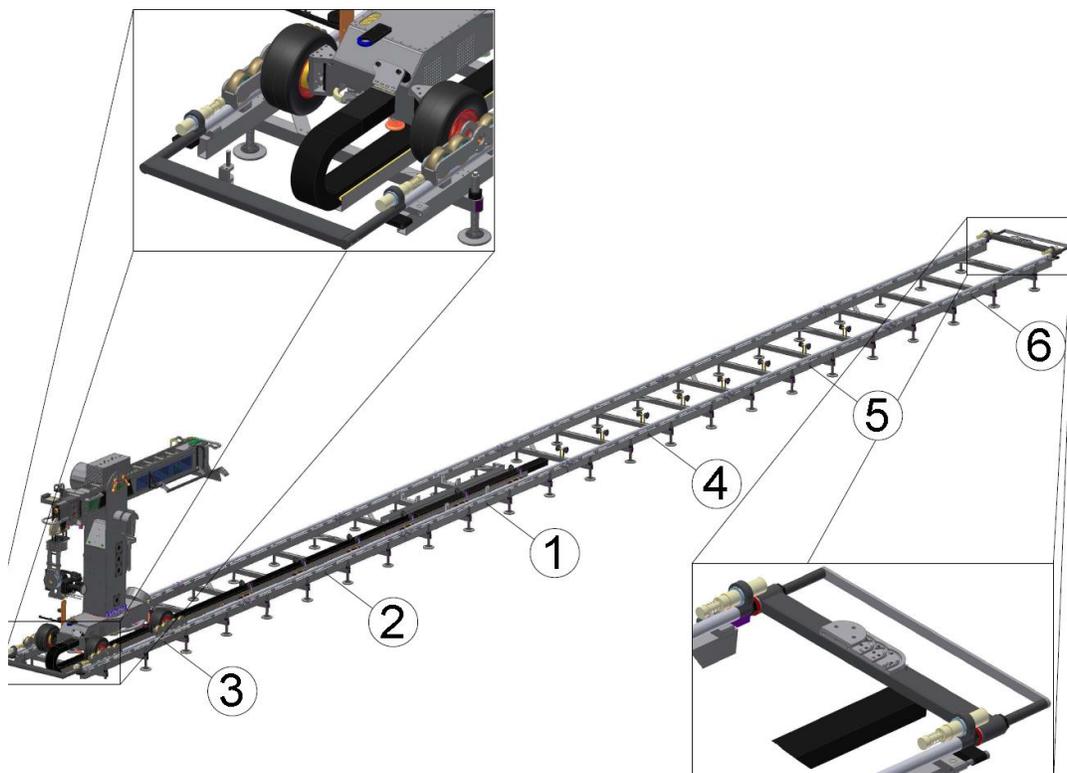
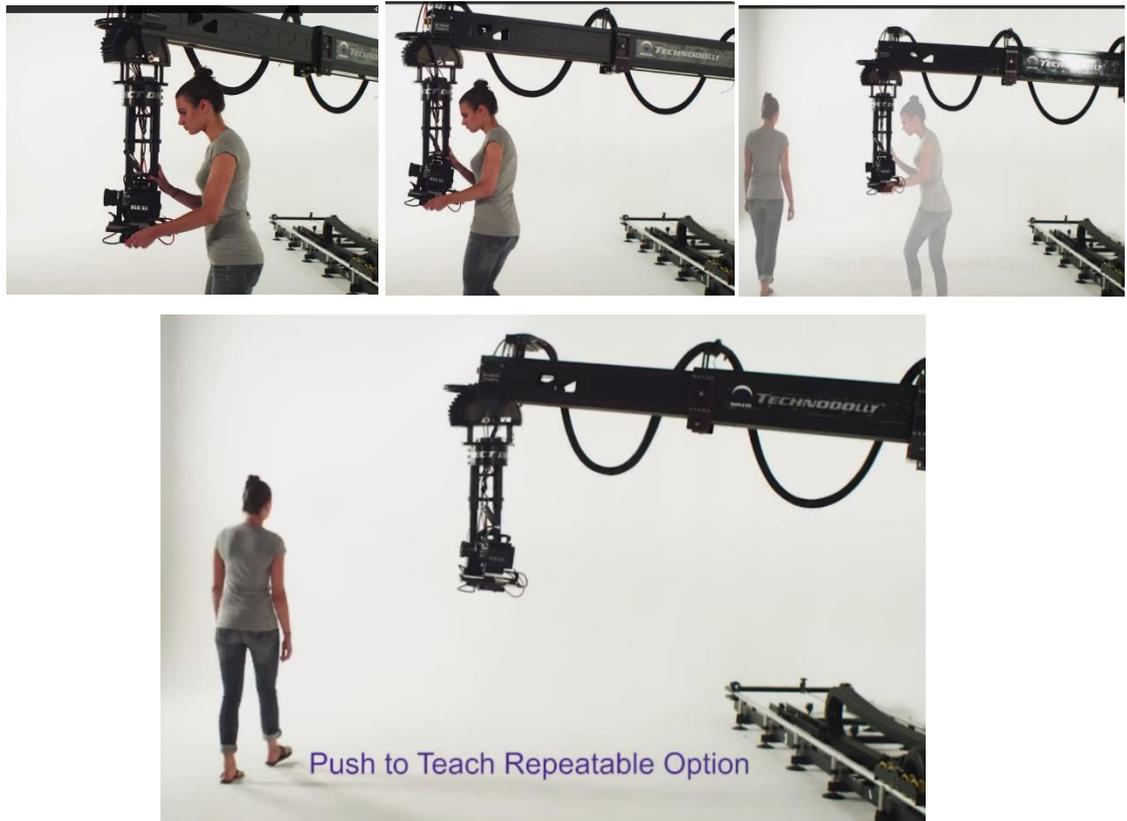


Figure 3-3 18 meter track, consist of 6 parts, 3 m each

## 3.2 Technodolly® 's crane features

### 3.2.1 Teach in

One of the most interesting options of the TECHNODOLLY®, is that it can memorize motion of camera made by human holding the camera, so-called “teach-in”. For example, you hold the camera that is installed to the crane and make a certain motion. TECHNODOLLY then enables to repeat precisely that motion. For better clarity see Figures 3-4



Figures 3-4 Teach in. [26]

### 3.2.2 Memory

Maximum amount of input positions is limited by 99 movements. All these positions can be memorized by Technodolly®. Any one of these 99 camera movements can be entered directly and replayed or altered. [11, p. 47]

### 3.2.3 Camera holder mechanism

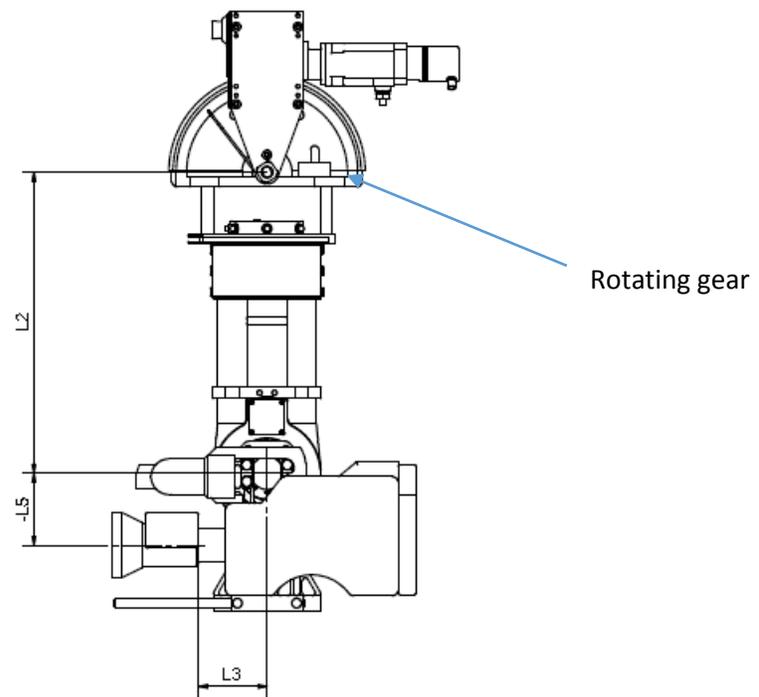


Figure 3-5 Camera holder

Rotating controlled gear is shown in Figure 3-5 and it is set to be always parallel to the ground, in order to simplify the operation of TECHNODOLLY®.

To calculate and program TECHNODOLLY®, developer should become familiar with robotics and programming (generally C/C++, C#)

## Chapter 4 Fundamentals of kinematic theory.

### 4.1 Introduction

**Mechanics** - it is the field of science that deals with energy, forces, moments and torques and their effect on the motion of mechanical system. Mechanics can be divided into 3 important areas: statics, kinematics, and dynamics. [12, p. 45]

**Mechanism** – it is set of rigid bodies, which are jointed (connected) together, usually by kinematic pair joints (see below), in order to make the needed motion, force transmission, energy transmission, motion transmission or each of them separately.

**Kinematic pairs** – are formed by two bodies whose relative motion is bounded by a constraint. There exist several kinds of constraints for connection of rigid bodies. The main difference between different types of kinematic pairs (joints) is the number of degrees of freedom i.e. what kind of motion does this joint allow to do. In 3D space, if body has no constraints and can move freely anywhere, it means that it has 6 degrees of freedom. It can move in x-direction, y-direction, z- direction and can rotate around each of these axis. 6 degrees of freedom is the maximum. In 2D this number is 3. X-direction, Y – direction, and rotation motion.

*j-class of joint* – is a number of motions that kinematic pair prevents. If a joint has  $n$  degrees of freedom ( $n$  relative allowable motions), then the class of joint is determined as  $j=(6-n)$  in 3D space,  $j=(3-n)$  in 2D space

- 1) Revolute joint – It is a type of kinematic pair that allows only single relative rotation between two bodies. All other rotations and translations are prevented. Ideal revolute joint is 5<sup>th</sup> class joint.

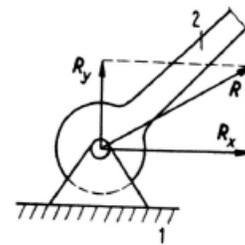
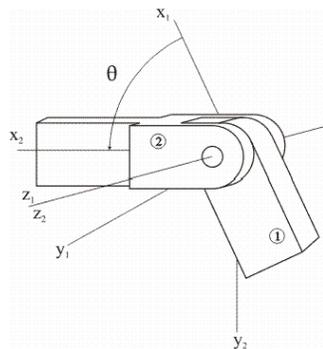


Figure 4-1a Revolute Joint 3D representation      Figure 4-1b Revolute joint 2D representation [13, p. 86]

- 2) Prismatic joint – or also known as sliding joint, translational joint. It allows two connected elements to slide with respect to each other along an axis that is defined by the geometry of the kinematic pair. [12, p. 7]. Ideal sliding pair is the 5<sup>th</sup> class joint.

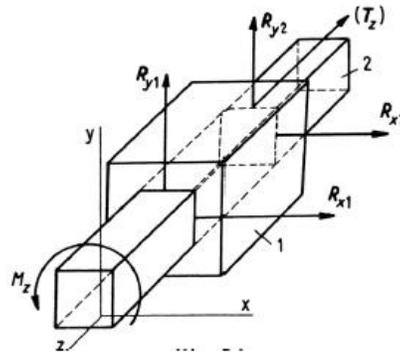


Figure 4-2 Prismatic joint [13, p. 87]

- 3) Cylindrical joint – allows to rotate connected body and make a translation motion on the defined axis of joint geometry. Ideal cylindrical joint is the 4<sup>th</sup> class joint.

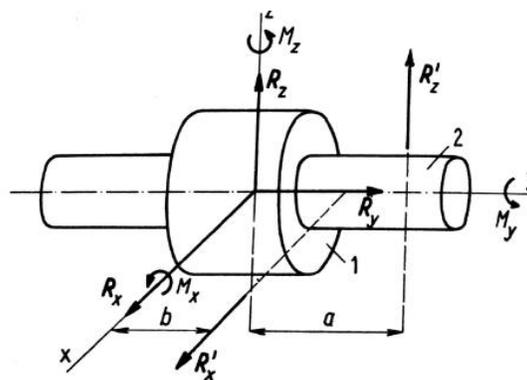


Figure 4-3 Cylindrical joint

The most common type of joints are along with their classifications are shown in Table 2.

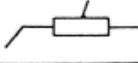
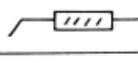
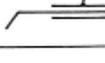
Pair name	Pair chart symbol	Pair symbol	Degree of freedom	Class
revolute		R	1	5
prismatic		P	1	5
helical		H	1	5
cylindrical		C	2	4
spherical		S	3	3
flat		F	3	3
general		G	5	1
arbitrary		A	i	6 - i

Table 2 Types of joints [14, p. 23]

**Degrees of freedom** – number of independent coordinates (parameters) needed to uniquely determine the configuration of a mechanism.

Gruebler-Kutzbach formula to determine degrees of freedom for mechanism [14]:

$$i = m(n - 1) - \sum_{j=1}^m j d_j \quad (4.1)$$

Where: m is the maximum allowable motion of one body.

m= 6 in 3D, m=3 in 2D

n – is number of elements

j – class of joint

d<sub>j</sub> – number of joints of specific kind.

**Kinematic chain** – is a set of links jointed together. Individual body of a multi body system counts as one element. One element connects with neighborhood elements by 1 or more kinematic joints. A number of bodies connected with each other by kinematic joints form *kinematic chains*.

*Closed loop kinematic chain* is a kinematic chain if there is one or more loops. Loops forms if every link is connected to every other link by at least two distinct path. See Figure 4-5

*Open kinematic chain* is a kinematic chain, where there is no loops. See Figure 4-4

*Mixed kinematic chain* is a kinematic chain where is a combination of closed loop and open kinematic chains.

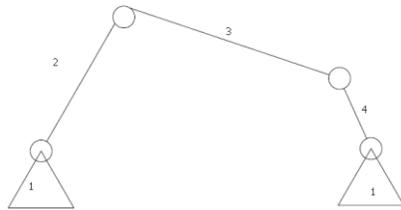


Figure 4-5 Closed loop chain

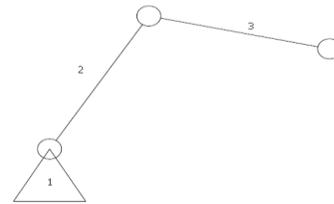


Figure 4-4 Open kinematic chain

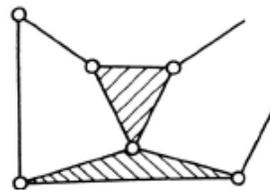
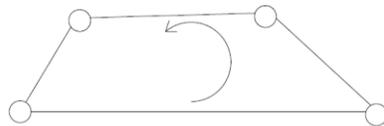


Figure 4-6 Mixed kinematic chain

Number of independent kinematic loops of mechanism is given

$$l = d - n + 1 \quad (4.2)$$

$d$  number of kinematics joint,

$n$  degrees of freedom.

**Multi body system** – or simply *MBS* is assembly of bodies connected with each other by kinematic joints.

## 4.2 Kinematics

Kinematics deals with motion analysis without considering masses, forces, moments and torques. Goal of kinematics is to analyse position, velocity and acceleration of point (body or multi body system) and their mutual interrelationships.

### 4.2.1 Position

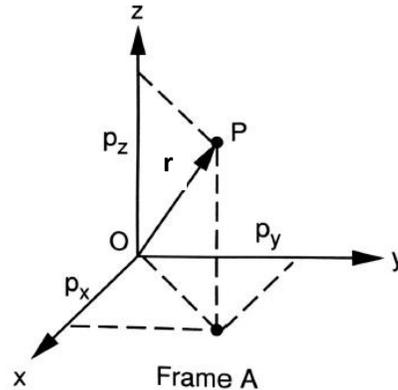


Figure 4-7 Point P in space with reference A

Position of a point with respect to some relative reference frame given by position vector, and it can be stated as 3 x 1 matrix. For example position of point P is described by radius vector  $r$ .

$${}^A\mathbf{r} = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} \quad (4.3)$$

Subscripts  $x, y, z$  represent a projection of the radius vector onto the three coordinates axes of reference frame.

### 4.2.2 Orientation

The orientation in fixed space of some certain rigid body can be described by several ways. The widely used description are Euler and Cardan angles.

Since the rotation is motion with 3 degrees of freedom, we need 3 independent parameters to describe the orientation of a body [15]. Let introduce 3 basic rotation matrices from which an Euler angles can be derived in terms of their product.

If a body performs rotation around z-axis, it means that  $s_x=s_y=0$  and  $s_z=1$ . Hence:

$$S_z(\varphi_z) = \begin{bmatrix} c\varphi_z & -s\varphi_z & 0 \\ s\varphi_z & c\varphi_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.4)$$

If rotation is happening around x-axis, analogically as in previous one  $s_z=s_y=0$  and  $s_x=1$ .

$$S_x(\varphi_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\varphi_x & -s\varphi_x \\ 0 & s\varphi_x & c\varphi_x \end{bmatrix} \quad (4.5)$$

Around y-axis,  $s_x=s_z=0$  and  $s_y=1$ .

$$S_y(\varphi_y) = \begin{bmatrix} c\varphi_y & 0 & s\varphi_y \\ 0 & 1 & 0 \\ -s\varphi_y & 0 & c\varphi_y \end{bmatrix} \quad (4.6)$$

Successive rotations around the axes z-x-z are the so-called Euler angles. We have two frames: Fixed frame A with coordinate axes x,y,z and moving frame B with coordinate axes u,v,w. Let's consider three rotation of a body about the coordinate axes of moving frame B. Beginning with frame B being coinciding with the fixed frame A, we rotate B about the body-attached w-axis by an angle  $\varphi_w$ . Second rotation is rotation of angle  $\varphi_{u'}$  about u' axis. Third rotation of  $\varphi_{w''}$  is being about w'' axis. Procedure is shown in Figure 4-8.

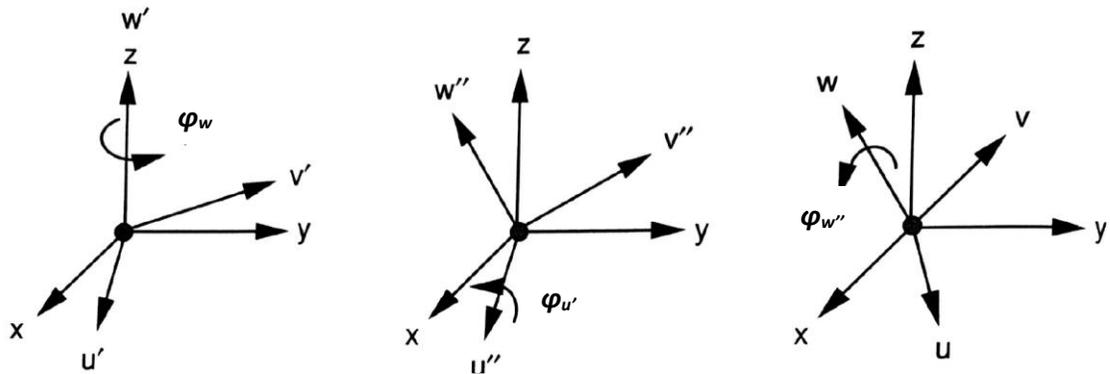


Figure 4-8 rotation about the moving coordinate axes. z-x-z

<sup>2</sup>  $c\varphi = \cos \varphi$ ,  $s\varphi = \sin \varphi$

The resultant rotation matrix can be expressed as:

$$\mathbf{S}_{AB}(\varphi_w, \varphi_{u'}, \varphi_{w''}) = \mathbf{S}_w(\varphi_w) \mathbf{S}_u(\varphi_{u'}) \mathbf{S}_w(\varphi_{w''})$$

$$\mathbf{S}_{AB} = \begin{bmatrix} c\varphi_w c\varphi_{w''} - s\varphi_w c\varphi_{u'} s\varphi_{w''} & -c\varphi_w s\varphi_{w''} - s\varphi_w c\varphi_{u'} c\varphi_{w''} & s\varphi_w s\varphi_{u'} \\ s\varphi_w c\varphi_{w''} + c\varphi_w c\varphi_{u'} s\varphi_{w''} & -s\varphi_w s\varphi_{w''} + c\varphi_w c\varphi_{u'} c\varphi_{w''} & -c\varphi_w s\varphi_{u'} \\ s\varphi_{u'} s\varphi_{w''} & s\varphi_{u'} c\varphi_{w''} & c\varphi_{u'} \end{bmatrix} \quad (4.7)$$

Another type of Euler angles consist of rotation of angle  $\varphi_w$  about w-axis, then second rotation of angle  $\varphi_{v'}$  about v'-axis, followed by a third rotation of  $\varphi_{w''}$  about w''-axis. The resulting rotation matrix is obtained by multiplication of basic rotation matrices:

$$\mathbf{S}_{AB}(\varphi_w, \varphi_{v'}, \varphi_{w''}) = \mathbf{S}_w(\varphi_w) \mathbf{S}_v(\varphi_{v'}) \mathbf{S}_w(\varphi_{w''})$$

$$\mathbf{S}_{AB} = \begin{bmatrix} c\varphi_w c\varphi_{v'} c\varphi_{w''} - s\varphi_w s\varphi_{w''} & c\varphi_w c\varphi_{v'} s\varphi_{w''} - s\varphi_w c\varphi_{w''} & c\varphi_w s\varphi_{v'} \\ s\varphi_w c\varphi_{v'} c\varphi_{w''} + c\varphi_w s\varphi_{w''} & -s\varphi_w c\varphi_{v'} s\varphi_{w''} + c\varphi_w c\varphi_{w''} & s\varphi_w s\varphi_{v'} \\ -s\varphi_{v'} c\varphi_{w''} & s\varphi_{v'} s\varphi_{w''} & c\varphi_{v'} \end{bmatrix} \quad (4.8)$$

Roll-Pitch-Yaw angles or also known as Cardan angles. It is rotation of moving frame B about coordinate axes of the fixed frame A. Starting with frame B coinciding with frame A, we rotate B about x-axis by an angle  $\varphi_x$ , then we rotate B about y-axis by an angle  $\varphi_y$  and last we rotate B about z-axis by an angle  $\varphi_z$ . Since all rotations take a place about axes of fixed frame A, the resulting rotation matrix is:

$$\mathbf{S}(\varphi_x, \varphi_y, \varphi_z) = \mathbf{S}_z(\varphi_z) \mathbf{S}_y(\varphi_y) \mathbf{S}_x(\varphi_x)$$

$$\begin{bmatrix} c\varphi_z c\varphi_y & c\varphi_z s\varphi_y s\varphi_x - s\varphi_z c\varphi_x & c\varphi_z s\varphi_y c\varphi_x + s\varphi_z s\varphi_x \\ s\varphi_z c\varphi_y & s\varphi_z s\varphi_y s\varphi_x + c\varphi_z c\varphi_x & s\varphi_z s\varphi_y c\varphi_x - c\varphi_z s\varphi_x \\ -s\varphi_y & c\varphi_y s\varphi_x & c\varphi_y c\varphi_x \end{bmatrix} \quad (4.9)$$

#### 4.2.3 Homogeneous transformation matrices

Homogeneous transformation matrices serve for concise description of respective motion (rotation and relative displacement) between respective reference frames.

Homogeneous transformation matrix can generally be expressed as 4x4 matrix:

$$\mathbf{T} = \begin{bmatrix} \mathbf{S}_{ij} & \mathbf{r} \\ \mathbf{0} & 1 \end{bmatrix} \quad (4.10)$$

Basic motions can be described as follows.

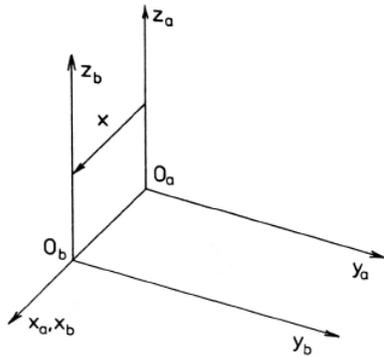


Figure 4-9 Basic motion along X

Translation motion along x-axis

$$T_x = \begin{bmatrix} 1 & 0 & 1 & x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

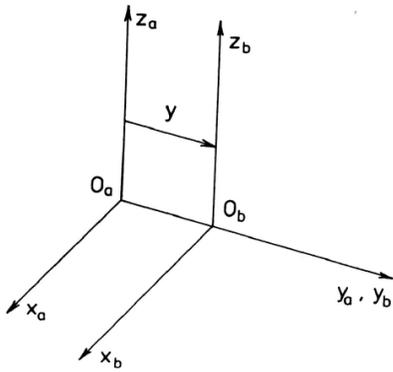


Figure 4-10 Basic motion along Y

Translation motion along y-axis

$$T_y = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.12)$$

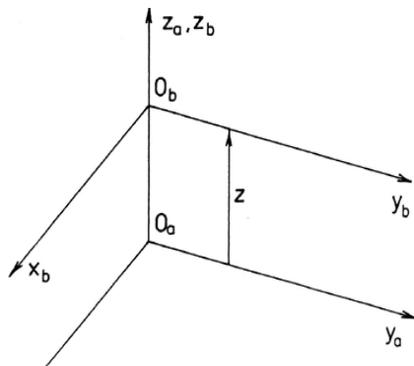


Figure 4-11 Basic motion along Z

Translation motion along z-axis

$$T_z = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.13)$$

### Revolute motion around x-axis

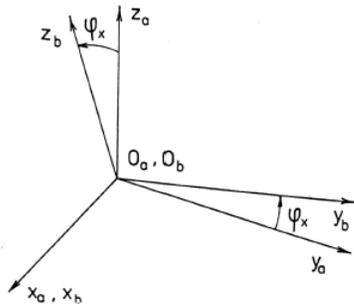


Figure 4-12 Revolution around X

$$T_{\theta_x} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\varphi_x & -s\varphi_x & 0 \\ 0 & s\varphi_x & c\varphi_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

### Revolute around y-axis

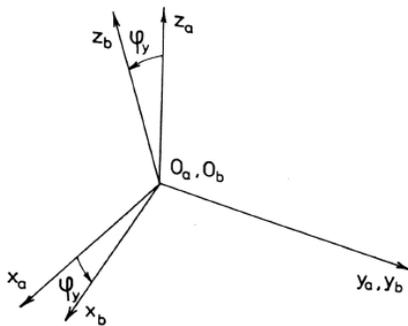


Figure 4-13 Revolute around Y

$$T_{\theta_y} = \begin{bmatrix} c\varphi_y & 0 & s\varphi_y & 0 \\ 0 & 1 & 0 & 0 \\ -s\varphi_y & 0 & c\varphi_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.15)$$

### Revolute around z-axis

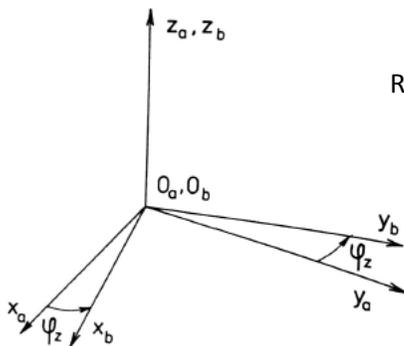


Figure 4-14 Revolute around Z

$$T_{\theta_z} = \begin{bmatrix} c\varphi_z & -s\varphi_z & 0 & 0 \\ s\varphi_z & c\varphi_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.16)$$

Position of a certain point can be expressed through basic matrices expressing the resultant position of a certain coordinate system body as a sequence of imagined basic motions from the initial to the final position regardless of whether or not these motions are realized by kinematic pair. [14, p. 152]

The motion of point L in basic space can then be expressed as

$${}^1\mathbf{r}_L = \prod_{j=1}^N \mathbf{T}_{k_j}(p_j) {}^n\mathbf{r}_L \quad (4.17)$$

### 4.3 Matrix method for solving Position Problem

Matrix method provides equations necessary to compute dependent coordinates  $\mathbf{z}$  as a function of independent coordinates  $\mathbf{q}$ .

From the total amount of coordinates we have to distinguish independent coordinates from dependent coordinates. Independent coordinates would be denoted as  $\mathbf{q}$  and dependent coordinates will be denoted as  $\mathbf{z}$ .  $\mathbf{q}$  are the known parameters and  $\mathbf{z}$  are dependent coordinates of driven kinematic pairs to be found.

$$z_i = z_i(\mathbf{q}), \quad i = 1 \text{ to } 6 \quad (4.18)$$

**Closed Loop Method** is one of the ways to find the set of equations in order to compute dependent coordinates as a function of  $\mathbf{q}$

$$\mathbf{T}_{1n} = \mathbf{T}_{12} \mathbf{T}_{23} \dots \mathbf{T}_{n-1,n} \quad (4.19)$$

Closed loop formula can be rearranged as :

$$\mathbf{T}_{12} \mathbf{T}_{23} \dots \mathbf{T}_{n-1,n} \mathbf{T}_{n1} = \mathbf{E}_4 \quad (4.20)$$

Where  $\mathbf{E}_4$  is identity matrix 4 x 4.

Formula (4.21) says that mechanism contains a closed loop.

Left hand side of the equation (4.21) is 4 x 4 matrix consisting of  $\mathbf{q}$  and  $\mathbf{z}$ . Equating elements of the matrix with identical elements of  $\mathbf{E}_4$ , we would get 16 equations, out of which 4 are identities. (see Figure 4-15 )

$$\begin{bmatrix} s_{11} & s_{12} & s_{13} & d_{14} \\ s_{21} & s_{22} & s_{23} & d_{24} \\ s_{31} & s_{32} & s_{33} & d_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad s_{12} = 0, d_{24} = 0, \dots$$

Figure 4-15  $f(q,z)=0$  construction

Out of remaining 12 equations, only 6 equations are independent due to orthonormality of  $S_{ij}$  matrix.

Then we obtain nonlinear equations for solving kinematic problem

$$f_k(\mathbf{z}, \mathbf{q}) = 0, \quad k = 1 \text{ to } 6 \quad (4.21)$$

## 4.4 Numerical solution (Theory)

After development of equations (4.21) we can begin to solve this set of equations, in order to find a set of dependent coordinates  $\mathbf{z}$ , by Newton-Raphson method. The system of equations is solved until each of the residual errors are less than required number  $\epsilon$ . [16, p. 124]

The Newton-Raphson method is based on a linearization of the system (4.22) and consists in replacing this system of equations with the first two terms of its expansion of a Taylor series around a certain estimated approximation  $\mathbf{z}_i$ . [17, p. 74]

We have

$$\mathbf{z}^{(k)} = [z_1^{(k)} \ z_2^{(k)} \ \dots \ z_n^{(k)}]^T$$

$$f(\mathbf{z}^{(k)}, \mathbf{q}) = f(\mathbf{z}^{(k)}, \mathbf{q}) + \frac{\partial f}{\partial \mathbf{z}^T} \Delta \mathbf{z}^{(k)} = 0 \quad (4.22)$$

For initial (first iteration loop) estimate  $k=0$ .

Jacobian of the system (4.21) according with respect to  $\mathbf{z}$  values:

$$\Phi_{\mathbf{z}}(\mathbf{z}, \mathbf{q}) = \frac{\partial f(\mathbf{z}, \mathbf{q})}{\partial \mathbf{z}^T}$$

$$\Phi_{\mathbf{z}} = \Phi_{\mathbf{z}}(\mathbf{z}^{(k)}, \mathbf{q}) \quad (4.23)$$

$$\Phi_{\mathbf{z}} \Delta \mathbf{z}^{(k)} = -f(\mathbf{z}^{(k)}, \mathbf{q}) \quad (4.24)$$

If the Jacobian  $\Phi_{\mathbf{z}}$  is non-singular matrix, system will be solved as follows.

$$\Delta \mathbf{z}^{(k)} = -\Phi_{\mathbf{z}}^{-1} f(\mathbf{z}^{(k)}, \mathbf{q}) \quad (4.25)$$

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \Delta \mathbf{z}^{(k)} \quad (4.26)$$

The whole procedure is to be iterated again with  $k=k+1$

However Newton-Raphson's method converges only if we start with proper initial (correctly found) values of  $\mathbf{z}^{(0)}$ . Using values that are not precise enough might lead to divergence or a jump to another solution. A very advantageous modification is made by adding "line-search" parameter lambda

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \lambda \cdot \Delta \mathbf{z}^{(k)} \quad (4.27)$$

Where  $\lambda$  is a scalar parameter, and its value obtained from one-dimensional minimization (linear search):

$$|\mathbf{f}(\mathbf{z}^{(k+1)}, \mathbf{q})| = \sqrt{\sum_{i=1}^n f_i^2(\mathbf{z}^{(k+1)}, \mathbf{q})} \quad (4.28)$$

which is function of parameter  $\lambda$  only. Procedure of calculation is simple. First we start with  $\lambda=1$ , then if calculation didn't give us required values (conditions weren't satisfied),  $\Delta \mathbf{z}^{(k)} \leq \varepsilon$ , next step of iteration will be with  $\lambda=\lambda/2$ . And so on, till we didn't reach needed goal.

## Chapter 5 Numerical solution

During the work the base of the TECHNODOLLY® has to be fixed by leveling jacks or to be on the railway track. Otherwise the dolly will not be stable. When it is fixed, the base must be positioned horizontally and the wheel steering must be blocked with the locking rod (Figure 5-1). [18]

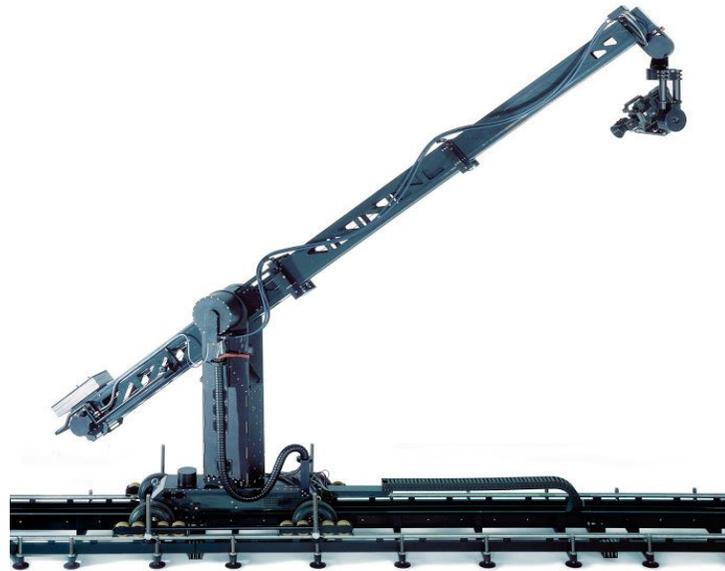


*Figure 5-1 Base with jacks and locking rod*

Modification with base standing on the track shown in Figure 5-2 and Figure 5-3.



*Figure 5-2 Track wheels on track [11, p. 20]*



*Figure 5-3 Technodolly on track*

We are going to choose modification of Technodolly, what it is on the railway track, as the main task. The goal is to solve assembled kinematic model by numeric solution.

First of all we need to determine what are dependent coordinates and independent coordinates. For this purpose we are going to use Figure 5-4.

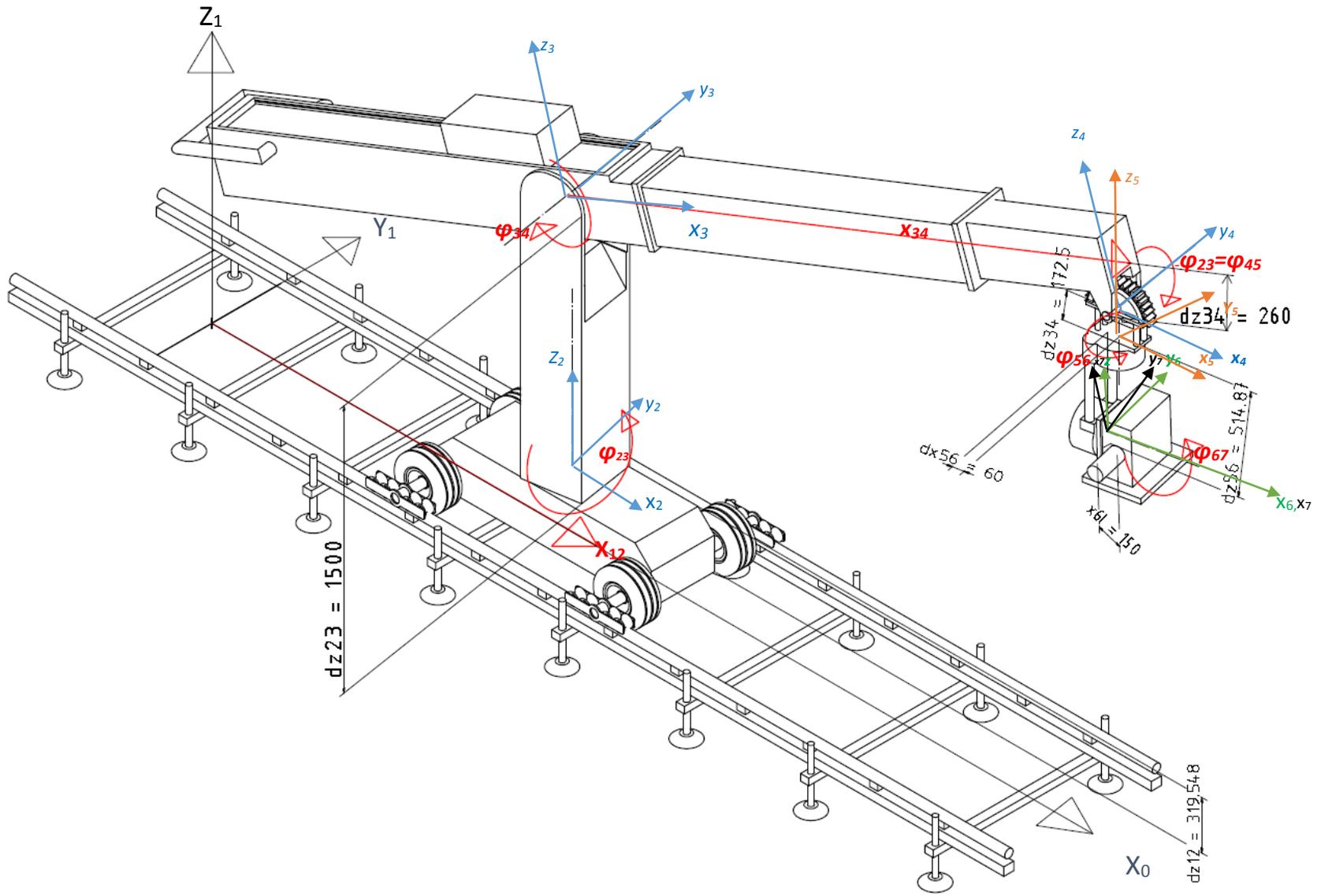


Figure 5-4 Dolly on track

Constant parameters for this case are:

$$\begin{aligned} d_{z12} = 319.548 [mm], d_{z23} = 1500[mm], d_{z34} = 260[mm], d_{z45} = 172.5 [mm], \\ d_{z56} = 514.87 [mm], d_{x56} = 60 [mm], d_{x6L} = 150 [mm], d_{y6L} = 320 [mm], d_{z6L} = \\ 35 [mm] \end{aligned} \quad (5.1)$$

$x_{12}, \varphi_{12}, \varphi_{23}, x_{34}, \varphi_{45}, \varphi_{56}, \varphi_{6L}$ : are to be computed parameters  $\mathbf{z}$ , they are unknown and they are colored red in the Figure 5-4. The number of dependent coordinates (unknowns) is seven. In order to eliminate this problem we decide angle  $\varphi_{45}$  to be always equal to  $-\varphi_{34}$ , as it was mentioned in Chapter 3, Section 3.2.3.

Using this, we obtain 6 dependent parameters, instead of 7. They will be represented in the matrix form as:

$$\mathbf{z} = [x_{12} \ \varphi_{12} \ \varphi_{23} \ x_{34} \ \varphi_{56} \ \varphi_{67}]^T \quad (5.2)$$

For mechanism shown in Figure 5-4 transformation matrices are listed below:

$$\mathbf{T}_{12} = \begin{bmatrix} 1 & 0 & 0 & x_{12} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_{z12} \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.3)$$

$$\mathbf{T}_{23} = \begin{bmatrix} c\varphi_{23} & -s\varphi_{23} & 0 & 0 \\ s\varphi_{23} & c\varphi_{23} & 0 & 0 \\ 0 & 0 & 1 & d_{z23} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.4)$$

$$\mathbf{T}_{34} = \begin{bmatrix} c\varphi_{34} & 0 & s\varphi_{34} & x_{34} \\ 0 & 1 & 0 & 0 \\ -s\varphi_{34} & 0 & c\varphi_{34} & -d_{z34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.5)$$

$$\mathbf{T}_{45} = \begin{bmatrix} c(-\varphi_{34}) & 0 & s(-\varphi_{34}) & 0 \\ 0 & 1 & 0 & 0 \\ -s(-\varphi_{34}) & 0 & c(-\varphi_{34}) & -d_{z45} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

$$\mathbf{T}_{56} = \begin{bmatrix} c\varphi_{56} & -s\varphi_{56} & 0 & -d_{x56} \\ s\varphi_{56} & c\varphi_{56} & 0 & 0 \\ 0 & 0 & 1 & -d_{z56} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.7)$$

$$\mathbf{T}_{67} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c\varphi_{67} & -s\varphi_{67} & 0 \\ 0 & s\varphi_{67} & c\varphi_{67} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.8)$$

$${}^7\mathbf{r}_L = \begin{bmatrix} d_{x7L} \\ d_{y7L} \\ d_{z7L} \end{bmatrix} \quad (5.9)$$

The motion of end point L, according to the formula (4.18), can be represented as a sequence of transformation matrices (5.3-5.10) from the initial position  $(x_0, y_0, z_0)$  to the end-point  $(x_L, y_L, z_L)$ .

$${}^1\mathbf{r}_L = \mathbf{T}_{12} \mathbf{T}_{23} \mathbf{T}_{34} \mathbf{T}_{45} \mathbf{T}_{56} \mathbf{T}_{67} {}^7\mathbf{r}_L \quad (5.10)$$

Independent coordinates of driving kinematic pairs are denoted as  $\mathbf{q}$ . In our specific case those parameters are end-effector coordinates in base frame. These are  $q_x, q_y, q_z, \varphi_x, \varphi_y, \varphi_z$

Where  $q_x, q_y, q_z$  are coordinates (distances) of end point.

$\varphi_x, \varphi_y, \varphi_z$  are responsible for camera orientation.  $\varphi_x$  is rotation around x-axis,  $\varphi_y$  is rotation around y-axis and  $\varphi_z$  is rotation around z-axis.

$$\begin{aligned} \mathbf{q} &= [q_1, q_2, \dots, q_n]^T \\ \mathbf{q} &= [q_x, q_y, q_z, \varphi_x, \varphi_y, \varphi_z]^T \end{aligned} \quad (5.11)$$

The general motion equation is expressed as:

$$\mathbf{T}_{17} = \mathbf{T}_{12} \mathbf{T}_{23} \mathbf{T}_{34} \mathbf{T}_{45} \mathbf{T}_{56} \mathbf{T}_{67} \quad (5.12)$$

$$\mathbf{T}_{17} = \begin{bmatrix} c(\varphi_{56} + \varphi_{23}) & -c(\varphi_{67})s(\varphi_{56} + \varphi_{23}) & s(\varphi_{6L})s(\varphi_{56} + \varphi_{23}) & p_x \\ s(\varphi_{56} + \varphi_{23}) & c(\varphi_{67})c(\varphi_{56} + \varphi_{23}) & -s(\varphi_{6L})c(\varphi_{56} + \varphi_{23}) & p_y \\ 0 & s(\varphi_{67}) & c(\varphi_{67}) & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.13)$$

Where:

$$p_x = x_{12} - d_{x56} \cdot c({}^z\varphi_{56} + {}^z\varphi_{23}) + c({}^z\varphi_{23}) c({}^y\varphi_{34})x_{34} - c({}^z\varphi_{23}) s({}^y\varphi_{34}) \cdot d_{z34}$$

$$p_y = -d_{x56} \cdot s({}^z\varphi_{56} + {}^z\varphi_{23}) + s({}^z\varphi_{23}) c({}^y\varphi_{34}) \cdot x_{34} - s({}^z\varphi_{23}) s({}^y\varphi_{34}) \cdot d_{z34}$$

$$p_z = -d_{z34} \cdot c\varphi_{34} - x_{34} \cdot s\varphi_{34} + d_{z12} + d_{z23} - d_{z45} - d_{z56}$$

Due to virtual closed kinematic loop it has to be equal to the matrix  $\mathbf{H}$  which is function of variables  $\mathbf{q}$ .

$$\mathbf{H} = \mathbf{T}_x \mathbf{T}_y \mathbf{T}_z \mathbf{T}_{\varphi_z} \mathbf{T}_{\varphi_y} \mathbf{T}_{\varphi_x}$$

$$\mathbf{H} = \begin{bmatrix} c\varphi_y c\varphi_z & s\varphi_x s\varphi_y - c\varphi_y c\varphi_x s\varphi_z & c\varphi_x s\varphi_y + c\varphi_y s\varphi_x s\varphi_z & q_x \\ s\varphi_z & c\varphi_x c\varphi_z & -c\varphi_z s\varphi_x & q_y \\ 0 & s\varphi_x & c\varphi_x & q_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.14)$$

Closed loop equation for selected mechanism is represented in equation:

$$\mathbf{T}_{17} = \mathbf{H} \quad (5.15)$$

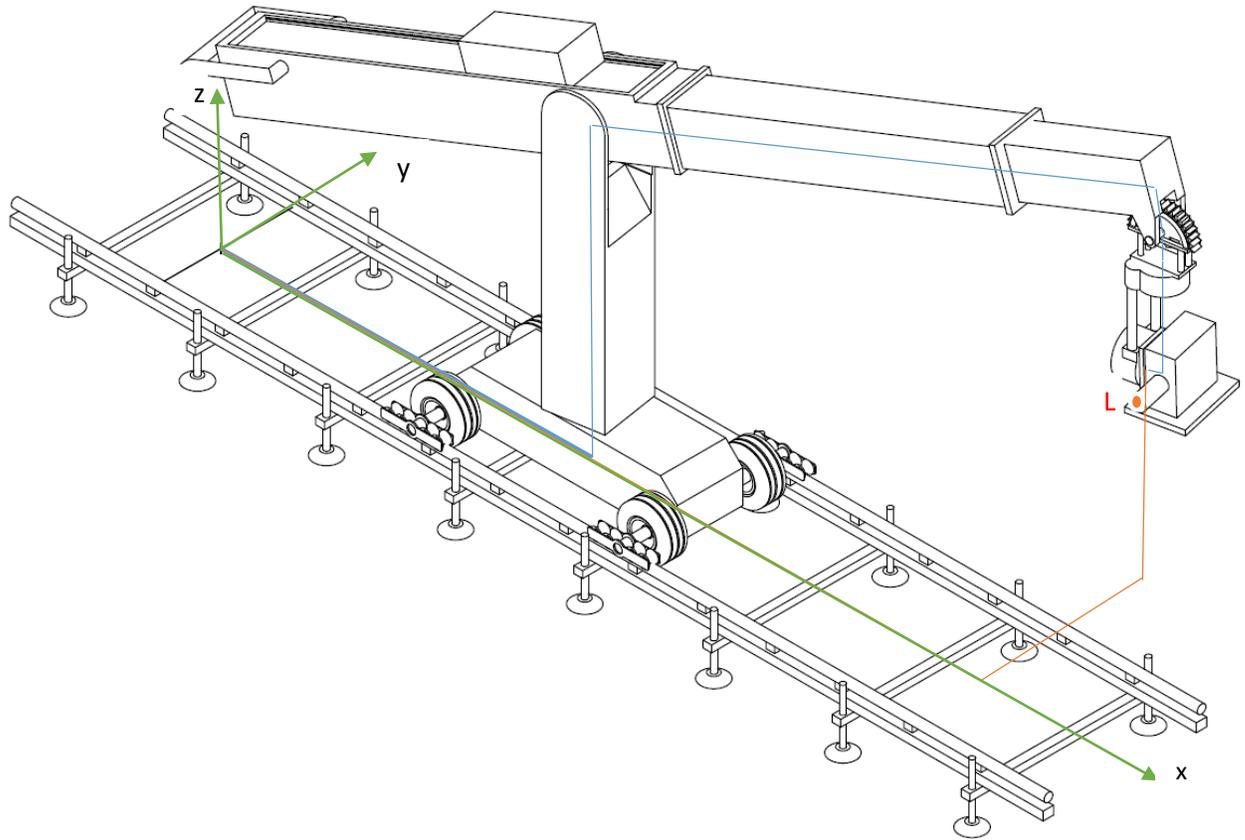


Figure 5-5 Loop of dolly on the track

- Path to L point through all mechanism (**z**)
- Path to L point immediately from base (**q**)

If we modify equation (5.15) in to form (4.21)

$$\mathbf{H}^{-1} \cdot \mathbf{T}_{17} = \mathbf{E}_4 \quad (5.16)$$

Where  $\mathbf{E}_4$  is identity matrix 4x4.

$H^{-1} \cdot T_{17} = E_4$  is then expresses as:

$$\begin{bmatrix} s_{11} & s_{12} & s_{13} & d_{14} \\ s_{21} & s_{22} & s_{23} & d_{24} \\ s_{31} & s_{32} & s_{33} & d_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.17)$$

Where:

$$s_{11} = s\varphi_z s(\varphi_{56} + \varphi_{23}) + c\varphi_y c\varphi_z c(\varphi_{56} + \varphi_{23})$$

$$s_{21} = s(\varphi_{56} + \varphi_{23})(c\varphi_x c\varphi_z + s\varphi_x s\varphi_z s\varphi_y) - c(\varphi_{56} + \varphi_{23})(c\varphi_x s\varphi_z - c\varphi_z s\varphi_x s\varphi_y)$$

$$s_{31} = c(\varphi_{56} + \varphi_{23})(s\varphi_x s\varphi_z + c\varphi_x c\varphi_z s\varphi_y) - s(\varphi_{56} + \varphi_{23})(c\varphi_z s\varphi_x - c\varphi_x s\varphi_y s\varphi_z)$$

$$s_{12} = c\varphi_y c\varphi_{67} s\varphi_z c(\varphi_{56} + \varphi_{23}) - c\varphi_y c\varphi_z c\varphi_{67} s(\varphi_{56} + \varphi_{23}) - s\varphi_y s\varphi_{67}$$

$$s_{22} = c\varphi_{67} s(\varphi_{56} + \varphi_{23})(c\varphi_x s\varphi_z - c\varphi_z s\varphi_x s\varphi_y) + c\varphi_{67} c(\varphi_{56} + \varphi_{23})(c\varphi_x \varphi_z + s\varphi_x s\varphi_z s\varphi_y) + c\varphi_y s\varphi_x s\varphi_{67}$$

$$s_{32} = c\varphi_x c\varphi_y s\varphi_{67} - c\varphi_{67} c(\varphi_{56} + \varphi_{23})(c\varphi_z s\varphi_x - c\varphi_x s\varphi_y s\varphi_z) - c\varphi_{67} s(\varphi_{56} + \varphi_{23})(s\varphi_x s\varphi_z + c\varphi_x c\varphi_z s\varphi_y)$$

$$s_{13} = c\varphi_y c\varphi_z s\varphi_{67} s(\varphi_{56} + \varphi_{23}) - c\varphi_{67} s\varphi_y - c\varphi_y s\varphi_z s\varphi_{67} c(\varphi_{56} + \varphi_{23})$$

$$s_{23} = c\varphi_y c\varphi_{67} s\varphi_x - s\varphi_{67} c(\varphi_{56} + \varphi_{23})(c\varphi_x c\varphi_z + s\varphi_x s\varphi_z s\varphi_y) - s\varphi_{67} s(\varphi_{56} + \varphi_{23})(c\varphi_x s\varphi_z - c\varphi_z s\varphi_x s\varphi_y)$$

$$s_{33} = s\varphi_{67} s(\varphi_{56} + \varphi_{23})(s\varphi_x s\varphi_z + c\varphi_x c\varphi_z s\varphi_y) + s\varphi_{67} c(\varphi_{56} + \varphi_{12})(c\varphi_z s\varphi_x - c\varphi_x s\varphi_y s\varphi_z) + c\varphi_x c\varphi_y c\varphi_{67}$$

$$d_{14} = [s\varphi_y(d_{z45} - d_{z23} - d_{z12} + d_{z34}c\varphi_{34} + d_{z56} + x_{34}s\varphi_{34})] - [q_x c\varphi_y c\varphi_z + q_y c\varphi_y s\varphi_z - q_z s\varphi_y] - [c\varphi_y s\varphi_z(d_{x56}s(\varphi_{56} + \varphi_{23}) + d_{z34}s\varphi_{23}s\varphi_{34} - x_{34}c\varphi_{34}s\varphi_{23})] + [c\varphi_y c\varphi_z(x_{12} - d_{x56}c(\varphi_{56} + \varphi_{23}) - d_{z34}c\varphi_{23}s\varphi_{34} + x_{34}c\varphi_{23}c\varphi_{34})]$$

$$d_{24} = -[q_x c\varphi_z s\varphi_x s\varphi_y + q_y s\varphi_x s\varphi_y s\varphi_z + q_y c\varphi_x c\varphi_z - q_x \varphi_x s\varphi_z + q_z c\varphi_y s\varphi_x] - [(c\varphi_x s\varphi_z - c\varphi_z s\varphi_x s\varphi_y)(x_{12} - d_{x56}c(\varphi_{56} + \varphi_{23}) - d_{z34}c\varphi_{23}s\varphi_{34} + x_{34}c\varphi_{23}c\varphi_{34})] - [(c\varphi_x c\varphi_z + s\varphi_x s\varphi_z s\varphi_y)(d_{x56}s(\varphi_{56} + \varphi_{23}) + d_{z34}s\varphi_{34}s\varphi_{23} - x_{34}s\varphi_{23}c\varphi_{34})] - [c\varphi_y s\varphi_x(d_{z45} - d_{z23} - d_{z12} + d_{z34}c\varphi_{34} + d_{z56} + x_{34}s\varphi_{34})]$$

$$\begin{aligned}
d_{34} = & [(c\varphi_z s\varphi_x - c\varphi_x s\varphi_y s\varphi_z)(d_{x56} s(\varphi_{56} + \varphi_{23}) + d_{z34} s\varphi_{23} s\varphi_{34} - x_{34} s\varphi_{23} c\varphi_{34})] \\
& - [q_x c\varphi_z s\varphi_y + q_y c\varphi_x s\varphi_y s\varphi_z + q_z c\varphi_x c\varphi_y - q_y c\varphi_z s\varphi_x + q_x s\varphi_x s\varphi_z] \\
& + [(s\varphi_x s\varphi_z + c\varphi_x c\varphi_z s\varphi_y)(x_{12} - d_{x56} c(\varphi_{56} + \varphi_{23}) - d_{z34} c\varphi_{23} s\varphi_{34} \\
& + x_{34} c\varphi_{23} c\varphi_{34})] \\
& - [c\varphi_x c\varphi_y (d_{z45} - d_{z23} - d_{z12} + d_{z34} c\varphi_{34} + x_{34} s\varphi_{34} + d_{z56})]
\end{aligned}$$

Out of the 16 nonlinear equations, only 6 are independent as mentioned in Section 4.3. Independent equations are functions of dependent and independent coordinates, that appeared by equating elements of left hand side equation (5.17) with analogical elements of right hand side.

These 6 equations are:

$$\begin{aligned}
s_{12} &= 0; \quad s_{21} = 0; \quad s_{23} = 0 \\
d_{14} &= 0; \quad d_{24} = 0; \quad d_{34} = 0
\end{aligned} \tag{5.18}$$

(5.18) can be written in concise form as:

$$f_k(\mathbf{q}, \mathbf{z}) = 0, \quad k = 1 \text{ to } 6 \tag{5.19}$$

To solve this problem numerically, we should apply Newton-Raphson method. This method was described in the Theoretical part of this work, see Chapter 4, Section 4.4.

First, we set Camera Dolly mechanism to some initial position. On this position we obtain independent and dependent values.

Coordinates and orientations angles were obtained from software Autodesk® **Inventor**® 2016 Professional, where mechanism was drawn.

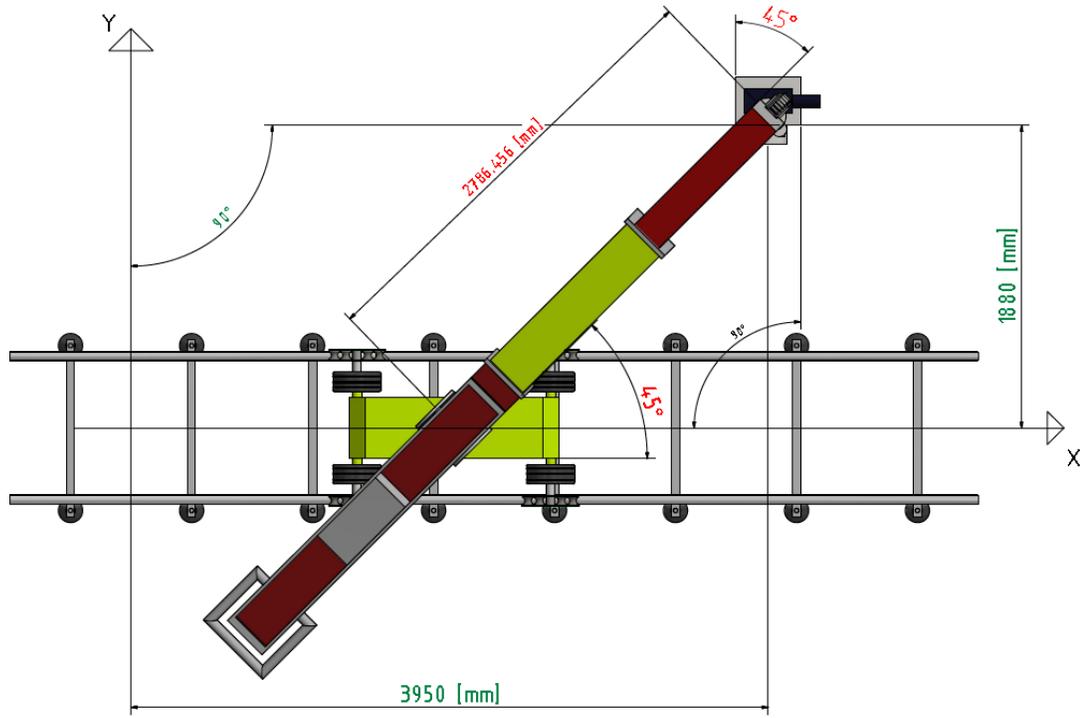


Figure 5-6 XY view of certain initial position

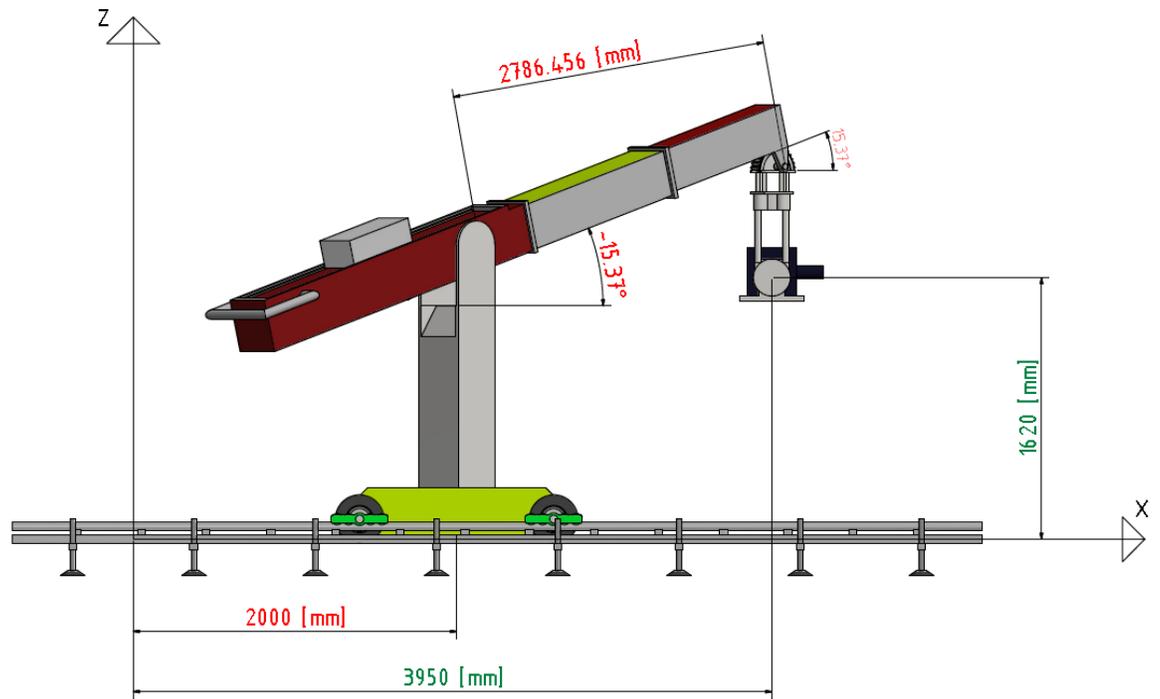


Figure 5-7 XZ view for certain initial position

**q** coordinates are green. Consequently:

$$q_x = 3950 \text{ [mm];}$$

$$q_y = 1880 \text{ [mm];}$$

$$q_z = 1620 \text{ [mm];}$$

$$\varphi_x = 0 \text{ [}^\circ\text{]}$$

$$\varphi_y = 0 \text{ [}^\circ\text{]}$$

$$\varphi_z = 90 \text{ [}^\circ\text{]}$$

**z** coordinates for this **q** are red:

$$z_1 = x_{12} = 2000 \text{ [mm]}, \quad z_2 = \varphi_{12} = 45 \text{ [}^\circ\text{]}, \quad z_3 = \varphi_{23} = -15.37 \text{ [}^\circ\text{]},$$

$$z_4 = x_{34} = 2786.456 \text{ [mm]}, \quad z_5 = \varphi_{56} = 45 \text{ [}^\circ\text{]}, \quad z_6 = \varphi_{6L} = 0 \text{ [}^\circ\text{]}$$

Initial values of **q** and **z** can be written in matrix form.

$$\mathbf{q}^{(0)} = [q_x^{(0)}, q_y^{(0)}, q_z^{(0)}, \varphi_x^{(0)}, \varphi_y^{(0)}, \varphi_z^{(0)}]^T$$

$$\mathbf{q}^{(0)} = [3950 \quad 1880 \quad 1620 \quad 0^\circ \quad 0^\circ \quad 90^\circ]^T \quad (5.20)$$

$$\mathbf{z}^{(0)} = [z_1^{(0)}, z_2^{(0)}, z_3^{(0)}, z_4^{(0)}, z_5^{(0)}, z_6^{(0)}]^T$$

$$\mathbf{z}^{(0)} = [2000 \quad 45^\circ \quad -15.37^\circ \quad 2786.456 \quad 45^\circ \quad 0^\circ]^T \quad (5.21)$$

This estimates of dependent coordinates  $\mathbf{z}^{(0)}$  need to be numerically iterated to the precision required by Newton-Raphson method. Structure of this procedure is shown in Diagram 1.

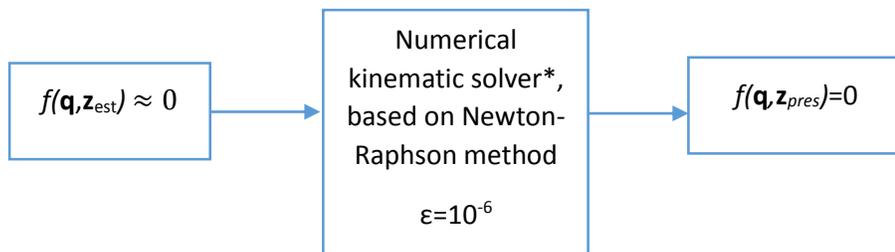


Diagram 1 Newton-Raphson method (short form)

\* Numerical kinematic solver was written in MATLAB® and can be found on the enclosed CD. File name ksolver.m

## 5.1 Detailed description of the Newton-Raphson method

First of all, we evaluate the set of equations (5.19)

$$\mathbf{R} = f(\mathbf{q}, \mathbf{z}) \quad (5.22)$$

Ideally  $\mathbf{R} = 0$ .

Using estimated values of  $\mathbf{q}^{(0)}$  and  $\mathbf{z}^{(0)}$  in equation (5.20), will not provide us exactly  $f(\mathbf{q}, \mathbf{z}) = 0$ , instead of, it provides numbers relatively close to zero. Specifically in case of values (5.20) and (5.21):

$$f(\mathbf{q}^{(0)}, \mathbf{z}^{(0)}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1.4189 \\ 1.4189 \\ 0.308 \end{bmatrix} \quad (5.23)$$

Our goal is to reach  $|\mathbf{R}| = 0$ . But in practice it is extremely rare situation when it is possible to reach ideal zero number for matrix norm value. And in fact we don't need such a small accuracy. It is enough if  $|\mathbf{R}|$  will be less or equal than some small positive number. In this case we set:

$$\varepsilon = 10^{-6} \quad (5.24)$$

The value of  $\varepsilon$  chosen to be  $10^{-6}$ , because we are working in millimeters and degrees (or radians).  $10^{-6}$ , means 0.000001, in millimetre scale it means that we eventually reach nanometer accuracy. For Dolly camera mechanism, this accuracy is enough. Same for angles.

Conditions are:

$$f(\mathbf{q}, \mathbf{z}) = \mathbf{R} \leq \varepsilon = 10^{-6} \quad (5.25)$$

Values of  $\mathbf{z}$  for such condition can be reached by applying iteration formula (4.28),

$$\mathbf{z}^{(k+1)} = \mathbf{z}^{(k)} + \lambda \Delta \mathbf{z}^{(k)}$$

Where  $\Delta \mathbf{z}^{(k)}$  is (4.26) and  $\lambda$  is described on the (p.38)

There is a very high probability that after first iteration, we won't get required values for given condition (5.25). That's why, we should assume that obtained value  $\mathbf{z}^{(k+1)}$  is  $\mathbf{z}^{(k)}$  and repeat procedure (4.28) again. If we still haven't reached the goal that has been needed, we should repeat process several times until our condition will be satisfied. If number of repetitions is too high we should stop iteration, because system doesn't converge. It can happen if obtained values are wrong or estimated equations do not correspond to required mechanism.

The Newton's method needs the Jacobian of the system (5.19).

$$\Phi_z = (z, q) = \frac{\partial f(z, q)}{\partial z^T}$$

$$\Phi_z = \Phi_z(z, q) \tag{5.26}$$

In this case MATLAB special function *jacobian(function, variable)*, for calculation of Jacobian is used.

For better understanding the procedure of applying Newton-Raphson's method in MATLAB® check the Diagram 2.

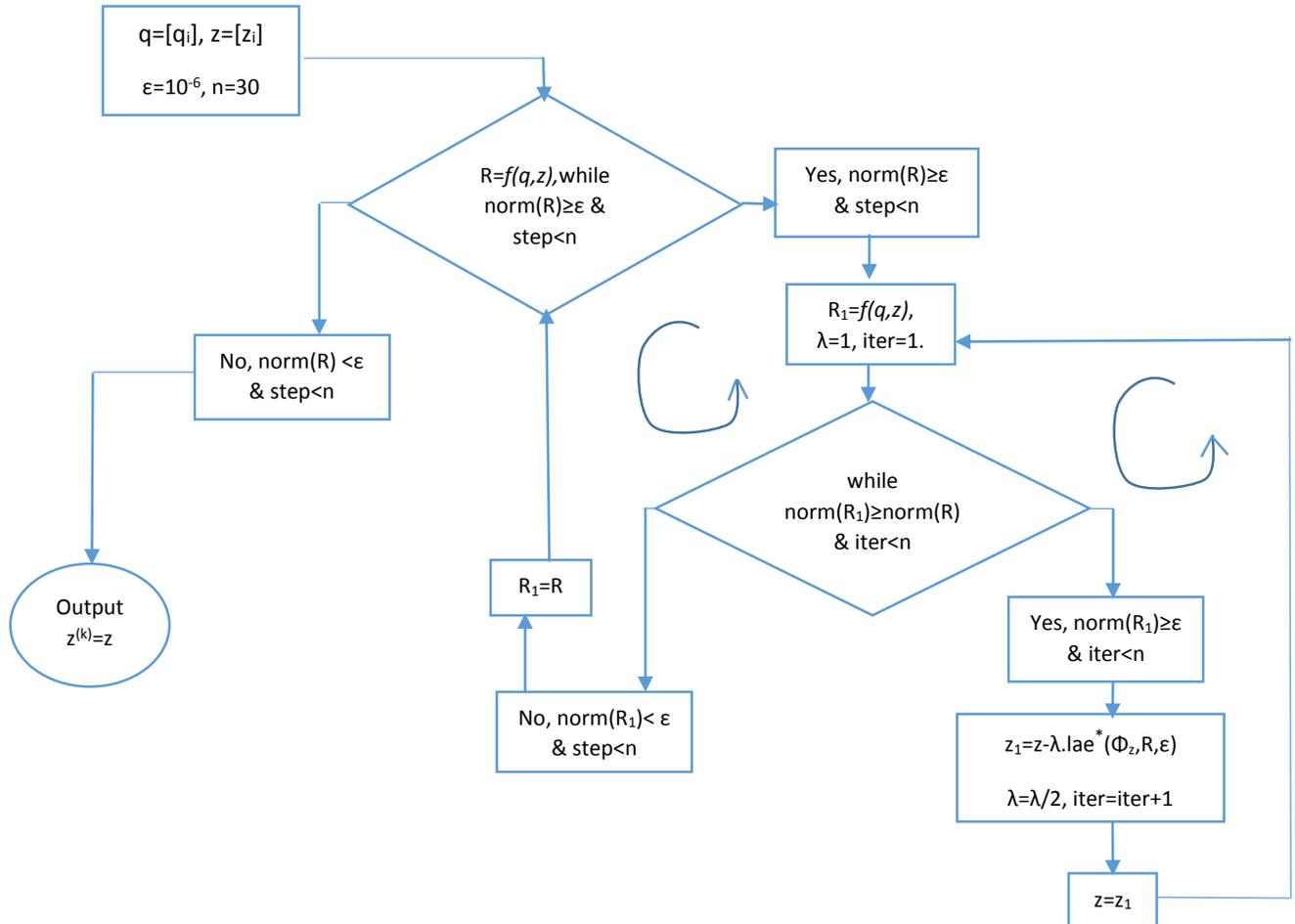


Diagram 2 Newton-Raphson's structure for MATLAB®

\* lae... solves  $A^*x=B$  for any dimensions (m,n) of A with the accuracy eps (ε). Appendix 4

Applying Kinematic solver to given values (5.21) and (5.22), we obtain following  $\mathbf{z}$  solutions

$$\mathbf{z}_{pres} = \begin{bmatrix} x_{12pres} \\ \varphi_{12pres} \\ \varphi_{23pres} \\ x_{34pres} \\ \varphi_{56pres} \\ \varphi_{6Lpres} \end{bmatrix} = \begin{bmatrix} 2000.93 \\ 45.0137^\circ \\ -15.362^\circ \\ 2787.78 \\ 44.986^\circ \\ 0 \end{bmatrix} \quad (5.27)$$

$$f(\mathbf{q}^{(0)}, \mathbf{z}_{pres}^{(0)}) = 10^{-12} \cdot \begin{bmatrix} 0.00004979 \\ -0.00016 \\ 0.000000000072 \\ 0.01449 \\ -0.339 \\ 0 \end{bmatrix} \quad (5.28)$$

## 5.2 Animation of Camera Dolly

To simulate a kinematic motion of our camera dolly mechanism, first of all we have to prescribe the independent coordinates  $\mathbf{q}$ , from initial position to required final position. Then, we have to obtain  $\mathbf{z}$  for initial  $\mathbf{q}$ .

Prescribed  $\mathbf{q}$  for chosen initial and final position:

$$\mathbf{q}_i = [4120[mm] \quad 1880[mm] \quad 1620[mm] \quad 0^\circ \quad 0^\circ \quad 90^\circ]^T$$

$$\mathbf{q}_f = [3170 [mm] \quad -1240 [mm] \quad 1074[mm] \quad 0^\circ \quad 0^\circ \quad 0^\circ]^T$$

Required motion time is 13 minutes (5.29)

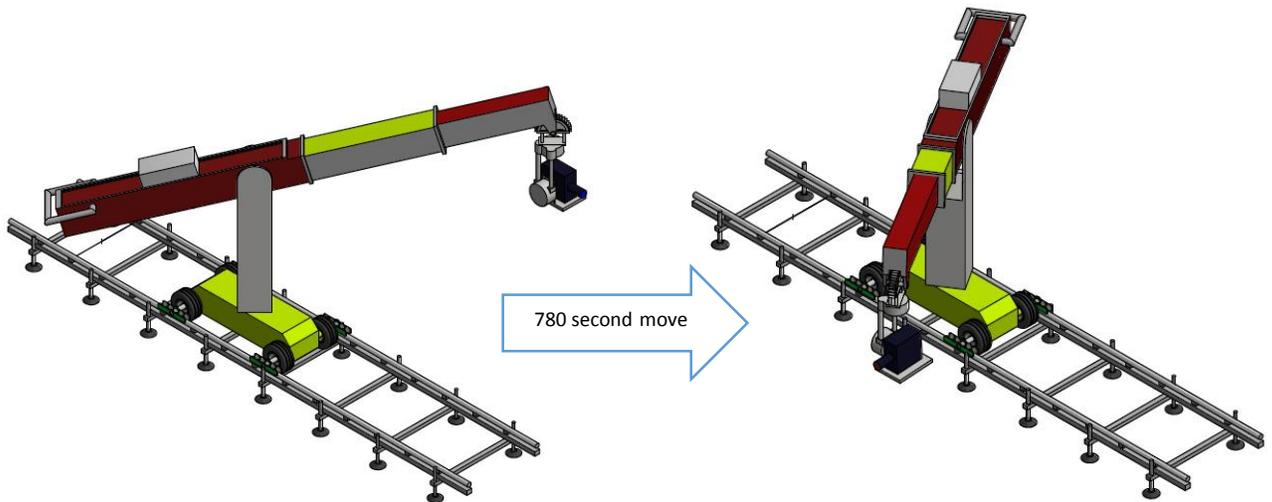


Figure 5-9 Camera dolly in  $\mathbf{q}_i$  position

Figure 5-8 Camera dolly in  $\mathbf{q}_f$  position

In MATLAB® representation

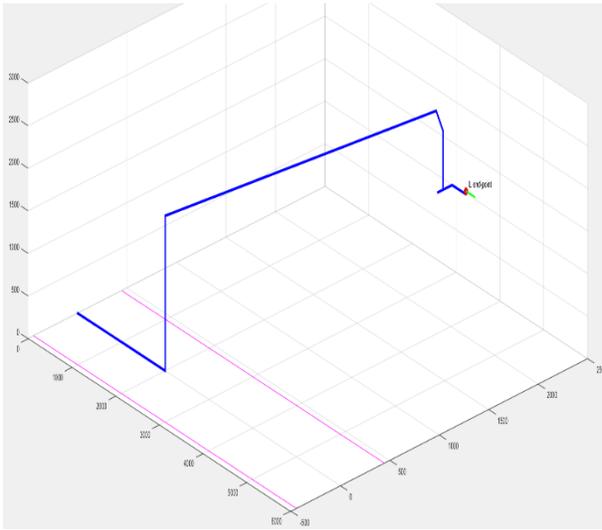


Figure 5-10 Camera dolly in  $q_f$  position MATLAB® 5

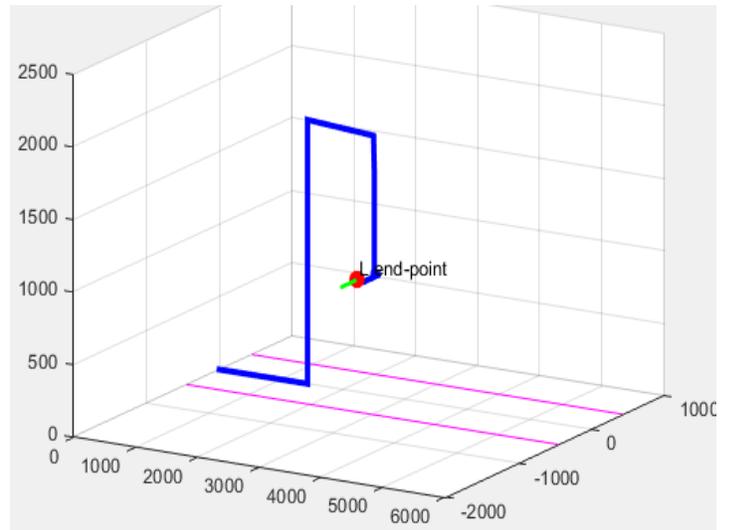


Figure 5-11 Camera Dolly in  $q_f$  position. MATLAB®

Let consider that, mechanism will move with constant velocity, with zero acceleration.

Real velocities can be computed as:

$$q_x^{(f)} = q_x^{(i)} + v_x \cdot 780 \rightarrow v_x = \frac{q_x^{(f)} - q_x^{(i)}}{780} \rightarrow v_x = \frac{3170 - 4170}{780} = -1 \text{ [mm.s}^{-1}\text{]}$$

$$v_y = \frac{q_y^{(f)} - q_y^{(i)}}{780} = \frac{-1880 - 1240}{780} = -4 \text{ [mm.s}^{-1}\text{]}$$

$$v_z = \frac{q_z^{(f)} - q_z^{(i)}}{780} = \frac{1074 - 1620}{780} = -0.7 \text{ [mm.s}^{-1}\text{]}$$

$$\varphi_z^{(f)} = \varphi_z^{(i)} + \omega_z \cdot 780 \rightarrow \omega_z = \frac{\varphi_z^{(f)} - \varphi_z^{(i)}}{780} \rightarrow \omega_z = \frac{0 - \pi/2}{780} = -\frac{\pi}{1560} \text{ [rad.s}^{-1}\text{]} \quad (5.30)$$

Using (5.30) position of crane can be prescribed every 1 second. In total have 780 steps equal to 780 animation frames

<sup>5</sup> Check Section 5.3 for color designation.

The overall procedure is illustrated in Diagram 3.

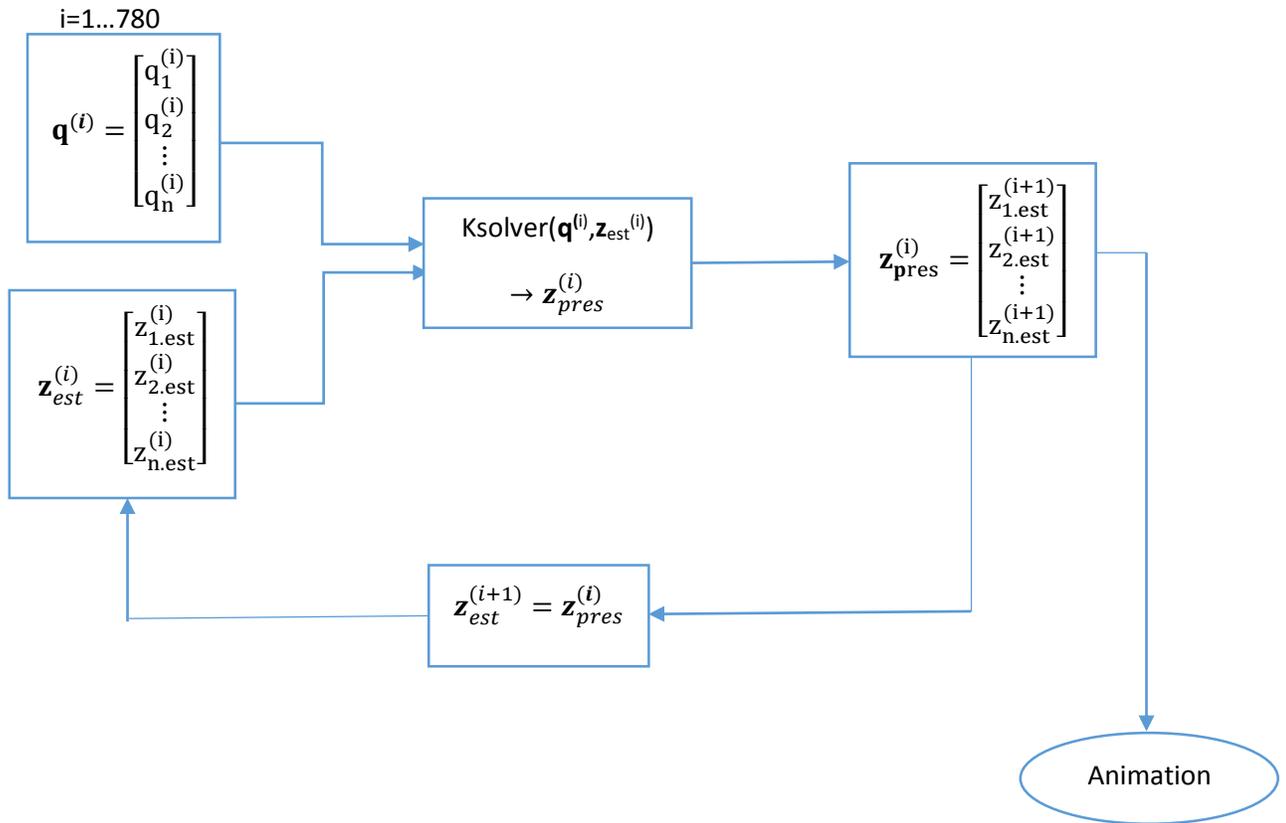


Diagram 3 Kinematic solution

Below in Section 5.4 you can see the obtained result as screenshots from evaluated results from MATLAB®.

### 5.3. Colors of MATLAB® drawings

For better understanding color designation is listed:

**Magenta** colored lines are showing railways.

The end point is **Red** (position of lens of cinema camera).

**Red** line represents a trajectory of end point changes.

**Blue** is the mechanism structure, based on constants dimensions of Camera motion-control mechanism.

**Green** shows the camera orientation.

### 5.4. Figures from MATLAB®

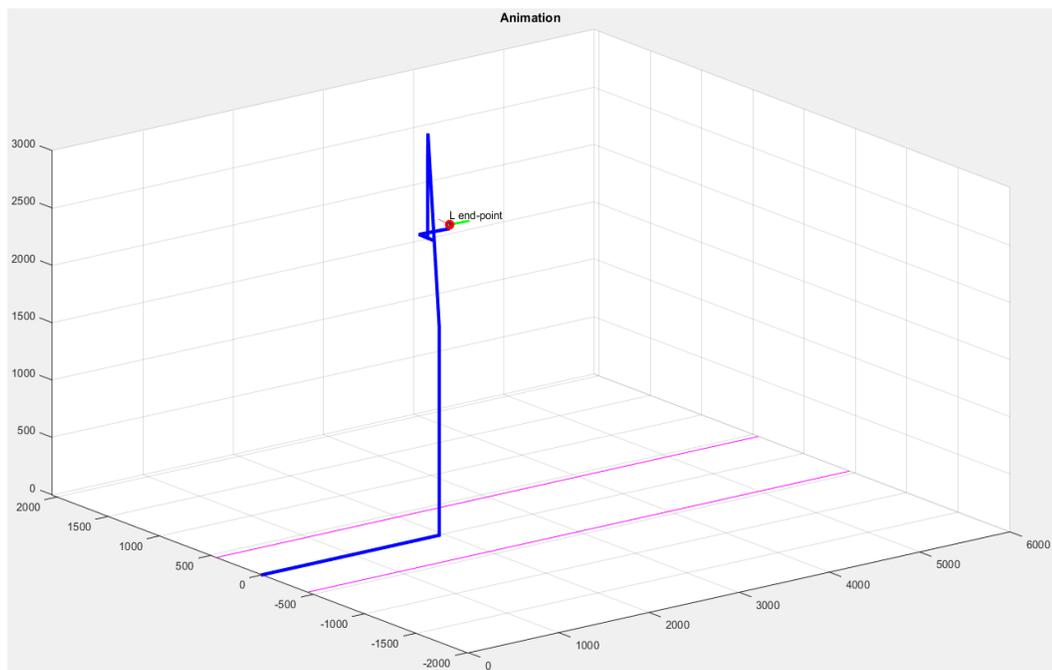


Figure 5-12 Position at 25th second

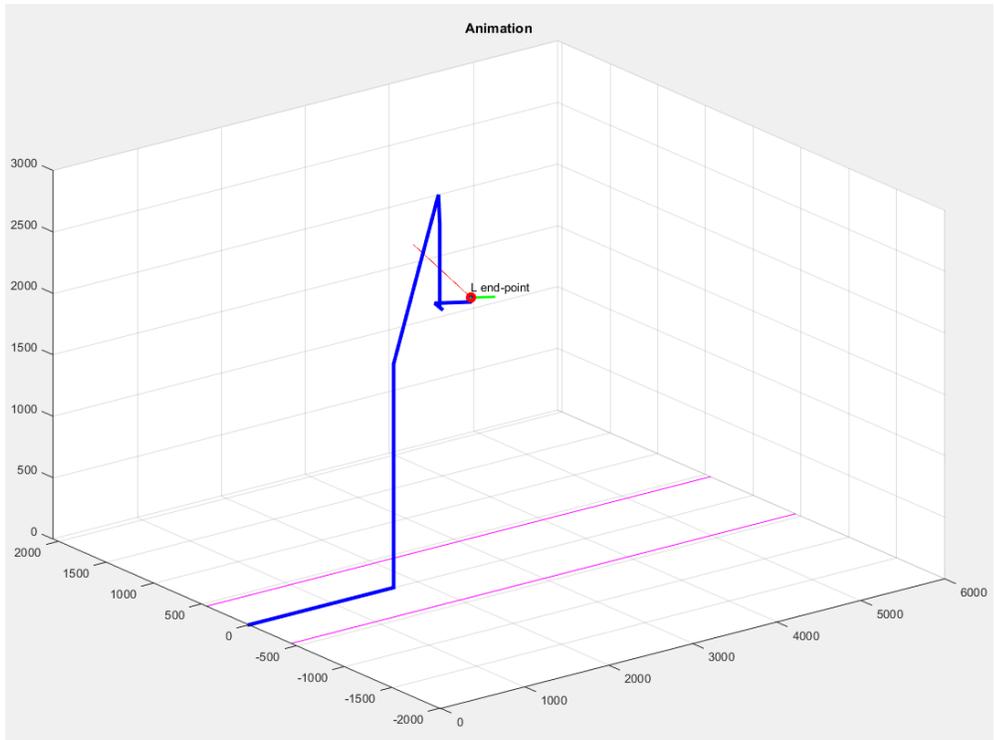


Figure 5-13 Mechanism at 195th seconds

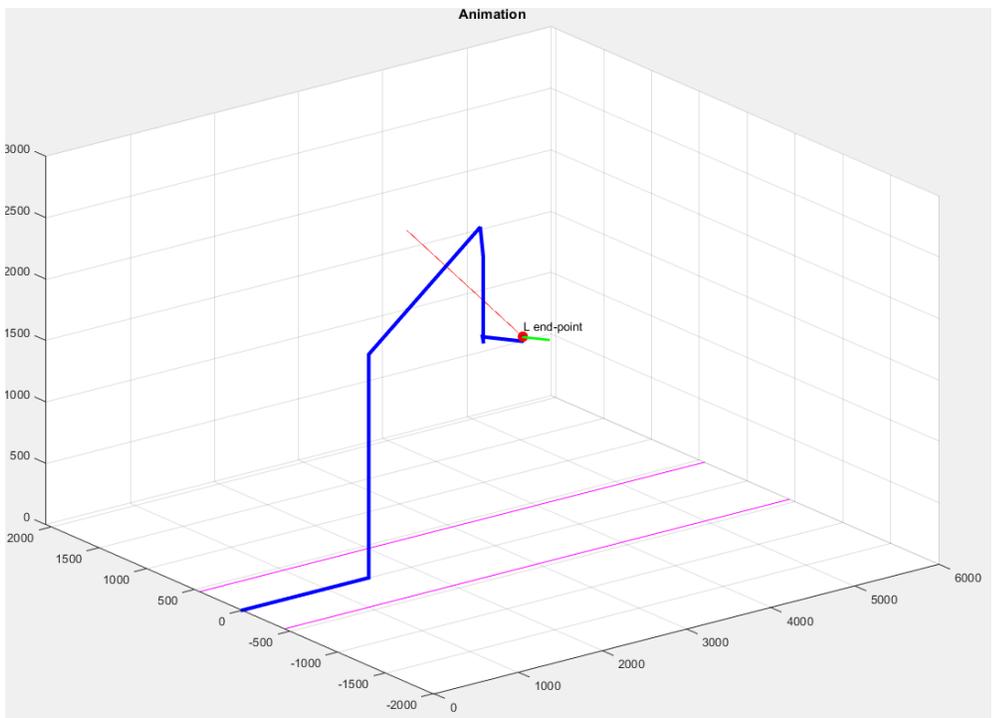


Figure 5-14 Mechanism at 391st second

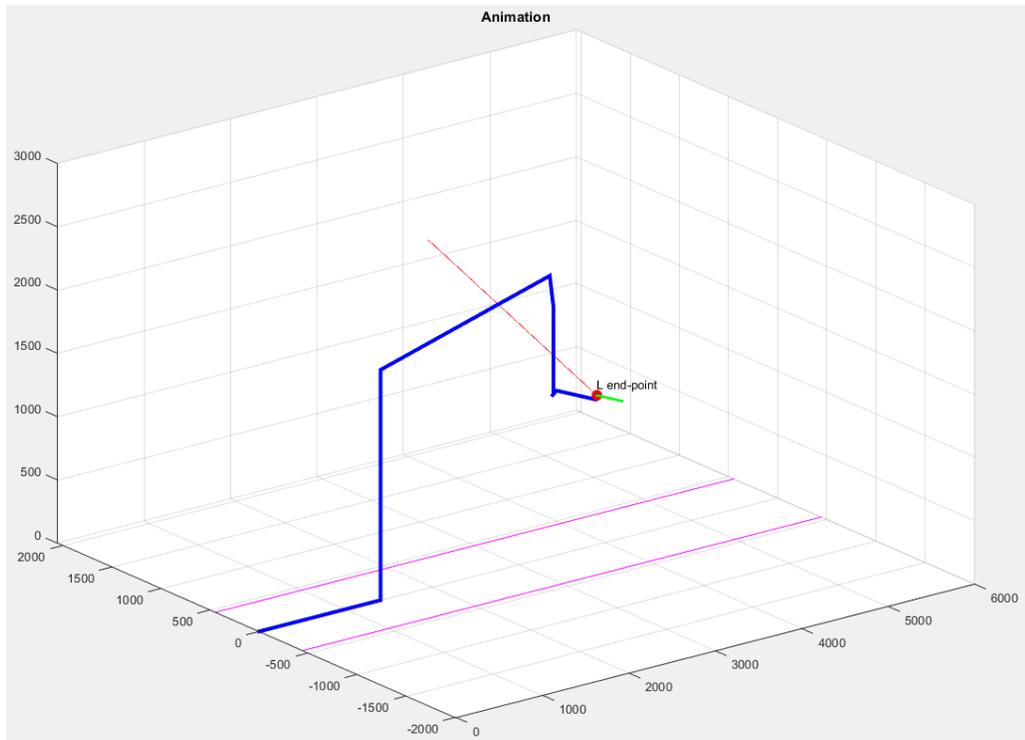


Figure 5-15 Mechanism at 552st second

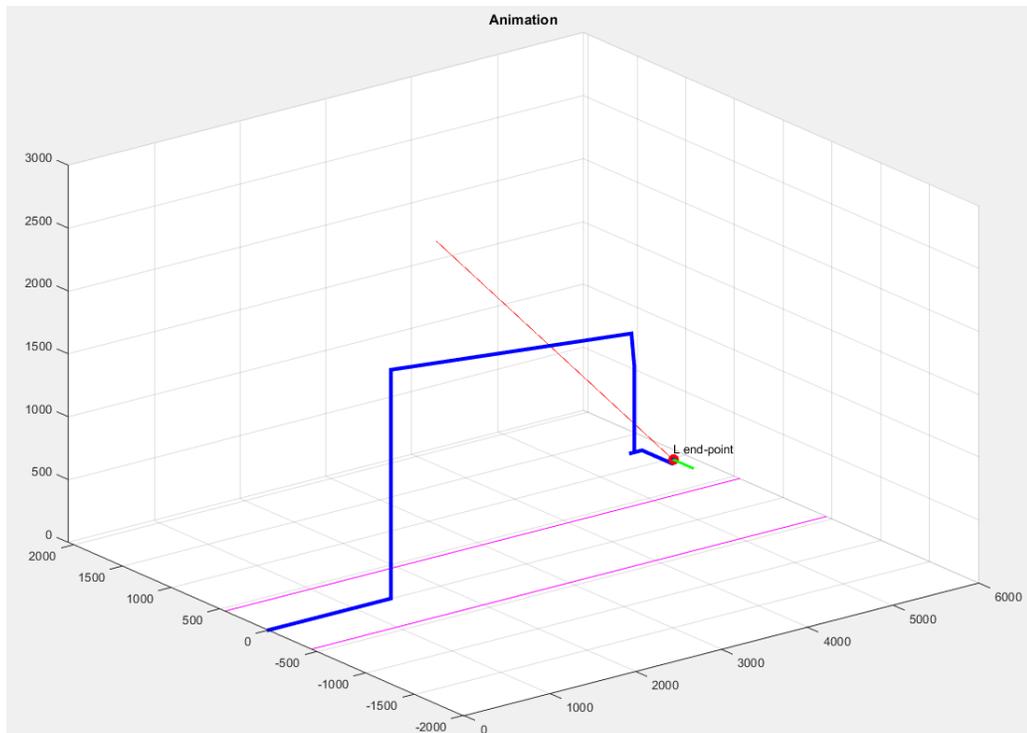


Figure 5-16 Mechanism at 779th second

## Chapter 6 Conclusion

In the first Chapter simple mechanisms used in movie industry were reviewed along with the history of their invention. Subsequently in Chapter two more complex and sophisticated mechanisms used in movie industry were described. The third Chapter focused on creation of 3D virtual CAD model of Technodolly camera crane (camera motion control). The crane was described, its history and use were also mentioned. The Chapter four dealt in detail with theory of kinematics, specifically matrix method and the way of solving resulting equations using Newton-Raphson iteration method. Chapter five specifically explains construction of kinematic equations and their subsequent solution by Newton Raphson iteration method in MATLAB®. It also illustrates the obtained results.

From the five tasks, according to assignment, three of those (1,2,3) were fully completed, part four was done partially (model was animated only in MATLAB®). Part five, due to its direct connection with part four, wasn't completed fully as well. Kinematic solution with its virtual simulation was done by regular MATLAB® graphing capabilities. Apart from the animations, all of the most vital parts of the work were successfully solved.

With the development of the robotics the mechanisms used in the movie industry can perform a set of much more complex actions, which could be an object of investigation. As a future challenge dynamic solution can be provided for camera motion control mechanism.

## Appendix

### 1 Vectors and matrices

Transpose of matrix  $A$  is making another matrix by  $A^T$  reflecting values along diagonal. Or make column matrix  $A$ , analogical new row matrix  $A^T$ . Or from row matrix  $A$ , makes column matrix  $A^T$ .

$$A = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad A = [x_1 \quad \dots \quad x_n]^T$$

Inverse of a matrix  $A$  exists and it is unique, if and only if  $A$  has a rank  $n$ , equivalently if and only if the determinant  $\det(A)$  is not zero. Inverse matrix satisfies

$$A^{-1} \cdot A = E$$
$$E = \begin{bmatrix} 1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 1 \end{bmatrix}$$

### 2 Jacobian

In vector calculus, the **Jacobian matrix** is the matrix of all first-order partial derivatives of a vector-valued function. Specifically, suppose  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$  is a function which takes as input the vector  $\mathbf{x} \in \mathbb{R}^n$  and produces as output the vector  $\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m$ . Then the Jacobian matrix  $\mathbf{J}$  of  $\mathbf{f}$  is an  $m \times n$  matrix, usually defined and arranged as follows [19]:

$$\Phi = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \dots & \frac{\partial f}{\partial x_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

### 3 Matrix norm

Given a square complex or real matrix  $\mathbf{A}$ , a matrix norm  $\|\mathbf{A}\|$  is a nonnegative number associated with  $\mathbf{A}$  having the properties

$\|\mathbf{A}\| > 0$  when  $\mathbf{A} \neq 0$  and  $\|\mathbf{A}\| = 0$  if  $\mathbf{A} = 0$ ,

$\|k\mathbf{A}\| = |k| \|\mathbf{A}\|$  for any scalar  $k$ ,

$\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$ ,

$\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\|$ .

Let  $\lambda_1, \dots, \lambda_n$  be the eigenvalues of  $\mathbf{A}$ , then

$$\frac{1}{\|\mathbf{A}^{-1}\|} \leq |\lambda| \leq \|\mathbf{A}\|$$

The matrix  $p$ -norm is defined for a real number  $1 \leq p \leq \infty$  and a matrix  $\mathbf{A}$  by

$$\|\mathbf{A}\|_p = \max_{\|\mathbf{x}\|_p=1} \|\mathbf{Ax}\|_p$$

where  $\|\mathbf{x}\|_p$  is a vector norm. The task of computing a matrix  $p$ -norm is difficult for  $p > 1$  since it is a nonlinear optimization problem with constraints.

Matrix norms are implemented as  $\text{Norm}[m, p]$ , where  $p$  may be 1, 2, Infinity, or "Frobenius".

The maximum absolute column sum norm  $\|\mathbf{A}\|_1$  is defined as

$$\|\mathbf{A}\|_1 = \max_j \sum_{i=1}^n |a_{ij}|$$

The spectral norm  $\|\mathbf{A}\|_2$ , which is the square root of the maximum eigenvalue of  $\mathbf{A}^H\mathbf{A}$  (where  $\mathbf{A}^H$  is the conjugate transpose),

$$\|\mathbf{A}\|_2 = (\text{maximum eigenvalue of } \mathbf{A}^H\mathbf{A})^{1/2}$$

is often referred to as "the" matrix norm.

The maximum absolute row sum norm is defined by

$$\|\mathbf{A}\|_\infty = \max_i \sum_{j=1}^n |a_{ij}|$$

$\|\mathbf{A}\|_1$ ,  $\|\mathbf{A}\|_2$  and  $\|\mathbf{A}\|_\infty$  satisfy the inequality

$$\|\mathbf{A}\|_2^2 \leq \|\mathbf{A}\|_1 \|\mathbf{A}\|_\infty \quad [20]$$

#### 4 lae function MATLAB® (from MATLAB® command: help lae)

```
function [x,nok,rA,VI,cI]=lae(A,B,eps)
%LAE solves A*x=B for any dimensions (m,n) of A with the accuracy
eps
%x - solution
%nok - 0 if OK, otherwise nonzero describing how many dependent
equations not fulfilled
%rA - rank of A computed by SVD with accuracy eps
%VI - enables to compute all solutions if many exists
% the solutions are x = x + VI*yI where yI is arbitrary vector of
dimensions (n-rA,1)
%cI - cI=0 describes the fulfillment of dependent equations
%
[m,n]=size(A);
[U,S,V]=svd(A);
S=diag(S);
rA=sum(S>eps);
c=U'*B;
if m>rA,
    cI=c((rA+1):m,1);
    nok=sum(cI>eps);
else
    cI=[];
    nok=0;
end;
S=S(1:rA,1);
cD=c(1:rA,1);
yD=cD./S;
if n>rA,
    VI=V(:,(rA+1):n);
    yI=zeros(n-rA,1);
else
    VI=[];
    yI=[];
end;
y=[yD;yI];
x=V*y;
```

## Bibliography

- [1] D. Bordwell and K. Thompson, *Film art: An introduction*, 8th ed., New York: McGraw-Hill Inc., 2008.
- [2] J. Anderson and B. Anderson, "The Myth of Persistence of Vision Revisited," *Journal of Film and Video*, vol. 45, no. 1, 1993.
- [3] Wikipedia, "Wikipedia," 5 6 2015. [Online]. Available: [https://en.wikipedia.org/wiki/Geneva\\_drive#cite\\_note-1](https://en.wikipedia.org/wiki/Geneva_drive#cite_note-1).
- [4] J. H. Bickford, *Mechanisms for intermittent motion*, New York: Industrial Press Inc., 1972, p. Chapter 9.
- [5] D. M. Mullen and K. Malkiewicz, *Cinematography*, New York: Simon & Schuster, 2005.
- [6] A. Kindelmann, J. Pearlman and W. Ostrander, "Claw and Registration pin mechanism". Patent US 2129093 A, 6 September 1938.
- [7] D. Trumbull, "Creating Special Effects for "2001: A Space Odyssey"," *Visual Memory*, [Online]. Available: <http://www.visual-memory.co.uk/sk/2001a/page3.html>.
- [8] Wikipedia, "The free encyclopedia," 29 March 2014. [Online]. Available: [https://en.wikipedia.org/wiki/Robert\\_A.\\_Mattey](https://en.wikipedia.org/wiki/Robert_A._Mattey).
- [9] H. Burbulla, "Camera crane". USA, Germany Patent 101 47 602.7,, 24 April 2003.
- [10] *OSCAR ceremony - Horst Burbulla and SUPERTECHNO crane*. [Film]. supertechnocom, 2005.
- [11] Technocrane s.r.o, "Technodolly products," [Online]. Available: <http://www.supertechno.com/product/technodolly/pdf/TECHNODOLLY%20-%20manual.pdf>.
- [12] L.-W. Tsai, *Robot analysis: the mechanics of serial and parallel manipulators*, New York: John Wiley & Sons, Inc., 1999.
- [13] K. Julis, R. Brepta and a. kol, *Mechanika 1. Díl: statika a kinematika*, Praha: SNTL -Nakladatelství Technické literatury, 1986.
- [14] V. Stejskal and M. Valasek, *Kinematics And Dynamics of Machinery*, New York: MARCEL DEKKER, INC., 1996.
- [15] O. V. Golubeva, *Theoretical mechanics (Теоретическая Механикаа)*, Moscow (Москва): Gos.izd. Fiziko-Matematicheskoy literatury, 1961.
- [16] E. J. Haug, *Computer aided kinematics and dynamics of mechanical systems*, Needham Heights: Allyn and Bacon, 1989.

- [17] J. García de Jalón and E. Bayo, Kinematic and dynamic simulation of multibody system: the real time challenge, Berlin: Springer Verlag, 2001.
- [18] Technocrane s.r.o., "TECHNODOLLY," Technocrane s.r.o., [Online]. Available: <http://www.supertechno.com/product/technodolly.html>.
- [19] Wikipedia, "Jacobian matrix and determinant," [Online]. Available: [https://en.wikipedia.org/wiki/Jacobian\\_matrix\\_and\\_determinant](https://en.wikipedia.org/wiki/Jacobian_matrix_and_determinant).
- [20] E. W. Weisstein, "Matrix Norm," MathWorld, [Online]. Available: <http://mathworld.wolfram.com/MatrixNorm.html>.
- [21] M. W. Spong, S. Hutchinson and M. Vidyasagar, Robot Modeling and Control, Hoboken: John Wiley & Sons, Inc., 2006, pp. 78-80.
- [22] S. Kucuk and Z. Bingul, Robot Kinematics: Forward and Inverse Kinematics, Industrial Robotics: Theory, Modelling and Control, Abu Dhabi: Sam Cubero (Ed.), 2006.
- [23] В. Миславский и В. Гергеша, Механик–изобретатель Иосиф Тимченко в документах и воспоминаниях, Kharkiv (Харьков): Factor (ФАКТОР), 2012.
- [24] E. Pärt-Enander, A. Sjöberg, B. Melin and P. Isaksson, The MATLAB® Handbook, Harlow: Addison-Wesley, 1996.
- [25] R. M. Murray, Z. Li and S. S. Sastry, A mathematical introduction to robotics, Boca Raton: CRC Press, 1994.
- [26] Introduction video to the Technodolly. [Online] [YOU TUBE]. URL: [https://www.youtube.com/watch?v=hKfCuA2l8c8&list=PLS\\_PhHiHpxkt0kCMeZndYcDCHd0FIMgbx](https://www.youtube.com/watch?v=hKfCuA2l8c8&list=PLS_PhHiHpxkt0kCMeZndYcDCHd0FIMgbx)