

Travail de fin d'études et stage[BR]- Travail de fin d'études : Design of an Aseptic Filling Station with Grade A GMP Compliance and Multiformat Capability Using an 8-Linkage Box Transport Mechanism[BR]- Stage d'insertion professionnelle : Citius Engineering

Auteur : Keutgen, Cédric

Promoteur(s) : Bruls, Olivier

Faculté : Faculté des Sciences appliquées

Diplôme : Master : ingénieur civil mécanicien, à finalité spécialisée en mécatronique

Année académique : 2022-2023

URI/URL : <http://hdl.handle.net/2268.2/17877>

Avertissement à l'attention des usagers :

Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.

University of Liege
Faculty of Applied Sciences

Design of an Aseptic Filling Station with Grade A GMP
Compliance and Multiformat Capability Using an 8-
Linkage Box Transport Mechanism

*An end of study work carried out with a view to obtaining the
Master's degree in Civil Mechatronic Engineering*



Faculty of Applied Sciences

Submitted by

Cédric Keutgen

Reg n° S180548

Under the guidance of

Olivier Brüls

Industrial promoter: **Laura Prijot**

Jury members: **Tristan Gilet, Marteen Arnst**



Academic year 2022-2023

Abstract

This Master thesis addresses the challenge of achieving multiformat filling of vials in sizes 2R, 15R, and 30R (ISO 8362-1:2018) by proposing a solution that utilizes an 8-linkage box transport mechanism. The primary objective is to achieve a flow rate of 300 mL per minute or higher while operating within the controlled environment of an isolator that complies with GMP grade A standards. The mechanical design takes into consideration various specific constraints, including maintaining a crossing air velocity of 0.45 m/s above the vials without causing disruption, ensuring a pressure gradient of 100 Pa between the isolator and its external environment, and using furniture materials compatible with the H₂O₂ sterilizing agent.

This study achieves its objectives by incorporating kinematic analysis into custom-made Python codes, available in Appendix 7.14. These codes serve the purpose of generating and analyzing 8-linkage box transport mechanisms, focusing on their kinematics. Moreover, an additional Python code is introduced to integrate these analysis tools into an optimization problem solved using a genetic algorithm. By adopting this approach, the developed solution exhibits remarkable flexibility and transferability, making it suitable for addressing diverse problems associated with box transport mechanisms. This adaptability extends to scenarios with varying linear displacement, dimensions, and engagement constraints.

Acknowledgements

I would like to first thank my academic and industrial supervisors, Professor Olivier Bröls and Mrs Laura Prijot, for their guidance, attention, and availability.

I would like to extend a special thanks to Alexis Courtejoie for providing valuable advice and sharing his expertise.

A big thank you to Mr Guillaume Laloux for his availability and precious expertise in the mechanical design of the box transport system.

I would also like to thank Mr Grégory Reichling and Mr Fabien Defays for allowing me to carry out this work at Cilyx Engineering, and thanks to my colleagues for their fantastic and warm welcome.

Lastly, I would like to express my gratitude to my family and all my loved ones who supported me throughout the completion of this work.

Table of content

1.	Environmental study	3
1.1	Overview of the main cGMP grade A characteristics	3
1.2	Cleanrooms	4
1.3	Norms and grades defining cleanliness	5
1.4	Airflow	7
1.4.1	Airflow pattern	7
1.4.2	Airflow velocity	9
1.4.3	Airflow Visualisation Studies	10
1.4.4	Airflow Velocity Measurements	10
1.5	Furniture material	12
1.6	Pressure cascade	13
1.7	Cleanroom isolator	16
1.7.1	Isolator decontamination	17
1.7.2	Difference between isolator and RABS	17
1.7.3	Isolator openings	19
1.7.4	Manual operations	20
1.7.5	Summarizing the advantages and limitations of isolators	21
1.8	Vials	22
1.8.1	Ready-to-Use (RTU) packaging	23
2.	State of the art	24
2.1	Filling line processes: overview	24
2.2	Bulk infeed	26
2.3	RTU infeed	27
2.3.1	Debugging	29
2.3.2	Delidding	30
2.3.3	Denesting	31
2.4	Filling and stoppering technologies	33
2.4.1	Small scale manufacturing: 30 – 120 VPM	33
2.4.2	Medium scale manufacturing: 120-150 VPM	36
2.4.3	High scale manufacturing: 200-600 VPM	37

2.5	Key features summary -----	41
2.5.1	Regarding infeed of the filling station -----	41
2.5.2	Regarding the filling and stoppering themselves -----	41
3.	Solution development -----	42
3.1	Choice of an 8-linkage box transfer mechanism -----	43
3.2	Filling circuit and stroke definition -----	44
3.2.1	Reaching 300 mL/min for 2R filling configuration -----	46
3.2.2	Extension to 15R and 30R filling configuration -----	47
3.3	Design parameters -----	52
3.3.1	Kinematic Analysis of an 8-Linkage Box Transport Mechanism -----	52
3.3.2	Python implementation -----	54
3.3.3	Standard scaled box transport mechanism: -----	55
3.4	Design criteria and Python trajectory analysis tool -----	57
3.5	Trajectory planning -----	62
3.5.1	Stroke: Velocity profiles strategies -----	63
3.5.2	Retreat: Velocity profile strategies -----	69
3.5.3	Overall motor requirements -----	71
3.6	Optimization problem -----	72
3.7	Formulation of the optimization problem -----	72
3.7.1	Design criteria -----	72
3.7.2	Design parameters: setting the domain -----	73
3.7.3	Objective (fitness) function: -----	73
3.7.4	Non-linear constraints: -----	74
3.8	Choice of the optimization method -----	74
3.8.1	Stochastic vs deterministic method -----	74
3.8.2	Choice of the genetic stochastic algorithm -----	75
3.8.3	General principle the multi-start genetic algorithm -----	76
3.8.4	Practical implementation -----	77
3.8.5	Results -----	80
3.9	Mechanical design for multiformat -----	82
3.9.1	Comb (format piece) -----	83
3.9.2	Guide rails -----	85
3.9.3	Global view of the proposed solution -----	88

3.10	Sealing the interfaces -----	89
3.10.1	General guidelines for seals selection -----	90
3.10.2	Static interfaces -----	91
3.10.3	Dynamic interfaces -----	93
4.	Future Perspectives -----	95
4.1	Validations -----	95
4.2	Improvements: -----	96
5.	Conclusion -----	97
6.	Bibliography -----	99
7.	Appendix -----	109
7.1	History of the 60-90FPM airflow velocity -----	109
7.2	2D Layouts of global filling processes -----	111
7.3	Filling components: technical specifications -----	113
7.4	E-beam tunnel and pulsed-light sanitization -----	119
7.5	H2O2 compatibility charts -----	122
7.6	Pressure cascade: guidance values justification -----	127
7.7	ISO 8536-1:2011: Infusion vials -----	128
7.8	ISO 8362-4:2011: Injection vials moulded from borosilicate -----	129
7.9	Debugging -----	131
7.10	Denesting -----	132
7.11	Kinematic analysis of the box transport mechanism -----	135
7.12	Pfizer vials datasheet -----	139
7.13	O'ring material compatibility with H2O2 -----	142
7.14	Python code -----	143
7.14.1	Filling configurations -----	143
7.14.2	Kinematics of the box transport mechanism -----	145
7.14.3	Trajectory analysis -----	157
7.14.4	Motion profiles analysis -----	164
7.14.5	Multi-start genetic algorithm applied to the box transport mechanism -----	171

Table of figures

<i>Figure 1 - Illustration of an GMP grade A isolator [1]</i>	3
<i>Figure 2 – Cleanroom representation [5]</i>	4
<i>Figure 3 - particle level for each GMP Grade [3]</i>	5
<i>Figure 4 - Airflow diagram in a cleanroom [60]</i>	7
<i>Figure 5 - HEPA filter example [59]</i>	7
<i>Figure 6 - Mixed airflow [8]</i>	7
<i>Figure 7 – Non-unidirectional airflow [8]</i>	7
<i>Figure 8 - Unidirectional airflow (UDA) [8]</i>	8
<i>Figure 9 - Turbulent flow pattern (left) versus lamnar flow pattern (right) [61]</i>	8
<i>Figure 10 - Flow past a circular cylinder [62]</i>	10
<i>Figure 11 - Stagnation point in groundwater [63]</i>	10
<i>Figure 12 - Working height in aseptic filling according to EU guidance [64] [14]</i>	10
<i>Figure 13 - Airflow supply in a GSK facility [66]</i>	11
<i>Figure 14 - Airflow velocity analysis in a GSK facility [66]</i>	11
<i>Figure 15 - SKAN example of pressure cascade for non-toxic product [1]</i>	13
<i>Figure 16 - SKAN example of pressure cascade for toxic product [1]</i>	14
<i>Figure 17 - Example of a multi-chamber cleanroom isolator [68]</i>	16
<i>Figure 18 – Illustration of the environment of an GMP Grade A isolator [1]</i>	16
<i>Figure 19 – Difference of airflow system between an open active RABS (left) and an isolator (right) [69]</i>	18
<i>Figure 20 - COMPARISON BETWEEN ISOLATOR AND RABS [27]</i>	18
<i>Figure 21 - Manually active mousehole [30]</i>	19
<i>Figure 22 - Passive mousehole with seal [30]</i>	19
<i>Figure 23 - Inside view of an isolator [30]</i>	19
<i>Figure 24 – Gloves used in isolator [70]</i>	20
<i>Figure 25 - Half-suit for isolator [70]</i>	20
<i>Figure 26 – Slide on airflow disturbance of the FDA-EMA Aseptic Requirements annexe 1 powerpoint presentation they made in Australia on the 20th September 2019 [71]</i>	21
<i>Figure 27 - SGD Pharma catalog of ISO 8362-1:2018 compliant vials. In red are the vials targetted by this project, in green their bottom diameter. [72]</i>	22
<i>Figure 28 - RTU vials packaging showing vials lying on a nest, covered by a tyvek. the whole is sealed in a tub. [45]</i>	23
<i>Figure 29 - Overall operations performed on vials [73]</i>	24
<i>Figure 30 - EXAMPLE OF DARA COMBI LINE FOR ASEPTIC FILLING [74]</i>	25
<i>Figure 31 - Depyrogenation tunnel illustration [75]</i>	26
<i>Figure 32 - Dara depyrogenation tunnel integration [76]</i>	26
<i>Figure 33 - EXAMPLES OF DEPYROGENATION TUNNEL OUTPUT MANAGEMENT [77] [78] [79]</i>	27
<i>Figure 34 - STEPS OF RTU TRANSFER [42]</i>	28
<i>Figure 35 - Typical manual handling of RTU, with grade zone specifications. NTT = no touch transfer [41]</i>	28
<i>Figure 36 - DARA RABS MODEL DB/A [80]</i>	29
<i>Figure 37 - Inside of the DARA model DB/A+ [44]</i>	29
<i>Figure 38 - DARA DL/A lid heating step [45]</i>	30
<i>Figure 39 - DARA DL/A LID removal step [45]</i>	30

Figure 40 - DARA DL/A INNER TYVEK REMOVAL STEP [45]	30
Figure 41 - Example of vial picking for denesting.....	31
Figure 42 - Example of denested vials directly placed on an endless-screw conveyor	31
Figure 43 - example of empty nest disposal [46].....	32
Figure 44 - Versynta machine achieving 30 VPM [81].....	33
Figure 45 - Versynta stoppering station [81].....	33
Figure 46 - Flexicon FPC60 stoppering axis [82].....	34
Figure 47 - Flexicon FPC60 filling andstoppering [82]	34
Figure 48 - Versynta machine achieving 60 VPM [83].....	34
Figure 49 - Watson-Marlow FMB210 [84]	35
Figure 50- Jerempli NFL/1-2-RDL fillig in nest station [85].....	35
Figure 51 - Jerempli SX-220-PP filling station [86]	36
Figure 52 - SX-220-PP stoppering station [86]	35
Figure 53 - jerempli SX-310-PP filling station [87].....	36
Figure 54 - Tofflon 120 VPM filling line [88].....	37
Figure 55 - Cook Pharmacica 150 VPM filling line [89]	37
Figure 56 - Bosch FLC 3080 filling station [90]	38
Figure 57 - Bosch FLC 3080 stoppering station [90].....	38
Figure 58 - Jerempli NFL/5-10 filling station [91]	38
Figure 59 - Jerempli HSL-PP filling station [92].....	39
Figure 60 - DARA 600 VPM filling line [93].....	39
Figure 61 - M.A.R 600 VPM filling line [94]	40
Figure 62 – Example of an 8-linkage transfer box [49]	43
Figure 63 - Flexicon PF7 peristaltic pump [95]	44
Figure 64 - PF7 capacity diagram (based on water) [96]	45
Figure 65 - Excel table for finding needles parameters	45
Figure 66 - Flexicon fillig nozzle geometric footprint [97].....	47
Figure 67 – Illustration of the needle’s geometric footprint	47
Figure 68 - Visualization of the problem on needles locations. $(a + 17)$; $(b + 25)$ and $(c + 31)$ are respectively entrax of 2R; 15R and 30R vials. a ; b and c represent the teeth length between vials	48
Figure 69- Triple loop of the python code finding needles locations.....	48
Figure 70 – Conditions for a solution to be admissible. $\text{min_tooth} = 5 \text{ mm}$, $\text{needles_dim} = 14 \text{ mm}$	49
Figure 71 - Interesting solutions for the number of vials to fill per cycle	51
Figure 72 - 8-linkage transfer box: geometry and denominations. [49]	52
Figure 73 - Reference trajectory.....	55
Figure 74 - Reference output.....	55
Figure 75 - Stroke of 216 mm reached by multypling reference parameters by $a = 1.322$	56
Figure 76 - Example of several stroke greater or equal to 216 mm, with a maximum deviation of 10 mm with respect to the actual trajectory.....	57
Figure 77 - Demonstration of a maximum deviation of 5 mm between a stroke of 216 mm and the actual trajectory.....	58
Figure 78 - Schematic representation of the vial engagement problem	59
Figure 79 - Illustration of the engagement and disengagement design criteria	60
Figure 80 - Tooth width required for the scaled box transport mechanism	61
Figure 81 - Implementatio of the velocity design parameters in the python code.....	63

Figure 82 - Motor requirement when applying a quintic motion for the stroke. $j_{max} = 5000$; $v_{max} = 300$; $a_{max} = 6000$	64
Figure 83 Quintic profile applied to stroke: motor view.....	64
Figure 84 - Quintic profile applied to stroke: comb view.....	64
Figure 85 - Motor requirements when applying a trapezoidal motion to the stroke. $v_{max} = 300$; $a_{max} = 700$	65
Figure 86 - Quintic profile applied to stroke: comb view. $j_{max} = 5000$; $v_{max} = 300$; $a_{max} = 6000$	65
Figure 87 - Quintic profile applied to stroke: motor view. $j_{max} = 5000$; $v_{max} = 300$; $a_{max} = 6000$	65
Figure 88 - Impact of trapezoidal and s-curve velocity profile on a glass of wine at the end of the motion [111]	66
Figure 89 - Comparison between an S curve velocity profile and a bang bang profile [51]	67
Figure 90 - MOTOR REQUIREMENTS WHEN APPLYING AN S-CURVE PROFILE TO THE STROKE. $J_{MAX} = 10\ 000$; $V_{MAX} = 300$; $A_{MAX} = 6000$	68
Figure 91 - Motor requirements when applying an s-curve profile to the stroke. $j_{max} = 5\ 000$; $v_{max} = 300$; $a_{max} = 6000$	68
Figure 92 - S-curve profile applied to stroke: motor view. $j_{max} = 5000$; $v_{max} = 300$; $a_{max} = 6000$	68
Figure 93 - S-curve profile applied to stroke: comb view. $j_{max} = 5000$; $v_{max} = 300$; $a_{max} = 6000$	68
Figure 94 - Definition of the waypoints parameters in the python code.....	69
Figure 95 - Lspb for retreat: comb view. Waypoints crossed at 40 and 60% of the retreat time. $t_{acc} = 0.1\ s$...	70
Figure 96 - Lspb for retreat: motor view. Waypoints crossed at 40 and 60% of the retreat time. $t_{acc} = 0.1\ s$...	70
Figure 97 - Lspb for retreat: motor requirements. Waypoints crossed at 40 and 60% of the retreat time. $t_{acc} = 0.1\ s$	71
Figure 98 - Design criteria definition in the Python tool.....	72
Figure 99 - Design parameters of an 8-linkage box-transport mechanism	73
Figure 100 - Bounding of the design parameters.....	73
Figure 101 - Comparison between deterministic and stochastic optimization methods [51]	74
Figure 102 - Census of the Scopus database regarding the number of appearances of stochastic algorithms in scientific papers between 2008 and 2018 [51]	75
Figure 103 - Steps detailed of the genetic algorithm operations [51].....	76
Figure 104 - Structure of the multi-start genetic algorithm [51].....	76
Figure 105 - Genetic optimization parameters	77
Figure 106 - Objective_function	78
Figure 107 - non_linear_constraints_function	78
Figure 108 - penalty_fuction	78
Figure 109 - Evaluation_function	79
Figure 110 - Error in the code that requires correction.	79
Figure 111 - trajectory explored	80
Figure 112 - Ergonomic trajectory.....	81
Figure 113 - Comb dimensions compliant with the solution	83
Figure 114 - Dimensions of gaps and teeth between teeth for the 2R configuration	83
Figure 115 - Needles inter-distance and perfect match with the 2R comb	84
Figure 116 - interface between the combs and the support in the system. The black circle highlights an edge that would have been exposed to particle accumulations if the comb were not designed to pass over the support until the red arrow	84
Figure 117 - Oblong holes for particles removal	85
Figure 118 - Captive knob for format changeover	85
Figure 119 - Global view of the conveying system	86
Figure 120 - Pfizer 2R vial nominal dimensions.....	86

Figure 121 - C-shaped guide rail dimensions.....	86
Figure 122 - Second guide rail to ensure a straight track.....	87
Figure 123 - Adjusting knob	87
Figure 124 - Adjusting knobs at the filling and stoppering station	87
Figure 125 - Mechanical design for the global solution for grade A aseptic multiformat filling.....	88
Figure 126 - Seal selection guideline [52].....	90
Figure 127 - Seal classification [52].....	90
Figure 128 - Example of a gasket used to seal to flanges [55].....	91
Figure 129 - O'ring working principle [99].....	92
Figure 130 - Virgin grade PTFE gasket [57]	92
Figure 131 - Types of double lips shaft seals suited for linear motion [102]	93
Figure 132 - Fonctionning of a TC4 lip seal [104]	94
Figure 133 - Choise of a TC4 lip seal from "France joint" catalog	94
Figure 134 - Flexicon filling nozzles technical specifications	117
Figure 135 - Fluid path components according to PF7 manual	118
Figure 136 - Best practice for RTU handling.....	119
Figure 137 – Inside of SKAN e-beam tunnel	120
Figure 138 - SKAN E-beam tunnel	120
Figure 139 - Comparison between active sanitization rocess for RTU	121
Figure 140 - Pulsed-light sanitization efficiency depending on surface complexity	121
Figure 141 - Steriline RTDS2 pulsed-light sterilisation station.....	121
Figure 142 - MATERIALS COMPATIBLE WITH H2O2 CHEMICAL AGENT	123
Figure 143 - Materials compatible with H2O2 chemical agent.....	124
Figure 144 - Materials compatible with H2O2 chemical agent.....	125
Figure 145 - Materials prohibited in a Grade A aseptic environment	126
Figure 146 - ISO 8536-1 :2011 COLOURLESS COMPLIANT VIALS.....	128
Figure 147 – ISO 8362-4:2011 colourless compliant vials	129
Figure 148 – ISO 8362-4:2011 Ambered compliant vials	130
Figure 149 - Automated de-bagging in DARA RABS DB/A.....	131
Figure 150 - Manual de-bagging example	131
Figure 151 - DARA vial de-nesting example 2	132
Figure 152 - DARA Vial denesting DN/E	132
Figure 153 - 8-linkage transfer box: geometry and denominations.....	135
Figure 154 - O'ring material compatibility with H2O2	142

Introduction

Since the onset of the Coronavirus crisis, the pharmaceutical industry has made remarkable strides, particularly in response to the global demand for aseptic filling stations enclosed in Grade A GMP isolators. These advancements are driven by the urgent need to maintain sterility and ensure product safety in pharmaceutical manufacturing processes.

To address the critical need for aseptic filling processes and prevent the introduction of contaminants into sterile drug products, the industry has established stringent guidelines for aseptic processing based on Good Manufacturing Practices (GMP). The pharmaceutical industry has embraced the use of Grade A GMP isolators as effective solutions in maintaining sterility throughout the filling process.

These Grade A GMP isolators, functioning as hermetically sealed enclosures, create controlled environments where sterility is strictly maintained. Strategies such as laminar airflow and pressure gradients are implemented within these isolators to prevent the entry of particles into the system. By acting as physical barriers between operators and the product, these isolators effectively minimize the risks associated with human-borne contamination and cross-contamination.

Previous to the controlled environment of a Grade A GMP isolator, aseptic vials undergo a series of preparation steps to maintain their integrity and sterility. These steps include washing, depyrogenation, and sterilization. Once the vials enter the isolator, they transition into a state of controlled cleanliness, shielding them from potential contamination. Stringent procedures for aseptic filling, including filling, capping, and labeling, are followed within the isolator, adhering to the highest standards of aseptic conditions.

In the context of conveying vials between filling, control, and stoppering stations, medium-scale installations commonly adopt an intermittent motion approach. This approach involves engaging and disengaging vials using combs attached to servomotors, with conveying typically accomplished using cylinders. While this system follows a precise rectangular path, its effective implementation requires complex machinery and automation.

This master thesis aims to overcome the drawbacks of the intermittent motion approach by exploring a costless alternative that achieves intermittent motion: the 8-linkage box transport mechanism. The primary objective is to develop a suitable multiformat solution for aseptic filling of vials in sizes 2R, 15R, and 30R, as per the ISO 8362-1:2018 standard.

Additionally, the present thesis seeks to implement the kinematic approach into custom-made Python tools, making it easier to develop box transport mechanisms for future projects involving varying linear displacement, dimensions, and engagement constraints that Cilyx Engineering may encounter. The Python code for these tools is available in Appendix 7.14.

To achieve those goals, this project is divided into three main chapters:

1. Environmental study:

This section explores the key features encountered in aseptic isolators. It thoroughly examines and defines the guidelines and constraints that the final solution must comply with. By understanding the environment within the isolator, the thesis lays the foundation for developing an effective box transport mechanism.

2. State of the art:

The second chapter of this thesis is dedicated to orienting the research towards an ergonomics solution by analyzing the existing proposals from Cilyx's competitors in the pharmaceutical market. This analysis involves examining the material infeed processes and correlating them with an evaluation of filling lines across different scales, ranging from small-scale to high-scale operations. By conducting this analysis, Cilyx Engineering can make informed decisions regarding the specific requirements of the project and gain valuable insights from the industry's current practices.

3. Solution development:

In this section, the thesis aims to develop a box transport mechanism that meets the project's requirements. The first step involves studying the kinematics and defining the design parameters and criteria necessary to fulfill the motion requirements. Throughout these studies, the chosen approach is implemented in Python, making it easily adaptable to any box-transport mechanisms. The second part of this chapter focuses on proposing mechanical design and sealing solutions for the chosen mechanism, ensuring its practical implementation.

It is important to note that the outcome solution of this thesis relies on a theoretical background and mathematical development. Therefore, it should be considered as a foundation for further empirical validations and refinements before being considered a ready-to-use solution.

1. Environmental study

Although this project does not undertake the conception of a grade A aseptic environment, it implements an automated filling method that must respect the constraints it imposes. Acquiring knowledge about such environments is therefore more than mandatory to ensure the ergonomics and compliance of the final solution. The reader should not neglect this chapter as it lays the foundation of all the reasoning that will be presented in this work.

1.1 Overview of the main cGMP grade A characteristics

Figure 1 below highlights all the major information which will be discussed in the next chapters. These are the constraints that the technical solutions will have to face to comply with a cGMP Grade A cleanliness. [1]

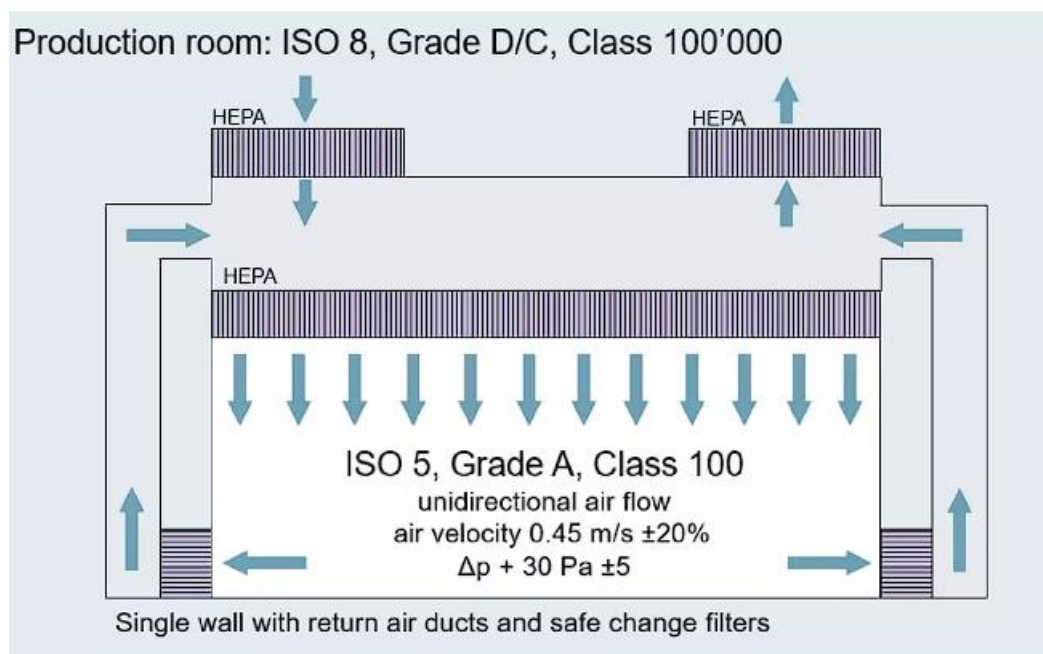


FIGURE 1 - ILLUSTRATION OF AN GMP GRADE A ISOLATOR [1]

1. GMP Grade A = ISO 4.8. facilities
2. Characterized by a unidirectional and uniform airflow throughout the entire space
3. Steady velocity and approximately parallel streamlines of 0.45 m/s ±20%.
4. Obstacles should preserve smoothness of the airflow pattern.
5. A pressure cascade with final pressure about 30 Pa ± 5 in the filling chamber.

1.2 Cleanrooms

A cleanroom, is illustrated in Figure 2, is a controlled environment designed to filter out particles and maintain a high level of cleanliness. It consists of specialized chambers where various processes are conducted. By controlling the level of pollutants, a precise degree of cleanliness is achieved. Additionally, cleanrooms often regulate other important parameters including temperature, air flow, and humidity to ensure optimal conditions for the processes carried out within them. [2]



FIGURE 2 – CLEANROOM REPRESENTATION [5]

Even if such chambers are used for various manufacturing products, this project mainly focuses on processes set up in pharmaceutical ones so that the solution it proposes complies with their constraints.

1.3 Norms and grades defining cleanliness

[CLASSIFICATIONS DE SALLE BLANCHE \(wiskindcleanroom.com\)](http://wiskindcleanroom.com)

Cleanrooms are currently controlled by the norm ISO 14644 which classifies their cleanliness. Based on the number and size of particles per cubic meter of space, norm 14644-1:2015 sort chambers from ISO-9 to ISO-1, i.e., from dirtiest to cleanest as shown in Table 1 and Figure 3. [3]

Cleanliness level	ISO class number	Maximum concentration limits (particles/m ³ of air) for particles equal to and larger than the considered sizes shown					
		≥0.1 μm	≥0.2 μm	≥0.3 μm	≥0.5 μm	≥1 μm	≥5 μm
Extremely clean	ISO 1	10	2				
	ISO 2	100	24	10	4		
	ISO 3	1,000	237	102	35	8	
	ISO 4	10,000	2,370	1,020	352	83	
	ISO 5	100,000	23,700	10,200	3,520	832	29
	ISO 6	1,000,000	237,000	102,000	35,200	8,320	293
	ISO 7				352,000	83,200	2,930
	ISO 8				3,520,000	832,000	29,300
Clean	ISO 9				35,200,000	8,320,000	293,000

TABLE 1- NORM ISO ON CLEANLINESS [3]

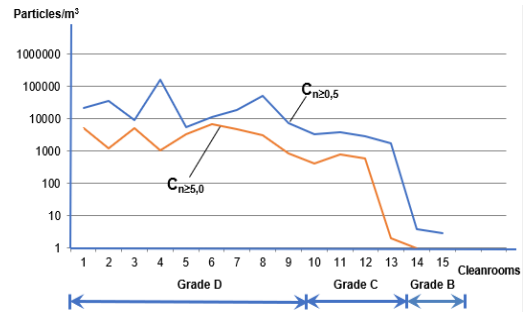


FIGURE 3 - PARTICLE LEVEL FOR EACH GMP GRADE [3]

The automated filling station this project aims to implement is in an environment in which sterile drugs and biological products are manufactured. In this particular case of aseptic filling, the FDA as well as the European Current Good Manufacturing Practices (EU cGMP) require that the Grade A and ISO 5 constraints of listed in Table and Table 3 be met. [3] [4]

Level	The maximum allowable number of particles greater than or equal to the size in the table per m ³			
	At rest ^a		In operation ^b	
	0,5 μm	0,5 μm	0,5 μm	0,5 μm
A	3 520	20	3 520	20
B	3 520	29	352 000	2 900
C	352 000	2 900	3 520 000	29 000
D	3 520 000	29 000	Not defined	Not defined

TABLE 2 - GMP GRADE [4]

	EU GMP		FDA
	Grade	ISO Class	
Filling – Aseptic	A	ISO 4,8	ISO 5
Filling – Terminal Sterilization	A	ISO 4,8	-
Background for Grade A – Aseptic	B	ISO 7	ISO 7
Background for Grade A – Terminal Sterilization	C	ISO 8	-
Supporting Clean Areas	D	No control, risk analysis	ISO 8

TABLE 3 - COMPARISON BETWEEN ISO AND GMP GRADE [58]

Operations undertaken by this work must be performed in a cGMP Grade A chamber. To achieve this level of cleanliness, 5 mechanisms are often set up: [5]

	Gowning	Sticky flooring	Airflow	Furniture material	Pressure cascade
Chapter:	/	/	Airflow	Furniture material	Pressure cascade

As gowning and sticky flooring cannot be disturbed by the design of the filling station, they will not be developed in this paper. This is not the case for the last three means of action which will impact the design of the filling station's elements, their materials as well as their sealing solutions. To ensure the compliance of the eventual solution with those mechanisms, they will be studied in more details in the following chapters.

1.4 Airflow

1.4.1 Airflow pattern

Both at the beginning and at the end of its journey through the cleanroom, the air must first be filtered. To do this, it passes through a HEPA¹ depicted in Figure 5 or ULPA² filter made of borosilicate glass fiber. In practice, HEPA filters make up 80% of the cleanroom ceiling as shown in Figure 4. [6]



FIGURE 5 - HEPA FILTER EXAMPLE [59]

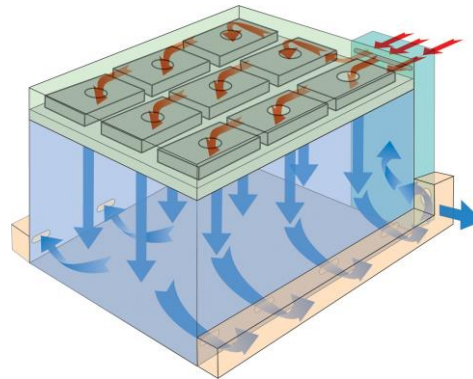


FIGURE 4 - AIRFLOW DIAGRAM IN A CLEANROOM [60]

Cleanrooms can be either unidirectional or non-unidirectional depending on the airflow crossing them. Non-unidirectional chambers make use of turbulent airflow systems to reach ISO 6-9 cleanliness. This airflow can cause random particle motion, hence making them more difficult to separate from the rest of the air. However, this randomness contributes to separating particles from the air through the filter. **As ISO 5 is not reached, non-unidirectional airflow is not the type of airflow considered for this project environment.** [7] [8] [9]

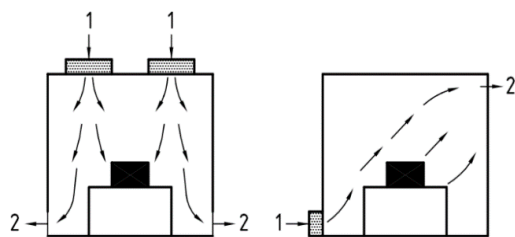


FIGURE 7 – NON-UNIDIRECTIONAL AIRFLOW [8]

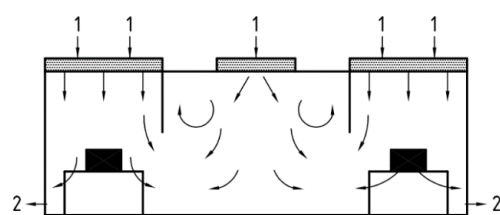


FIGURE 6 - MIXED AIRFLOW [8]

¹ High Efficiency Particulate Air.

² Ultra-Low Particulate Air.

According to cGMP norm: “All aseptic manipulations, including processing of sterile materials, filling, and closing (e.g., placement and sealing of stoppers on vials) **should be performed under unidirectional airflow that is ISO 5 or better**”. Such airflow can be visualized on Figure 8. [8] [9]

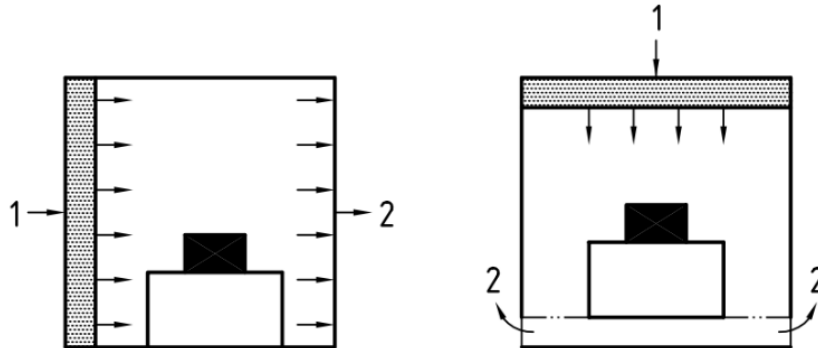


FIGURE 8 - UNIDIRECTIONAL AIRFLOW (UDA) [8]

In this case, air flow hoods direct air jets in a straight path. Laminar airflow thus created crosses a HEPA or ULPA filter before following a straight and unimpeded path through the whole chamber. Eventually, particles get stuck to the randomly placed fibres within the filter and removed from the air entering the room. This goal is usually achieved using air velocity ranging between **60 – 90 FPM³**, enough to sweep away particles while remaining within **18 degrees of parallel**. [8] [9]

Main differences between non-unidirectional airflow and UDA are illustrated in Figure 9.

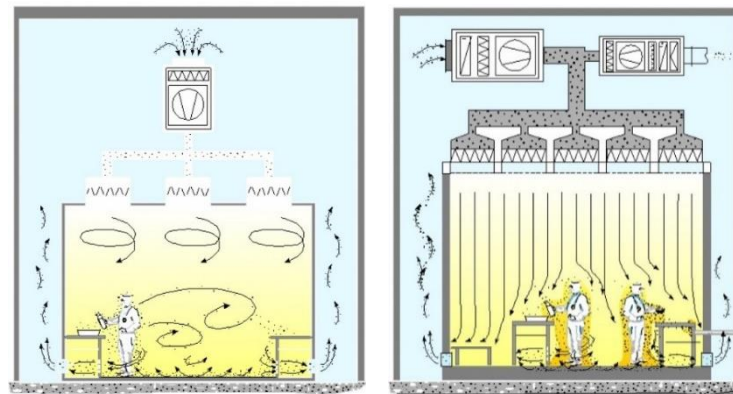


FIGURE 9 - TURBULENT FLOW PATTERN (LEFT) VERSUS LAMNAR FLOW PATTERN (RIGHT) [61]

By preventing particle mixing, **UDA can reach the ISO 1-5 standard and will therefore be considered in this project**. Their velocity will be studied in more detail in the next subsection as this airflow shall not be too disturbed by the filling station. [8] [9]

³ Feet per minute. Unit of measure for air velocity. 60-90 FPM = 0.3-0.45 m/s. This value will be discussed in more details in Section Airflow velocity

1.4.2 Airflow velocity

Nowadays, EU GMP Annex 1 propose this value of 0.45 m/s ±20% as a guidance in its statement “Laminar air flow systems should provide a homogeneous air speed in a range of 0.36 to 0.54 m/s (guidance value) at the working position in open clean room applications”. From THE FDA 2004⁴ side, more flexibility is provided as it states: “at a velocity sufficient to sweep particles away from the filling / closing operation and maintain unidirectional airflow during operation.” However, it still proposes this value in its footnote through “A velocity from 90 feet per minute is generally established, with a range of ±20% around the set point. Higher velocities may be appropriate in operations generating high levels of particulates.” Those pieces of information are summed up on Table 1. [10] [11] [12]

Velocity Threshold	ISO 5	ISO 4	ISO 3	ISO 2
Low-End Velocity (Meters/Second)	0.2	0.3	0.3	0.3
High-End Velocity (Meters/Second)	0.5	0.5	0.5	0.5

TABLE 1 - VELOCITY TRESHOLD REGARDING CLEANLINESS [12]

Conclusion: The solution developed later in this project should not disturb the laminarity of a vertical air flow of speed 0.45 m/s ±20%. A more detailed history on how and why this value have been established as a guidance is available in 7.1. In broad terms, it has been found empirically and is the result of compromises between the system's energy consumption, the cleanliness it achieves and its impact on the comfort of operators working in the white zone.

⁴ FDA guidance are the US equivalency of GMP. The objective of the FDA "Annual Product Review". The objective of the EU "Product Quality Review" (PQR). [65]

1.4.3 Airflow Visualisation Studies

Temporal and spatial characteristics of an airflow can be visualized by **injecting smoke in the chamber**. Indeed, the behaviour study of the smoke gives important information regarding the air-flows smoothness and pattern. And one can therefore assess risk for the airflows to draw potentially contaminated air towards critical places. Another piece of information that this study can provide is whether certain obstacles in the air flow cause contamination by changing the direction of the air. When strike by a unidirectional air flow, an object creates a 'wake region' as shown in Figure 10. [13]

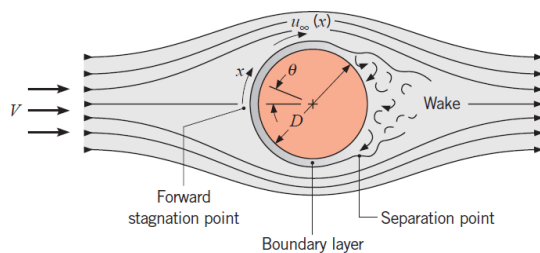


FIGURE 10 - FLOW PAST A CIRCULAR CYLINDER [62]

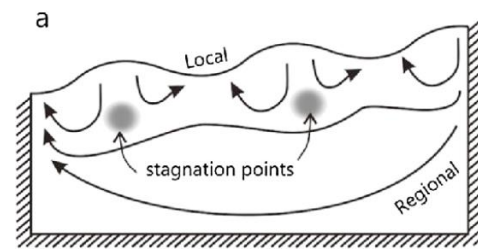


FIGURE 11 - STAGNATION POINT IN GROUNDWATER [63]

Such regions should be avoided as much as possible. Indeed, they are characterized with vortices and eddies which lead to pockets of stagnation and thus particle accumulation as shown in Figure 11. At the interface between GMP grade B, such wake can even drag ambient air into the grade A chamber and compromise it. **The final design of the filling station will have to be subjected to this test to ensure that it does not interfere with the flow.** [13]

1.4.4 Airflow Velocity Measurements

By now, airflow velocity and patterns of the environment have been established. However, an important piece of the puzzle is still missing. Where should those parameters be measured? **According to European regulatory guidance, airflow velocities should be measured at working height.** What is it? Is it a relevant location to assess airflow pattern described hereabove? **For aseptic filling, a widely used definition of working height is the point just above the vial neck opening as shown in Figure 12.** [14]



FIGURE 12 - WORKING HEIGHT IN ASEPTIC FILLING ACCORDING TO EU GUIDANCE [64] [14]

Although this location ensure that the airstream do not bring any particles in the open neck, it is inherent to vial size and thus may vary within the same line. One can argue with the coherency of this choice of point as it can leads to significant variation of the measurement depending on size and configuration of the equipment within the unidirectional airflow. These measurements aim at reflecting the flow pattern which is of most importance than the fluid velocity. In the other hand, the FDA advice to take velocity measurements directly below the HEPA filter face. Under this guidance point, measurements tend to be more accurate predictor of airflow patterns than for the European choice of working height. Examples of such airflow velocity analysis are given in Figure 14 and Figure 13. [14]

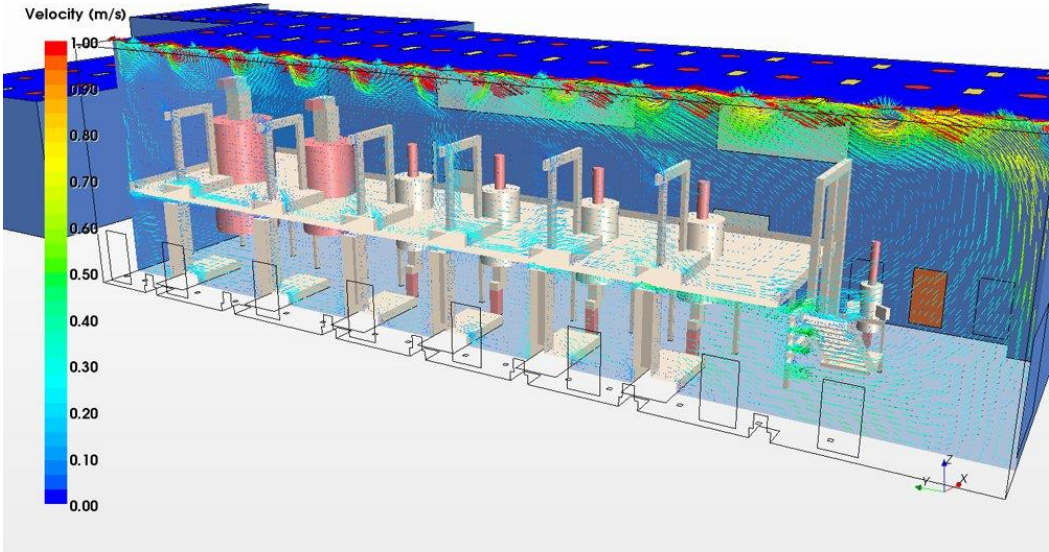


FIGURE 14 - AIRFLOW VELOCITY ANALYSIS IN A GSK FACILITY [66]

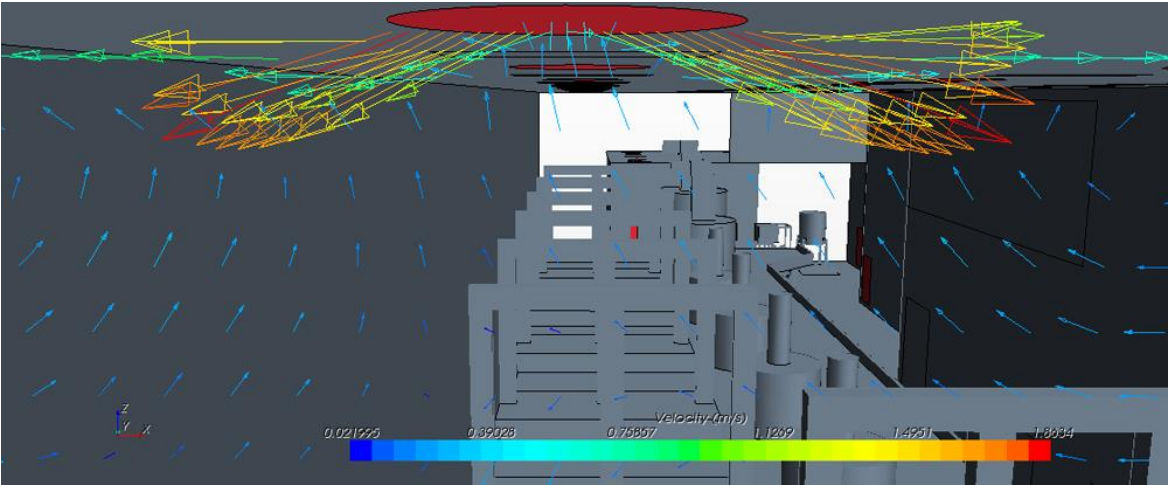


FIGURE 13 - AIRFLOW SUPPLY IN A GSK FACILITY [66]

1.5 Furniture material

According to the Salipur Institute of Pharmacy and Technology, it is essential for all material surfaces used in the isolator to be smooth, free of crevices, and easily cleanable and sterilizable to prevent the accumulation of particles. Additionally, these surfaces must be resistant to the effects of cleaning agents and disinfectants. In this project, H₂O₂ is later chosen as the cleaning agent in Section 1.7.1. Material selection should therefore be based on the **hydrogen peroxide material compatibility chart** provided by Utah University, available in Appendix 7.6. [15] [16]

Table 2 from this compatibility chart emphasizes the **suitability of stainless steel** for use in contact with hydrogen peroxide. Stainless steel is known for its excellent resistance to corrosion and chemical agents, making it an appropriate choice for the project.

Material	90% H ₂ O ₂	98% H ₂ O ₂	Notes
301	Class 2 / 3	Class 2 / 3	Depends on surface treatment
302	Class 2 / 3	Class 2 / 3	Depends on surface treatment
304	Class 2	Class 2	
316	Class 2	Class 2	
329	Class 3 / 4	Class 3 / 4	Depends on surface treatment
347	Class 2	Class 2	
443	Class 4	Class 4	
446	Class 4	Class 4	

- Class 1: Materials Satisfactory for Unrestricted use with H₂O₂ – Class 2: Materials Satisfactory for Repeated Short-Time Contact with H₂O₂.
 - Maximum of 4 hours at 160 F or 1 Week at 70 F.
- Class 3: Materials Satisfactory for Short-Term Contact Only.
 - Less than 1 minute at 160 F and 1 hour at 70 F for unpressurized systems. Single use only.
- Class 4: Not recommended for use with H₂O₂.

TABLE 2 - STAINLESS STEEL COMPATIBLY WITH H₂O₂ [100]

When considering components that come into **contact with glass vials, Polyacetal (POM)** is a frequently chosen material in aseptic installations. This selection is based on the advantageous properties of POM, such as its high chemical resistance, excellent sliding properties, and low friction coefficients. [17]

1.6 Pressure cascade

EudraLex GMP, Vol 4 Annex 1 §53 indicates: 'A filtered air supply should maintain a positive pressure and an air flow relative to the surrounding areas of a lower Cleanroom grade under all operational conditions and should flush the area effectively'. [18]

The main purpose of maintaining a defined pressure and controlling the air flow in a cleanroom environment is to prevent cross-contamination between a grade A chamber/isolator and its surroundings. Unintended air leakage, whether it's from door openings or other sources, can lead to cross-contamination. By establishing a pressure gradient, the direction of the air flow can be managed effectively. [18] [19] [20]

The specific scenarios discussed below are based on the practical implementation of pressure cascade, as illustrated in Figure 15 and Figure 16 slides provided by SKAN, a leading company in cleanroom and isolator technologies. [21] [1]

Scenario depicted in Figure 17: **non-toxic product**

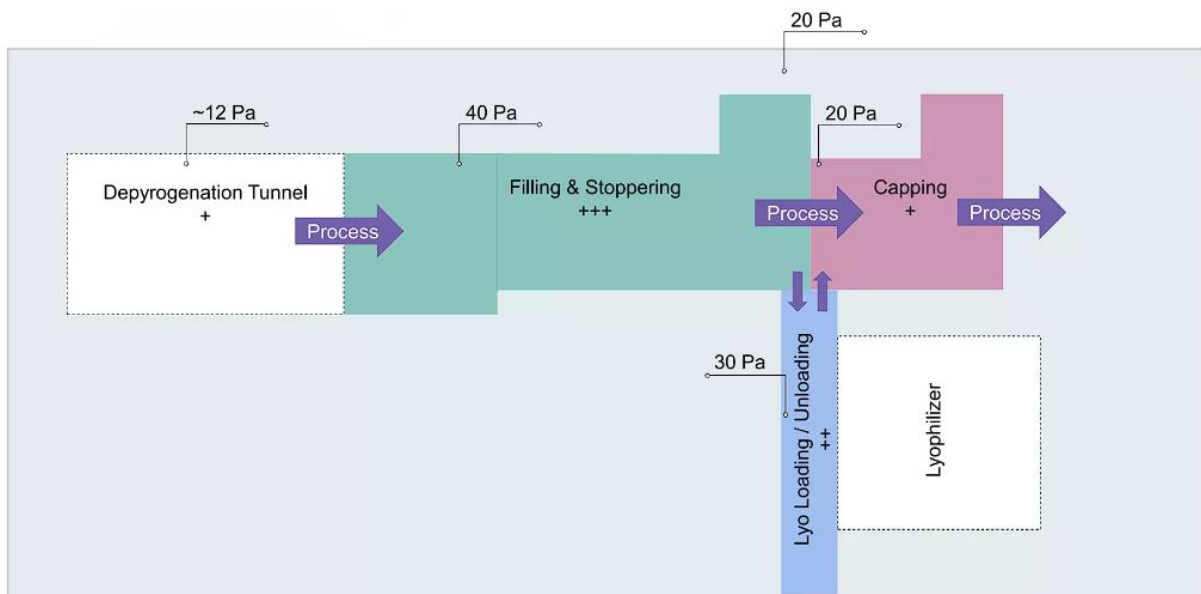


FIGURE 15 - SKAN EXAMPLE OF PRESSURE CASCADE FOR NON-TOXIC PRODUCT [1]

In this scenario, the **product being filled is non-toxic**. The filling and stoppering station exhibit a **positive pressure gradient** compared to the surrounding areas. To prevent cross-contamination, a cascade of increasing cleanliness areas is implemented. As the areas become cleaner, the pressure within them is gradually increased. The transitional room, acting as an airlock, ensures that any air entering the chamber is significantly cleaner and less contaminating. This setup helps maintain a controlled environment, minimizing the risk of contamination during the filling process. [19]

- Scenario depicted in Figure 18: **toxic/harmful product**

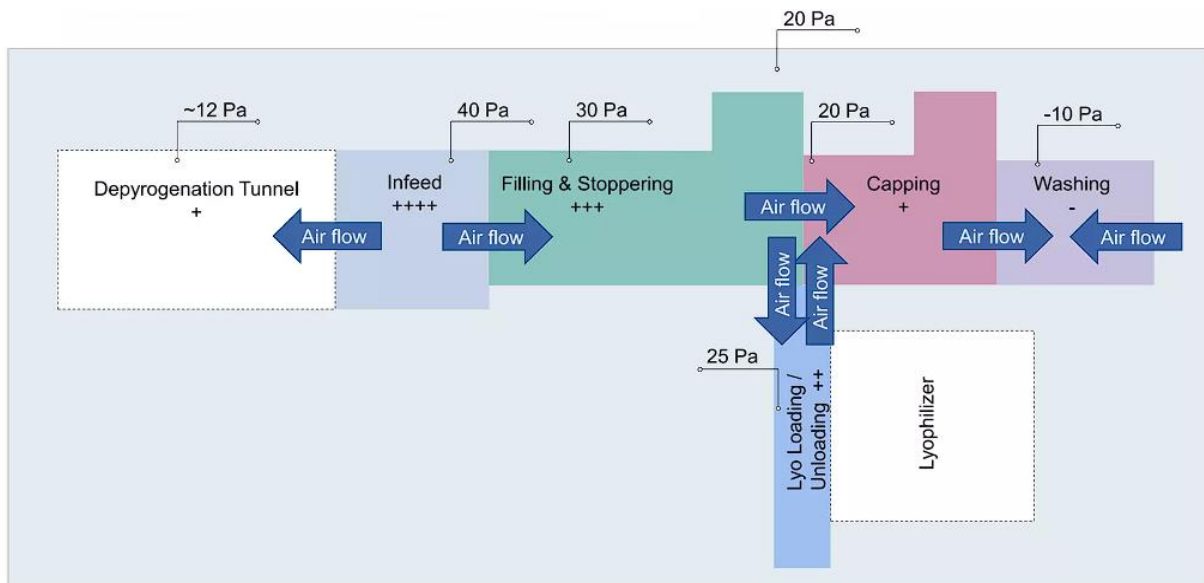


FIGURE 16 - SKAN EXAMPLE OF PRESSURE CASCADE FOR TOXIC PRODUCT [1]

In this scenario, the **product being filled is toxic or potentially harmful**. The filling and stoppering station have a **negative pressure gradient** relative to the surrounding areas. The purpose of this setup is to prevent the product from contaminating operators. By establishing a negative pressure gradient, any potential leakage of air from the chamber is directed inward, reducing the risk of exposure to harmful substances for the operators. The transitional room or airlock, in this case, acts as a containment barrier, ensuring the safety of personnel. [18] [19] [22]

These scenarios exemplify practical implementations of pressure cascade in cleanroom environments, demonstrating how pressure differentials and airlocks play a crucial role in preventing cross-contamination and ensuring the safety and quality of the products being handled. **In both cases, the pressure gradient between the insulator and the cleanroom is of the order of ten pascals.** [19] [23] [24]

- Are the SKAN values for the pressure gradient really representative of what shall be found for real?

Indeed, they are supported by the US Aseptic Processing Guidance which requires a static pressure gradient of 12.5 Pa for both “controlled and critical” zone. They are also supported by **the European Community (EC) GMPs which gives a range of 10 to 15 pascals**⁵. [25]

⁵ According to physics, a door opening lead to a temporary pressure impulse. Pressure is then balanced between both rooms and the airflow velocity only depends on their differential supply and exhaust-air. To date, the

Conclusion:

for the continuation of this project, an isolator pressure of 30 Pa with a positive pressure gradient of 15 Pa with respect to the cleanroom will be considered. To validate the practical relevance of the guidelines mentioned, a thorough examination of their origins is conducted in Appendix 7.8. This detailed study highlights that the guidelines are derived from a balance between energy consumption and efficiency considerations. Therefore, a solution developed based on these guidelines is expected to fulfil the requirements of the potential customers of Cilyx who are the target audience for this project.

competent authorities unfairly link the pressure differential with the air flow crossing opened doors between the chambers. [21]

1.7 Cleanroom isolator

Now that we have identified the constraints related to Grade A environments, it is crucial to consider the additional limitations associated with implementing filling processes within isolators. This aspect is particularly relevant to the current project. Eliminating direct operator involvement in the process is an effective approach to minimize product biocontamination. This concept has led to the adoption of isolators and restricted access barrier systems (RABS) for aseptic filling, as shown in Figure 17. [1]

An illustration of a Grade A isolator can be seen in Figure 18. In general, isolators are fully sealed units that are completely isolated from the external environment. Like the previously discussed clean chambers, they utilize independent air filtration systems to reduce particle levels within the enclosed space. Differential pressure is also utilized to mitigate any potential leakage, ensuring a higher level of cleanliness. [1]

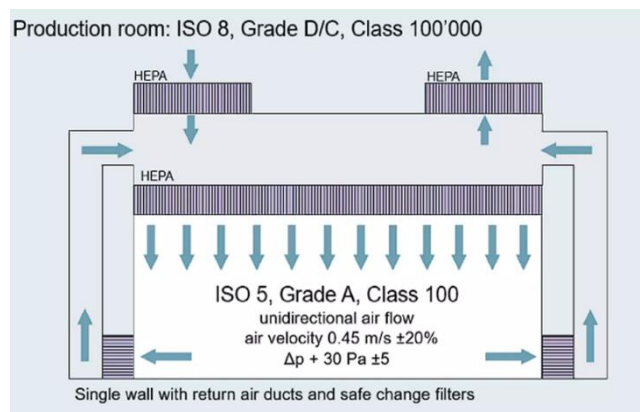


FIGURE 18 – ILLUSTRATION OF THE ENVIRONMENT OF AN GMP GRADE A ISOLATOR [1]



FIGURE 17 - EXAMPLE OF A MULTI-CHAMBER CLEANROOM ISOLATOR [68]

1.7.1 Isolator decontamination

When considering product changeovers, **specifically in this project focusing on filling multiformat vials**, isolators necessitate thorough **decontamination through H₂O₂ gassing**. Consequently, isolators are ideally suited for large-scale production runs of single products. The selection of vaporized hydrogen peroxide (VHP) H₂O₂ is driven by its favorable characteristics, which include: [26] [27] [28]

- Decomposition into water and oxygen
- Reasonable exposure times
- Rapid aeration capability
- Lower toxicity compared to alternative agents
- Compatibility with a wide range of materials

1.7.2 Difference between isolator and RABS

While both isolators and restricted access barrier systems (RABS) can create Grade A environments for aseptic manufacturing, they have distinct differences. RABS necessitates a Grade B (ISO 7) or better environment with fully gowned operators, whereas isolators provide superior product protection in an outdoor environment that can be downgraded to ISO 8 (Grade D). In addition, isolators have their own decontamination system. [26] [13]

In terms of energy consumption, RABS requires additional areas like airlocks and gowning rooms, and they do not recirculate fresh air. In comparison to cleanrooms, RABS are estimated to save 30% of energy, whereas isolators offer a greater energy savings of 65%. This further **supports the consideration of isolators**, not only in this study but also in most pharmaceutical companies. Comparative analysis of isolators and RABS can be found in Figure 19 and Figure 20. [26] [13]

RABS versus Isolator



RABS

- Surrounding Grade B
- Decontamination together with the surrounding room



Isolator

- Surrounding Grade D or better
- Integrated Decontamination System

FIGURE 20 - COMPARISON BETWEEN ISOLATOR AND RABS [27]

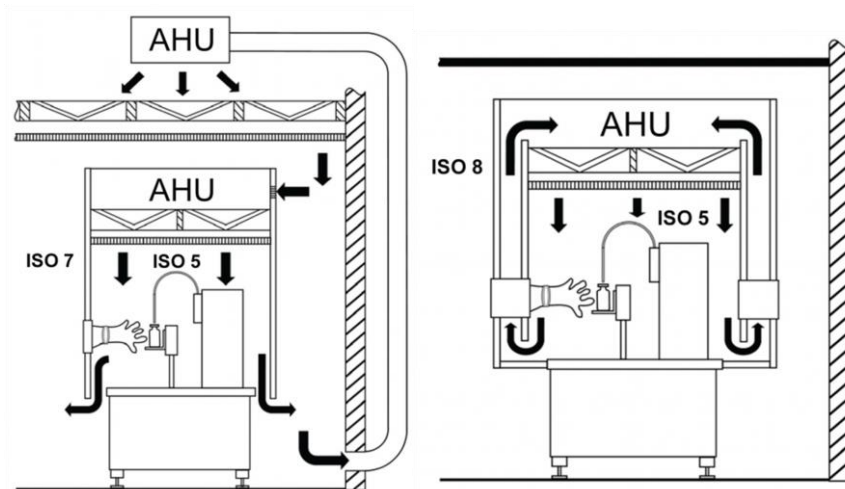


FIGURE 19 – DIFFERENCE OF AIRFLOW SYSTEM BETWEEN AN OPEN ACTIVE RABS (LEFT) AND AN ISOLATOR (RIGHT) [69]

1.7.3 Isolator openings

Openings, also called mouseholes, are said to be active if they open and close during the aseptic processing. Otherwise, they are said to be passive. To protect mouseholes from external environment, **RABS are often used as airlocks** to prevent any contamination of the isolator. In either case, openings are sealed when the isolator is being decontaminated. Figure 21 illustrates examples of an active mousehole, while Figure 22 showcases a passive mousehole. A more comprehensive view by the inside of an isolator is shown in Figure 23. [29]



FIGURE 21 - MANUALLY ACTIVE MOUSEHOLE [30]

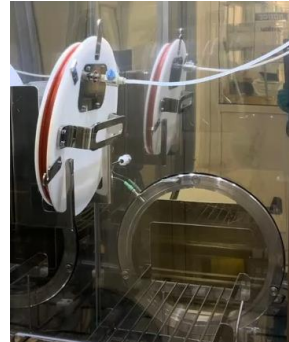


FIGURE 22 - PASSIVE MOUSEHOLE WITH SEAL [30]



FIGURE 23 - INSIDE VIEW OF AN ISOLATOR [30]

1.7.4 Manual operations

Even within an isolator, there are situations where manual control is necessary inside the enclosure, such as when changing the system configuration. Typically, this is accomplished through the use of **glove ports or half-suits, which impose geometric constraints on equipment accessibility**. These manual access systems enable operators to perform adjustments, maintenance, and even equipment repairs. [29] [30]

As shown in Figure 24, Glove ports are sealed sleeves through which operators can insert their arms to handle components within the enclosure. **Standard isolators usually have gloves of approximately 81 cm in length,** ([A Guide to Isolator Gloves | Fabtech \(fabtechnologies.com\)](#)) made from materials selected based on the specific products being handled. However, glove ports are a common source of contamination, as they can develop rips or pinholes in the gloves. Therefore, it is recommended to **minimize their usage to reduce contamination risks**. [31]



FIGURE 24 – GLOVES USED IN ISOLATOR [70]

To address the limitations of glove ports, an alternative solution is the use of a half-suit. As shown in Figure 25, a half-suit is a protective garment that is connected to the isolator chamber. It is airtight and accessible from below, enabling the operator to handle objects that may be out of reach for the gloves due to their limited length. [31]



FIGURE 25 - HALF-SUIT FOR ISOLATOR [70]

However, it's important to note that **half-suits** have large geometrical dimensions and are less commonly utilized. Therefore, they **will not be considered** in the development of the solution, given their rarity and limited practical implementation.

1.7.5 Summarizing the advantages and limitations of isolators

To summarize the advantages and limitations of isolators:

Advantages:

- Isolators offer enhanced product protection in a controlled and sealed environment.
- Isolators can provide a Grade A environment while surrounded by a Grade C or D environment.
- They ensure operator safety by minimizing exposure to hazardous materials.
- Isolators reduce the risk of product contamination.
- They are energy-efficient, resulting in cost savings and environmental benefits.
- They are easy to monitor and decontaminate.

Limitations:

- Isolators may have restricted accessibility, requiring the **use of glove ports** or half-suits.
- Due to this restricted accessibility, the enclosed components of an isolator must be reliable and **easily accessible for maintenance and adjustment**.
- Product changeovers within isolators can be time-consuming due to decontamination procedures.

Overall, isolators provide significant advantages in terms of product protection and operator safety, despite limitations related to accessibility and changeover procedures. To this day, isolators are still considered **the most economical and ergonomic solution for aseptic filling as long as they comply with Figure 26**.

Grade A Unidirectional Air Flow (Aseptic Critical Zone Filling Line)




- Unidirectional airflow shall not be disturbed above open container like vials, syringes etc.
- Air should not return from areas below to the filling zone or open containers.

FIGURE 26 – SLIDE ON AIRFLOW DISTURBANCE OF THE FDA-EMA ASEPTIC REQUIREMENTS ANNEXE 1 POWERPOINT PRESENTATION THEY MADE IN AUSTRALIA ON THE 20TH SEPTEMBER 2019 [71]

1.8 Vials

While aseptic filling processes commonly involve syringes and cartridges, this project will specifically focus on **glass vials**, which are also widely used. This project undertakes vials provided by Pfizer company, for which dimensions are available in Appendix 7.12. In general, glass vials can be categorized into two main groups: infusion vials and injection vials. However, the project will concentrate on injection vials. In either case, the shape and dimensions of these containers are regulated by their respective ISO standards to ensure interchangeability. Appendix 7.7 provides information on infusion bottles. [32] [33]

For injection bottles made from glass tubes, which are the specific focus of this project, ISO 8362-1:2018 standard is applicable, independent of surface treatment. This standard pertains to containers intended for the packaging, storage, or transportation of injection products. Its denomination is as follows: a 2R vial refers to a vial intended to be filled with a volume of 2 mL. For injection bottles moulded from borosilicate or soda-lime glass, ISO 8362-4:2011 is the applicable standard, and more information can be found in Appendix 7.8. [34]



Reference	Nominal capacity	Brimful capacity	Weight	Standard neck-finish	Height	Ø	Thickness
2R	3.4	4	4.4	13 mm	35	16	1
4R	4.8	6	5.7	13 mm	45	16	1
6R	8.3	10	7.9	20 mm	40	22	1
8R	9.8	11.5	8.7	20 mm	45	22	1
10R	11.5	13.5	9.5	20 mm	45	24	1
15R	17.3	19	12	20 mm	60	24	1
20R	21.7	26	16.2	20 mm	55	30	1.2
25R	27	32.5	18.9	20 mm	75	30	1.2
30R	33	37.5	21.9	20 mm	75	30	1.2
50R	49	62	34.5	20 mm	73	40	1.5
100R	75	123	60	20 mm	100	47	1.7

Figure 27 illustrates the ISO 8362-1:2018 norm mentioned above. Among the various types of vials, it has been decided in Section 3 on solution development to focus on filling **2R, 15R, and 30R vials (= 2 mL, 15 mL and 30 mL vials)**. By selecting these specific sizes, a wide range of bottom diameters can be accommodated, and the solution can be easily adapted to intermediate diameters as well. [35]

FIGURE 27 - SGD PHARMA CATALOG OF ISO 8362-1:2018 COMPLIANT VIALS. IN RED ARE THE VIALS TARGETTED BY THIS PROJECT, IN GREEN THEIR BOTTOM DIAMETER. [72]

1.8.1 Ready-to-Use (RTU) packaging

This project focuses on the filling process, assuming that the vials entering the filling cell have already undergone sterilization and depyrogenation. However, if these steps are not performed earlier in the production line, a common solution is to use Ready-to-Use (RTU) vials. These vials are already sterilized and are packaged in nests contained within tubs and covered with Tyvek material. This type of packaging shown on Figure 28. can maintain sterility for a minimum of 5 years. To make the vials accessible for filling, several operations impacting the material flow and discussed in Section 2.3 on RTU infeed are required. As a general rule, vials placed in a nest can be accessed from the bottom and can be easily removed by applying pressure. [36] [37]



FIGURE 28 - RTU VIALS PACKAGING SHOWING VIALS LYING ON A NEST, COVERED BY A TYVEK. THE WHOLE IS SEALED IN A TUB. [45]

2. State of the art

To design an effective 2D layout for the aseptic filling line, it is crucial to consider the level of automation and the desired production rate. The transfer of sterilized containers into the filling unit is a key aspect of the filling process. Depending on the desired production rate, containers can be supplied in bulk for high-speed operations or in Ready-to-Use (RTU) format for smaller volumes. Therefore, a thorough analysis of this aspect is conducted as a preliminary step in the state of the art. Once this analysis is completed and the knowledge is acquired, a more focused and detailed state of the art on the filling station is performed. This enables the selection of the final solution that aligns with the chosen production rate and glass format.

2.1 Filling line processes: overview

A filling line combines RABS, isolators, and, occasionally, other elements such as a depyrogenation tunnel to perform operations on vials shown in Figure 29. This project specifically concentrates on vial filling and stopper insertion processes.

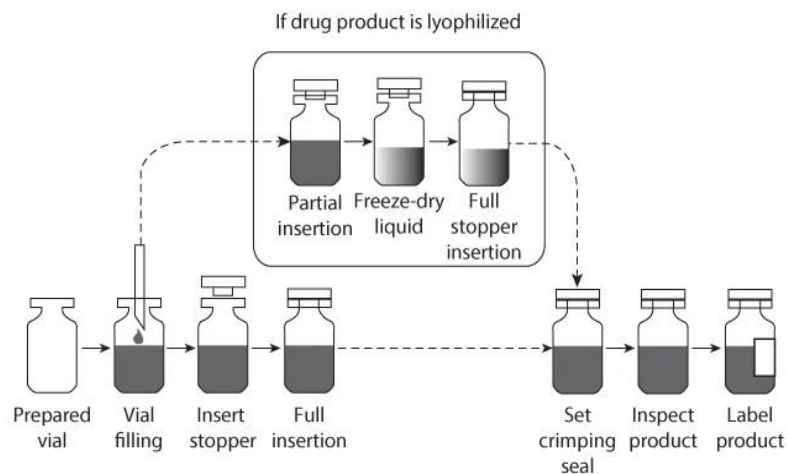


FIGURE 29 - OVERALL OPERATIONS PERFORMED ON VIALS [73]

Figure 30 and Table 3 display an example of a highly flexible filling line available on the market and the maximum cadence it can reach. In the following sections, each component of the filling line, beginning with the infeed and progressing to the filling and stoppering units, will be examined. Operations downstream from these processes fall outside the scope of this work and will not be addressed.

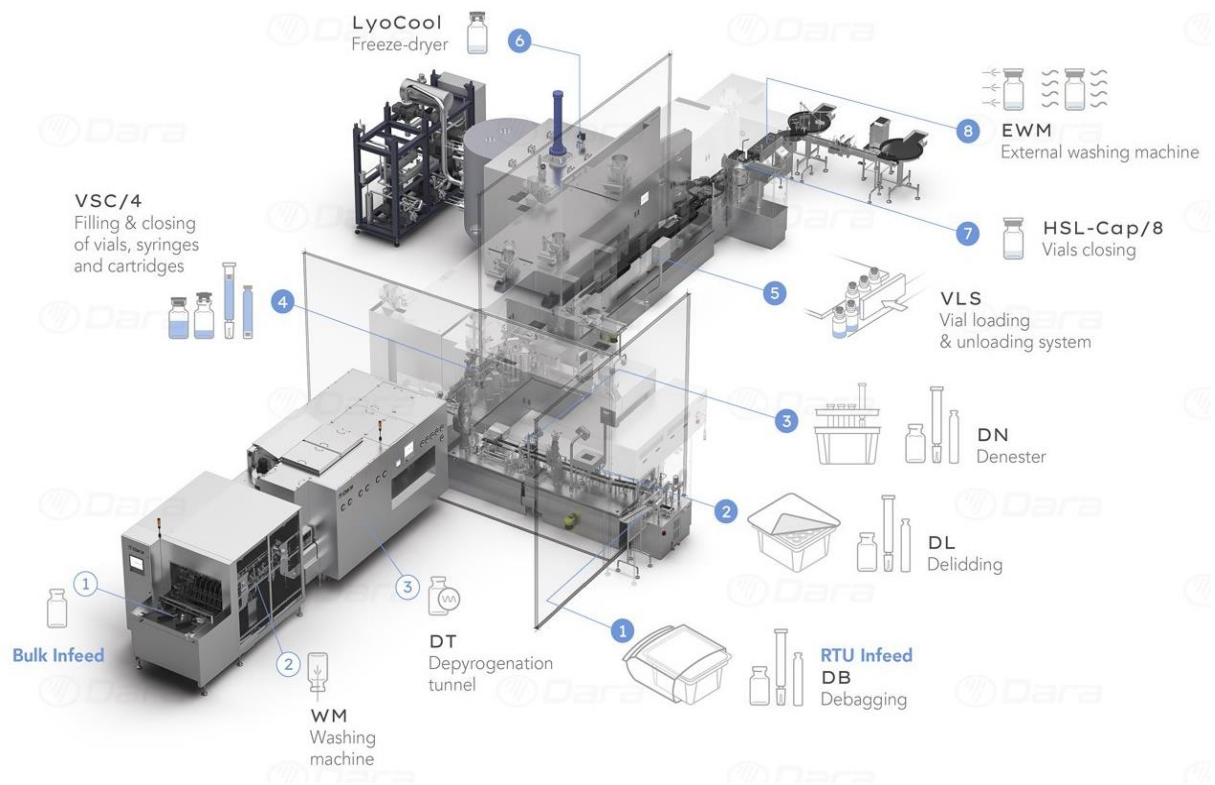


FIGURE 30 - EXAMPLE OF DARA COMBI LINE FOR ASEPTIC FILLING [74]

Complete line		WM-20	DT-20	DB/A	DL/A	DN/E	VS-Lyo/4	VLS	LyoCool	HSL-CAP/8
Process		Washing machine	Depyrogenation tunnel	Debagging	Delidding	Vials & syringe denester	Filling & closing	Vial loading & unloading system to the freeze dryer	Freeze dryer	Vials closing
Output max.	Vial Ø 36 - h 75 mm max.		12,000 uph	115 Tubs/h	115 Tubs/h			12,000 uph		
	Syringe 0,5 - 5 ml	-	-	115 Tubs/h	115 Tubs/h			12,000 uph		
	Cartridge 3 ml	-	-	115 Tubs/h	115 Tubs/h			12,000 uph		

TABLE 3 - COMPONENTS AND MAXIMUM CADENCE OF THE DARA COMBI LINE [74]

2.2 Bulk infeed

Incorporating a depyrogenation tunnel in the filling line provides a reliable method for achieving sterilization. Despite the high equipment cost, such tunnel offers the advantage of ensuring a **consistent and high-speed infeed process while maintaining a Grade A environment**. This process ensures the sterilization and removal of potential contaminants that could compromise the safety and quality of the final product. Indeed, it effectively eliminates pyrogens, which are fever-inducing substances, from their surfaces through controlled heat exposure. [38] [39]

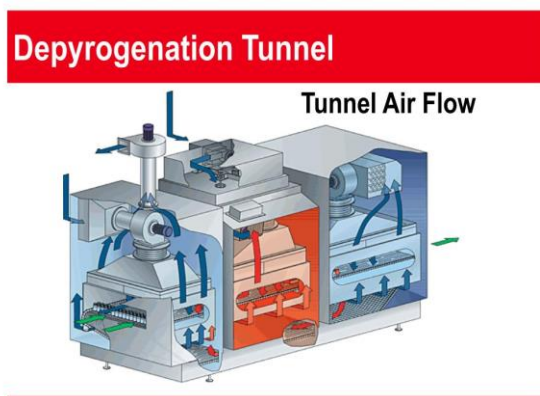


FIGURE 31 - DEPYROGENATION TUNNEL ILLUSTRATION [75]

Figure 31 and Figure 32 highlight the functionality and integration of the depyrogenation tunnel. Throughout the process, washed glass ampoules and vials are conveyed on a stainless-steel wire conveyor, starting from the non-sterile automatic washing machines and ending at the downstream Grade A filling line. [40]

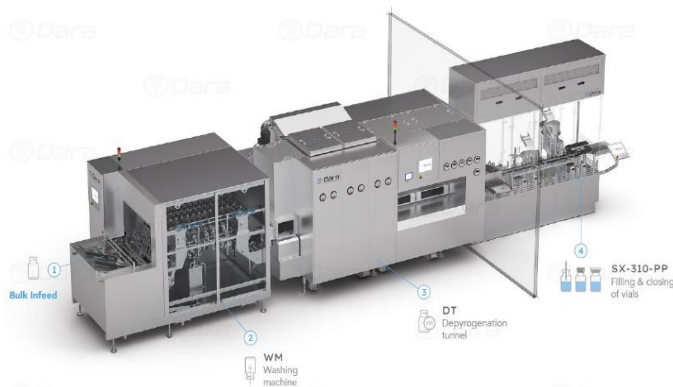


FIGURE 32 - DARA DEPYROGENATION TUNNEL INTEGRATION [76]

This continuous conveying method facilitates a seamless and uninterrupted supply of vials to the filling line, ensuring efficiency and maintaining the desired sterility throughout the production process. [40]

The various machines offered by DARA company are capable of achieving the maximum cadence indicated in Table 4. When it comes to interfacing the output flux with the filling station, Figure 33 below showcases the **utilization of accumulation tables, endless screws and wheels**.

Model		DT-9	DT-20	DT-26	DT-40	DT-50
Output max.	2R	12,000 uph	29,000 uph	36,000 uph	> 36,000 uph	> 36,000 uph
	10R	6,350 uph	13,100 uph	18,850 uph	26,000 uph	32,000 uph
	30R	4,000 uph	8,400 uph	11,875 uph	16,500 uph	22,500 uph
	100H	690 uph	1,650 uph	2,055 uph	3,050 uph	3,600 uph

TABLE 4 - DARA CATALOGUE FOR DEPYROGENATION TUNNEL [76]

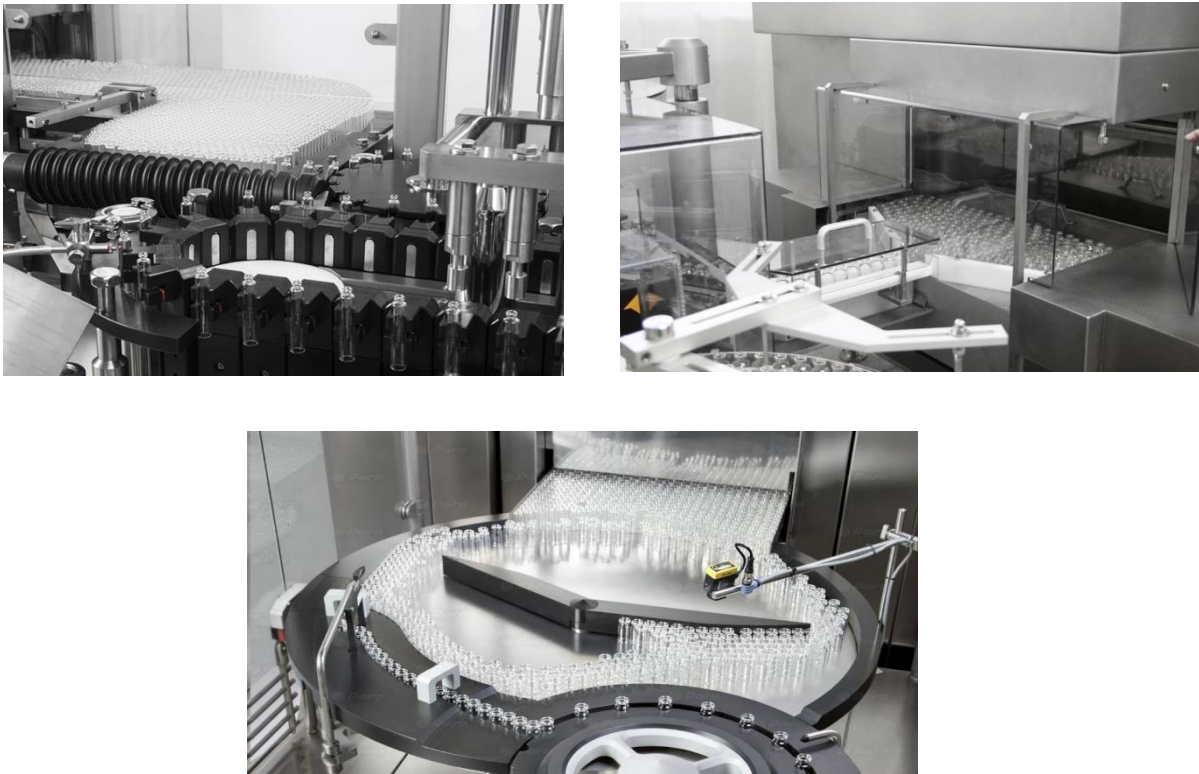


FIGURE 33 - EXAMPLES OF DEPYROGENATION TUNNEL OUTPUT MANAGEMENT [77] [78] [79]

2.3 RTU infeed

An alternative to sterilizing vials within the filling line is the use of Ready-to-Use (RTU) packaging. This approach eliminates the need for a depyrogenation tunnel. While the facility cost can be reduced, additional processes are involved and may result in a lower production cadence. Throughout this chapter, the processes depicted in Figure 35 and Figure 34 will be explored in further detail to determine their achievable levels of cadence.

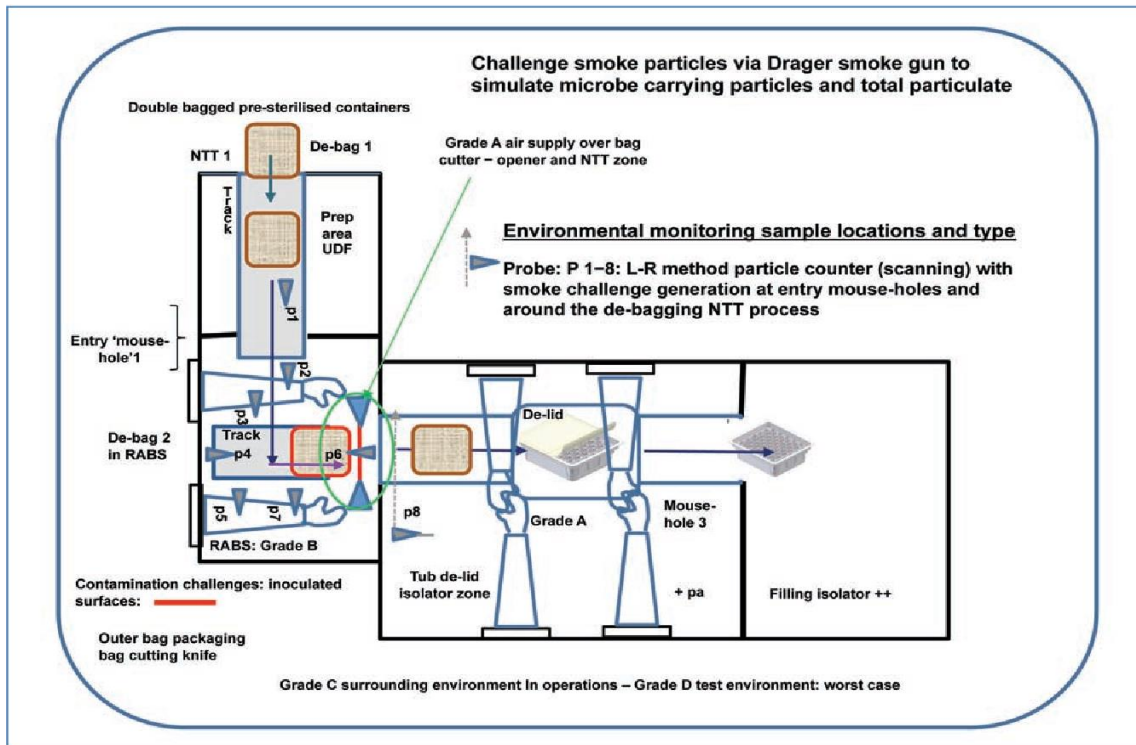


FIGURE 35 - TYPICAL MANUAL HANDLING OF RTU, WITH GRADE ZONE SPECIFICATIONS. NTT = NO TOUCH TRANSFER [41]

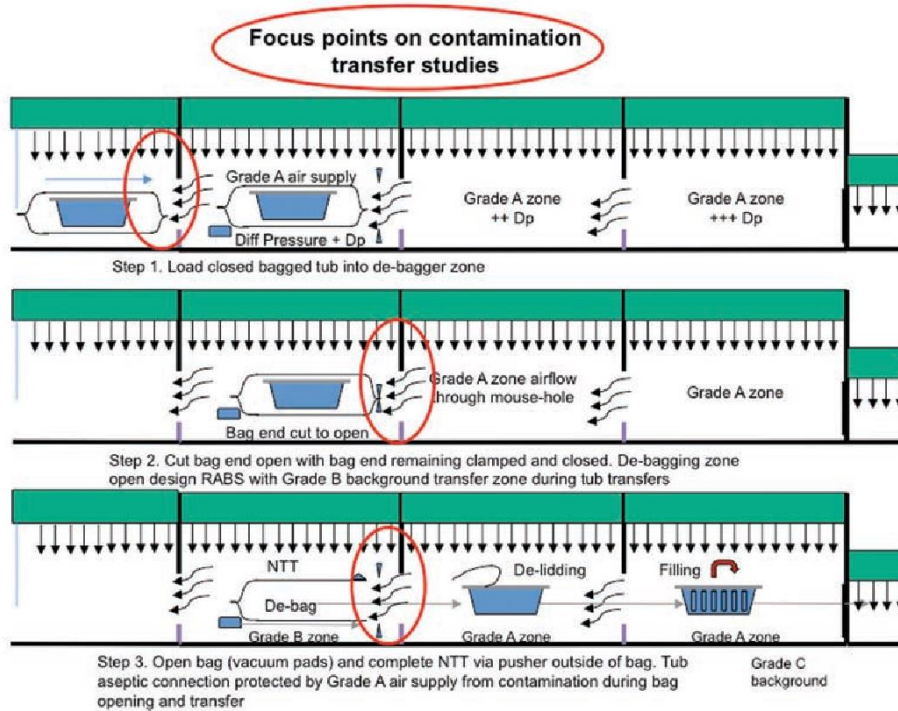


FIGURE 34 - STEPS OF RTU TRANSFER [42]

2.3.1 Debagging

Debagging refers to the process of removing the bag from the RTU packaging. It is typically carried out within an open RABS as the one illustrated in Figure 36. During debagging, it is important to ensure a Grade A air supply is positioned above the bag cutter. Additionally, the back of the bag should be securely clamped from the start of the cutting process until the transfer of the tub to the next chamber is completed. These pieces of information are illustrated on the previous Figure 35 and Figure 34. [40] [41]

Figure 37 depicts the interior of the DARA RABS model DB/A+ specifically designed for automatic debagging. The maximum cadence of the automatic solution provided by DARA is outlined in Table 5. For additional examples and further information on debagging, please refer to Appendix 7.9.



FIGURE 36 - DARA RABS MODEL DB/A [80]

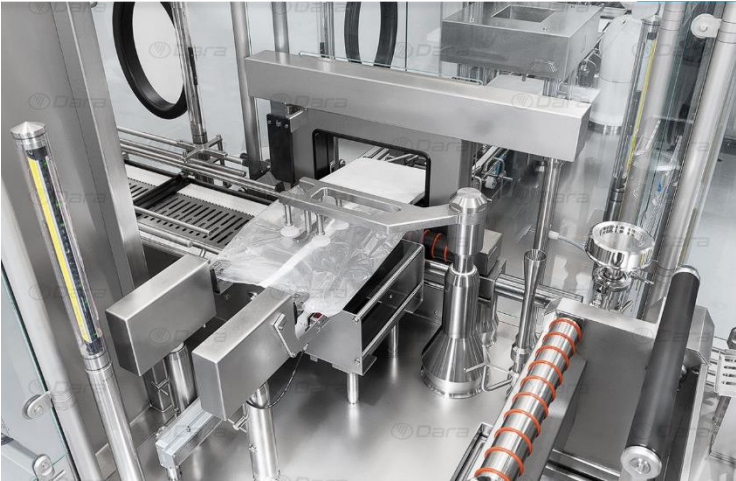


FIGURE 37 - INSIDE OF THE DARA MODEL DB/A+ [44]

Model	DB/A	DB/A+
Output max.	115 Tubs/h	225 Tubs/h

TABLE 5 - DARA DEBAGGING MACHINERY MAXIMUM CADENCE [80]

2.3.2 Delidding

De-lidding refers to the process of removing the top trim from tubs. This process consists of several steps, as illustrated below, using the DARA DL/A automatic de-lidding solution. The maximum achievable cadence of DARA de-lidding machinery is provided in Table 6. [42] [43] [44]



FIGURE 38 - DARA DL/A LID HEATING STEP [45]

Step 2: Lid removal

In the example shown in Figure 39, a Staübli robotic arm grasps the lid at one of the corners of the tub and pulls it out, effectively removing the lid.



FIGURE 39 - DARA DL/A LID REMOVAL STEP [45]



FIGURE 40 - DARA DL/A INNER TYVEK REMOVAL STEP [45]

Step 3: Inner Tyvek removal

This process depicted in Figure 40 is consistently carried out using suction, ensuring effective and efficient removal of the Tyvek material.

Model	DL/A	DL/A'
Output max.	115 Tubs/h	225 Tubs/h

TABLE 6 - DARA DL MACHINERY MAXIMUM CADENCE [45]

2.3.3 Denesting

Denesting refers to the extraction of containers from their nests. This process involves multiple steps, as demonstrated below, using the DARA DN/A automatic denesting solution. The maximum achievable cadence of DARA denesting machinery is presented in Table 7. [45]

Although filling containers directly within the nests is possible, it is not recommended due to difficulties in sampling, labeling, and managing individual containers. As the denesting process is of interest for Cilyx, more detailed information can be found in Appendix 7.10. [45]



FIGURE 41 - EXAMPLE OF VIAL PICKING FOR DENESTING

Step 1: Vial picking

The first step shown in Figure 41 involves placing the nest on a shallow plate with perforations that allow the vials to protrude. A robotic arm is used to pick up the vials, either by applying pressure or using suction.

Step 2: Outfeed

After the vials have been picked, they are transferred and placed onto a conveyor system, as depicted in Figure 42. This can be accomplished either directly by the robotic arm or by utilizing a combination of a gravity rail and an accumulation table before reaching the conveyor. Although this latter method allows for a higher processing speed, it does result in a compromise on the level of contact between the vials.



FIGURE 42 - EXAMPLE OF DENESTED VIALS DIRECTLY PLACED ON AN ENDLESS-SCREW CONVEYOR

Step 3: Nest Disposal

The final step, illustrated in Figure 43, involves the removal of the empty nest. To accomplish this, the empty nest is placed onto a rail system that carries it out of the isolator.



FIGURE 43 - EXAMPLE OF EMPTY NEST DISPOSAL [46]

Model	DN/N	DN/E	DN/E*
Output max.	9,000 uph	12,000 uph	24,000 uph

TABLE 7 - DARA DENESTING CATALOG [46]

Optionnal: E-beam tunnel and pulsed light sanitization

In the case of RTU supply, it is possible to incorporate an extra active decontamination process before the filling line to eliminate any bioburden that might have entered the system. Common techniques used for this purpose are electron beam tunnel and pulsed light sanitization process. While these processes can affect the throughput, they are not universally implemented. For the sake of brevity, detailed explanations about these processes can be found in Appendix 7.4. [46]

2.4 Filling and stoppering technologies

This section, which forms the essence of this project, will comprehensively analyze numerous filling and stoppering stations in terms of their achievable cadence. After gaining insights into the vial filling rates currently provided by leading industry players⁶, the objective is to determine the specific market that Cilyx Engineering aims to target and devise an effective solution to establish a foothold in that market.

2.4.1 Small scale manufacturing: 30 – 120 VPM

Versynta	30 VPM	S-C system	1 by 1, one step
----------	--------	------------	------------------



FIGURE 44 - VERSYNTA MACHINE ACHIEVING 30 VPM [81]

In the first showcased example by Versynta, as depicted in Figure 44, vials are **picked up using servomotors, transported in a straight line by a cylinder, and then released using servomotors.** This creates an **intermittent walking beam-like motion**, enabling the **individual retrieval** of vials from the accumulation table.



FIGURE 45 - VERSYNTA STOPPERING STATION [81]

Figure 45 demonstrates the stoppering process, where a shaft with **vertical and rotational movement** is utilized. The sealing mechanism appears to involve the use of a **lip seal**.

In the subsequent state-of-the-art study, the system that combines servomotors with a cylinder to achieve linear stroke and intermittent motion will be referred to as an "**S-C system**," which stands for "**Servo-cylinder system**."

Flexicon FPC60	45 VPM	S-C system, 0.2 to 100mL	1 by 1, one step
----------------	--------	--------------------------	------------------

⁶ Regarding vial filling lines, several major players in the industry include Jerempli, DARA, Syntegon, Groninger, Tofflon, Bausch-Ströbel, and Watson Marlow (formerly Flexicon).

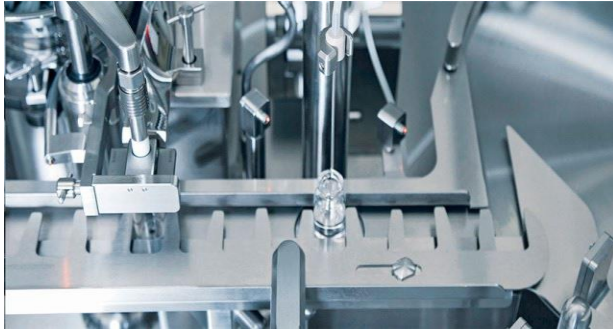


FIGURE 47 - FLEXICON FPC60 FILLING AND STOPPERING [82]

In the case of the example from Flexicon depicted in Figure 47, the output cadence is achieved through **manual denesting**. The operator pushes the vials onto an **accumulation table**, and they are subsequently picked up **one by one** by an **S-C system**. The filling process is carried out individually.



FIGURE 46 - FLEXICON FPC60 STOPPERING AXIS [82]

As illustrated in Figure 46, the stoppering process is conducted in a similar manner to the previous example. The stoppering robot axis is sealed using a **lip seal**.

Versynta	60 VPM	3 robotic arms	2 by 2, one step
----------	--------	----------------	------------------



FIGURE 48 - VERSYNTA MACHINE ACHIEVING 60 VPM [83]

The Versynta machine depicted in Figure 48 utilizes **three robotic arms** to transfer vials from the nest to the filling station and the stoppering station. This configuration allows for small batch sizes and enables thorough analysis. However, due to its **high cost, this solution is not suitable for the current project.**

Watson-Marlow FMB210	75 VPM	Non-continuous rotative wheel	2 by 2, two steps
----------------------	--------	-------------------------------	-------------------



FIGURE 49 - WATSON-MARLOW FMB210 [84]

The Watson-Marlow FMB210 filling station of Figure 49 is **compact** and uses **manual denesting**. Vials are placed on an **accumulation table** and **transferred individually to a rotating wheel using an endless screw**. **Filling occurs in two steps, followed by stoppering in one step.**

The stoppering process utilizes a vertical plunger. However, this method does not align with the sealing management requirements of this project.

Jerempli NFL/1-2-RDL	85 VPM	In nest 20 mL	2 by 2, one step
----------------------	--------	---------------	------------------



FIGURE 50- JEREMPLI NFL/1-2-RDL FILLING IN NEST STATION [85]

For small nest-filled batches, an example is the NFL/1-2-RDL filling machine from Jerempli shown in Figure 50. However, it is not suitable for in-process control (IPC) as a filling issue could contaminate the entire tub. **Consequently, nest filling will not be considered as a viable solution for this project.**

The NFL/1-2-RDL filling station incorporates a shaft that combines both rotation and translation movements. To maintain tightness, **a lip seal** is utilized.

Jerempli SX-220-PP	100 VPM 0.1 to 250 mL	intermittent rotating wheel 2 by 2 entrance	2 by 2, three steps
--------------------	-----------------------	--	---------------------



FIGURE 51 - SX-220-PP STOPPERING STATION [86]

For stoppering, as depicted in Figure 51, a shaft with two pairs of grippers is employed. The shaft rotates and then moves vertically, enabling one pair of grippers to place stoppers on the vials while the other pair retrieves new ones.

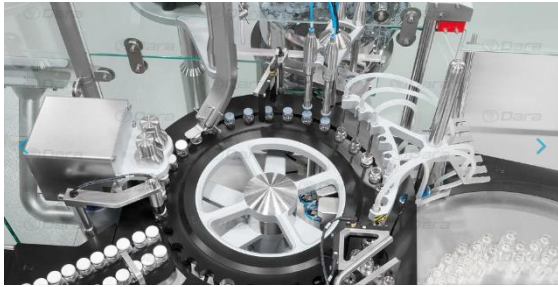


FIGURE 52 - JEREMPLI SX-220-PP FILLING STATION [86]

The SX-220-PP filling station shown in Figure 52 also utilizes an **intermittent rotating wheel**, similar to the FMB210 filling station. It achieves a higher speed of 100 VPM by **filling vials in pairs in 3 steps** and **capping them in pairs in one step**.

2.4.2 Medium scale manufacturing: 120-150 VPM

Jeremphi SX-310-PP	120 VPM 0.1 to 250 mL	S-C system	2 by 2, 2 steps
--------------------	-----------------------	------------	-----------------

The SX-310-PP filling station, illustrated in Figure 53, employs an **S-C system** to transfer vials from **two slots** to the right. Pairs of vials are **then filled in two stages and stoppered in one**. The stoppering process is executed by a **rotating shaft that also moves vertically**. **Lip seals** are utilized to maintain tightness.



FIGURE 53 - JEREMPLI SX-310-PP FILLING STATION [87]

Tofflon	120 VPM	S-C system	3 by 3, one step
---------	---------	------------	------------------

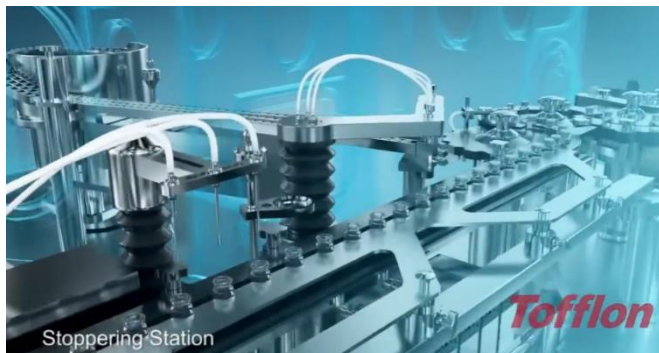


FIGURE 54 - TOFFLON 120 VPM FILLING LINE [88]

In the filling line depicted in Figure 54, a **S-C system** is employed to **fill the vials in groups of three**.

The stoppering station utilizes a **shaft that rotates and translates**, while maintaining a seal with the help of a **bellow**.

Cook Pharmica	150 VPM for 2 mL vials	S-C system	4 by 4, one step
---------------	------------------------	------------	------------------



FIGURE 55 - COOK PHARMACICA 150 VPM FILLING LINE [89]

In the filling line illustrated in Figure 55, a **S-C system** is utilized to move and fill vials **in batches of four**. During motion, **the vials are secured between a track and a stainless-steel framework**.

The stoppering station in this setup employs a sealed shaft that combines rotational and translational movements. The sealing is ensured by a **bellow**.

2.4.3 High scale manufacturing: 200-600 VPM

Bosch FLC 3080	200 VPM	Continuous conveyor belt	8 by 8, one step
----------------	---------	--------------------------	------------------

The Bosch FLC 3080 filling station, depicted in Figure 56, utilizes a **plastic conveyor** to handle and transport bottles by their necks continuously. Multiple wheels along the line are capable of picking up five vials at a time for in-process control (IPC) before returning them to the conveyor.



FIGURE 56 - BOSCH FLC 3080 FILLING STATION [90]



FIGURE 57 - BOSCH FLC 3080 STOPPERING STATION [90]

While the filling process is carried out in **batches of 8 bottles** at a time, **the stoppering process is performed using a wheel** that is not perfectly aligned on a horizontal plane. As the wheel rotates over the bottles, it inserts the stoppers inside them. This stoppering process is illustrated in Figure 57.

Jerempli NFL/5-10	375 VPM	In nest	5 by 5, one step
-------------------	---------	---------	------------------



FIGURE 58 - JEREMPLI NFL/5-10 FILLING STATION [91]

Although filling in nests was not considered of interest in this project, Figure 58 is still relevant as it demonstrates that **a shaft with combined rotation and translation can be effectively sealed by a lip seal, even at high frequencies.**

Jerempli HSL-PP	400 VPM	S-C system	4 by 4, two steps
-----------------	---------	------------	-------------------



FIGURE 59 - JEREMPLI HSL-PP FILLING STATION [92]

The Jerempli filling station depicted in Figure 59 features a **S-C system** for filling and stoppering. The **filling process is conducted in two steps**, while stoppering is performed with a **shaft that combines rotation and translation, sealed with a lip seal**. The station allows for 4 by 4 stoppering.

DARA	600 VPM 0.1 to 540 mL	S-C system	8 by 8, one step
------	-----------------------	------------	------------------



FIGURE 60 - DARA 600 VPM FILLING LINE [93]

The Dara filling station shown in Figure 60 is similar to the HSL-PP station, but it achieves a higher output by filling and stoppering **8 vials simultaneously**.

Watson-Marlow	600 VPM	Continuous conveyor belt	12 by 12, one step
---------------	---------	--------------------------	--------------------

The Watson-Marlow filling station depicted in Figure 61 utilizes a **plastic endless screw** to transfer the bottles onto a **plastic wheel conveyor**. From there, they are taken by a **belt conveyor** and **filled continuously using a 12-needle** filling station. Stoppering is carried out in motion by a **vertically rotating wheel**.

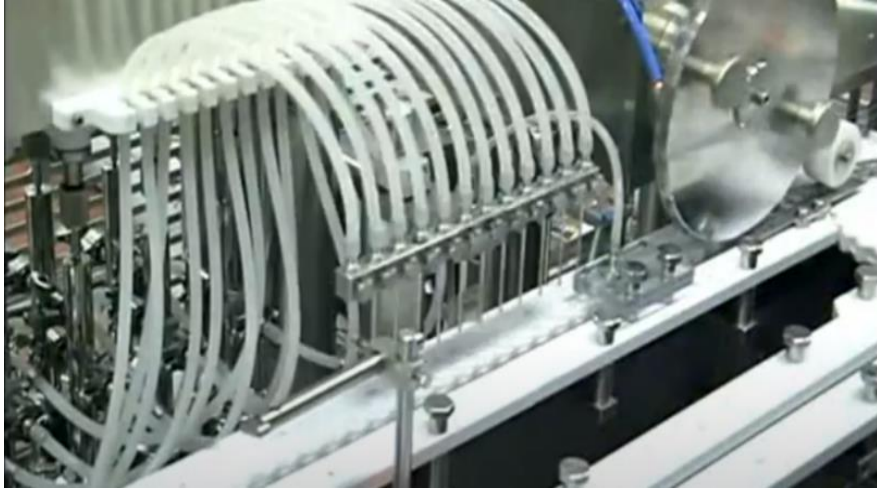


FIGURE 61 - M.A.R 600 VPM FILLING LINE [94]

2.5 Key features summary

2.5.1 Regarding infeed of the filling station

As the project excludes filling in nests, the filling stations considered will either involve bulk infeed via depyrogenation tunnel or denesting. According to Table 7, the achievable cadence for 2R vials using the cheapest DARA depyrogenation tunnel is 12,000 units per hour, which translates to **200 vials per minute**. On the other hand, their cheapest automated denesting station has a maximum output of **150 VPM for 2R vials**.

Model		DT-9	DT-20	DT-26	DT-40	DT-50
Output max.	2R	12,000 uph	29,000 uph	36,000 uph	> 36,000 uph	> 36,000 uph
	10R	6,350 uph	13,100 uph	18,850 uph	26,000 uph	32,000 uph
	30R	4,000 uph	8,400 uph	11,875 uph	16,500 uph	22,500 uph
	100H	690 uph	1,650 uph	2,055 uph	3,050 uph	3,600 uph
Model		DL/A			DL/A*	
Output max.		115 Tubs/h			225 Tubs/h	
Model		DN/N		DN/E		DN/E*
Output max.		9,000 uph		12,000 uph		24,000 uph

TABLE 8 - GENERAL SUMMARY OF THE MAXIMUM CADENCE AVAILABLE IN DARA COMPANY ACHIEVABLE THROUGH THEIR DEPYROGENATION TUNNEL (DT), READY-TO-USE PROCESSES (RTU) AND DENESTING (DN) [46] [45] [76]

2.5.2 Regarding the filling and stoppering themselves

Output max	Denesting	Conveyors	Needles	Seals
Small scale : 25-100 VPM	Manual Automated	S-C system, Wheels	1-2 1-2 steps	Lip seal
Medium scale : 120-150 VPM	Automated	S-C system	3-4 1-2 steps	Lip seal Bellow
High scale : 200-600 VPM	Automated	Conveyor belt, S-C system	8-12, 1-2 steps	Lip seal

In all previous cases of the state-of-the-art solutions, components in contact with the vials during their movement are **made of POM**. The remaining parts are made of **stainless steel**.

Furthermore, during the retraction of the S-C systems and the filling process, **a comb with triangular teeth is used** to securely hold the vials against a guide rail, ensuring their stability and precise positioning.

In the majority of the previous installations featuring a shaft that combines both rotation and translation, **lip seals are employed** to ensure proper sealing.

3. Solution development

After discussions with Cilyx, it was decided to achieve a speed of 150 VPM for 2R vials using a walking beam mechanism, enabling intermittent motion without requiring the S-C system. Filling will occur on a vertical axis, ensuring a filling volume accuracy of 0.5%. The stoppering process will involve a shaft that combines rotation and translation. The solution should accommodate multiple formats, including 15R and 30R vials, while maintaining a minimum flow rate of 300 mL/min, which is determined by the constraints on the 2R cadence.

There are several motivations behind these choices. Firstly, Cilyx already has experience implementing a wheeled conveyor and seeks to expand its knowledge of the walking beam mechanism to broaden its range of solutions. Secondly, the company already possesses solutions for high-speed operations, and since low speeds are less automation-oriented, a moderate performance of 150 VPM was requested. A figure that already appeared in the analysis of Table 7 in the previous chapter.

The requirement for accuracy in the filled volume stems from a previous offer made to Cilyx, which sought a filling solution at a rate of 25 VPM for 2R vials while maintaining a 0.5% volume accuracy.

Lastly, Cilyx lacks a comprehensive understanding of sealing in an aseptic environment, leading to the selection of a shaft which combines rotation and translation for the stoppering station.

The development of the solution will be structured as follow:

- Section 3.1: Justification of the choice of the box transport mechanism.
- Section 3.2: Implementation of a filling circuit compliant with the requirements and defining the stroke.
- Section 3.3: Presentation of kinematic equations implemented in Python, leading to a scaled standard box transport mechanism.
- Section 3.4: Design criteria definition and utilization of a custom Python tool to analyse the trajectory of the scaled standard box transport mechanism in relation to them.
- Section 3.5: Proposal of velocity speed profiles for the system motorization.
- Section 3.6: Formulation of an optimization problem for finding an improved box transport and increasing the transposability of this project.
- Section 3.7: Proposal of Grade A-compliant mechanical designs for the solution.
- Section 3.8: Proposal of sealing solutions for the environmental interfaces.

3.1 Choice of an 8-linkage box transfer mechanism

Conveying vials using a box transport mechanism, such as the one illustrated in Figure 62, offers several advantages over other conveyor systems. Firstly, compared to a belt conveyor, a box transport mechanism allows for intermittent motion. This simplifies the filling process as it only requires vertical motion of the needles. However, this design choice limits the achievable output to lower levels. [47]

When compared to the S-C system that utilizes servomotors and cylinders, the box transport mechanism provides several advantages, which are listed below: [47]

- No requirement for lubrication
- Simple construction
- Low-speed motor is sufficient
- Only one actuator is needed
- Easy maintenance

However, these advantages come at the expense of system compactness and the perfect engagement/disengagement that servomotors offer.

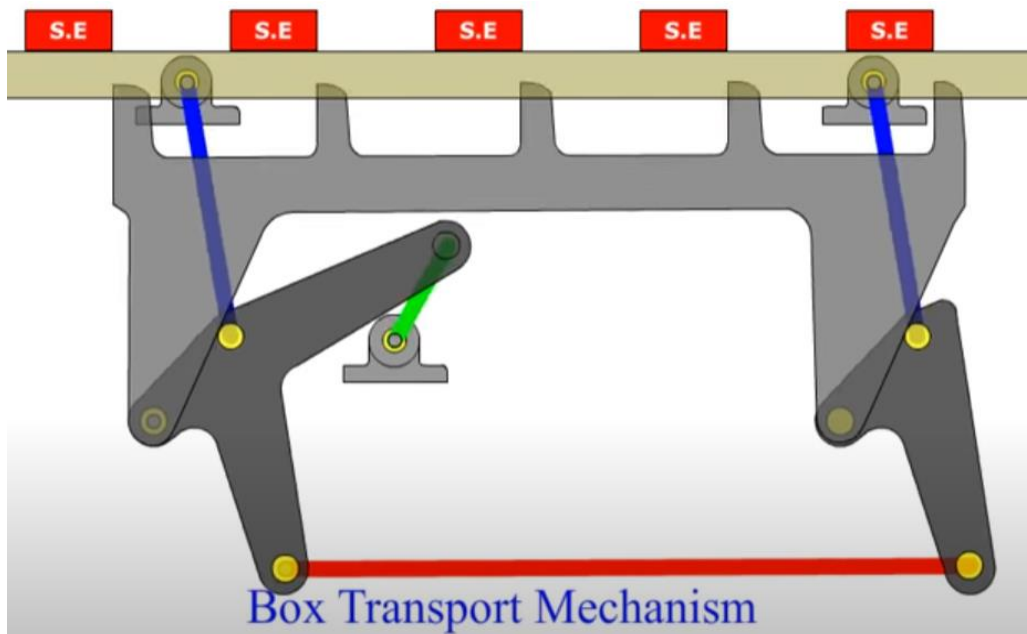


FIGURE 62 – EXAMPLE OF AN 8-LINKAGE TRANSFER BOX [49]

3.2 Filling circuit and stroke definition

Cilyx, having previously resolved filling challenges, has imposed a constraint on this project to utilize aseptic filling components from their partner Flexicon. These components will be selected to meet the solution requirements. This section takes precedence in the global solution development as the choices made here will impose physical constraints on the walking beam mechanism, specifically its stroke and spacing between vials.

Problem addressed in this section:

By the conclusion of this section, the appropriate needle diameters, quantities, and placements will be determined to achieve a minimum flow rate of 300 mL per minute for both 2R, 15R, and 30R vial filling configurations, while ensuring a precision of 0.5% in the filled volume.

Outcome:

Utilizing a custom Python code developed for this purpose and available in Appendix 7.14.1, a filling circuit that does not require format-specific parts will be identified. Additionally, the proposed solution guarantees a minimum tooth width of 5 mm between vials for manufacturing considerations.

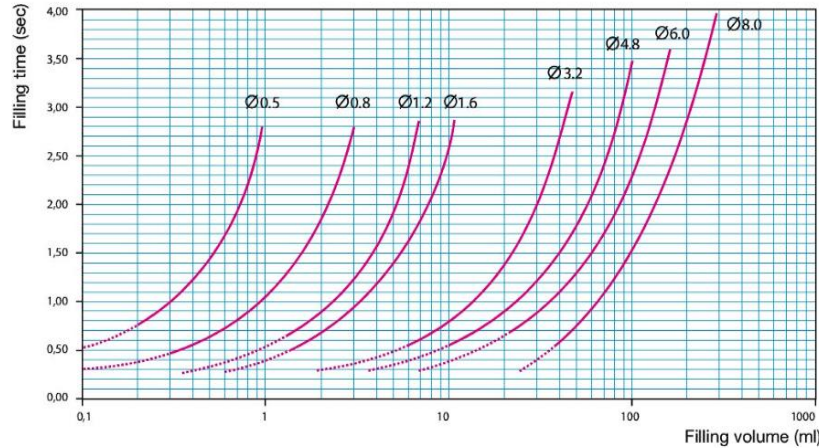
PF7 peristaltic pump



FIGURE 63 - FLEXICON PF7 PERISTALTIC PUMP [95]

The Flexicon PF7 peristaltic pump, shown in Figure 63, is recommended by Flexicon for aseptic filling applications. It offers high accuracy, achieving a filling volume accuracy of 0.5% or better. Each pump is capable of handling a maximum of 2 needles. However, these pumps are costly, and the objective is to minimize their number while meeting the previously mentioned requirements. The PF7 datasheet is available in Appendix **Error!**

To maximize equipment utilization, an ideal solution would involve using the same number of pumps for filling 2R, 15R, and 30R vials. The feasibility of this solution will be further analyzed in Figure 71, as it may affect the physical length of the system due to the stroke requirements. [87]



The selection of needles and tube diameters to be associated with these pumps is made using the PF7 pump capacity diagram shown in Figure 64 as well as an Excel table developed and shown in Figure 65. If the end customer wishes to fill a more viscous liquid, a discussion with Flexicon should be held to establish a new capacity chart. [96]

The Excel table in Figure 65 below is explained in more detail on the next page.

Case	Filling volume [mL]	Vial bottom diameter + 1 [mm]	Cadence min [VPM]	Tubes diameter	# Needles	Cadence max [VPM]	# Pumps	Filling time + 0,5 [s]	Time for stroke [s] (fixed)	Stroke length [mm]
1	2	17	150	1,6	6	150,0	3	1,2	1,2	6 (x1+17)
2	15	25	20	1,6	6	64,3	3	4,4	1,2	6 (x2+25)
3	30	31	10	1,6	6	37,9	3	2,5	1,2	6 (x3+31)
Case 1: 2R										
	Volume [mL]	Vial bottom diameter + 1 [mm]	Cadence min [VPM]	Tubes diameter	# Needles	Cadence max [VPM]	# Pumps	Filling time + 0,5 [s]	Time for stroke [s] (fixed)	Stroke length [mm]
	2	17	150	0,5	18	147,9	9	6,1	1,2	18 (x1+17)
	2	17	150	0,8	9	152,1	5	2,35	1,2	9 (x1+17)
	2	17	150	1,2	7	161,5	4	1,4	1,2	7 (x1+17)
	2	17	150	1,6	6	150,0	3	1,2	1,2	6 (x1+17)
	Cannot go for higher diameter for such small volumes !					Minimum cycle time = 2,4 s				
Case 2: 15R										
	Volume [mL]	Vial bottom diameter + 1 [mm]	Cadence min [VPM]	Tubes diameter	# Needles	Cadence max [VPM]	# Pumps	Filling time + 0,5 [s]	Time for stroke [s] (fixed)	Stroke length [mm]
	15	25	20	0,8	6	22,9	3	14,5	1,2	6 (x2+25)
	15	25	20	1,2	6	48,6	3	6,2	1,2	6 (x2+25)
	15	25	20	1,6	6	64,3	3	4,4	1,2	6 (x2+25)
	15	25	20	3,2	6	133,3	3	1,5	1,2	6 (x2+25)
	15	25	20	4,8	6	144,0	3	1,3	1,2	6 (x2+25)
	15	25	20	6	6	171,4	3	0,9	1,2	6 (x2+25)
Case 3: 30R										
	Volume [mL]	Vial bottom diameter + 1 [mm]	Cadence min [VPM]	Tubes diameter	# Needles	Cadence max [VPM]	# Pumps	Filling time + 0,5 [s]	Time for stroke [s] (fixed)	Stroke length [mm]
	30	31	10	1,2	6	27,5	3	11,9	1,2	6 (x3+31)
	30	31	10	1,6	6	37,9	3	8,3	1,2	6 (x3+31)
	30	31	10	3,2	6	97,3	3	2,5	1,2	6 (x3+31)
	30	31	10	4,8	6	124,1	3	1,7	1,2	6 (x3+31)
	30	31	10	6	6	138,5	3	1,4	1,2	6 (x3+31)

FIGURE 65 - EXCEL TABLE FOR FINDING NEEDLES PARAMETERS

3.2.1 Reaching 300 mL/min for 2R filling configuration

The Excel table in Figure 65 utilizes the capacity diagram of the PF7 pump shown in Figure 64 to determine configurations that meet the specific requirements of Cilyx. Several parameters are considered in this analysis, and it is important to provide further details about them:

1. **Filling time:** This is the duration required for a needle to fill a vial. The filling time in the table has been incremented by 0.5 seconds from the capacity chart to account for the time it takes for the needle to descend into the vial before filling and ascend back out once the filling is complete. It should be noted that the needles retract as the bottles are being filled. During the filling time, the mobile beam mechanism moves backward to prepare for filling the next set of vials.
2. **Time for stroke:** This refers to the time taken by the mechanism to transport vials from one position to the next. The duration of the time stroke impacts the required filling time to achieve the desired cadence. If the filling time is too short, the moving beam may not have sufficient time to move backward. Therefore, a balance needs to be struck. In this case, a value of 1.2 seconds for the time stroke corresponds to a filling time of 1.2 seconds for the 2R vials. These values are obtained through an iterative sizing process that considers the velocity profiles discussed in Section 3.5 to ensure the feasibility of an appropriate motorization system.

The optimal 2R filling configuration was determined by selecting needles with the largest diameter of 1.6 mm, which ensures a 0.5% accuracy on the filled volume. The choice of these needles was based on the consideration that needles have a geometric footprint of 13 mm (see Figure 66) independent of their nozzle diameter. By using the largest needles, the number of needles required was minimized without any compromise on the geometric footprint, resulting in a smaller stroke and overall system dimensions (as shown in Figure 67). Furthermore, this configuration requires the fewest number of pumps, making it the most cost-effective option for 2R filling using Flexicon components. To achieve the desired flow rate of 300 mL/min (equivalent to a cadence of 150 VPM), **the number of 1.6 mm needles was increased to six**. Filling nozzle datasheet and fluid path guidelines are available in Appendix 7.3.

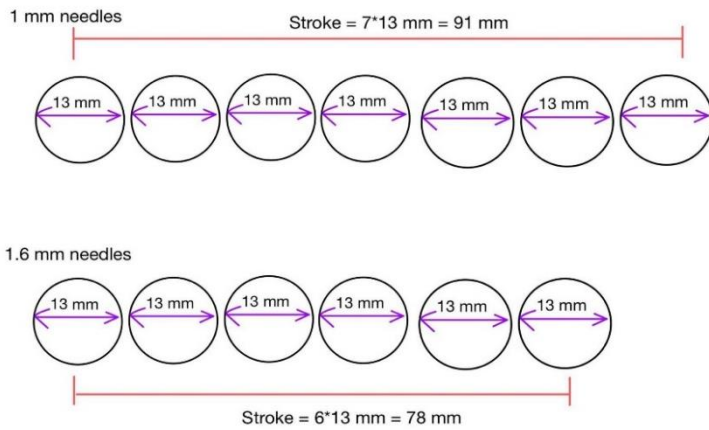


FIGURE 67 – ILLUSTRATION OF THE NEEDLE'S GEOMETRIC FOOTPRINT

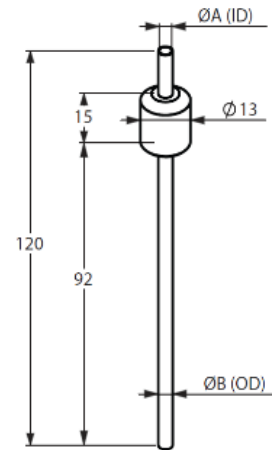


FIGURE 66 - FLEXICON FILLIG NOZZLE GEOMETRIC FOOTPRINT [97]

3.2.2 Extension to 15R and 30R filling configuration

For the filling of the 15R and 30R vials, the same 1.6 mm needle diameters as in the 2R configuration are used. Although this may result in longer filling times and a reduction in achievable cadence for these vials, it ensures consistency and eliminates the possibility of operator errors when switching between different configurations. If smaller diameter needles had been chosen for the 2R vials, the increase in wasted filling time would have been even greater.

Now, the question arises of how many needles should be used for these configurations. To address this, several constraints need to be considered. As depicted in Figure 68, each configuration must have the same stroke, as the box transport mechanism is designed to remain unchanged between configurations.

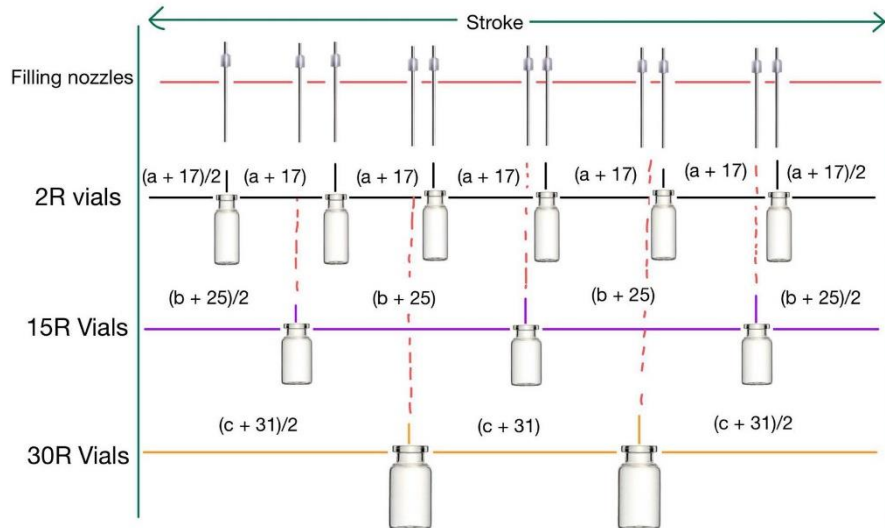


FIGURE 68 - VISUALIZATION OF THE PROBLEM ON NEEDLES LOCATIONS. $(A + 17)$; $(B + 25)$ AND $(C + 31)$ ARE RESPECTIVELY ENTRAX OF 2R; 15R AND 30R VIALS. A; B AND C REPRESENT THE TEETH LENGTH BETWEEN VIALS

This problem admits the following mathematical formulation:

$$\text{Stroke} = \text{number}_{2R} * (a + 17) \quad (1)$$

$$\text{Stroke} = \text{number}_{15R} * (b + 25) \quad (2)$$

$$\text{Stroke} = \text{number}_{30R} * (c + 31) \quad (3)$$

With $\text{number}_{2R-15R-30R}$ the number of vials filled per cycle for each configuration. The previous equations lead to

$$b = \frac{\text{number}_{2R}}{\text{number}_{15R}} * (a + 17) - 25 \quad (4)$$

$$c = \frac{\text{number}_{2R}}{\text{number}_{30R}} * (a + 17) - 31 \quad (5)$$

This problem can be easily solved using a Python code found in Appendix Filling configurations. This code assumes that $\text{number}_{2R} = 6$ and uses a triple loop shown in Figure 69 to reduce the number of unknowns to the number of equations. The python code then computes the

```
for number_15R in range(1, 7):
    for number_30R in range(1, 7):
        for Teeth_width_2R in range(1, 70):
```

FIGURE 69- TRIPLE LOOP OF THE PYTHON CODE FINDING NEEDLES LOCATIONS

solutions of Equations (4) and (5) to calculate the location of the needles for each configuration.

Consideration on multiformat:

In the context of minimizing the dimensions of the system, the design consideration for the support of the needles can be approached as a format part. This approach allows for more flexibility in adjusting the teeth width for each configuration. By treating the constraint on the needle spacing independently from one configuration to another while maintaining the constraint of identical stroke, the system dimensions can be further optimized.

However, in this project, the decision was made to have the support as a **non-format piece**. This compromises on the overall system dimensions but reduces the potential for operator errors when modifying the system configuration.

Back to the python code:

The feasibility of the solution is determined by satisfying the conditions outlined in Figure 70, which are explained hereafter:

1. **Spacing between needle locations:** In the chosen non-format piece configuration, it is necessary to maintain a minimum spacing of 14 mm between the needle locations within each configuration as well as between them. However, it should be noted that a spacing of 0 mm, meaning that the needles are located at the same position, is considered acceptable and should not be disregarded.
2. **Teeth width:** For each configuration, the width of the teeth must be greater than 4 mm to ensure reliable machining of POM. Additionally, the teeth width should be smaller than 1.5 times the vial dimension to avoid wasting excessive space.

```
not_ok_1 = dist_needles_2R_15R[(dist_needles_2R_15R <= needles_dim) & (dist_needles_2R_15R != 0)]
not_ok_2 = dist_needles_2R_30R[(dist_needles_2R_30R <= needles_dim) & (dist_needles_2R_30R != 0)]
not_ok_3 = dist_needles_15R_30R[(dist_needles_15R_30R <= needles_dim) & (dist_needles_15R_30R != 0)]

if not_ok_1.size + not_ok_2.size + not_ok_3.size == 0\
    and min_tooth < Teeth_width_30R < 1.5 * vial_dim_30R\
    and min_tooth < Teeth_width_15R < 1.5 * vial_dim_15R\
    and min_tooth < Teeth_width_2R < 1.5 * vial_dim_2R:
```

FIGURE 70 – CONDITIONS FOR A SOLUTION TO BE ADMISSIBLE. MIN_TOOTH = 5 MM, NEEDLES_DIM = 14 MM.

Length of the system:

Once a feasible solution has been found, the length of the mechanism can be computed by **multiplying the stroke by the desired number of stations**. In the present project, it is assumed that 5 stations are required.

- 2 stations are dedicated to the loading and unloading of vials onto the mechanism.
- 1 station is designated for the filling process, where the vials are filled with the desired substance.
- 1 station is allocated for the stoppering operation
- Finally, there is one additional station located between the filling and stoppering stations. This intermediate station is used for performing quality control and inspection on the filled vials.

Results:

Figure 71 presents three notable solutions identified by the algorithm that meet the previously imposed constraints. These solutions highlight a clear trade-off between the achievable cadence for configurations 2 and 3, and the dimensions of the teeth on the comb responsible for conveying the 2R vials, as well as the stroke and overall size of the system.

```

stroke = 174.00 | Number of vials | Teeth width [mm] |
-----|-----|-----|
17 mm vials | 6 | 12 |
-----|-----|-----|
25 mm vials | 3 | 33 |
-----|-----|-----|
31 mm vials | 3 | 27 |
-----|-----|-----|

Needles locations:
needles 17 = [ 14.5 43.5 72.5 101.5 130.5 159.5]
needles 25 = [ 29. 87. 145.]
needles 31 = [ 29. 87. 145.]

Total length of box transfer mechanism : 87.00 [cm]
-----|-----|-----|

stroke = 180.00 | Number of vials | Teeth width [mm] |
-----|-----|-----|
17 mm vials | 6 | 13 |
-----|-----|-----|
25 mm vials | 6 | 5 |
-----|-----|-----|
31 mm vials | 3 | 29 |
-----|-----|-----|

Needles locations:
needles 17 = [ 15. 45. 75. 105. 135. 165.]
needles 25 = [ 15. 45. 75. 105. 135. 165.]
needles 31 = [ 30. 90. 150.]

Total length of box transfer mechanism : 90.00 [cm]
-----|-----|-----|

stroke = 216.00 | Number of vials | Teeth width [mm] |
-----|-----|-----|
17 mm vials | 6 | 19 |
-----|-----|-----|
25 mm vials | 6 | 11 |
-----|-----|-----|
31 mm vials | 6 | 5 |
-----|-----|-----|

Needles locations:
needles 17 = [ 18. 54. 90. 126. 162. 198.]
needles 25 = [ 18. 54. 90. 126. 162. 198.]
needles 31 = [ 18. 54. 90. 126. 162. 198.]

Total length of box transfer mechanism : 108.00 [cm]
-----|-----|-----|

```

FIGURE 71 - INTERESTING SOLUTIONS FOR THE NUMBER OF VIALS TO FILL PER CYCLE

Impacts on the length of the system:

While doubling the output of the 15R vials costs the system 3 cm in additional length (a 3.45% increase), simultaneously doubling the output of the 15R and 30R vials from 3 to 6 vials per cycle lengthens the system by 21 cm. This is an increase of 24.14%.

Impacts on the parameters of the box transport mechanism:

The same analysis can be performed, this time taking into account the stroke. While the second solution proposes to double the rate of the 15R vials at the cost of 6 centimeters of additional stroke (i.e. an increase of 3.4%), the third solution doubles the rate of the 15R and 30R vials at the cost of a 24.13% increase in stroke length.

Final choice of the solution:

The final solution chosen for the rest of this project is the last one, **with a stroke of 216 mm**. Not only does it achieve the **highest throughput** for both vials, but it also has the **smallest average tooth width**. An additional criterion in favor of this solution is that each configuration uses the same number of needles in the same locations. No configuration leaves empty needle holders. At the end of this chapter, a solution has been found that leaves no possibility of error for the operator when changing the system configuration.

3.3 Design parameters

The conclusion of this section includes the introduction of a Python code, provided in Appendix 7.14.2, which establishes a connection between design parameters and the corresponding trajectories and velocity profiles. This Python tool is utilized to showcase a "standard scaled box transport mechanism" that fulfills the specified requirements. However, further analysis reveals potential limitations in terms of ergonomic efficiency, as discussed in Section 3.4. In response to this, Section 3.6 approaches the problem as an optimization challenge, seeking to identify enhanced design parameters that can improve overall performance.

3.3.1 Kinematic Analysis of an 8-Linkage Box Transport Mechanism

Problem Statement:

The objective is to find design parameters of an 8-linkage box transport mechanism with a stroke of 216 mm. To achieve this, a comprehensive kinematics analysis of the system is conducted based on the scientific paper "Box Transport Mechanism" [48]. While the detailed calculations can be found in Appendix 7.11, this section focuses on highlighting key features of the analysis.

Denomination and axis imposition

The first step involves imposing denominations and axis, as shown in Figure 72. This step establishes the reference frame and coordinate system for further analysis.

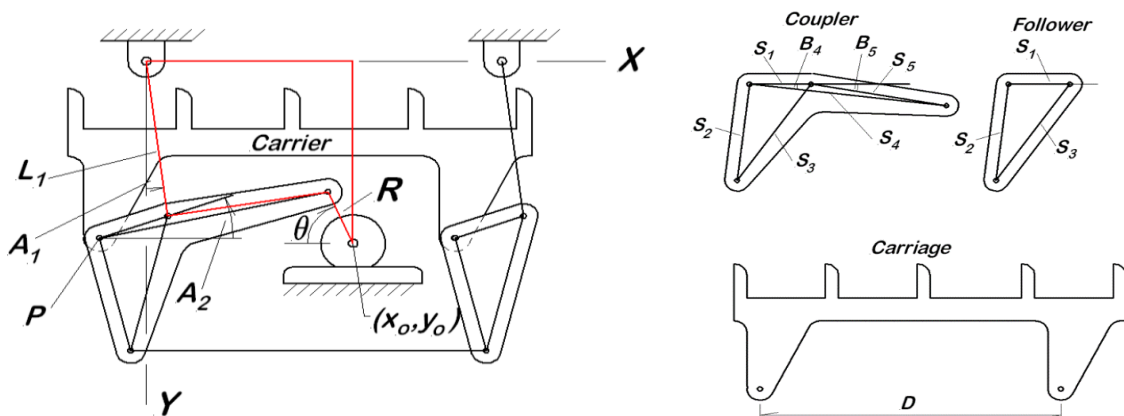


FIGURE 72 - 8-LINKAGE TRANSFER BOX: GEOMETRY AND DENOMINATIONS. [49]

Ensuring Carrier Blade Stability:

To prevent tilting of the carrier blades during the cycle, it is necessary for the follower triangle to be isometric to the coupler triangle S1-S2-S3. Once this condition is met, the size of the triangle has no impact on the kinematics of the system. Therefore, the mathematical analysis can be limited to the triangle S1-S4-S5. By studying the displacement of point P, which corresponds to the displacement of each point on the carriage, Equations (6) and (7) are derived:

$$x_p = L_1 \sin(A_1) - S_1 \cos(A_2) \quad (6)$$

$$y_p = L_1 \cos(A_1) + S_1 \sin(A_2) \quad (7)$$

Velocity analysis

The velocity of point P is expressed using Equations (8) and (9):

$$\dot{x}_p = \dot{\theta} \cdot K_{px}(\theta) \quad (8)$$

$$\dot{y}_p = \dot{\theta} \cdot K_{py}(\theta) \quad (9)$$

These equations demonstrate the strong dependence of the velocities on the crank angle through the velocity speed coefficients $K_{px}(\theta)$ and $K_{py}(\theta)$. It is important to note that even at a constant crank speed, the speed of the comb varies, which needs to be considered in Section **Error! Reference source not found.** when discussing the motorization velocity profile.

The velocity speed coefficients correspond to the derivatives of the angle rates with respect to the crank speed, as shown in Equations (10) and (11):

$$K_{px}(\theta) = \frac{dx_p}{d\theta} = \frac{d}{d\theta}(L_1 \sin(A_1) - S_1 \cos(A_2)) \quad (10)$$

$$K_{py}(\theta) = \frac{dy_p}{d\theta} = \frac{d}{d\theta}(L_1 \cos(A_1) + S_1 \sin(A_2)) \quad (11)$$

Acceleration analysis

To perform an acceleration analysis, Equations (8) and (9) are differentiated. The acceleration of point P is expressed using Equations (12) and (13):

$$\ddot{x}_p = \frac{d}{dt}(\dot{\theta} K_{px}) = \ddot{\theta} K_{px} + \dot{\theta}^2 L_{px} \quad (12)$$

$$\ddot{y}_p = \frac{d}{dt}(\dot{\theta} K_{py}) = \ddot{\theta} K_{py} + \dot{\theta}^2 L_{py} \quad (13)$$

Velocity coefficient derivatives, L_{px} and L_{py} , are introduced and calculated using Equations 14 and 15:

$$L_{px}(\theta) = \frac{dK_{px}(\theta)}{d\theta} = \frac{d^2 x_p}{d\theta^2} = \frac{d^2}{d\theta^2}(L_1 \sin(A_1) - S_1 \cos(A_2)) \quad (14)$$

$$L_{py}(\theta) = \frac{dK_{py}(\theta)}{d\theta} = \frac{d^2 y_p}{d\theta^2} = \frac{d^2}{d\theta^2} (L_1 \cos(A_1) + S_1 \sin(A_2)) \quad (15)$$

Note: velocity coefficients derivatives should not be considered as **acceleration coefficients**. Indeed, in the previous equations (12) and (13), they never multiply the acceleration.

Design parameters:

From the previous analysis of the 8-linkage box transport mechanism, the parameters that impact the trajectories, velocities, and accelerations of the system are as follows:

Design parameters	S_1	S_4	S_5	x_0	y_0	L_1	R
-------------------	-------	-------	-------	-------	-------	-------	-----

3.3.2 Python implementation

The complete kinematics equations were implemented in Python and solved using the fsolve function. The code, along with the necessary functions and parameters, can be found in Appendix Kinematics of the box transport mechanism. By running the code, the results can be obtained and displayed.

To ensure the accuracy of the implementation, a comparison was made between the results obtained from the Python code and the results displayed in Figure 73 and Figure 74 of the scientific paper "Box Transport Mechanism" [48]. The comparison revealed that the results from the Python code were consistent and matched the results presented in the paper.

Stroke = 185.42 mm	S_1	S_4	S_5	x_0	y_0	L_1	R
Values [mm]	108	319	217	198	240	234.75	91.95

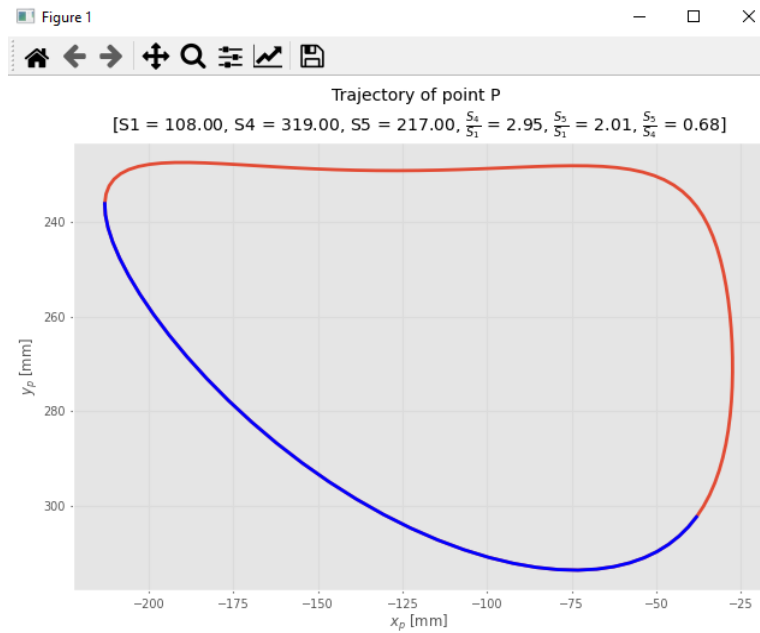


FIGURE 73 - REFERENCE TRAJECTORY

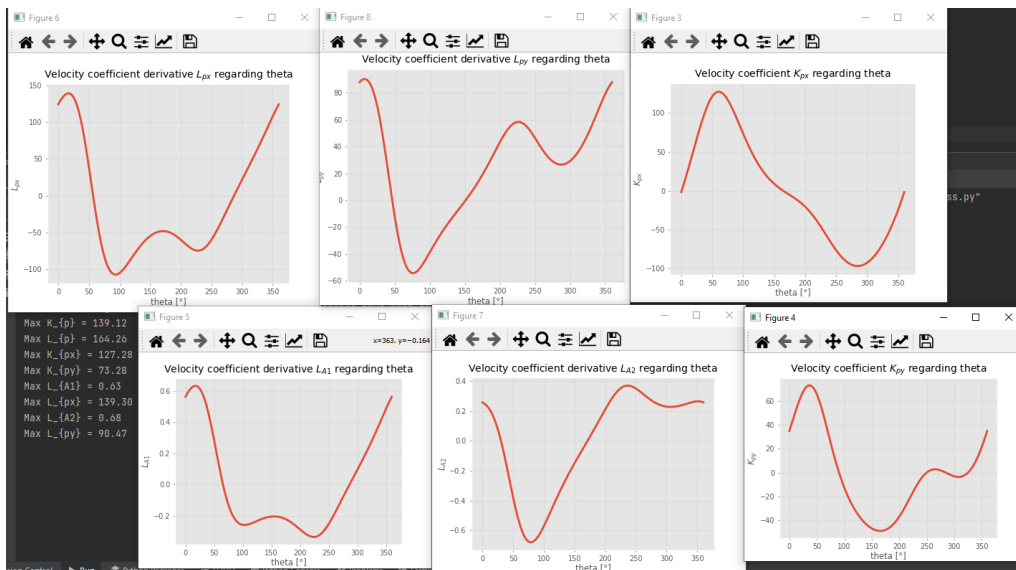


FIGURE 74 - REFERENCE OUTPUT

3.3.3 Standard scaled box transport mechanism:

To achieve a different stroke solution, the shape of the box transport mechanism can be preserved by scaling all its dimensions uniformly. For instance, to reach the target stroke of 216 mm, all parameters can be multiplied by a constant coefficient, such as $a = 1.322$.

Stroke = 216 mm	S_1	S_4	S_5	x_0	y_0	L_1	R
-----------------	-------	-------	-------	-------	-------	-------	-----

Values [mm]	143	422	287	262	317	310	122
-------------	-----	-----	-----	-----	-----	-----	-----

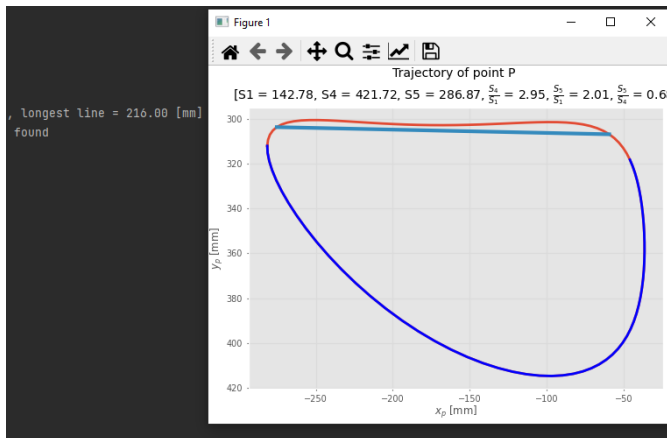


FIGURE 75 - STROKE OF 216 MM REACHED BY MULTIPLYING REFERENCE PARAMETERS BY A = 1.322

Figure 75 demonstrates that the use of a box transport mechanism does not result in perfectly rectangular trajectories. To address this, design criteria will be established in the next subsection and will show that this trajectory actually answers the requirements of this project.

Analysis of this solution:

Based on Figure 72, the resulting box transport mechanism, depicted in Figure 75, appears to meet the stroke requirements while maintaining acceptable design parameter values. This conclusion is drawn based on the following considerations:

- Horizontal space required: The approximate horizontal space required can be calculated as $S4 + S1 = 422 + 143 = 565$ mm, which is less than the length of the required box transport comb (1080 mm).
- Vertical space required: The vertical distance required can be approximated as $L1 = 310$ mm, which is considered acceptable in terms of geometric footprint.

Problem induced by scaling: lack of flexibility

Although the scaling method effectively achieves the desired stroke, it has the disadvantage of increasing the values of all design parameters as well as the recoil. This unnecessary increase in recoil can have negative implications for the overall geometric footprint of the system.

To address this issue, Section 3.6 introduces a more flexible approach to finding an optimal box transport mechanism based on the design criteria developed in the subsequent section. This approach involves exploring different design parameters and trajectory shapes that still meet the required specifications while offering improved ergonomics. By adopting this approach, the project aims to provide Cilyx with a solution that is both ergonomics and adaptable to other box transport mechanism requirements they may encounter.

3.4 Design criteria and Python trajectory analysis tool

Problem: non-rectangular trajectory

To ensure the proper functioning of vial transport, trajectories must adhere to several design criteria, particularly compensating for the non-rectangular trajectory. These criteria will be presented and illustrated in this section using a custom Python trajectory analysis tool, which is available in Appendix 7.14.3. The tool enables a detailed analysis of the trajectories to evaluate their compliance with the established criteria.

Outcome:

The conclusion of this section will entail the establishment of design criteria for trajectories, taking into consideration the stroke, as well as the engagement and disengagement of vials. A customized Python tool, specifically developed to analyse trajectories in relation to these design criteria, will be showcased on the scaled box-transport mechanism found in the previous section.

1. Stroke: 216 mm

The stroke is defined as the length of the linear motion performed by the box transport mechanism. In a trajectory, the stroke can be defined at any point along the path, which may include multiple conforming linear segments. Figure 76 illustrates this scenario. To address this issue and ensure proper engagement and disengagement of the mechanism, additional parameters will be introduced. These parameters will help

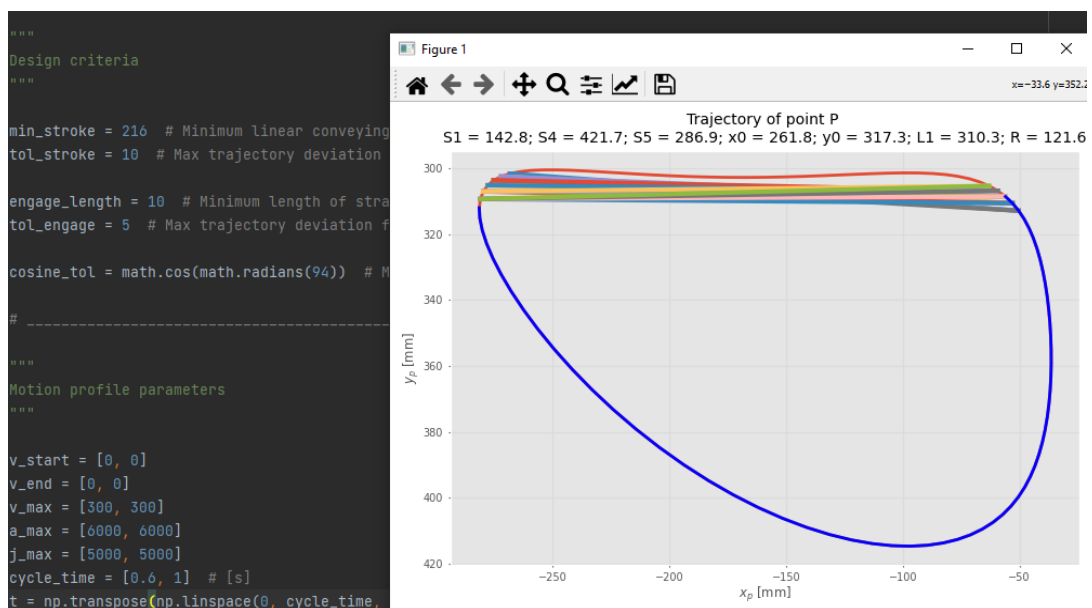
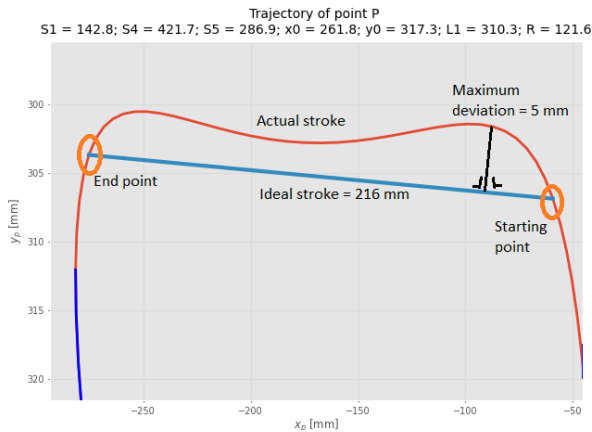


FIGURE 76 - EXAMPLE OF SEVERAL STROKE GREATER OR EQUAL TO 216 MM, WITH A MAXIMUM DEVIATION OF 10 MM WITH RESPECT TO THE ACTUAL TRAJECTORY.

select a unique stroke that is close to 216 mm while maintaining the required functionality.

2. Maximum stroke deviation: 5 mm



The non-linear travel exhibited by the box transport mechanisms, as shown in Figure 77, raises the question of its impact on the functionality of the system.

FIGURE 77 - DEMONSTRATION OF A MAXIMUM DEVIATION OF 5 MM BETWEEN A STROKE OF 216 MM AND THE ACTUAL TRAJECTORY

From the perspective of operations requiring precise positioning of the vials, the non-linear travel of the box transport mechanisms in Figure 77 is not a problem. The filling and stoppering operations are performed at specific starting and ending points along the defined stroke. As long as the vials successfully reach these points, the operations will be successful regardless of the intermediate trajectory.

To ensure that the vials indeed reach these points, two solutions are available:

Solution 1: Gripping System

One approach is to incorporate a gripping system, such as suction cups, into the comb. This ensures that the vials follow the exact path of the box transport mechanism. However, this solution is complex to implement and requires the sliding surface to have a width equal to the vials' width plus twice the maximum stroke deviation. As the deviations between the box transport mechanism and the ideal stroke decrease, the width of the track can be narrower, resulting in a smaller geometric footprint.

Solution 2: Guide Rails

Another solution, chosen for its simplicity in this project, is to add guide rails that keep the vials on a straight line regardless of the comb's relative motion. With this approach, the track width can be sized to match the vials' diameter.

Due to the contact and friction between the comb and the pushed bottles, the comb's movement relative to the vials can generate torque on the vials. Minimizing the

deviation between the box transport mechanism and the ideal stroke reduces this phenomenon.

In accordance with the comb gaps design developed in the project's Section 3.7, a stroke deviation of 5 mm was arbitrarily selected. This value was chosen to minimize the height of the tooth design while still being comparable to the 8 mm radius of the 2R flasks and the 31 mm width required for the track⁷ of 30R vials. The objective was to find the smallest possible stroke deviation that meets the requirements of the system and ensure the existence of the solution.

Note: To achieve maximum accuracy in bottle positioning during filling and stoppering, a second triangular-toothed comb that locks the bottles against the guide rail immediately after the comb retracts and throughout the entire vial stop must be installed. This ensures the final location of the vials.

3. Engagement and disengagement length: 15 mm

Figure 78 illustrates that to avoid unwanted collision between the teeth of the comb and the vials, a minimum engagement length of $\sqrt{R^2 + C^2}$ should be chosen, where R is the radius of the vial and C is a **design parameter** representing the additional gap between the comb's teeth width and the vial diameter (2R). In the case of the 30R vials, with a radius R of 15 mm and a **design parameter of C = 1 mm**, the minimum engagement length is approximately 15 mm and can be kept for engaging the smaller diameter vials.

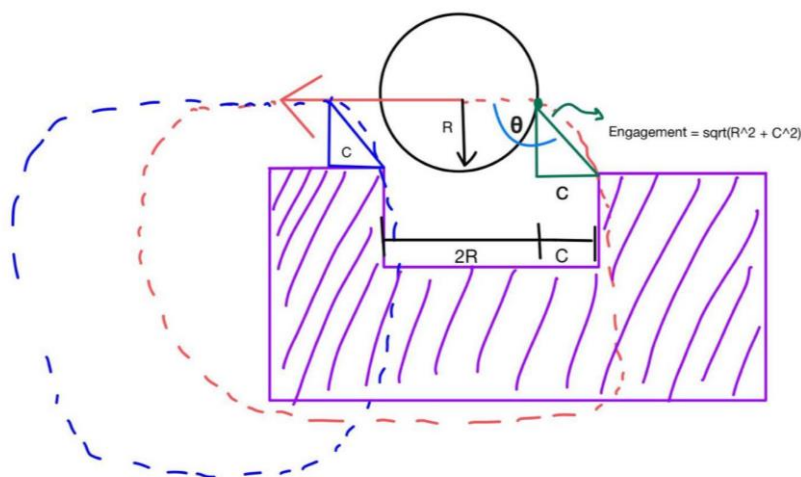


FIGURE 78 - SCHEMATIC REPRESENTATION OF THE VIAL ENGAGEMENT PROBLEM

⁷ In Section Mechanical design for multiformat of the project, it will be explained in more detail that the track used is not a customized piece and remains the same for all vial filling configurations. Its width is set at 32 mm to accommodate the 30R vial diameter, which is equal to 31 mm. For other filling configurations, the width of the track is adjusted by using moving guide rails to accommodate different vial sizes.

4. Maximum engagement and disengagement deviations: 5 mm

The parameter for maximum engagement and disengagement deviations, set to 5 mm, is chosen arbitrarily to ensure that the engagement and disengagement lengths accurately reflect the real trajectory of the box transport mechanism.

5. Engagement and disengagement angle: $90^\circ < \theta < 94^\circ$.

The angle theta (θ) represents the angle between the stroke and the engagement/disengagement lines. It can be used to define the value of parameter C, or conversely, the value of C can be used to define the maximum value of theta ensuring proper engagement. The relationship between these two parameters is defined by Equation (16).

$$\theta = 90 + \text{Atan}\left(\frac{C}{R}\right) \quad (16)$$

For instance, when considering an additional 1 mm width for 30R vials ($C = 1$ mm), it results in an angle of 94° .

In the Python code shown in Figure 79, no specific constraints have been applied to the engagement angle. As a result, the code provides information on the angles of engagement and disengagement lengths that are greater than or equal to 15 mm, as long as they do not deviate by more than 5 mm from the actual trajectory.



FIGURE 79 - ILLUSTRATION OF THE ENGAGEMENT AND DISENGAGEMENT DESIGN CRITERIA

By calculating the cosine of the engagement line to be 0.77, it indicates that the actual trajectory engages the stroke at an angle of $\cos^{-1}(-0.77) = 140^\circ$. As illustrated in Figure 80, this would require a correction of the design parameter C to be 13.4 mm. However, considering the 30 mm diameter of 30R vials, this correction is deemed excessive. Hence, it emphasizes the need for further exploration to find a more optimal solution.

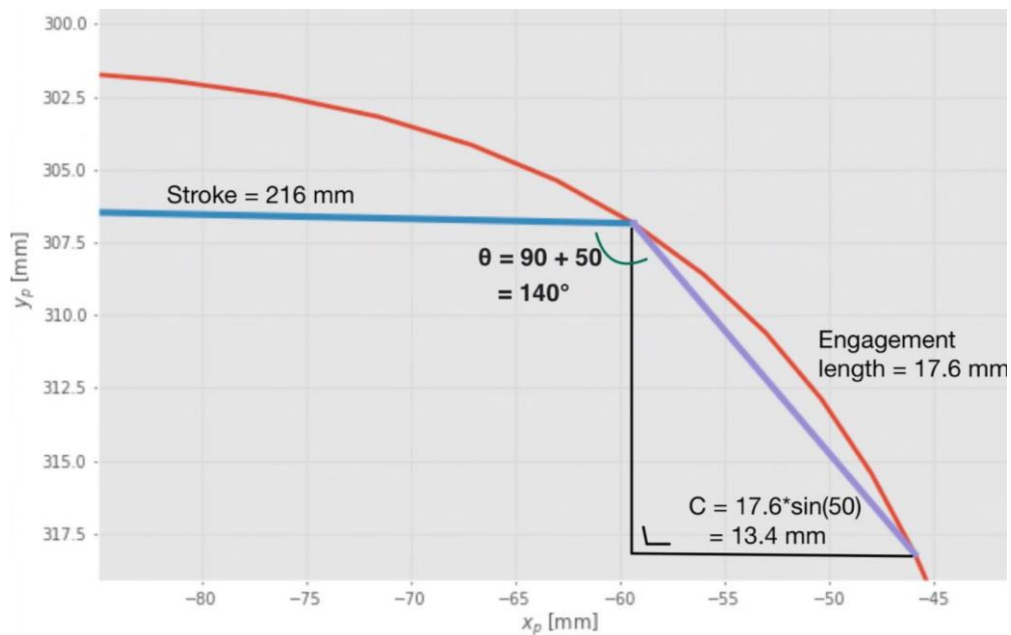


FIGURE 80 - TOOTH WIDTH REQUIRED FOR THE SCALED BOX TRANSPORT MECHANISM

By reversing the trajectory direction, it is possible to invert the engagement and disengagement points. This change results in a better engagement angle of $\cos^{-1}(-0.29) = 107^\circ$, which leads to a reduced required value of $C = 4.61$ mm.

3.5 Trajectory planning

Problem to solve in this section:

In the previous chapters, Python tools were introduced for generating the box transport mechanism and analyzing the shape of the resulting trajectory. However, there is still a need for a tool to assess the feasibility of the motorization system with respect to the motion constraints outlined in Section 3.2.1. Among those constraints, the most stringent are imposed by the 2R vials filling configuration. These constraints are as follows:

- **Filling time = 1.2 seconds:**
The time required to fill the 2R vials with the desired substance.
- **Retreat time = 1.2 seconds:**
The time required for the mechanism to retreat from the 2R vials after filling.

Not only do the speed and torque requirements for the motor depend on the set of design parameters, but they also heavily depend on the chosen trajectory planning strategy. To assess the feasibility of the motorization system and make informed decisions for optimizing the design, this section will implement several of those strategies and compare their requirements based on the standard scaled box transport mechanism previously found.

As a reminder, Equations (8) and (9) describe the relationship between the velocity and acceleration of the comb and the motorization system, specifically through the velocity coefficient and velocity derivative coefficient:

$$\dot{p}(\theta) = \dot{\theta} K_p(\theta) \quad (17)$$

$$\ddot{p}(\theta) = \ddot{\theta} K_p(\theta) + \dot{\theta}^2 L_p(\theta) \quad (18)$$

By reverting those equations, velocity required on the motor can be found according to Equations 19 and 20

$$\dot{\theta}(\theta) = \frac{\dot{p}}{K_p(\theta)} \quad (19)$$

$$\ddot{\theta}(\theta) = \frac{(\ddot{p} - \dot{\theta}^2 L_p(\theta))}{K_p(\theta)} \quad (20)$$

All along this section, the following set of parameters is considered:

Stroke = 216 mm	S_1	S_4	S_5	x_0	y_0	L_1	R
Values [mm]	143	422	287	262	317	310	122

3.5.1 Stroke: Velocity profiles strategies

The goal of this section is to explore the trade-off between trajectory efficiency and smoothness, specifically in the context of a straight-line stroke without the inclusion of via points. The section examines three commonly used strategies for achieving such motion. These strategies are first applied to the comb piece and then converted into motor velocity profiles⁸. Finally, a comparison between them will be performed. [49]

The following analysis was performed using the velocity design parameters shown in Figure 81. The complete custom-made Python codes for motion analysis of a box transport mechanism are available in the Appendix 7.14.4.

```

"""
Motion profile parameters
"""
v_start = [0, 0]
v_end = [0, 0]
v_max = [300, 300]
a_max = [6000, 6000]
j_max = [10000, 10000]
cycle_time = [1.2, 1.2] # [Time for stroke, Time for retreat]

```

FIGURE 81 - IMPLEMENTATIO OF THE VELOCITY DESIGN PARAMETERS IN THE PYTHON CODE

Smooth motion: quintic profile

Quintic motion profile refers to the use of a 5th order polynomial on which boundary conditions on velocity and accelerations can be imposed.

$$p(t) = At^5 + Bt^4 + Ct^3 + Dt^2 + Et + F \quad (21)$$

To ensure a slow and safe engagement and disengagement of vials, this project imposes constraints of zero speed and acceleration at the beginning and end of the stroke. These constraints are necessary to solve the quintic polynomial and determine the desired motion profile.

$$p(0) = p_{engage} \quad (22)$$

$$p(t_{filling}) = p_{disengage} \quad (23)$$

$$\dot{p}(0) = 0 \quad (24)$$

$$\dot{p}(t_{filling}) = 0 \quad (25)$$

$$\ddot{p}(0) = 0 \quad (26)$$

$$\ddot{p}(t_{filling}) = 0 \quad (27)$$

⁸ Velocity profiles have been implemented using the Peter Corke's Robotic Toolbox [106] and the PyScurvePython library. [107]

Once applied to the 216 mm stroke of the standard box transport mechanism, the resulting motion profiles are depicted in Figure 83 and Figure 84. As expected, the motion appears smooth both from the comb's perspective and the motor's perspective.

```

Quintic for stroke:

Max  $\theta$  speed = 5.92 [rad/s] = 56.52 RPM
Max  $\theta$  acc = 76.47 [rad/s2] = 43813.14 RPM2

Max comb speed = 337.44 [mm/s]
Max comb acc = 866.03 [mm/s2]
    
```

FIGURE 82 - MOTOR REQUIREMENT WHEN APPLYING A QUINTIC MOTION FOR THE STROKE. J_MAX = 5000; V_MAX = 300; A_MAX = 6000

As discussed in Section 3.1, Figure 82 highlights that a low-speed motor with a rotation rate of **56 RPM** is sufficient to achieve the desired motion. The vials will experience a maximum acceleration of **866 mm/s²**. While this appears to be suitable, empirical tests are required to validate these findings.

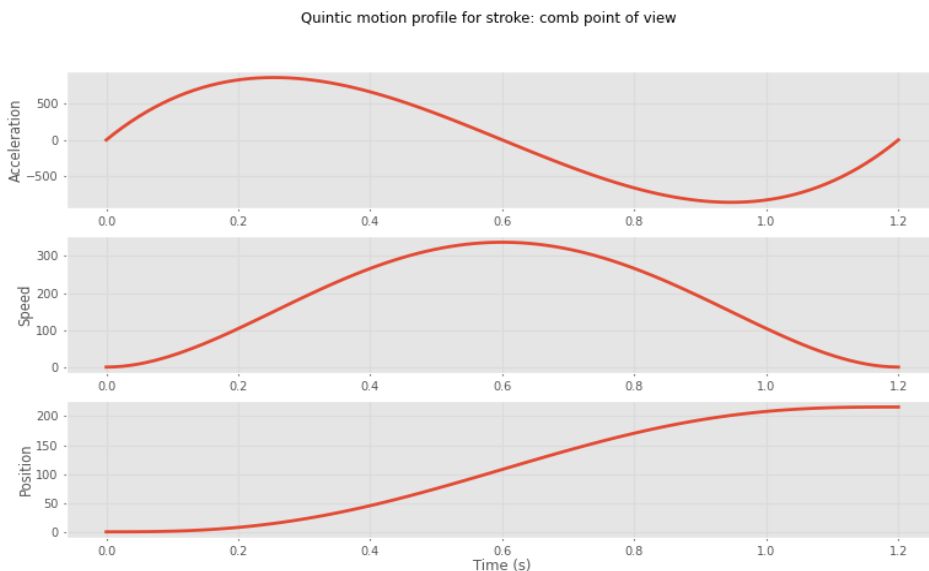


FIGURE 84 - QUINTIC PROFILE APPLIED TO STROKE: COMB VIEW.

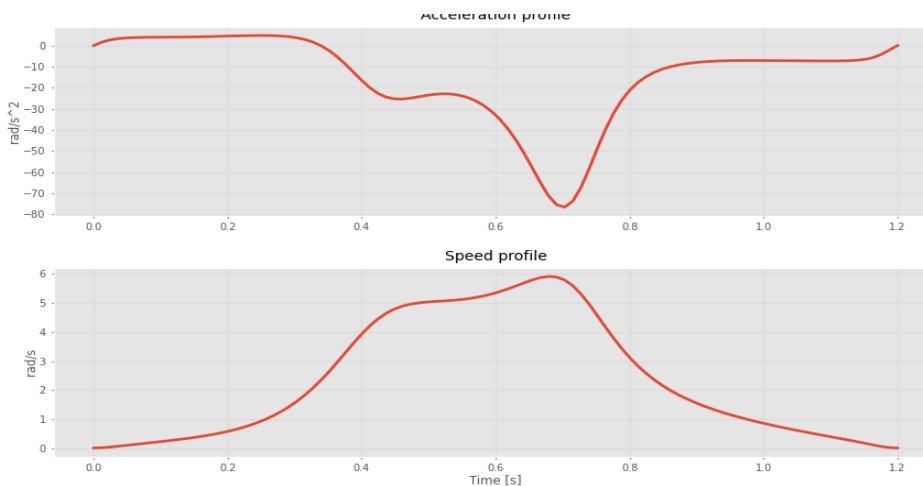


FIGURE 83 QUINTIC PROFILE APPLIED TO STROKE: MOTOR VIEW.

Minimizing the velocity: trapezoidal velocity profile

Trapezoidal velocity profiles aim to utilize the maximum allowed acceleration of the vials to determine the minimum required motor speed. While the exact maximum acceleration is unknown, as discussed earlier, it is straightforward to implement and allows for setting a maximum acceleration as long as it does not render the movement infeasible. [49]

When applied to the requirements of this project, with a maximum acceleration of 700 mm/s^2 , the results are depicted in Figure 85 and Figure 86.

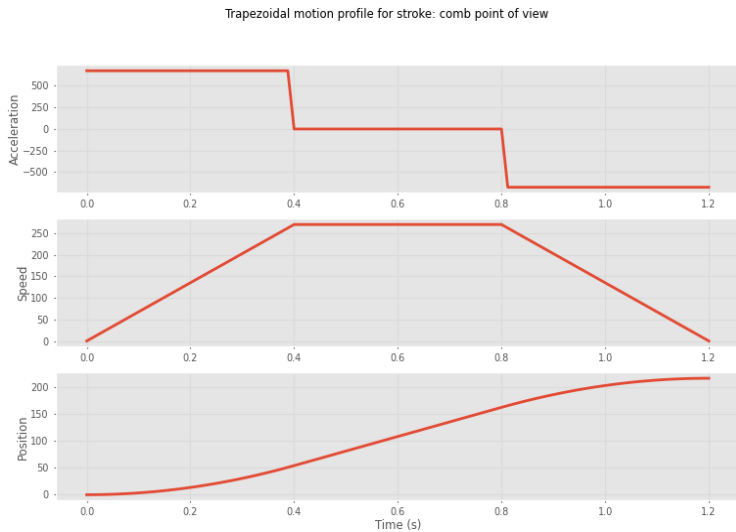


FIGURE 86 - QUINTIC PROFILE APPLIED TO STROKE: COMB VIEW. $J_{MAX} = 5000$; $V_{MAX} = 300$; $A_{MAX} = 6000$

```
Trapezoidal for stroke:
Max  $\theta$  speed = 4.94 [rad/s] = 47.20 RPM
Max  $\theta$  acc = 50.23 [rad/s^2] = 28781.50 RPM^2
Max comb speed = 270.00 [mm/s]
Max comb acc = 675.01 [mm/s^2]
```

FIGURE 85 - MOTOR REQUIREMENTS WHEN APPLYING A TRAPEZOIDAL MOTION TO THE STROKE. $V_{MAX} = 300$; $A_{MAX} = 700$

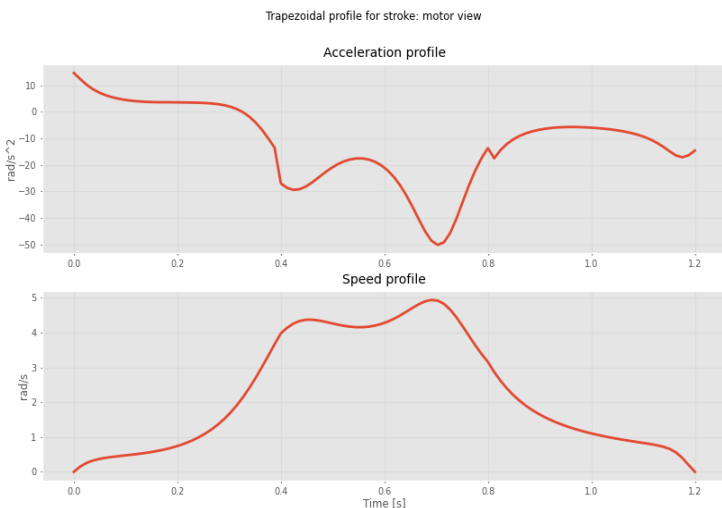


FIGURE 87 - QUINTIC PROFILE APPLIED TO STROKE: MOTOR VIEW. $J_{MAX} = 5000$; $V_{MAX} = 300$; $A_{MAX} = 6000$

As illustrated in Figure 85, it is observed that a **47 RPM** motor is sufficient to meet the requirements. The vials will experience a maximum acceleration of **675 mm/s^2** .

As depicted in Figure 85, it is evident that the trapezoidal motion profiles offer improved efficiency compared to the quintic motion profiles. However, this efficiency gain comes at the

cost of reduced smoothness. The trapezoidal profiles exhibit abrupt changes in acceleration, even at the start and end of the motion, which results in infinite jerk. Those abrupt changes in acceleration cause shocks and vibrations in the system during motion, increasing the wear of its components. Additionally, it can be problematic when handling filled vials, as they should be moved gently to avoid any potential damage or splash.

Bounding the jerk for filled containers: S-curve velocity profile

While the quintic profile achieves high smoothness, and the trapezoidal profile achieves high efficiency, a combination of both would be advantageous. This can be achieved using S-curve velocity profiles, which round the edges of the trapezoidal profile to comply with a constant defining the maximum jerk allowed. In the case of this project, the PyScurve Python library is used, which utilizes a three-order polynomial to achieve this objective. [50] [51] [52]

A comparison with the trapezoidal motion is presented in Figure 87, which illustrates the reduction in turbulence generated on the conveyed fluid when controlling the jerk.



FIGURE 88 - IMPACT OF TRAPEZOIDAL AND S-CURVE VELOCITY PROFILE ON A GLASS OF WINE AT THE END OF THE MOTION [111]

As it can be seen in Figure 88, implementing this strategy to limit the jerk results in an increased maximum acceleration reached during the motion. [51] [52]

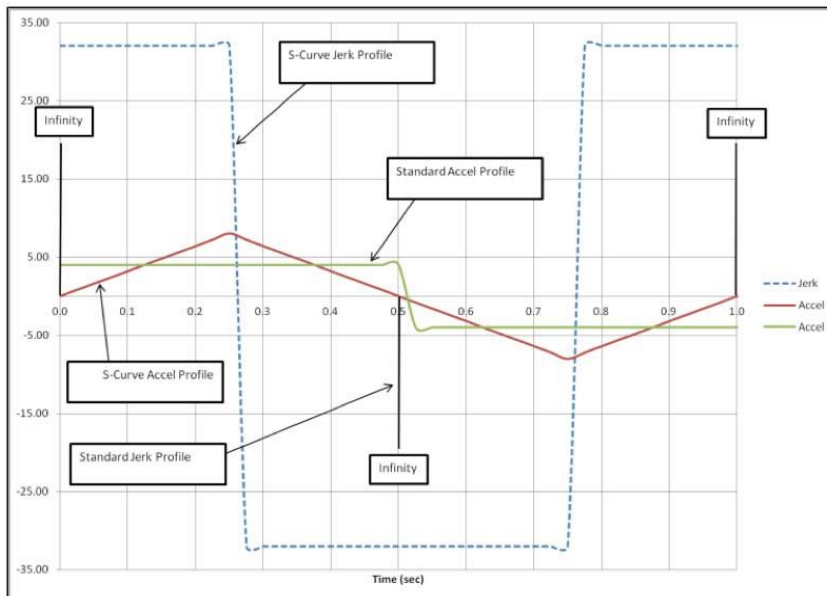
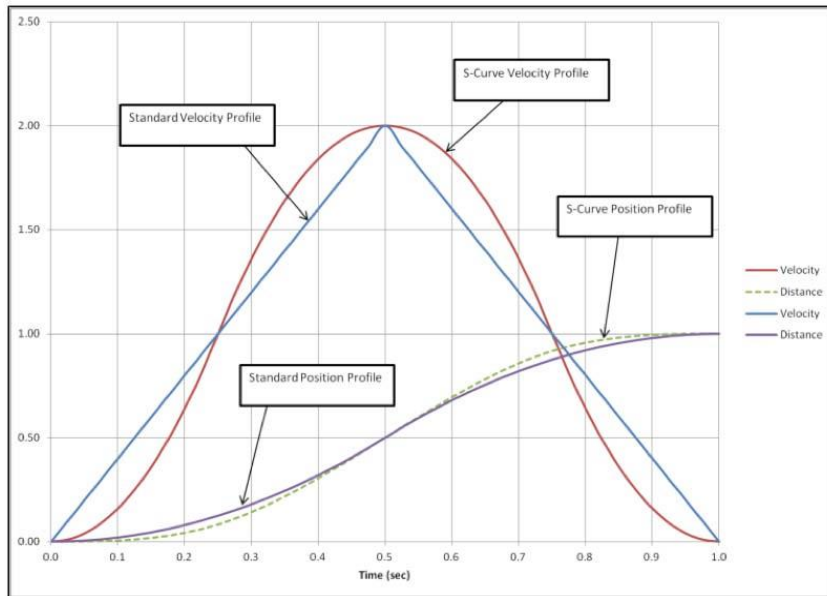
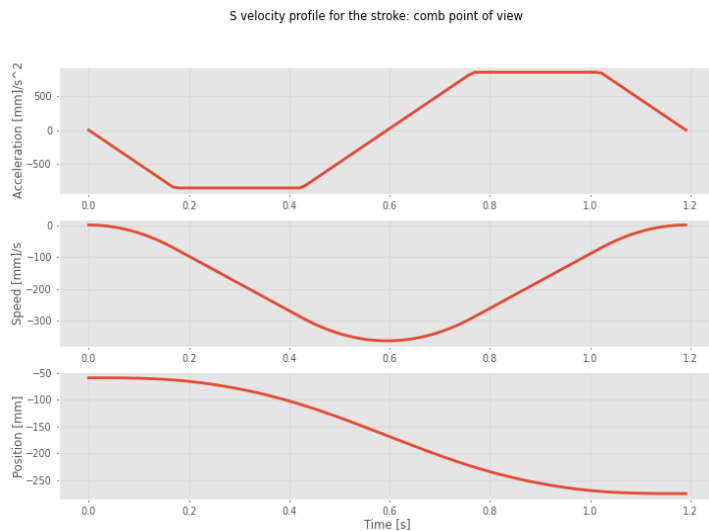


FIGURE 89 - COMPARISON BETWEEN AN S CURVE VELOCITY PROFILE AND A BANG BANG PROFILE [51]



```
S-curve for stroke:
Max θ speed = 6.31 [rad/s] = 60.23 RPM
Max θ acc = 69.95 [rad/s^2] = 40077.80 RPM^2
Max comb speed = 362.58 [mm/s]
Max comb acc = 853.84 [mm/s^2]
```

FIGURE 91 - MOTOR REQUIREMENTS WHEN APPLYING AN S-CURVE PROFILE TO THE STROKE. $J_MAX = 5000$; $V_MAX = 300$; $A_MAX = 6000$

As shown in Figure 91, implementing the S-curve profile requires a motor capable of reaching **60.23 RPM**, which is the highest among the profiles considered. However, the acceleration of **70 rad/s²** falls in the middle range between the values obtained for the quintic and trapezoidal profiles. It is worth noting that the S-curve profile results in vial acceleration of **854 mm/s²**.

FIGURE 93 - S-CURVE PROFILE APPLIED TO STROKE: COMB VIEW. $J_MAX = 5000$; $V_MAX = 300$; $A_MAX = 6000$

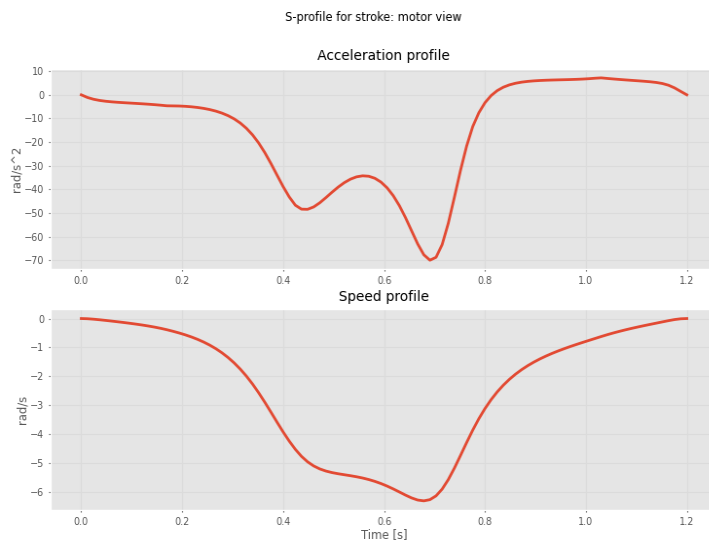


FIGURE 92 - S-CURVE PROFILE APPLIED TO STROKE: MOTOR VIEW. $J_MAX = 5000$; $V_MAX = 300$; $A_MAX = 6000$

```
S-curve for stroke:
Max θ speed = 6.07 [rad/s] = 57.93 RPM
Max θ acc = 57.22 [rad/s^2] = 32786.96 RPM^2
Max comb speed = 361.59 [mm/s]
Max comb acc = 684.47 [mm/s^2]
```

FIGURE 90 - MOTOR REQUIREMENTS WHEN APPLYING AN S-CURVE PROFILE TO THE STROKE. $J_MAX = 10000$; $V_MAX = 300$; $A_MAX = 6000$

Figure 90 demonstrates that by relaxing the constraint on the maximum jerk from 5000 to 10,000, similar results to those obtained with trapezoidal motion profiles can be achieved.

Summary

Motion profile	Maximum speed RPM	Maximum acceleration rad/s ²	Maximum acceleration mm/s ²
Quintic	56	76	866
Trapezoidal	47	50	675
S-curve	60	70	854

TABLE 9 - COMPARISON BETWEEN THE MOTION PROFILE OUTPUTS

Considering the negligible differences in velocity and acceleration among the previous conveying methods, the selection of the optimal solution should be based on the smoothness of the trajectory. In terms of fluid conveying, both the quintic and S-curve profiles appear to be the most favorable options. The S-curve profile offers added flexibility in design, making it a preferred choice for this application.

In order to meet the requirements of the problem, a motor with a minimum rotor speed of 100 RPM should be selected. The minimum torque required can be determined by correlating the acceleration values provided in Table 9 with the inertia of the final mechanical design.

3.5.2 Retreat: Velocity profile strategies

By employing a multipoint motion approach for the retreat of the walking beam, waypoints can be strategically positioned to regulate the velocity during both the engagement and disengagement phases. This allows for finer control on the engagement and disengagement of the vials.

Imposing waypoints: Linear segments with parabolic blend (Lspb)

In this section, the Linear Segments with Parabolic Blend method is employed. This motion profile is created by interpolating between via points, resulting in constant velocity segments. The via points are strategically placed so that each engagement and disengagement phase corresponds to 40% of the retreat time. This approach can be visualized through Figure 95 and Figure 96, where the blue crosses represent the via points. [49]

The via points for the Lspb method are determined as follows, with an acceleration time for rounding set to 0.1 second, as depicted Figure 94:[49]

```
def lspb(closer_line, t):  
  
    """Waypoints parameters"""  
  
    ts = [0.4*t[-1], (1-2*0.4)*t[-1], 0.4*t[-1]] # 40% of the total time is taken from engagement and disengagement  
    tacc = 0.1 # Acceleration time to round edged.
```

FIGURE 94 - DEFINITION OF THE WAYPOINTS PARAMETERS IN THE PYTHON CODE

- Via point 1: This point corresponds to reaching the end disengagement position after 40% of the retreat duration.
- Via point 2: This point corresponds to reaching the beginning of the engagement position after 60% of the retreat duration.

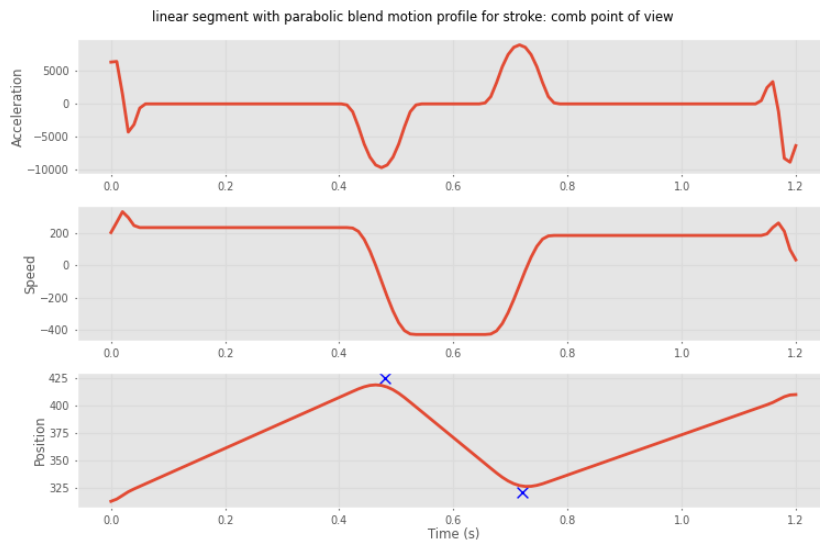


FIGURE 95 - LSPB FOR RETREAT: COMB VIEW. WAYPOINTS CROSSED AT 40 AND 60% OF THE RETREAT TIME. $T_{ACC} = 0.1$ S.

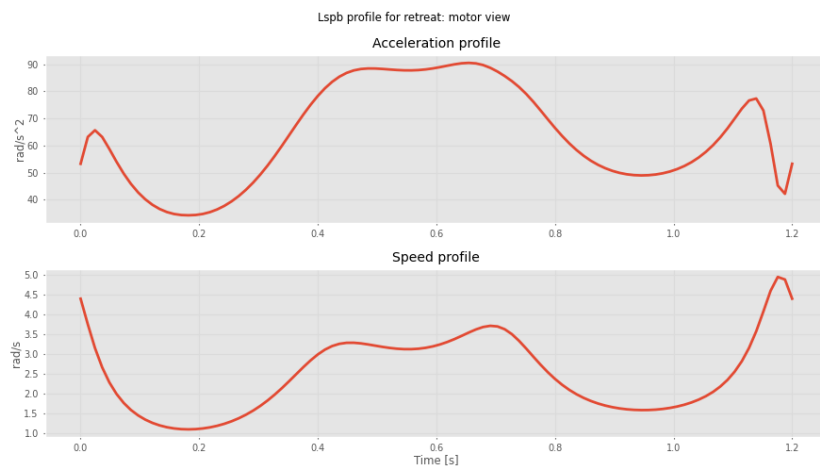


FIGURE 96 - LSPB FOR RETREAT: MOTOR VIEW. WAYPOINTS CROSSED AT 40 AND 60% OF THE RETREAT TIME. $T_{ACC} = 0.1$ S.

By utilizing Linear segments with parabolic blend (Lspb) method, the extremity constraints of null initial and ending velocities are imposed, similar to the stroke motion. This ensures continuity between the stroke motion and retreat motion.

The Lspb method, although less smooth than cubic splines, offers improved efficiency. While there may be jumps in acceleration, they are not a major concern as the vials are not expected to be engaged during the retreat. Therefore, the constraint on smoothness can be relaxed, and the Lspb method is a suitable choice. Alternatively, if stricter smoothness is required, the cubic spline interpolation method can be considered.[49]

The intermediate constraints, also known as waypoints, are approximately respected using the Lspb method, with an error dependent on the prescribed acceleration. These waypoints **introduce uncertainty in the velocities** near their locations due to the fixed trajectory of the box transport mechanism.[49]

Motor requirements for the retreat motion using the Lspb profile are provided in Figure 97. A 47 RPM motor is required. Although torque calculations need to be performed considering the system's design dimensions, it is not expected to hinder the existence of a suitable motorization system.

```
Lspb profile for retreat:  
  
Max  $\theta$  speed = 4.94 [rad/s] = 47.15 RPM  
Max  $\theta$  acc = 90.52 [rad/s2] = 51865.97 RPM2  
  
Max comb speed = 428.89 [mm/s]  
Max comb acc = 9665.34 [mm/s2]
```

FIGURE 97 - LSPB FOR RETREAT: MOTOR REQUIREMENTS. WAYPOINTS CROSSED AT 40 AND 60% OF THE RETREAT TIME. T_ACC = 0.1 s.

3.5.3 Overall motor requirements

Based on the previous analysis, achieving a filling and retreat time of 1.2 seconds is feasible for the standard scaled box transport mechanism. To accomplish this, a motor with the following characteristics is required:

1. **Maximum speed:** The motor should have a speed capability greater than 60 RPM to accommodate the desired motion profiles.
2. **Maximum acceleration:** The motor should be capable of providing an acceleration greater than 91 rad/s². Once the inertial properties of the system are known, this acceleration requirement can be converted into torque requirements using Newton's second law for rotation ($\tau = I\alpha$).
3. **Bandwidth:** Although no specific values are provided, it is advisable to choose a motor with a wide bandwidth to allow for efficient and responsive control.

3.6 Optimization problem

Goal of this section:

The python tools developed in the previous chapters provide a strong foundation for analyzing the kinematics of a standard-scaled box transport mechanism and evaluating its suitability. By leveraging these tools for optimization, a powerful and versatile python tool for finding optimal box transport mechanisms can be created. This approach not only enables the optimization of performance but also enhances the flexibility and adaptability of the solution proposed in this thesis. The resulting tool can be easily applied to different scenarios with varying linear displacement, dimensions, and engagement constraints, making it well-suited for addressing a wide range of problems associated with box transport mechanisms. Cilyx can benefit from this tool in their endeavors.

Outcome: A Python code in Appendix 7.14.5 for maximizing/minimizing an objective function using multi-start genetic algorithm.

3.6.1 Formulation of the optimization problem

Design criteria

As depicted in Figure 98, the design criteria for the box transport mechanism are defined in the Python code, aligning with the guidelines outlined in Section 3.4. These criteria will remain constant throughout the analysis in this section to ensure that the resulting solution meets the specific requirements of this project accurately and consistently.

```
"""
Design criteria
"""

min_stroke = 216 # Minimum linear conveying sought [mm]
tol_stroke = 5 # Max trajectory deviation from the straight line [mm]

engage_length = 15 # Minimum length of straight lines allowing to engage the linear trajectory
tol_engage = 5 # Max trajectory deviation from the straight line [mm]

cosine_tol = math.cos(math.radians(10)) # Maximum cosine between engagement lines and stroke (goal: near 90°)
```

FIGURE 98 - DESIGN CRITERIA DEFINITION IN THE PYTHON TOOL

Design parameters: setting the domain

Design parameters	S_1	S_4	S_5	x_0	y_0	L_1	R
-------------------	-------	-------	-------	-------	-------	-------	-----

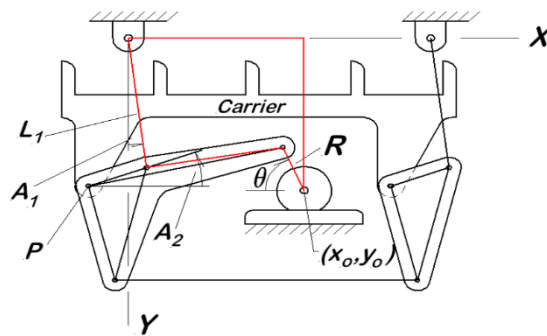


FIGURE 99 - DESIGN PARAMETERS OF AN 8-LINKAGE BOX-TRANSPORT MECHANISM

To ensure the suitability of the optimized solution, the design parameters depicted in Figure 99 should be constrained based on the specific requirements of the desired box-transport mechanism. In this case, as illustrated in Figure 100, the design parameters are limited to values that are close to the dimensions of the standard scaled box-transport mechanism. This constraint is necessary because a solution with larger overall dimensions may not be feasible or practical for the intended application.

```

'''
Design parameters: bound
'''
nv = 7 # Number of variables (S1, S4, S5, x0, y0, L1, v)
lb = [100, 100, 100, 0, 0, min_stroke/2, -40] # Lower bounds
ub = [200, 400, 400, 400, 400, 900, 40] # Upper bounds
ot = -1 # ot = 1 for maximisation, otherwise minimization

```

FIGURE 100 - BOUNDING OF THE DESIGN PARAMETERS

Objective (fitness) function:

In this project, the objective function for the optimization process is defined as the sum of the deviations in the stroke, engagement, and disengagement times, as well as a penalty term for non-compliance with the constraints and parameters boundaries. The fitness function is expressed in the following Equation (17):

$$fitness = \Delta(stroke) + \Delta(engage) + \Delta(disengage) + 100 * penalty \quad (28)$$

Note: this fitness function can be further customized or complexified based on specific requirements. For example, additional terms or factors, such as the size or geometric footprint of the design parameters, could be incorporated to account for other considerations in the evaluation of the mechanism's performance.

Non-linear constraints:

To ensure the physical integrity and proper functioning of the box-transport mechanism, it is necessary to satisfy two non-linear constraints known as the Grashof condition and the Assembly condition. These constraints, represented by Equations (12) and (13) respectively, ensure that the mechanism operates within the required range of motion and that all components can be assembled correctly.

$$\text{Grashoft:} \quad 2 * \max(S_5; L_1; R; L_4) - (S_5 + L_1 + R + L_4) < 0 \quad (29)$$

$$\text{Assembly:} \quad 2 * (\max(S_5; L_1; R; L_4) + \min(S_5; L_1; R; L_4)) - (S_5 + L_1 + R + L_4) < 0 \quad (30)$$

Note: $L_4 = \sqrt{x_0^2 + y_0^2}$

In the Python code, a method is implemented to penalize solutions that do not satisfy the Grashof and Assembly conditions. The deviation between the set of design parameters and the satisfaction of these constraints is multiplied by 100 and added to the fitness value. By doing so, as the objective is to minimize the fitness value, solutions that violate these constraints are effectively penalized.

3.6.2 Choice of the optimization method

Stochastic vs deterministic method

Figure 101 highlights the main reasons for selecting a stochastic optimization algorithm for this problem. Firstly, the problem involves seven parameters, making it a medium-sized problem. Exhaustively exploring the entire solution space proved to be time-consuming.

Comparing Optimization Methods	
Deterministic Optimization Methods	Stochastic Optimization Methods
Popular methods: Branch-and-Bound methods, Cutting Plane methods, Primal-Dual Decomposition methods, Outer Approximation methods, Inner Approximation methods, Difference of Convex methods, Reverse Convex methods, Reformulation-Linearization methods, Lipschitzian methods, Trajectory and Homotopy methods, Interval Analysis methods, etc.	Popular methods: Genetic Algorithm, Tabu Search, Particle Swarm Optimisation, Cuckoo Search, Ant Colony Optimisation, Simulated Annealing, Hill Climbing, Downhill Simplex, Artificial Bee Colony Algorithm, Swarm Intelligence, Differential Evolution Algorithm, etc.
Global optimal solution: 100% guaranteed	Global optimal solution: not 100% guaranteed (probabilistic guarantee and this probability will become 100% in infinite computing time)
Problem type: LP, MILP, NLP, MINLP	Problem type: Any types
Problem size (usually): Small	Problem size (usually): Small, medium, large
Computing time: Long and extremely long for medium and large-scale problems (not controllable)	Computing time: Can be short for medium and large-scale problems (controllable)

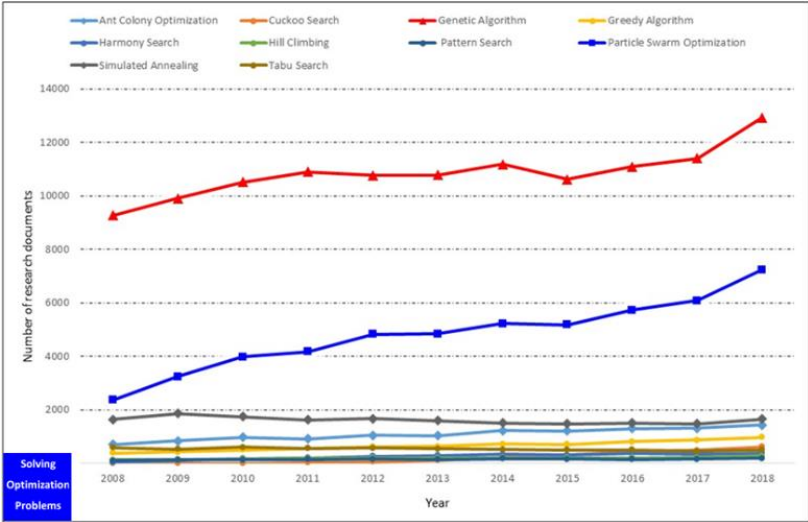
FIGURE 101 - COMPARISON BETWEEN DETERMINISTIC AND STOCHASTIC OPTIMIZATION METHODS [51]

Secondly, the mathematical relationships between the box transport parameters and the design parameters are not well-defined. Small changes in the box transport mechanism can lead to significant variations in the trajectory. The objective function is considered a black box.

Lastly, the Grashof's and assembly constraints are non-linear, which adds complexity to deterministic algorithms. Linearizing these constraints is possible but not necessary for the chosen stochastic algorithm. Indeed, such algorithms allow to simply penalize the objective function.

Although the stochastic method does not guarantee finding the global optimum, the problem's moderate number of parameters and state space increase the likelihood of obtaining a satisfactory solution.

Choice of the genetic stochastic algorithm



Among the various stochastic optimization algorithms, evolutionary algorithms have been deemed particularly suitable for solving the problem at hand. These algorithms exhibit high flexibility and have been proven effective in solving a wide range of diverse problems, as depicted in Figure 102.

FIGURE 102 - CENSUS OF THE SCOPPUS DATABASE REGARDING THE NUMBER OF APPEARANCES OF STOCHASTIC ALGORITHMS IN SCIENTIFIC PAPERS BETWEEN 2008 AND 2018 [51]

Although the particle swarm optimization algorithm is often considered more efficient than the genetic algorithm in finding the global optimal solution for large-scale problems, multiple studies have shown that there is no significant difference between the two algorithms for small-scale problems, such as the one considered in this section. Therefore, both algorithms are expected to yield comparable results in terms of finding optimal solutions and computation time. [53] [54]

The decision to choose the genetic algorithm over the particle swarm optimization algorithm was influenced by the availability and accessibility of a trusted source, a PhD researcher with a dedicated YouTube channel and a track record of winning prizes in the field. The expertise

and support provided by the researcher make the implementation and understanding of the genetic algorithm easier for this project. [53] [54]

General principle the multi-start genetic algorithm

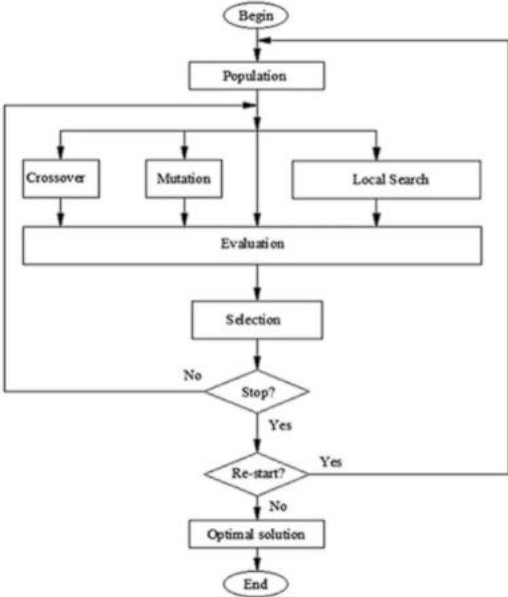


FIGURE 104 - STRUCTURE OF THE MULTI-START GENETIC ALGORITHM [51]

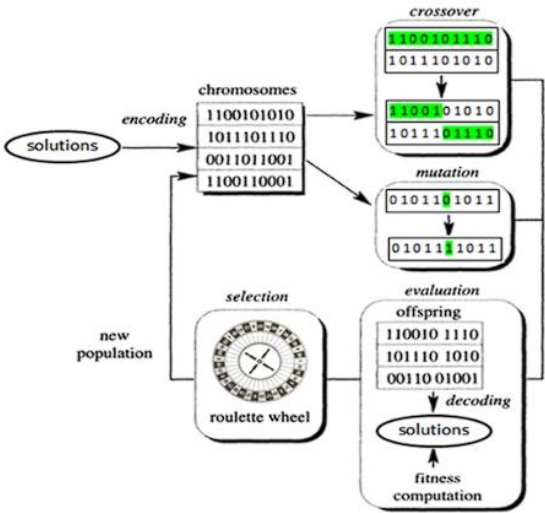


FIGURE 103 - STEPS DETAILED OF THE GENETIC ALGORITHM OPERATIONS [51]

The genetic optimization algorithm, as illustrated in Figure 104 and Figure 103, operates as follows: [54]

1. **Initialization:** An initial population is created, where each individual is called a chromosome and represents a set of solutions. In this case, chromosomes are arrays of 7 values representing the parameters of the box transport mechanism. The initial population can be defined or generated randomly.
2. **Evolution:** The chromosomes evolve through successive iterations known as generations. During each generation, the chromosomes are evaluated using the objective function and constraint functions to determine their fitness.
3. **Offspring Generation:** New chromosomes, called offspring, are created from the previous population. This can be done by merging two chromosomes from the current generation or by modifying a chromosome through a mutation operator.

4. **Selection:** A selection step is performed to choose which chromosomes will be retained for the next generation. Fitter chromosomes, those with higher fitness values, have a higher probability of being selected.
5. **Convergence:** The algorithm continues to iterate through generations, generating offspring and selecting the fittest chromosomes. Eventually, the algorithm is expected to converge to the best chromosome, representing the optimal or suboptimal solution to the problem.

The custom-made Python tool utilizes the optimization parameters shown in Figure 105, to control the number of chromosomes (i.e., sets of parameters) considered in each step.

```
"""
Optimization parameters
"""

pop_size = 20
cr = 4 # Crossover_rate
mr = 6 # Mutation_rate
sr = 5 # Local_search_rate
a = 6 # adaptive restart

computing_time = 5
no_gen = 3000000 # because we use computing time as termination criterion
```

FIGURE 105 - GENETIC OPTIMIZATION PARAMETERS

3.6.3 Practical implementation

In the particular case of this project, the previous algorithm has been implemented in Python. The code is available in Appendix 7.14.5 and this section aims at presenting all its customizable parts which can be modified to fit new problem requirements.

Objective function:

Figure 106 shows that if no straight line have been found in a trajectory, the objective function is set to 1E7 which is expected to be far greater than the other solution. By doing so, set of parameters which does not achieve minimal requirements on straight lines and engagement length are discarded.


```
def objective_function(pop, lb, ub, ot, min_stroke, engage_length, tol_stroke, tol_engage, cosine_tol):
    # Customize here to solve your problem. The rest will be handled automatically
    s = {'S1': pop[0], 'S4': pop[1], 'S5': pop[2]}
    x0 = pop[3]
    y0 = pop[4]
    l1 = pop[5]

    closer_line, motor_speed = box_transfer(s, x0, y0, l1, min_stroke, engage_length,
                                           tol_stroke, tol_engage, cosine_tol)

    if not closer_line:
        return 10**7

    objective = abs(closer_line.line.length-min_stroke) + abs(closer_line.engaging.length + # objective function here
                                                             closer_line.disengaging.length - 2 * engage_length)

    penalty = penalty_function(pop, lb, ub, min_stroke)

    # .....
    # .....

    if ot == 1:
        y = objective - 100 * penalty
    else:
        y = objective + 100 * penalty
```

FIGURE 106 - OBJECTIVE_FUNCTION

Penalty handling:

Figure 107 illustrates the implementation of the Grashof and assembly constraints in the code. The deviation between the set of design parameters and the satisfaction of these constraints is stored in variables c1 and c2, respectively. These deviations quantify the violation of the constraints.

```
def penalty_function(chromosome, lb, ub, min_stroke):

    c1, c2, ceq = nonlinear_constraints(chromosome, min_stroke)
    if c1 <= 0:
        c1 = 0
    else:
        c1 = c1
    if c2 <= 0:
        c2 = 0
    else:
        c2 = c2
    v = c1 + c2 + abs(ceq)

    # check the bounds

    y1 = chromosome.shape[0]

    h = np.zeros((2, y1))
    for i in range(y1):
        if chromosome[i] < lb[i]:
            h[0, i] = abs(chromosome[i] - lb[i])
        else:
            h[0, i] = 0
        if chromosome[i] > ub[i]:
            h[1, i] = abs(chromosome[i] - ub[i])
        else:
            h[1, i] = 0

    penalty = np.sum(np.sum(h)) + v # penalty
```

FIGURE 108 - PENALTY_FUCTION

To penalize the violation of the constraints, a penalty term is constructed as shown in Figure 108. This penalty term is obtained by adding the deviations c1 and c2 to the amount by which each design parameter deviates from its bounded domain. This approach ensures that solutions that do not satisfy the constraints are penalized in the fitness function. Indeed, for minimization, the penalty is multiplied by a factor of 100

```
def nonlinear_constraints(chromosome, min_stroke):
    s = {'S1': chromosome[0], 'S4': chromosome[1], 'S5': chromosome[2]}
    x0 = chromosome[3]
    y0 = chromosome[4]
    l1 = chromosome[5]
    r = (l1 * 3.14321 / 180) * math.degrees(math.asin(min_stroke / (2 * l1)))
    l4 = math.sqrt(x0 ** 2 + y0 ** 2)
    # Customize here to solve your problem. The rest will be handled automatically
    c1 = 2*max(l4, l1, s['S5'], r) - (l4 + l1 + s['S5'] + r) # <= 0 constraint
    c2 = 2*(max(l4, l1, s['S5'], r) + min(l4, l1, s['S5'], r)) - (l4 + l1 + s['S5'] + r)

    ceq = 0 # constraint

    return c1, c2, ceq
```

FIGURE 107 - NON_LINEAR_CONSTRAINTS_FUNCTION

and added to the fitness function.

Evaluation function: DOES NOT WORK WELL

The below evaluation function increments the real fitness value by a factor of 10E6 so that maximization and minimisation are well performed in the algorithm

```
def evaluation(pop, ot, lb, ub, min_stroke, engage_length, tol_stroke, tol_engage, cosine_tol):
    # ot == 1 = maximisation; ot == 0 for minimisation

    h = np.zeros((pop.shape[0], 1))
    for chromosome in range(pop.shape[0]):
        h[chromosome] = objective_function(pop[chromosome], lb, ub, ot, min_stroke,
                                          engage_length, tol_stroke, tol_engage, cosine_tol)

    # .....
    # .....

    if ot == 1:
        y = 10**6 + h
    else:
        y = 10**6 - h

    return y
```

FIGURE 109 - EVALUATION_FUNCTION

IMPORTANT REMARK:

A conflict arises when a generation contains chromosomes for which a stroke has been found and chromosomes for which no stroke has been found. This conflict is due to the 10E6 value of the evaluation function with respect to the 10E7 value returned by the objective function. This leads to the evaluation function producing both positive and negative fitness values within the same generation, as can be seen in Figure 110. As a result, when the selection function attempts to compute assignment probabilities, this conflict prevents the algorithm from producing accurate results.

```
[-9000000. ]
[-9000000. ]
[ 999996.70265436]
[ 999996.23690197]
[ 999996.26259882]
[ 999996.26798195]
[ 999996.26695212]]
Traceback (most recent call last):
  File "C:\Users\Cédric Keutgen\Desktop\TFE_15_05_2023\TFE\Python\Box_optimize_main.py", line 74, in <module>
    A = msga(pop_size, cr, mr, sr, nv, lb, ub, ot, a, no_gen, computing_time, min_stroke, engage_length,
  File "C:\Users\Cédric Keutgen\Desktop\TFE_15_05_2023\TFE\Python\Box_optimize.py", line 279, in msga_constrained_main
    pop = selection(extended_pop, fitness, pop_size)
  File "C:\Users\Cédric Keutgen\Desktop\TFE_15_05_2023\TFE\Python\Box_optimize.py", line 172, in selection
    index_selected = np.random.choice(index, size=pop_size - 1, replace=False, p=p)
  File "mtrand.pyx", line 956, in numpy.random.mtrand.RandomState.choice
ValueError: probabilities are not non-negative
```

FIGURE 110 - ERROR IN THE CODE THAT REQUIRES CORRECTION.

Despite this conflict, the trajectory analysis functions bypass the bug by displaying the design parameters that yield suitable trajectories on the terminal when they are found. This allows for the identification of good results even without the algorithm being able to produce accurate fitness values. **Indeed, it is observed that the algorithm even converge well to solutions which minimize the objective function.**

3.6.4 Results

Despite the previously discussed bug, the algorithm is capable of exploring various trajectories and incrementally improving the fitness value until it discovers a solution that precisely meets the design criteria. In the course of a single run, the algorithm explored the trajectory depicted in Figure 111, which initially had an unfavorable stroke length of 228 mm. However, it continuously refined the results, eventually identifying the trajectory shown in Figure 112 with a stroke length of 216 mm, which perfectly aligns with the optimum design criteria asked to the algorithm.

This demonstrates the algorithm's effectiveness in iteratively optimizing the design parameters of the box transport mechanism, allowing it to achieve the desired design criteria through continuous improvement.

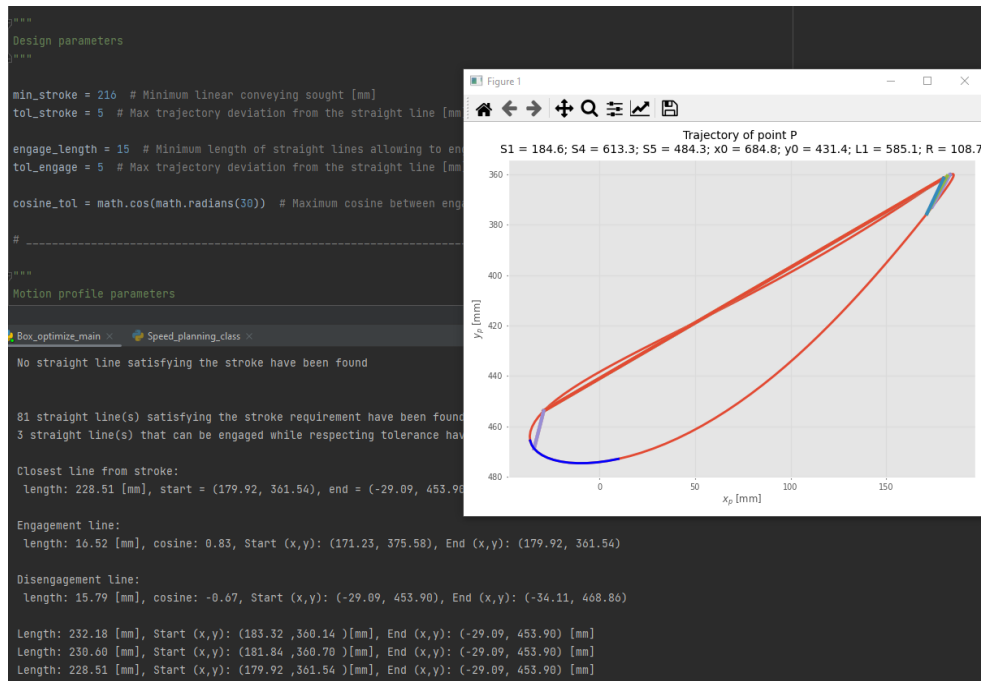


FIGURE 111 - TRAJECTORY EXPLORED

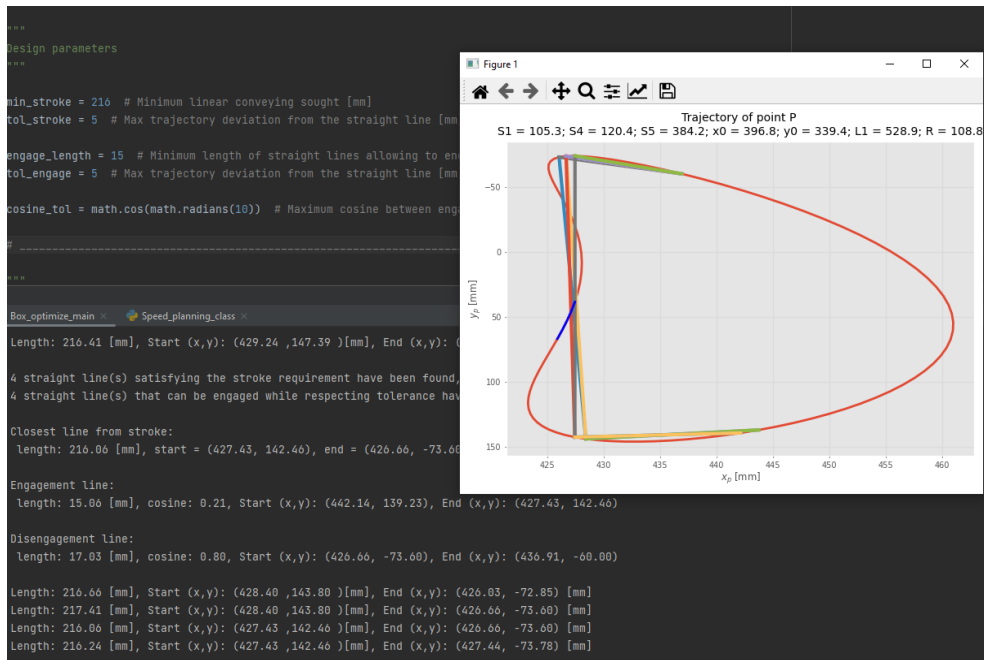


FIGURE 112 - ERGONOMIC TRAJECTORY

Optimized box transport mechanism:

Stroke = 216 mm	S_1	S_4	S_5	x_0	y_0	L_1	R
Values [mm]	105.3	120.4	384.2	396.8	339.4	528.9	108.8

Standard box transport mechanism:

Stroke = 216 mm	S_1	S_4	S_5	x_0	y_0	L_1	R
Values [mm]	143	422	287	262	317	310	122

Conclusion:

The optimized box transport mechanism surpasses the standard box transport mechanism only in terms of S_1 , S_4 , and R parameters. To enhance this Python tool, the objective function should be further developed to incorporate the length of design parameters. **Nevertheless, the Python tool created in this section successfully enhances the flexibility of designing an 8-linkage box transport mechanism.**

3.7 Mechanical design for multiformat

Problem addressed in this section:

The design of the system needs to accommodate several requirements listed below:

- The number of multiformat parts that need to be changed when switching between filling configurations should be minimised.
- A laminar airflow of 0.45 m/s as per the environmental study should be maintained at the working height.
- The overall geometric footprint of the system should be minimised.
- All components of the system should be accessible through glove ports with a length of 85 cm.
- Appropriate materials should be considered for machining, such as POM for parts in contact with vials and stainless steel for other components.
- No surface should allow particle accumulations.

Outcome:

This chapter presents a mechanical design solution for the 2R configuration, that allows the aseptic conveying system to handle three different vial sizes (2R, 15R, and 30R) while requiring only a single format part to be changed, namely the comb in contact with the vials. All appearing values have been established considering Pfizer' vial datasheets, available in Appendix Pfizer vials datasheet

The design incorporates guide rails that can be manually adjusted using a double-filet end screw and knob. All components are positioned to avoid disrupting the airflow at the working height, which is located at the upper neck of the vials. Additionally, sharp edges have been rounded with a minimum radius of 1 mm to prevent operator injuries and minimize airflow disruption.

3.7.1 Comb (format piece)

```

stroke = 216.00 | Number of vials | Teeth width [mm] |
-----|-----|-----|
17 mm vials   |      6      |      19   |
-----|-----|-----|
25 mm vials   |      6      |      11   |
-----|-----|-----|
31 mm vials   |      6      |      5    |
-----|-----|-----|

Needles locations:
needles 17 = [ 18. 54. 90. 126. 162. 198.]
needles 25 = [ 18. 54. 90. 126. 162. 198.]
needles 31 = [ 18. 54. 90. 126. 162. 198.]

Total length of box transfer mechanism : 108.00 [cm]
-----|-----|-----|

```

As a reminder, the dimensions of the combs are determined based on Figure 113, which was obtained in Section Filling circuit and stroke definition, Both combs will have a total length of 108 cm and will be made of POM.

FIGURE 113 - COMB DIMENSIONS COMPLIANT WITH THE SOLUTION

Interface with vials:

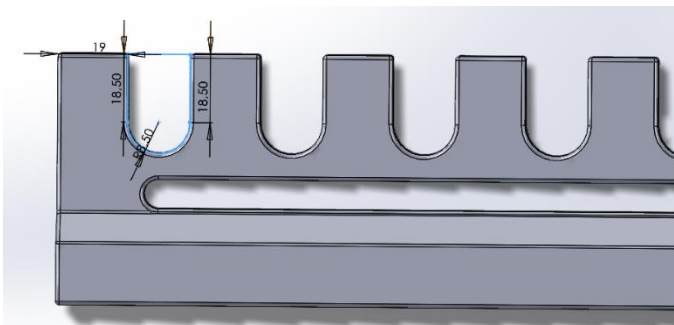


FIGURE 114 - DIMENSIONS OF GAPS AND TEETH BETWEEN TEETH FOR THE 2R CONFIGURATION

As depicted in Figure 114, the gaps between teeth in the comb have a width equal to the vial diameter plus $C = 1$ mm. This accounts for machining uncertainties and facilitates the engagement of vials.

The height of the gaps is equal to the vial diameter plus 10 mm. This allows the box transport mechanism to deviate from the straight line by 5 mm in both normal directions.

The nominal width of the teeth is set to 19 mm, as determined from Figure 113. Additionally, the sharp corners of the teeth are rounded with a radius of 1 mm.

Interface with filling and stoppering stations:

By designing the filling station with an inter-needle distance of 36 mm, as shown in Figure 113, a perfect alignment is achieved with the 2R, 15R, and 30R combs. This eliminates the need to change the needle support piece or adjust the needle locations. Indeed, the combs are also designed to have an inter-vial distance of 36 mm, as depicted in Figure 115, ensuring a perfect match between those components.

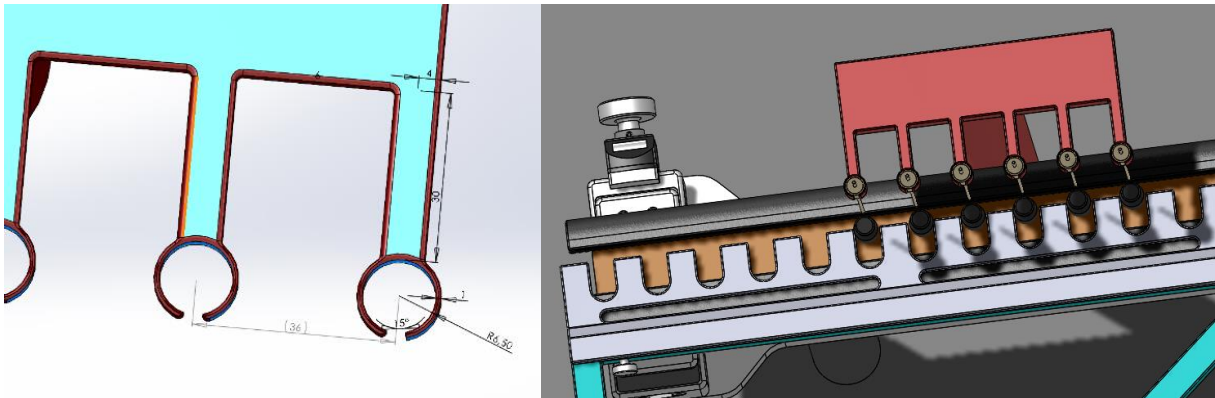


FIGURE 115 – NEEDLES INTER-DISTANCE AND PERFECT MATCH WITH THE 2R COMB

Interface with support:

Since stainless steel is a much stronger and more rigid material than POM, a portion of the support passes underneath the comb to absorb some of the mechanical forces acting on it, such as those generated by gravity, which could eventually lead to its bending.

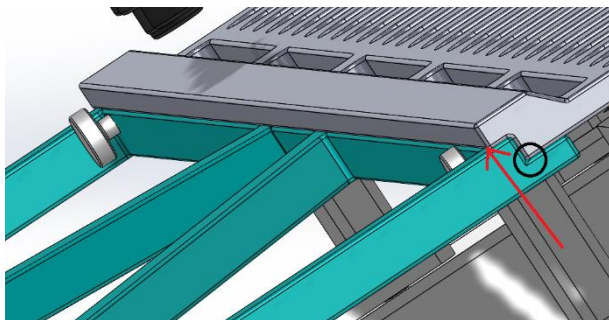


FIGURE 116 - INTERFACE BETWEEN THE COMBS AND THE SUPPORT IN THE SYSTEM. THE BLACK CIRCLE HIGHLIGHTS AN EDGE THAT WOULD HAVE BEEN EXPOSED TO PARTICLE ACCUMULATIONS IF THE COMB WERE NOT DESIGNED TO PASS OVER THE SUPPORT UNTIL THE RED ARROW

However, by passing the support underneath the comb, the contact surfaces between the two are poorly exposed to airflow, which could result in particle accumulation at the edge surrounded by a black circle in Figure 116.

To address this issue, the comb design has been conceived to pass over the support and reach the material recesses already present in it, indicated by the red arrow. By doing so, the particles should traverse the system, greatly reducing their probability of reaching the edge surrounded by a black circle.

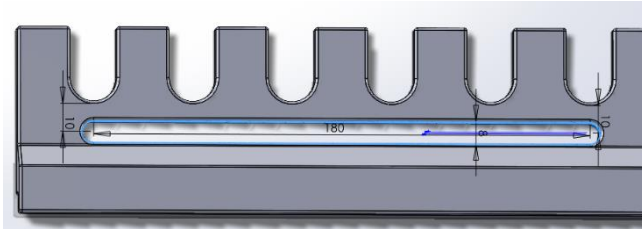


FIGURE 117 - OBLONG HOLES FOR PARTICLES REMOVAL

Passing the comb over the support poses a danger. A directed airflow towards the teeth will occur, which is hazardous if it carries particles along with it. To counter this issue, oblong holes depicted in Figure 117 have been placed just before the teeth to serve as particle outlets or vents. Those holes have been rounded with a radius of 2 mm for reduced turbulence generation.

Considerations for easy format changeover:

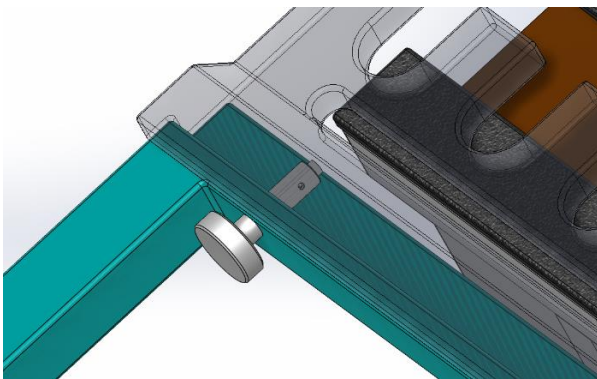


FIGURE 118 - CAPTIVE KNOB FOR FORMAT CHANGEOVER

As shown in Figure 118, a captive knob is integrated into the stainless-steel structure perpendicular to the airflow, for minimizing its disturbance. This design choice allows for easier part changeover.

However, a concern arises as the knob needs to be screwed into the combs, which are made of POM, a plastic material. To address this issue, metal inserts can be embedded in the format parts, providing a secure attachment point for the knob.

3.7.2 Guide rails

This sub-section presents the dimensions of each component in the conveying system shown in Figure 121. In this design, the heights and dimensions of the components have been calculated to leave a 0.5 mm gap between each of them. This approach is intended to accommodate for mechanical uncertainties during machining and assembly processes.

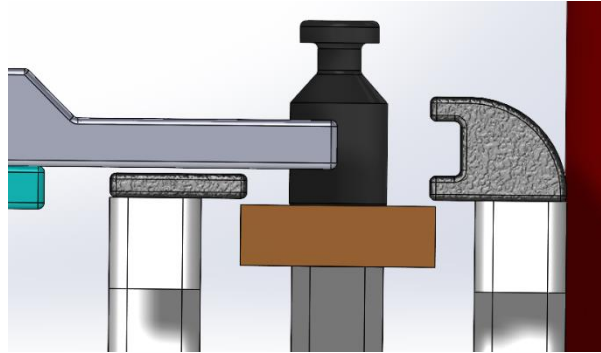


FIGURE 119 - GLOBAL VIEW OF THE CONVEYING SYSTEM

Coping for stroke deviations:

Below Figure 120 and Figure 121, the dimensions of a C-shaped guide rail are provided. This guide rail is designed to guide 2R vials for 8 mm of their height, which corresponds to 45% of their total height. It also takes into account the normal deviation of 5 mm of the box transport mechanism, which has an arbitrary thickness fixed at 8 mm in relation to the stroke. A compromise arises between the comb thickness, which is related to mechanical strength, and the contact length between the guide rail and the vials.

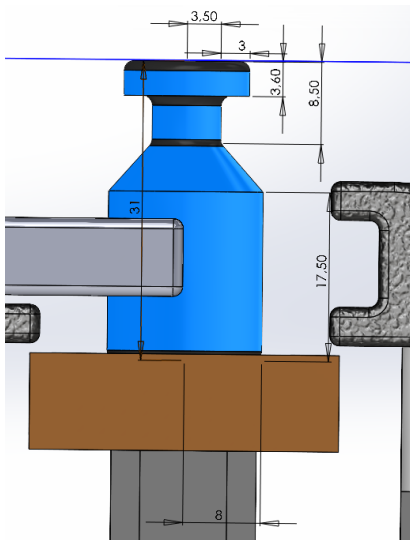


FIGURE 120 - PFIZER 2R VIAL NOMINAL DIMENSIONS

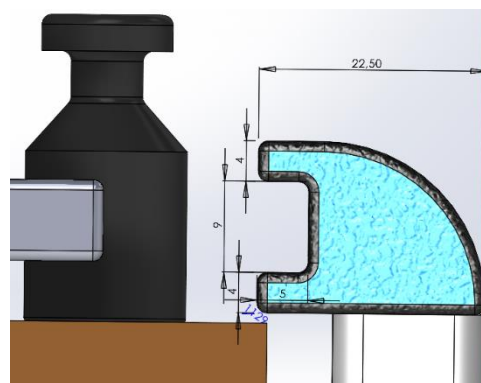


FIGURE 121 - C-SHAPED GUIDE RAIL DIMENSIONS

Why a C-shaped guide rail?

shaped

The guide rail is designed with a C-shaped structure to make contact with the vials both below and above their center of gravity. This design is intended to counteract any torque that may arise when the vials are pressed against the guide rail. The friction-induced pushing effect between the comb and the vials is expected to be minimal due to the smooth gliding properties of the POM material.

Alternative to triangular-tooth comb:

The project did not involve the design of a triangular-tooth comb for fixing the vials' positions during the engagement and retreat of the walking beam motion. Instead, a guide rail, as shown in Figure 122, has been implemented. Similar to the C-shaped guide rail, its purpose is to prevent any movement of the vials perpendicular to the stroke, but in the opposite direction.

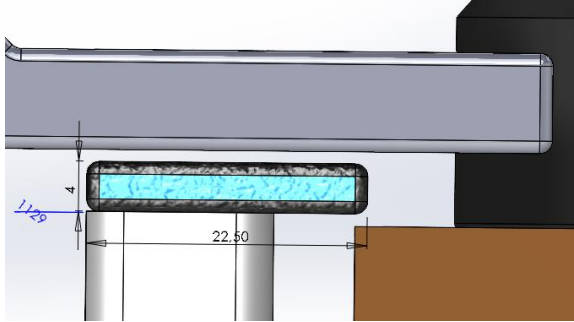


FIGURE 122 - SECOND GUIDE RAIL TO ENSURE A STRAIGHT TRACK

Adjusting the track for format changeover:

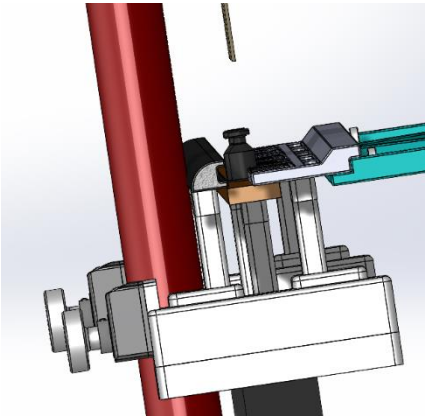


FIGURE 123 - ADJUSTING KNOB

To avoid considering the guide rails as format parts, they have been positioned on adjusting knobs located at the filling and stoppering station for maximum accuracy, as depicted in Figure 123.

As shown in Figure 124, these adjustment knobs have been strategically placed at the back of the system to ensure easy access for operators through the glove ports. The knobs are designed as endless screws with inverted threading, allowing the C-shaped guide rail and the other rail to move in opposite directions when the knob is turned.

This design enables the width of the conveying track to be adjusted according to the bottom diameter of each vial, ensuring maximum accuracy and adaptability.

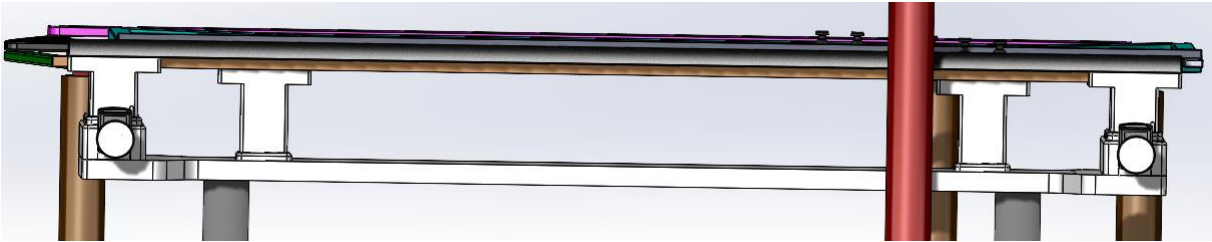


FIGURE 124 - ADJUSTING KNOBS AT THE FILLING AND STOPPERING STATION

3.7.3 Global view of the proposed solution

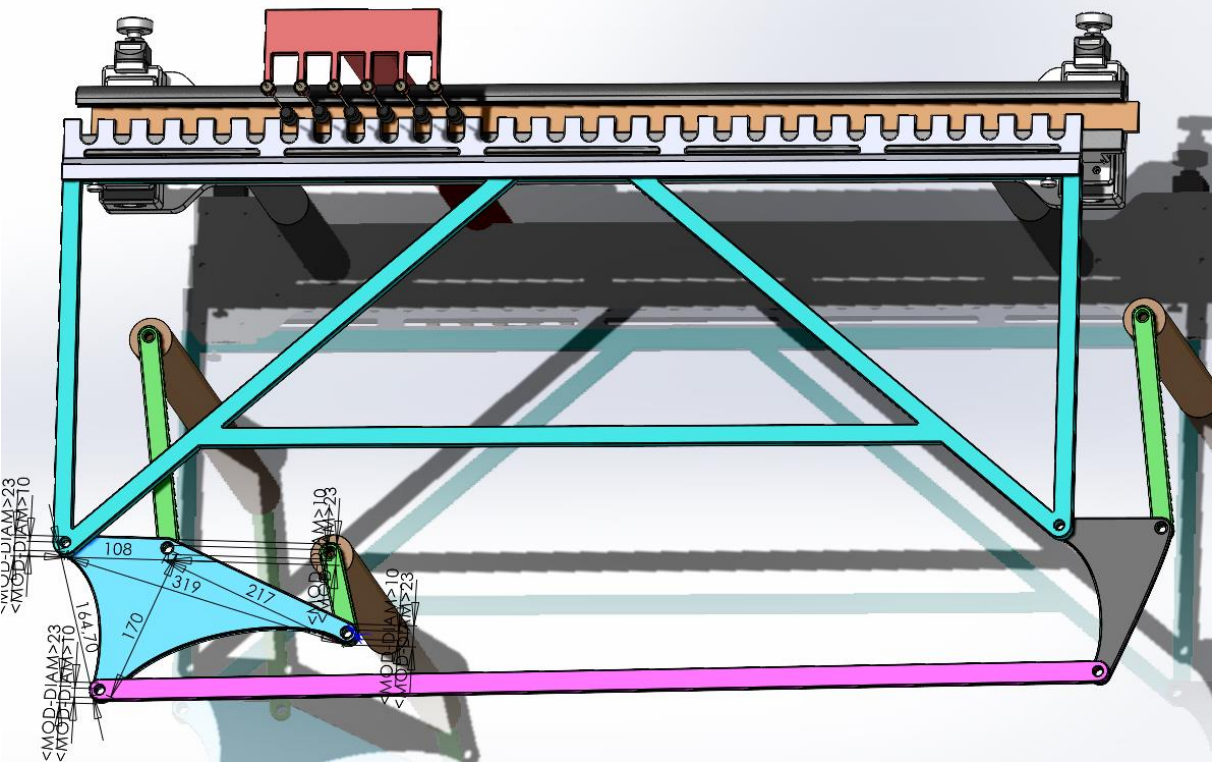


FIGURE 125 - MECHANICAL DESIGN FOR THE GLOBAL SOLUTION FOR GRADE A ASEPTIC MULTIFORMAT FILLING.

3.8 Sealing the interfaces

Problem addressed in this section:

The current design of the system is located within a Grade A GMP isolator, which requires strict adherence to sealing requirements. Two types of interfaces sealing problems are identified in the proposed solution from Section 3.7.3:

1. Static Interfaces: These interfaces primarily involve components that serve as supports for the system. Ensuring effective sealing at these interfaces is essential.
2. Dynamic Interfaces: These interfaces involve rotating and sliding shafts that require sealing to prevent any leakage or contamination.

The objective of this section is to propose effective sealing solutions for both static and dynamic interface sealing problems in the system. These solutions are designed to maintain the integrity and aseptic nature of the system within the Grade A GMP isolator, while taking into account specific constraints and requirements, including:

- Operation under a pressure gradient not exceeding 100 Pa.
- Material selection in accordance with Table 11 to ensure compatibility.

Material	90% H2O2	98% H2O2	Notes
Buna N	Class 4	Class 4	
Butyl Rubber	Class 4	Class 4	
<u>Delrin</u>	Class 4	Class 4	
<u>Kel-F</u>	Class 1 to 3	Class 1 to 3	Depends on actual grade
Polyethylene	Class 2 to 4	Class 2 to 4	Depends on actual grade / Vendor
Silicon	Class 2 to 4	Class 2 to 4	Depends on actual grade / Vendor
Teflon (Virgin)	Class 1	Class 1	
Viton A	Class 2 to 4	Class 2 to 4	Depends on actual grade / Vendor
Viton B (805)	Class 1	Unknown	From <u>Dupont</u> , 1967 Vintage

- Class 1: Materials Satisfactory for Unrestricted use with H2O2 – Class 2: Materials Satisfactory for Repeated Short-Time Contact with H2O2.
 - Maximum of 4 hours at 160 F or 1 Week at 70 F.
- Class 3: Materials Satisfactory for Short-Term Contact Only.
 - Less than 1 minute at 160 F and 1 hour at 70 F for unpressurized systems. Single use only.
- Class 4: Not recommended for use with H2O2.

TABLE 10 - SEALS AND ELASTOMERS COMPATIBILITY CHART WITH H2O2

- Ease of seal replacement.
- Cleanliness: the seal should not generate particles.

3.8.1 General guidelines for seals selection

In the pharmaceutical and medical industries, a very common approach is to use seal designs from industrial applications and select materials approved for food contact. [55]

Seals can be classified into two main categories: static seals and dynamic seals. Figure 127 and Figure 126 provide a visual representation of this classification, highlighting the different types of seals within each category and consideration that should be taken into account when selecting the right one. [56]

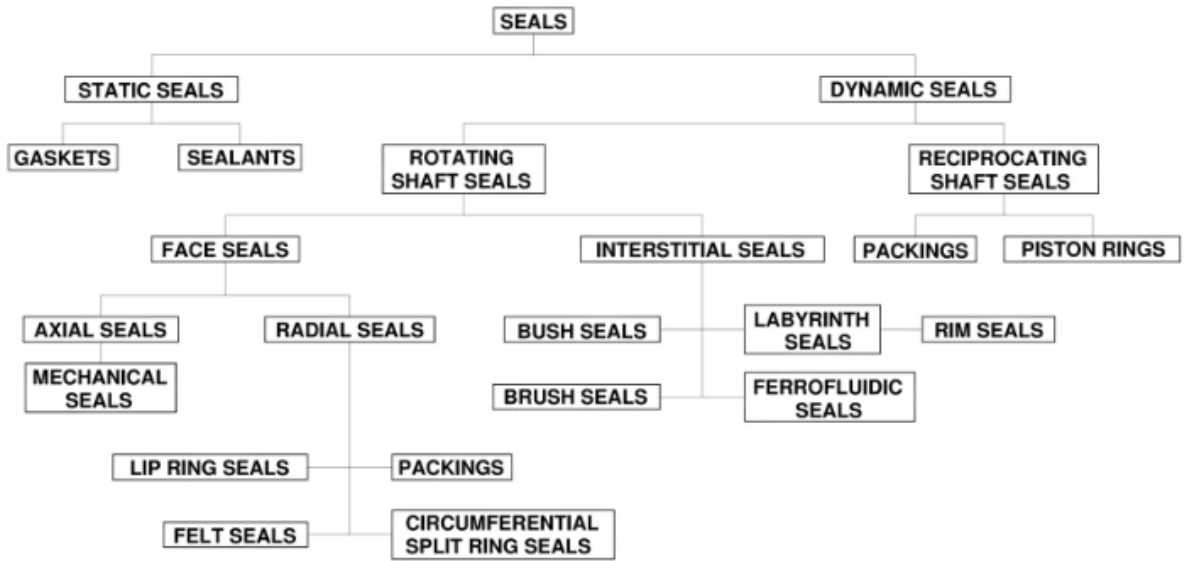


FIGURE 127 - SEAL CLASSIFICATION [52]

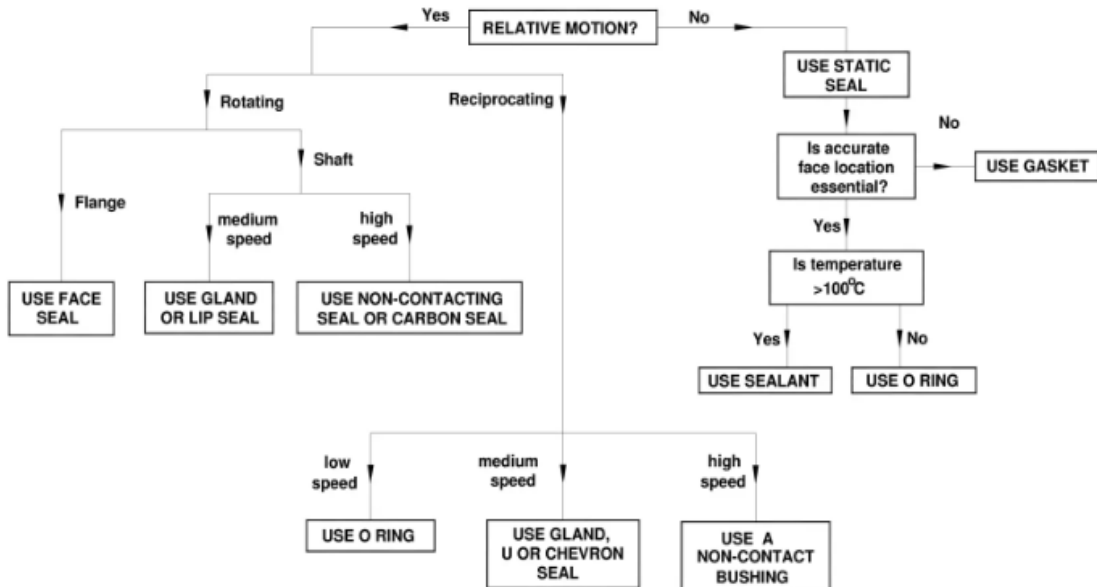


FIGURE 126 - SEAL SELECTION GUIDELINE [52]

3.8.2 Static interfaces

Unless the two surfaces in contact are perfectly flat, which is either impossible to achieve or too expensive, it is necessary to incorporate a seal into the system. Seals provide an effective solution for maintaining a tight and reliable connection between the mating surfaces. [57] [58]

Figure 104 provides a visual representation of different types of seals suitable for sealing a static joint in an aseptic isolator. Among these options, two solutions stand out: gaskets and O-rings. These sealing methods are widely used and offer the advantage of cost-effectiveness compared to extensive machining. [57] [58]

It is worth mentioning that welding is not considered as a solution in this context. This is primarily because welding has a significant drawback, which is its lack of flexibility when it comes to the assembly and disassembly of the system. Welding creates a permanent bond between the components, making it challenging to make any changes or perform maintenance tasks that may be required. [57] [58]

Gasket

As depicted in Figure 128, gaskets are designed as physical barriers to prevent any fluid leakage between two stationary components or flanges that have flat surfaces. When subjected to a compressive load, gaskets absorb energy and effectively compensate for any irregularities present on the mating surfaces. [98]

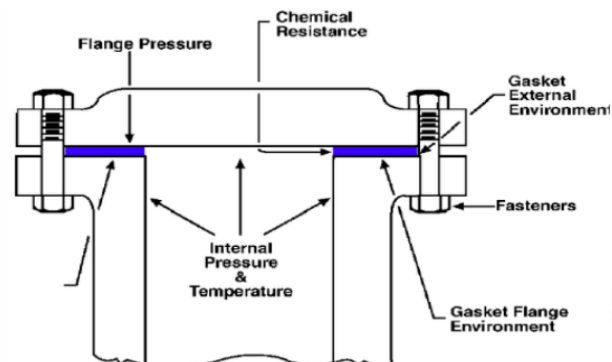


FIGURE 128 - EXAMPLE OF A GASKET USED TO SEAL TO FLANGES [55]

Indeed, gaskets are commonly made using materials that rely on the flexibility and resilience of rubber to achieve an effective seal. The specific properties of the rubber core can be modified to meet the following requirements of the sealing application previously mentioned. [59]

O'ring

An O-ring is a circular solid rubber seal that is used to block the passage of liquids or gases when compressed between two surfaces. It is typically made of elastomer material and is

placed in a specially designed groove where it is compressed between two or more parts to create an effective seal. [60]

When pressure is applied, the O-ring will shift towards the side with the least amount of pressure, which causes it to exert a tighter seal against the two components. This phenomenon is represented in Figure 129. [60]

For additional information regarding material compatibility with H₂O₂, please refer to the table available in Appendix 7.13. [61]

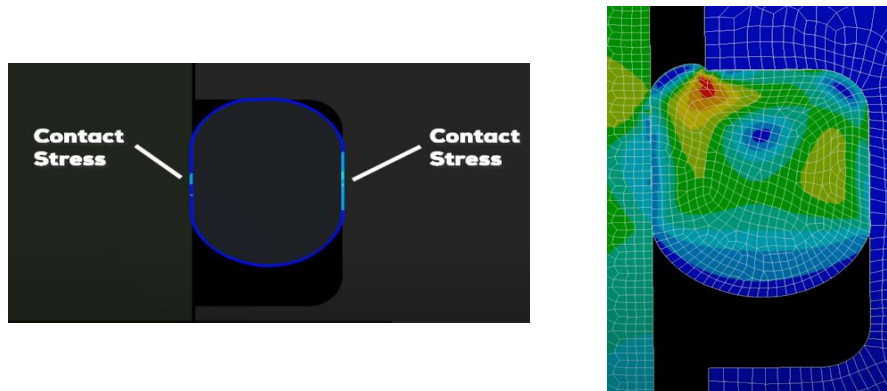


FIGURE 129 - O'RING WORKING PRINCIPLE [99]

Comparison and chosen solution:

While O-rings are more suitable for high-pressure applications, which is not necessary in this case as the maximum pressure gradient is 100 Pa, they do require machining a notch to insert them. To avoid this minor drawback, a Teflon gasket can be used as an alternative. **Virgin grade Teflon gaskets**, like the one depicted in Figure 130, offer compatibility with H₂O₂ and stainless steel. Additionally, they exhibit low particle generation and can withstand a temperature range of -200°C to 260°C while being easy to replace. **These qualities make Virgin grade PTFE gaskets a suitable choice for achieving effective sealing in the system.** [62]



FIGURE 130 - VIRGIN GRADE PTFE GASKET [57]

3.8.3 Dynamic interfaces

When it comes to sealing a shaft that undergoes both reciprocating and linear motion, one effective solution is to utilize double lip seals. Although lip seals are primarily designed for rotary motion sealing, they can be adapted to accommodate linear motion by modifying their primary lip. This adaptation has led the industry to develop six types of lip seals that are suitable for such applications, as depicted in Figure 131. [63]

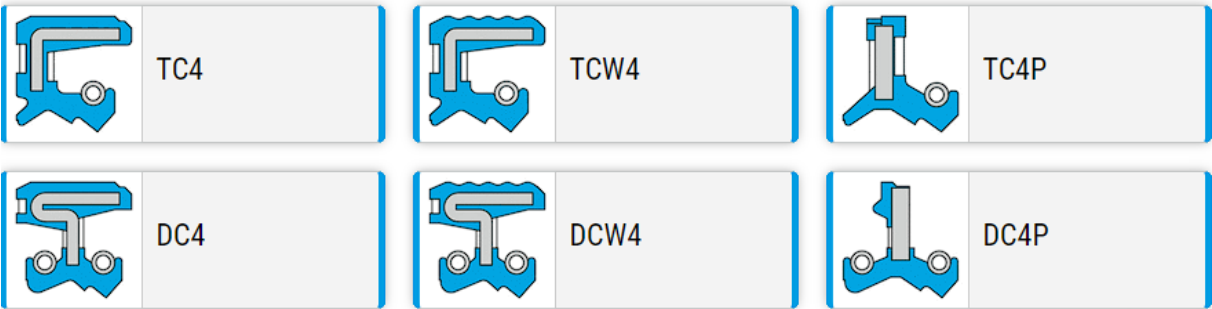


FIGURE 131 - TYPES OF DOUBLE LIPS SHAFT SEALS SUITED FOR LINEAR MOTION [102]

The metal casing of the rotary shaft seal is covered with rubber, which serves two purposes. Firstly, it protects the casing from rusting, ensuring its durability and longevity. Secondly, it helps prevent damage to the housing during the assembly process, minimizing the risk of any potential scratches or dents. [64]

By understanding the following nomenclature and the specific features of each type of lip seal, an informed decision can be made on selecting the most suitable seal for this project application, considering factors such as the nature of the media and the level of protection required. [64]

- **The single lip seals** are denoted as SC or VC. These seals have one lip that is designed to provide sealing against the internal media. [64]
- **The double lipped seals**, on the other hand, are named TC or KC. These seals have two lips, each serving a specific purpose. As shown in Figure 132, the first lip provides sealing against the internal media, while the second lip offers additional protection against dirt and dust. This design helps to maintain the integrity of the sealing and prevent any contaminants from entering the system. [64]
- For applications that require protection against two different media, the double lipped seal with dual protection is denoted as DC. These seals incorporate an additional spring to ensure effective sealing against both media. [64]



FIGURE 132 - FUNCTIONING OF A TC4 LIP SEAL [104]

Choice of the solution: TCA lip seal made of FKM

In this project, the objective is to protect the inside of the isolator against a single media. To achieve this, the chosen solution is to utilize the **TCA lip seal**. From the sealing catalog of "France Joint company," [63] the suitable material for this application is fluorinated rubber (**FKM**). This material has been selected based on its compatibility with H₂O₂, as indicated in the compatibility chart provided in Appendix 7.13. [63]

By implementing the solution depicted in the figure below, which incorporates the TC4 lip seal, a maximum linear speed of 1.5 m/s can be achieved while maintaining a pressure gradient of 100 Pa at both the stoppering and filling stations. [63]

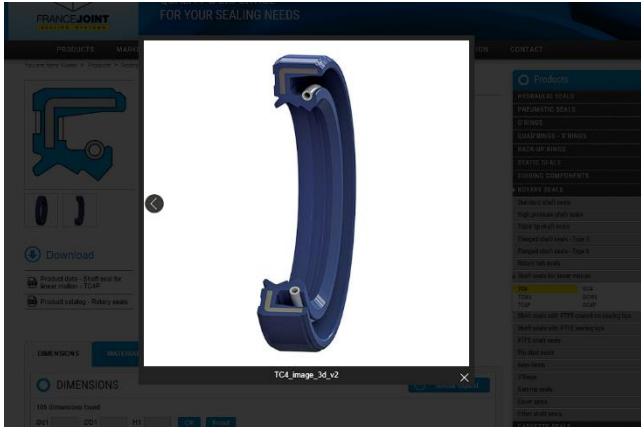
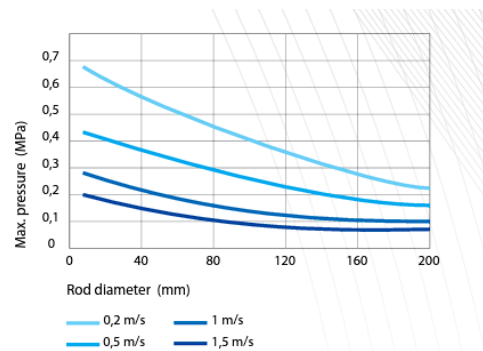


FIGURE 133 - CHOISE OF A TC4 LIP SEAL FROM "FRANCE JOINT" CATALOG



Note: For the purpose of brevity, this chapter has not delved into alternative solutions such as non-contact seals like labyrinth seals. These seals operate without physical contact between the sealing surfaces, effectively reducing particle generation.

4. Future Perspectives

4.1 Validations

To validate the global solution proposed in Section 3.7.3, the following operations should be performed:

- **Finite element analysis:**

A finite element analysis should be performed to ensure that the system withstands the acceleration it imposes to the vials.

- **Numerical simulations:**

Conduct motion simulations to verify the proper engagement and disengagement of the vials. If any issues are identified, such as inadequate engagement, the extra width of the comb gaps can be increased. This step allows for iterative improvements and the generation of a new filling station solution.

- **Fluidic analysis:**

Perform a fluidic analysis on the solution, both at rest and during motion, to ensure that no turbulence occurs on top of the vial neck (working height). If turbulence is observed due to the mechanism's motion, one potential solution could involve placing the box-transport mechanism below the comb, rather than on the same horizontal plane.

- **Prototype:**

If the previous conditions are satisfied, create a prototype and place it in an environment similar to that of an isolator. Inject smoke to analyze the smoothness and pattern of airflows.

4.2 Improvements:

- **Section 3.6:**

In this section, a bug related to the Python tool for finding optimal design parameters has been identified. While the tool is already powerful, it would be beneficial to address and correct the bug to further enhance its performance. By rectifying the bug, the tool can provide more accurate and reliable results, improving its overall effectiveness in optimizing the design parameters of the box transport mechanism.

- **Engagement/Disengagement velocities:**

The relationship between the non-respect of the via points and the speed during those phases can indeed be further explored.

- **Triangular comb:**

By converting the rotation motion of the box-transport motorization system into a linear motion, an additional comb with triangular tooth can be implemented to block vias against guide rails during the retreat of the system.

5. Conclusion

As soon as items must be moved in an intermittent motion, such as boxes, bottles, or containers, 8-linkage box transport mechanisms are likely to provide a simple and easy-to-maintain solution. Among these types of packaging, this work explored the possibility of implementing an 8-linkage box transport mechanism to address the problem of filling multiformat vials in an aseptic isolator while complying with its cleanliness level GMP Grade A.

To understand the constraints that such mechanisms would have to face, the first part of this work focused on a deep analysis of the constraints that must be respected inside a Grade A isolator. It has been highlighted that inside such a confined room, European Good Manufacturing Processes provide guidance to achieve the desired level of cleanliness..

Among those guidelines, it has been found that the 8-linkage box transport mechanism, as well as the global design of the filling station, must not disturb a laminar airflow velocity of 0.45 meters per second crossing the isolator. In addition to that mechanism, the isolator is submitted to a positive pressure gradient of the order of 50 Pa with respect to its surrounding environment. Thanks to this mechanism, cross-contamination through particles entering the system is reduced.

The sterilization mechanism of the isolator also appeared to deeply impact the material choices. Isolators have been commonly sterilized using H₂O₂. This consideration impacted the choice of materials when designing the filling station furniture. POM appeared to be perfectly suited thanks to its chemical resistance and high strength. Thanks to its high gliding properties and small friction coefficient, it has been chosen as the material for parts in contact with vials. For non-contact parts, stainless steel appeared to be a compliant choice. Regarding sealing solutions, Virgin PTFE gaskets and TCA lip seals have been mainly chosen due to this constraint.

The second part of this project involved the analysis of filling lines already present in the aseptic filling market. From an infeed analysis, as well as a filling technology perspective, this part highlighted that for production rates below 600 VPM, most of the filling stations use an intermittent motion mechanism based on servomotors and cylinders to achieve perfect triangular motion. The choice has, therefore, been made with Cilyx company to achieve a rate of 150 VPM for 2R vials while using a simpler mechanism, namely the 8-linkage box transport mechanism.

The third part of this work successfully implemented a theoretical solution. First, by considering the filling circuit as a non-format piece to avoid operator errors while changing configuration, a filling circuit filling 2R, 15R, and 30R vials using 6 1.6 mm needles while

keeping a retreat time of 1.2 seconds for the mechanism appeared to meet the requirements. Through an Excel table created for this purpose, this configuration was proven to achieve 150 VPM for 2R vials, 64 VPM for 15R vials, and 38 VPM for 30R vials.

Once this filling circuit was chosen, it imposed a conveying stroke of 216 mm and a minimum filling time of 1.2 seconds. To answer the question of whether the system using a cylinder and servomotors could be replaced by an 8-linkage box transport mechanism to achieve such motion, several python tools capable of performing a kinematic analysis of such a system have been developed and applied to a standard box transport mechanism.

This system effectively addressed the filling requirements regarding design criteria imposed on the shape of its trajectory. Regarding motion, a Python code has been developed to ensure the existence of a suitable motorization system. This tool compares several trajectory planning strategies, namely trapezoidal, quintic, and S-curve, on a box-transport trajectory and outputs the motor and cam maximum velocity and requirements. Through this analysis, it has been determined that a motor capable of reaching 70 RPM can meet the most stringent requirements regarding filling time.

Furthermore, the overall approach has been integrated into a final Python tool that provides an optimized box transport mechanism based on design criteria for its shape. While the output design parameters of this code are quite similar to the standard box transport mechanism, it allows for far greater flexibility in designing box transport mechanisms.

In the last step of the solution development, a mechanical design has been proposed, utilizing adjusting knobs and rails to ensure a linear motion of vials, regardless of their diameter.

Overall, this work has convincingly demonstrated the adaptability of the 8-linkage box transport mechanism and has provided the necessary tools to generate optimized solutions for a wide range of problems. The findings and methodologies presented in this thesis contribute to the advancement of packaging and filling technologies, particularly in the context of aseptic isolators and GMP Grade A environments. The research conducted here opens up possibilities for enhanced efficiency and precision in the pharmaceutical industry and beyond.

6. Bibliography

- [1] Reinraum-Lounge, "Aseptic toxic fill-finnish manufacturing Isolator SKAN Lehmann," SKAN, [Online]. Available: <https://www.slideshare.net/FrankMartinLehmann/2023-reinraumlounge-aseptic-toxic-fillfinnish-manufacturing-isolator-skan-lehmannpdf>. [Accessed 2023].
- [2] "What is a Cleanroom? - Angstrom Technology Cleanroom Solutions," Angstrom Technology, 21 April 2023. [Online]. Available: <https://angstromtechnology.com/what-is-a-cleanroom/#:~:text=A%20cleanroom%20is%20a%20controlled%20environment%20where%20pollutants,such%20as%20electronics%2C%20pharmaceutical%20products%2C%20and%20medical%20equipment..> [Accessed 2023].
- [3] «CLASSIFICATIONS DE SALLE BLANCHE,» Shandong Wiskind Clean Technology Co., Ltd., [En ligne]. Available: https://fr.wiskindcleanroom.com/blog/cleanroom-classifications_b65. [Accès le 2023].
- [4] «Cleanroom type according to EU GMP standard,» EU GMP Network, 08 August 2022. [En ligne]. Available: <https://gmpnetwork.eu/cleanroom-type-according-to-eu-gmp-standard/>. [Accès le 2023].
- [5] C. C. a. system, "Pharmaceutical Cleanroom Components," 01 November 2020. [Online]. Available: <https://www.cleanroom-industries.com/index.php/resources/pharmaceutical-cleanroom-components/494>. [Accessed 2023].
- [6] Thomasnet, «Cleanroom Air Flow Principles,» Thomasnet® - Product Sourcing and Supplier Discovery Platform - Find North American Manufacturers, Suppliers and Industrial Companies, [En ligne]. Available: <https://www.thomasnet.com/articles/automation-electronics/Cleanroom-Air-Flow-Principles/>.
- [7] SAFETYWISE, «Proper Use of Biological Safety Cabinets,» epublish, December 2012. [En ligne]. Available: <https://epublish.ust.hk/cgi-bin/eng/story.php?id=36&catid=29&sid=330&keycode=8c8612c2640554a4ced968458ec4eab7>. [Accès le 2023].
- [8] «The Truth about Cleanroom Air Change Rates, Charts & Methods of Calculation,» Technical Articles by Production Automation Corporation, 10 August 2020. [En ligne]. Available: <https://blog.gotopac.com/2019/05/24/the-truth-about-cleanroom-air-change-rates-calculations/#:~:text=Cleanroom%20airflow%20is%20unidirectional%20Airflow%20velocity%20is%20independent,approximately%20parallel%20streamlines%20of%2060%20%E2%80%93%2090%20FPM.> [Accès le 2023].
- [9] E. Pharma, «Unidirectional or Turbulent Airflow: Which is best for Sterile Pharmacy Compounding?,» Esco Pharma, [En ligne]. Available:

- <https://www.escopharma.com/news/unidirectional-or-turbulent-airflow-which-is-best-for-sterile-pharmacy-compounding>. [Accès le 2023].
- [10] T. C. Air, «Unidirectional Airflow Cleanroom,» Total Clean Air, 28 November 2018. [En ligne]. Available: <https://www.modular-cleanroom.net/cleanrooms/unidirectional-airflow/>. [Accès le 2023].
- [11] C. A. Velocities, «Cleanroom Airflow Velocities,» Cleanroom Airflow Velocities, 27 October 2019. [En ligne]. Available: <https://www.cleanroom-industries.com/index.php/resources/cleanroom-airflow-velocities/436>. [Accès le 2023].
- [12] ISPE, "Why is 90 FPM Considered Standard for Cleanroom Airflow?," ISPE, 01 May 2022. [Online]. Available: <https://ispe.org/pharmaceutical-engineering/march-april-2017/why-90-fpm-considered-standard-cleanroom-airflow>.
- [13] pharmaceuticalonline, "The Basics Of Cleanroom Design & Material Transfer For Microbial Control," The Basics Of Cleanroom Design & Material Transfer For Microbial Control, [Online]. Available: <https://www.pharmaceuticalonline.com/doc/the-basics-of-cleanroom-design-material-transfer-for-microbial-control-0001>. [Accessed 2023].
- [14] U. F. a. D. Administration, «Air Velocity Meters,» U.S. Food and Drug Administration, [En ligne]. Available: <https://www.fda.gov/inspections-compliance-enforcement-and-criminal-investigations/inspection-technical-guides/air-velocity-meters>. [Accès le 2023].
- [15] I. MED, "Hydrogen Peroxide Material Compatibility Chart - Industrial Spec," IS MED, 9 July 2020. [Online]. Available: <https://www.industrialspec.com/images/files/hydrogen-peroxide-material-compatibility-chart-from-ism.pdf>. [Accessed 2023].
- [16] iptsalipur, "BS5 BP303T PMB UNIT IV - iptsalipur.org," iptsalipur, [Online]. Available: https://www.iptsalipur.org/wp-content/uploads/2020/08/BP303T_PMB_UNIT_IV.pdf. [Accessed 2023].
- [17] Ensinger, «POM medical grade,» Ensinger, [En ligne]. Available: <https://www.ensingerplastics.com/en-us/shapes/biocompatible-medical-grade/pom-c-acetal#:~:text=POM%20intended%20for%20medical%20or,the%20medical%20industry's%20usual%20methods..> [Accès le 2023].
- [18] C. technology, "Air flow design: using the cascade approach," Cleanroom technology, [Online]. Available: https://www.cleanroomtechnology.com/news/article_page/Air_flow_design_using_the_cascade_approach/83305. [Accessed 2023].
- [19] E. Pharma, "Compounding Aseptic Isolator (CAI)," Esco Pharma, [Online]. Available: <https://www.escopharma.com/compounding-aseptic-isolator/compounding-aseptic-isolator-cai>. [Accessed 2023].

- [20] cleanroom-industries, «Cleanroom tightness and room pressurisation: Discrepancy?,» cleanroom-industries, [En ligne]. Available: <https://cleanroom-industries.com/en/resources/item/479-cleanroom-tightness-and-room-pressurisation-discrepancy>. [Accès le 2023].
- [21] C. C. a. system, "Bio-Pharmaceutical Cleanroom," CLIN Cleanroom architecture system, 27 October 2019. [Online]. Available: <https://www.cleanroom-industries.com/index.php/resources/bio-pharmaceutical-cleanroom/434>. [Accessed 2023].
- [22] cleanroom-industries, "Classification Impacts For Cleanroom," cleanroom-industries, 02 April 2020. [Online]. Available: <https://www.cleanroom-industries.com/index.php/resources/classification-impacts-for-cleanroom/459>. [Accessed 2023].
- [23] R. & D. World, «Cleanroom Trends: Roll-up and Bi-parting High-speed Doors,» Research & Development World, 26 March 2019. [En ligne]. Available: <https://www.rdworldonline.com/cleanroom-trends-roll-up-and-bi-parting-high-speed-doors/>.
- [24] S. Digest, "Design and operation of Biotechnology: Design and operation of biopharmaceutical airlocks," Semiconductor Digest, [Online]. Available: <https://sst.semiconductor-digest.com/2000/02/design-and-operation-of-biotechnology-design-and-operation-of-biopharmaceutical-airlocks/>. [Accessed 2023].
- [25] U. F. a. D. Administration, "Guidance for Industry - U.S. Food and Drug Administration," U.S. Food and Drug Administration, [Online]. Available: <https://www.fda.gov/media/71026/download>. [Accessed 2023].
- [26] W. P. Today, "rabs and isolator cleanroom technology combined," World Pharma Today, 15 July 2019. [Online]. Available: <https://www.worldpharmatoday.com/Articles/rabs-and-isolator-cleanroom-technology-combined-in-v-crt/>. [Accessed 2023].
- [27] C. Isolators, "Cleanroom Isolators," Thomasnet® - Product Sourcing and Supplier Discovery Platform - Find North American Manufacturers, Suppliers and Industrial Companies, [Online]. Available: <https://www.thomasnet.com/articles/automation-electronics/cleanroom-isolators/>. [Accessed 2023].
- [28] B. Sterile, "An Introduction to Isolator Technology," Berkshire Sterile, 12 October 2021. [Online]. Available: <https://www.youtube.com/watch?v=gyqRb6fgC1Y>. [Accessed 2023].
- [29] B. Sterile, «A validation approach to a multiple equipment complex filling line,» Berkshire Sterile, 15 September 2021. [En ligne]. Available: <https://www.youtube.com/watch?v=YbOtd0S3qXg>. [Accès le 2023].
- [30] D. Group, "Aseptic Filling Line Isolators," Dec Group, [Online]. Available: <https://www.dec-group.net/fillinglinesasepticcontained>. [Accessed 2023].
- [31] F. T. -. T. S. f. P. Manufacturing, "A Comprehensive Guide to Isolator Gloves," Fabtech Technologies - Turnkey Solutions for Pharma Manufacturing, 13 April 2023. [Online]. Available:

<https://www.fabtechnologies.com/quide-to-isolator-gloves/#:~:text=Port%20sizes%20come%20in%20standard%20sizes%20of%206%E2%80%B3%2C,while%20evaluating%20glove%20for%20oval%20ports%20circumference%20measurements..>
[Accessed 2023].

- [32] ISO, "ISO 8536-1:2011," ISO, 11 April 2022. [Online]. Available: <https://www.iso.org/standard/56076.html>. [Accessed 2023].
- [33] ISO, "ISO 8362-1:2018," ISO, 02 August 2018. [Online]. Available: <https://www.iso.org/standard/74398.html>.
- [34] ISO, "ISO 8362-4:2011," ISO, 11 April 2022. [Online]. Available: <https://www.iso.org/standard/57745.html>. [Accessed 2023].
- [35] T. h. ".-P. LLC, "Tubular glass vials for pharmaceutical preparations," Tubular glass vials for pharmaceutical preparations | Trading house "TD-PACK" LLC, [Online]. Available: <https://td-pack.com/en/catalog/tubular-glass-vials-pharmaceutical-preparations>. [Accessed 2023].
- [36] A. Caledonia, "Ready-to-Use Vials – Pharma Packaging Overview," Andersen Caledonia, 07 March 2023. [Online]. Available: <https://andersencaledonia.com/ready-to-use-vials-pharma-packaging-overview/#:~:text=Ready-to-Use%20%28RTU%29%20Pharmaceutical%20Glass%20Vials%20Ready-to-use%20vials%20are,chemicals%2C%20depyrogenated%20to%20remove%20endotoxins%20and%20terminally%20sterili>. [Accessed 2023].
- [37] Dupont, "Tyvek® Packaging for Pharmaceutical Protection," Dupont.co, [Online]. Available: <https://www.dupont.co.kr/packaging-materials-and-solutions/tyvek-packaging-for-pharmaceutical-protection.html>. [Accessed 2023].
- [38] P. M. & Technology, "Isolators in Aseptic Processing," Pharma Machines & Technology, 10 July 2021. [Online]. Available: <https://pharmamachines.com/isolators-in-aseptic-processing/>. [Accessed 2023].
- [39] A. P. Machines, "Sterilization and Depyrogenation Tunnel," Adinath Pharma Machines, [Online]. Available: <https://www.adinathmachines.com/autoclave-and-sterilizers/sterilization-and-depyrogenation-tunnel/#:~:text=Sterilization%20and%20Depyrogenation%20Tunnel%20or%20pharma%20sterilizing%20tunnel,%28non-sterile%29%20to%20the%20outlet%20filling%20line%20%28steril>. [Accessed 2023].
- [40] S. Scholar, "Figure 5 from Assessing contamination control of pre-sterilised container tub transfers into an aseptic manufacturing filling isolator via a de-bagging/no-touch-transfer process step | Semantic Scholar," Semantic Scholar, 01 January 1970. [Online]. Available: <https://www.semanticscholar.org/paper/Assessing-contamination-control-of-pre-sterilised-a>

Ljungqvist-Reinm%C3%BCller/e7b0d2119d4648e5d2c23e9ece6e6f560188d260/figure/8.
[Accessed 2023].

- [41] S. Scholar, " Assessing contamination control of pre-sterilised container tub transfers into an aseptic manufacturing filling isolator," Semantic Scholar, 01 January 1970. [Online]. Available: <https://www.semanticscholar.org/paper/Assessing-contamination-control-of-pre-sterilised-a-Ljungqvist-Reinm%C3%BCller/e7b0d2119d4648e5d2c23e9ece6e6f560188d260/figure/2>. [Accessed 2023].
- [42] Dara, "Dara Pharmaceutical Equipment in Virtual Pharma Expo 2022 together NJM packaging," Dara, 11 May 2022. [Online]. Available: <https://www.youtube.com/watch?v=Ac2fxrwIOSI>. [Accessed 2023].
- [43] Jerempli, "fill & finish with qualification," Jerempli, [Online]. Available: <https://www.jerempli.com/db>. [Accessed 2023].
- [44] Dara, "DL. Automatic delidding for RTU vials, syringes and cartridges," Dara, [Online]. Available: <https://www.dara-pharma.com/en/machinery-selector/1-dara-pharma-aseptic-filling-machinery/64-dl-automatic-delidding-for-rtu-vials-syringes-and-cartridges>. [Accessed 2023].
- [45] Jerempli, "fill & finish with qualification," Jerempli, [Online]. Available: <https://www.jerempli.com/dn-as>. [Accessed 2023].
- [46] B. Steriline, «Enhancing sterility assurance in isolator-based aseptic filling,» Berkshire Steriline, 02 June 2022. [En ligne]. Available: <https://www.youtube.com/watch?v=hXfykaGoxTw&t=463s>. [Accès le 2023].
- [47] IJARIIE, "DEVELOPMENT OF BOX TRANSFER MECHANISM," IJARIIE, [Online]. Available: https://ijariie.com/AdminUploadPdf/DEVELOPMENT_OF_BOX_TRANSPORT_MECHANISM_ijariie8138.pdf. [Accessed 2023].
- [48] mekanizmalar, "Box Transport Mechanism," mekanizmalar, [Online]. Available: <https://www.mekanizmalar.com/transport01.html>. [Accessed 2023].
- [49] O. Brüls, "Advanced industrial robotics course 4," Olivier Brüls. [Online]. [Accessed 2023].
- [50] nameofuser1, "py-scurve: Planning s-curve trajectories with python," nameofuser1, [Online]. Available: <https://github.com/nameofuser1/py-scurve>. [Accessed 2023].
- [51] E. e. academy, "Creating Through Points in Linear Function with Parabolic Blends Path," Engineering educator academy, [Online]. Available: for engagement and disengagement: need to control speed. Linear segment with parabolic blends . [Accessed 2023].

- [52] P. M. Devices, "Mathematics of Motion Control Profiles," Performance Motion Devices, [Online]. Available: <https://www.pmdcorp.com/resources/type/articles/get/mathematics-of-motion-control-profiles-article>. [Accessed 2023].
- [53] «A Comparative Study of Genetic Algorithm and the Particle Swarm optimization,» [En ligne]. Available: https://www.ripublication.com/irph/ijee16/ijeev9n2_06.pdf. [Accès le 2023].
- [54] L. W. Panda, "Comparing Different Characteristics of Deterministic and Stochastic Optimization Methods," Learn With Panda, 05 February 2021. [Online]. Available: <https://learnwithpanda.com/2020/05/12/comparing-different-characteristics-of-deterministic-and-stochastic-optimization-methods/>. [Accessed 2023].
- [55] A. Seals, "Pharmaceutical and Medical Seals: FDA Seals: American High Performance Seals," AHP Seals, 10 August 2017. [Online]. Available: <https://ahpseals.com/pharmaceutical-and-medical/>. [Accessed 2023].
- [56] S. a. D. K. o. SlideShare, "Chapter09 seals," Share and Discover Knowledge on SlideShare, [Online]. Available: <https://www.slideshare.net/mirhadizadeh/chapter09-seals>. [Accessed 2023].
- [57] I. Aero Rubber Company®, "What's the Difference Between a Gasket and a Seal?," Aero Rubber Company®, Inc., 05 June 2023. [Online]. Available: <https://aerorubber.com/2020/05/28/whats-the-difference-between-a-gasket-and-a-seal/>.
- [58] AZoM, "Engineering Seals - Sealing of Static Joints," AZoM.com, 11 June 2013. [Online]. Available: <https://www.azom.com/article.aspx?ArticleID=503>. [Accessed 2023].
- [59] PGC, "WHAT IS A GASKET? WHY ARE THEY USED? WHAT ARE THE CONSIDERATIONS?," PGC, 22 April 2019. [Online]. Available: <https://pgc-solutions.com/what-is-a-gasket-why-are-they-used-what-are-the-considerations/>. [Accessed 2023].
- [60] I. UC Components, "Understanding O-rings," UC Components, Inc., 03 November 2021. [Online]. Available: <https://www.uccomponents.com/understanding-o-rings/>. [Accessed 2023].
- [61] eFunda, "oring chemical compatibility with Hydrogen+Peroxide," eFunda, [Online]. Available: https://www.efunda.com/designstandards/oring/oring_chemical.cfm?SM=none&SC=Hydrogen%20Peroxide. [Accessed 2023].
- [62] gteek, "PTFE gaskets," gteek, [Online]. Available: <https://www.gteek.com/PTFE-gaskets-cutting>. [Accessed 2023].
- [63] F. JOINT, "TC4," FRANCE JOINT , [Online]. Available: <https://www.francejoint.com/shaft-seals-for-linear-motion/tc4/>. [Accessed 2023].
- [64] A. Seals, "Guide Seals: Quality, Design, Fitting and Leakage," Action Seals, [Online]. Available: <https://actionseals.co.uk/guide-to-seals>. [Accessed 2023].

- [65] PharmTech, «Understanding the Impact of Annex 1 on Isolator Design,» PharmTech, [En ligne]. Available: <https://www.pharmtech.com/view/understanding-the-impact-of-annex-1-on-isolator-design>. [Accès le 2023].
- [66] D. Filters, "HEPA Filter - 24 x 24 x 12 (11.5) 1000CFM Upstream 99.99%," DAMN Filters, [Online]. Available: <https://damnfilters.com/products/hepa-24-inch-merv-18-2000-cfm-filter>. [Accessed 2023].
- [67] Starrco, "Modular Cleanrooms and Cleanroom Wall Systems," Starrco, 20 March 2023. [Online]. Available: <https://www.starrco.com/modular-clean-rooms/>.
- [68] G. e. f. a. b. world, «Understanding aseptic filling technology,» GEA engineering for a better world. [En ligne]. [Accès le 2023].
- [69] Reddit, "Dive into anything," Reddit, [Online]. Available: https://www.reddit.com/r/EngineeringPorn/comments/3dndc4/flow_past_a_hollow_circular_cylinder/. [Accessed 2023].
- [70] ResearchGate, "Stagnation points and zones of high groundwater age in regional scale groundwater flow," ResearchGate, [Online]. Available: https://www.researchgate.net/figure/Stagnation-points-and-zones-of-high-groundwater-age-in-regional-scale-groundwater-flow_fig2_256199217. [Accessed 2023].
- [71] NAF, "30 ml injection vials (30R), dimensions \varnothing 30.00 x 75 x 1.20 mm., tubular glass, type 1.," NAF, [Online]. Available: <https://www.naf-vsm.com/flacone-di-iniezione-30-ml-30r-chiaro-borosilicato-tipo-vetro-1.html>. [Accessed 2023].
- [72] E. Academy, «What are the Differences between EU and FDA GMP?,» ECA Academy, 01 August 2018. [En ligne]. Available: <https://www.gmp-compliance.org/gmp-news/what-are-the-differences-between-eu-and-fda-gmp>.
- [73] GSK, "GSK Cleanrooms (CFD)," GSK Cleanrooms (CFD) | IES Case Study, [Online]. Available: <https://www.iesve.com/consulting/projects/2456/cfd-gsk-cleanroom-dublin>. [Accessed 2023].
- [74] T. C. Air, «Materials,» Total Clean Air, November 2018. [En ligne].
- [75] T. C. Air, "Materials," Total Clean Air, 29 November 2018. [Online]. Available: <https://www.modular-cleanroom.net/cleanrooms/materials/#:~:text=This%20includes%3A%20Paper%2C%20notebooks%2C%20pens%20and%20pencils,Wipes%20and%20detergents%20%20Solvents%20%20Plastic%20containers.>
- [76] WEBInformation, "Business Advice, Lifestyle Tips & Travel Guide Blog," WEBInformation, 14 June 2018. [Online]. Available: <https://www.webinformation.org/how-is-uncontaminated-drug-filling-possible-with-aseptic-filling-isolator/>. [Accessed 2023].

- [77] E. P. Review, "Advanced Aseptic Processing: RABS and Isolator Operations," European Pharmaceutical Review, 08 June 2017. [Online]. Available: <https://www.europeanpharmaceuticalreview.com/article/1372/advanced-aseptic-processing-rabs-and-isolator-operations/>. [Accessed 2023].
- [78] K. B. EMS-CHEMIE als Arbeitgeber: Gehalt, "EMS-CHEMIE als Arbeitgeber: Gehalt, Karriere, Benefits," kununu, [Online]. Available: <https://www.kununu.com/ch/ems-chemie3>. [Accessed 2023].
- [79] FDA-EMA, "FDA-EMA Aseptic Requirements Annex 1," FDA-EMA, [Online]. Available: https://www.pda.org/docs/default-source/website-document-library/chapters/presentations/australia/fda-ema-aseptic-requirements-annex-1.pdf?sfvrsn=ba0e9a8e_4. [Accessed 2023].
- [80] L. Hunan Grand Packaging Machinery Co., "2ml or 3ml Vial? No, It's ISO 2R - Hunan Grand Packaging Machinery Co.,Ltd," Hunan Grand Packaging Machinery Co.,Ltd, [Online]. Available: <https://www.grand-packing.com/2ml-or-3ml-vial-no-it-s-iso-2r.html>. [Accessed 2023].
- [81] B. Key, "Sterile Manufacturing Facilities," Basicmedical Key, 08 May 2017. [Online]. Available: <https://basicmedicalkey.com/sterile-manufacturing-facilities/>. [Accessed 2023].
- [82] Dara, "Line for vials and RTU syringes and cartridges," Dara, [Online]. Available: <https://www.dara-pharma.com/en/machinery-selector/1-dara-pharma-aseptic-filling-machinery/123-combi-line-for-processing-of-vials-and-rtu-syringes>. [Accessed 2023].
- [83] SlideServe, "Process Stages," SlideServe, 14 November 2014. [Online]. Available: <https://www.slideserve.com/keefe-cooley/process-stages>. [Accessed 2023].
- [84] Dara, "Complete lines for processing of vials. WM + DT + SX-310-PP," Dara, [Online]. Available: <https://www.dara-pharma.com/en/machinery-selector/1-dara-pharma-aseptic-filling-machinery/42-complete-lines-for-processing-of-vials>. [Accessed 2023].
- [85] U. S. D. T. I. K. M. A. 2017-2022, "United States Depyrogenation Tunnels Industry Key Manufacturers Analysis 2017-2022," United States Depyrogenation Tunnels Industry Key Manufacturers Analysis 2017-2022, 01 January 1970. [Online]. Available: <https://latestindustryreport.blogspot.com/2017/09/united-states-depyrogenation-tunnels.html>. [Accessed 2023].
- [86] L. Bio Pharma Systems, "WASHING/ DEPYRO TUNNELS," Bio Pharma Systems , LLC, 05 October 2018. [Online]. Available: <https://www.b-systems.com/washing-depyro-tunnels/>. [Accessed 2023].
- [87] Netsteril, "GMP production: GMP pharmaceutical quality," Netsteril, [Online]. Available: <https://netsteril.com/en/>. [Accessed 2023].

- [88] Dara, "DB/A+. Automatic debagging for RTU vials, syringes, cartridges Dara," Dara, [Online]. Available: <https://www.dara-pharma.com/en/machinery-selector/1-dara-pharma-aseptic-filling-machinery/121-db-a-automatic-debagging-for-rtu-vials-syringes-and-cartridges>. [Accessed 2023].
- [89] Versynta, "Small Batch Solutions from Syntegon: Versynta Isolated FLT," Versynta, 02 August 2022. [Online]. Available: <https://www.youtube.com/watch?v=j-g0lw4bD5s>. [Accessed 2023].
- [90] Flexicon, "FPC60: Products: Flexicon: WMFTS," WatsonMarlow, [Online]. Available: <https://www.wmfts.com/en/flexicon/fully-automatic-filling-systems/fpc60-fully-automatic-fillfinish-system/>. [Accessed 2023].
- [91] Versynta, "Versynta: Small Batch Solutions from Syntegon: Flexible Filling Platform (FFP)," Versynta, 25 January 2021. [Online]. Available: <https://www.youtube.com/watch?v=oaJHD9rGeQg>. [Accessed 2023].
- [92] Watson-Marlow, "FMB210 for filling, stoppering and crimp capping of vials," Watson-Marlow, 06 May 2014. [Online]. Available: <https://www.youtube.com/watch?v=SMiUKiaOleE>. [Accessed 2023].
- [93] Jerempli, "fill & finish with qualification NFL 1 - 2," Jerempli, [Online]. Available: <https://www.jerempli.com/nfl-1-2-rdl>. [Accessed 2023].
- [94] Jerempli, "fill & finish with qualification SX-220-PP," Jerempli, [Online]. Available: <https://www.jerempli.com/sx-220-pp>. [Accessed 2023].
- [95] Jerempli, "Jerempli sx-310-pp," Jerempli, [Online]. Available: <https://www.jerempli.com/sx-310-pp>. [Accessed 2023].
- [96] Tofflon, "Tofflon 120VPM Vial Filling Line," Tofflon, 25 May 2020. [Online]. Available: <https://www.youtube.com/watch?v=kPdpc0irHDo>. [Accessed 2023].
- [97] C. Pharmica, "Cook Pharmica Provides Liquid and Lyophilized Vial Filling and Finishing," Cook Pharmica, 16 June 2011. [Online]. Available: https://www.youtube.com/watch?v=20HbxkyW_pM. [Accessed 2023].
- [98] Klenzaims, "Klenzaims - Aseptic Filling & Closing," Klenzaims, 25 February 2020. [Online]. Available: <https://www.youtube.com/watch?v=61kYLs8LfGk>. [Accessed 2023].
- [99] Jerempli, "Jerempli NFL-5-10," Jerempli, [Online]. Available: <https://www.jerempli.com/NFL-5-10>. [Accessed 2023].
- [100] Jerempli, "Jerempli hsl-pp," Jerempli, [Online]. Available: <https://www.jerempli.com/hsl-pp>. [Accessed 2023].

- [101] Dara, "Dara Pharmaceutical Equipment - Filling and closing machines for injectable vials," Dara, 11 November 2022. [Online]. Available: <https://www.youtube.com/watch?v=H0Oq7JdU17c>. [Accessed 2023].
- [102] Watson-Marlow, "Liquid filling line - speed 600 vials/minute," Watson-Marlow, October 18 2016. [Online]. Available: https://www.youtube.com/watch?v=5a_J1kRiD3w. [Accessed 2023].
- [103] Flexicon, "PF7+ peristaltic filling machine: Flexicon: WMFTS Biopharm," WatsonMarlow, [Online]. Available: <https://www.wmfts.com/en/biopharm-products/flexicon/table-top-filling-machines/pf7pf7-peristaltic-filling-machine/>. [Accessed 2023].
- [104] WatsonMarlow, "PF7 user manual - WatsonMarlow," WatsonMarlow, [Online]. Available: <https://www.wmfts.com/siteassets/catalog/products/corporate/flexicon/table-top-filling-machines/7515/literature/m-flexicon-pf7-en2/>. [Accessed 2023].
- [105] WatsonMarlow, "Fluid path accessories: Products: Accessories: WMFTS Biopharm," WatsonMarlow, [Online]. Available: <https://www.wmfts.com/en/biopharm-products/flexicon/fluid-path-accessories/fluid-path-accessories/>. [Accessed 2023].
- [106] H. Solutions, "What are gasket seals and how can you use it?," HBS Solutions, [Online]. Available: <https://hbssolutions.eu/en/about-us/blog/what-are-gasket-seals-and-how-can-you-use-it/>. [Accessed 2023].
- [107] tarkka, "O-Rings? O-Yeah! How to Select, Design, and Install O-Ring Seals," tarkka, 24 March 2019. [Online]. Available: <https://www.youtube.com/watch?v=aweDWuNkPw0>. [Accessed 2023].
- [108] U. University, "http://mae-nas.eng.usu.edu/Peroxide_Web_Page/chapter_7.html," Utah University, [Online]. Available: http://mae-nas.eng.usu.edu/Peroxide_Web_Page/chapter_7.html. [Accessed 2023].
- [109] T. c. wijaya, "how oil seal works (animation)," Temmy candra wijaya, 25 February 2022. [Online]. Available: <https://www.youtube.com/watch?v=KcBX62wkVXw&t=324s>. [Accessed 2023].
- [110] P. Corke, "Trajectories - Robotics Toolbox for Python documentation," Peter Corke, May 15 2023. [Online]. Available: https://petercorke.github.io/robotics-toolbox-python/arm_trajectory.html#roboticstoolbox.tools.trajectory.trapezoidal. [Accessed 2023].
- [111] NextCobot, "S-Curve motion profile," NextCobot , 12 May 2020. [Online]. Available: <https://www.youtube.com/watch?v=LF0mAnpukho>. [Accessed 2023].

7. Appendix

7.1 History of the 60-90FPM airflow velocity

In 1961, a team from the Sandia Corporation based at Albuquerque, New Mexico, USA led by Willis Whitfield developed the first laminar flow technology in partnership with the U.S. Atomic Energy Commission. From their work came out the first specialised airflow cabinets whereby greater levels of cleanliness were achieved. Throughout their study, they found that an airflow velocity of 0.45 m/s (= 90 feet per minute) was sufficient to maintain unidirectional flow whilst reaching the necessary levels of particle. However, their optimal value has been criticised for a main reason. In fact, they only explored a small range of velocities from 70 to 100 FPM. Reasons for that range lies behind sound, comfort and motion of workers operating in the room. No consideration on efficiency were made.

Times having passed, technologies have evolved. Discussions of this optimal value is more than necessary to assess its current validity.

In 1963, the 90 feet per minute speed entered the first American standard for cleanrooms - Federal Standard FS209 (a precursor to the ISO 14644 standard for cleanrooms). In 1987, a kind of conflict arose between the US FDA's first guide to aseptic processing and FS 209C. While the former adopted the requirement for an air velocity of 90 feet per minute $\pm 20\%$, the latter removed mention of specific velocity altogether and emphasized air visualisation.

Nowadays, EU GMP Annex 1 propose this value of 0.45 m/s $\pm 20\%$ as a guidance in its statement "Laminar air flow systems should provide a homogeneous air speed in a range of 0.36 to 0.54 m/s (guidance value) at the working position in open clean room applications". From THE FDA 2004⁹ side, more flexibility is provided as it states: "at a velocity sufficient to sweep particles away from the filling / closing operation and maintain unidirectional airflow during operation." However, it still proposes this value in its footnote through "A velocity from 90 feet per minute is generally established, with a range of $\pm 20\%$ around the set point. Higher velocities may be appropriate in operations generating high levels of particulates."

The air velocity range cited in the regulatory documents being a "guidance value", it infers that higher or slower air velocities could be used under sufficient justification. However, one should keep in mind that lowering air velocity even if it reduces energy consumption, can lead to insufficient laminarity and reduced air cleanliness in the room. In the other hand, higher flow velocity can lead to turbulence and eddy formation. Nevertheless, it might be needed in

⁹ FDA guidance are the US equivalency of GMP. The objective of the FDA "**Annual Product Review**". The objective of the EU "**Product Quality Review**" (PQR). [What are the Differences between EU and FDA GMP? - ECA Academy \(gmp-compliance.org\)](http://www.euracadm.com/what-are-the-differences-between-eu-and-fda-gmp/)

highly particulate environment. A balance must be found to obtain an ergonomic solution, which achieves desired level of particles in the environment of interest whilst remaining cost efficient.

Regarding the 20% marge on the velocity, one can ask itself where does it come from? Why not 30%? Indeed, some people asked themselves the exact same question and made some experimentations. It is notably the case for William Whyte¹⁰ that studied a broader range of airflow velocities from 0.1 m/s to 0.6 m/s during a study on surgical operations ([233874.pdf \(gla.ac.uk\)](#)). Results he obtained have shown that below 0.3 m/s, a stable unidirectional airflow could not be obtained, and concentration of particles remained too high. Once 0.3 m/s were reached, results followed a law of 'diminished returns' meaning that the more additional energy was added to the system, the less it impacted particle levels. Hence, they concluded that 0.3 m/s was the optimal value for airflow velocity.

¹⁰ Researcher on infection control at university of Glasgow

7.2 2D Layouts of global filling processes

[A validation approach to a multiple equipment complex filling line - YouTube](#)

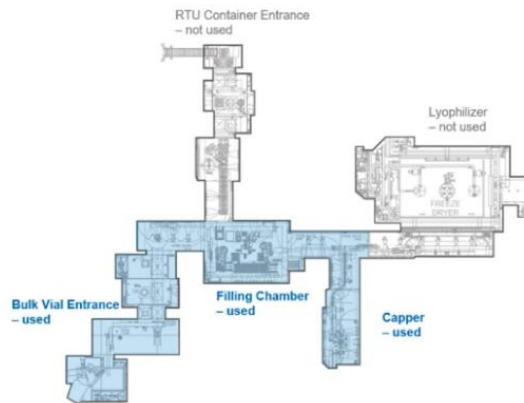
Berkshire flexible line: 3000 vials/h for 2R; 2000 vials/h for 10R

Supervisory PC

Operating Modes:

- A liquid fill in a bulk glass vial

Operation Mode (Liquid Fill in a Bulk Glass Vial):

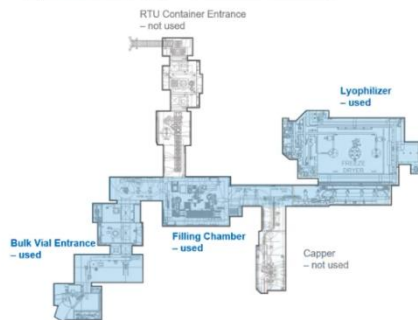


Supervisory PC

Operating Modes:

- A liquid fill in a bulk glass vial
- A lyo fill in a bulk glass vial

Operation Mode (Lyo Fill in a Bulk Glass Vial):

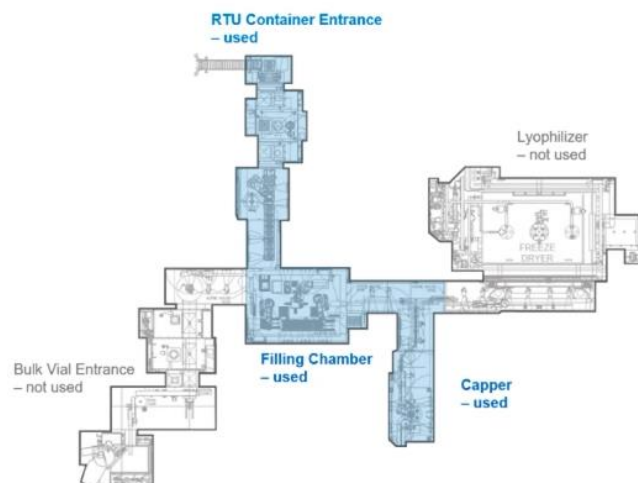


Supervisory PC

Operating Modes:

- A liquid fill in a bulk glass vial
- A lyo fill in a bulk glass vial
- A liquid fill in a RTU vial

Operation Mode (Liquid Fill in a RTU Vial):

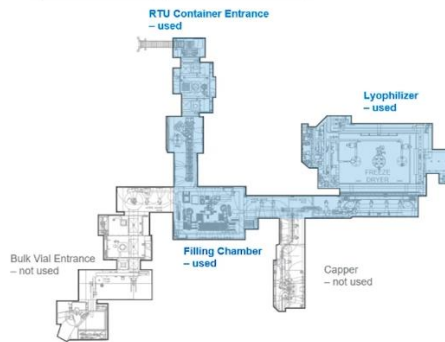


Supervisory PC

Operating Modes:

- A liquid fill in a bulk glass vial
- A lyo fill in a bulk glass vial
- A liquid fill in a RTU vial
- A lyo fill in a RTU vial

Operation Mode (Lyo Fill in a RTU Vial):

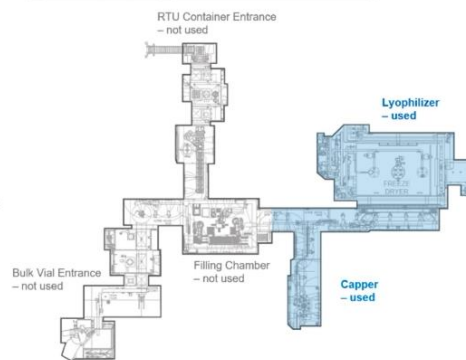


Supervisory PC

Operating Modes:

- A liquid fill in a bulk glass vial
- A lyo fill in a bulk glass vial
- A liquid fill in a RTU vial
- A lyo fill in a RTU vial
- A liquid fill in a RTU syringe or cartridge
- Unloading the lyophilizer

Operation Mode (Lyo Fill in a Bulk Glass Vial):

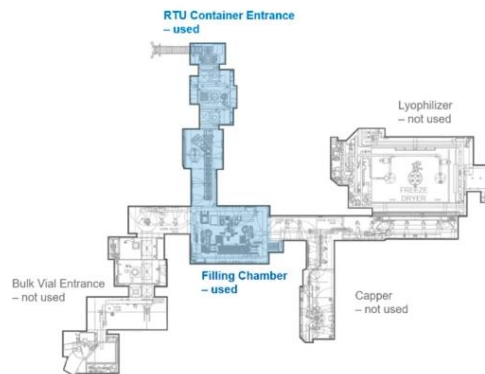


Supervisory PC

Operating Modes:

- A liquid fill in a bulk glass vial
- A lyo fill in a bulk glass vial
- A liquid fill in a RTU vial
- A lyo fill in a RTU vial
- A liquid fill in a RTU syringe or cartridge

Operation Mode (Liquid Fill in a RTU Syringe or Cartridge):



7.3 Filling components: technical specifications

<https://www.wmfts.com/siteassets/catalog/products/corporate/flexicon/table-top-filling-machines/7515/literature/wd-flexicon-pf7-fr.pdf>


Machine de remplissage péristaltique pour paillasse PF7

PF7

Flexicon Liquid Filling

CARACTÉRISTIQUES ET AVANTAGES

- Excellente reproductibilité, avec précision du remplissage supérieure à $\pm 0,5\%$
- Remplissage de précision de **microvolumes**
- Jusqu'à 200 et recettes 50 utilisateurs peuvent être sauvegardés en mémoire
- Interface claire et simple conçue pour les salles blanches
- Connectivité USB avec les balances et imprimantes pour une calibration sans erreur, et rapports de lot conformément aux bonnes pratiques de fabrication et aux exigences réglementaires
- Tête de pompe péristaltique à faible pulsation, optimisée pour les tubes **Accusil** de haute précision



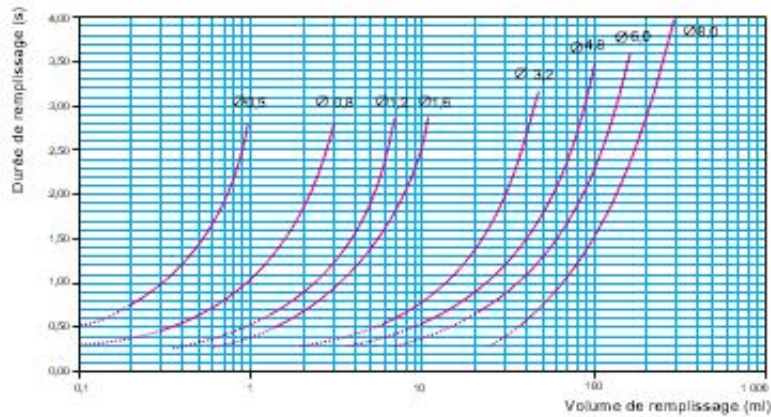
Flexicon...Filled with experience

CARACTÉRISTIQUES TECHNIQUES

PF7	
Boîtier	
Norme d'étanchéité	IP32/NEMA 2
Substances de nettoyage approuvées	Alcool à 70 % Désinfectant de surface contenant du formaldéhyde Concentration à 6 % de peroxyde d'hydrogène dans l'eau pour préparation injectable (EPI)
Contrôle	
Sélection des paramètres	Configuration manuelle ou assistée
Saisie de caractères	Alphanumérique par clavier
Contrôle à distance	Pédale, FlexFoot 15, 20 et 30
Communication numérique	
Imprimante thermique, balance	USB 2.0
Manipulation	
Types d'utilisateurs	Administrateur et opérateur
Sécurité	Connexion des utilisateurs protégée par code PIN
Restrictions	Tous les paramètres peuvent être restreints pour l'opérateur
Conformité réglementaire/normes	
Normes	CE, CE-UK , norme CAN/CSA C22.2 N° 61010-1 et norme UL 61010-1
Montage de tube	
Configuration de tubes à usage unique	Double alimentation avec verrouillages des tubes ou double Y

PERFORMANCE

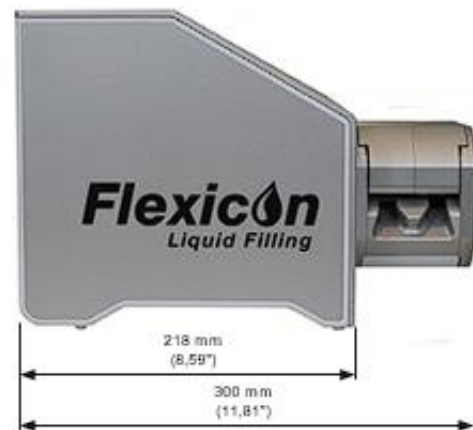
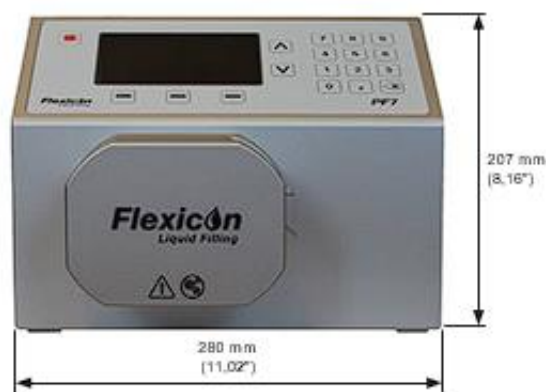
Diagramme de capacité PF7 (basé sur l'eau)



Performance

Volume de remplissage (ml)	Diamètre intérieur / mm (pouces)	Durée de remplissage (s)
0,2	0,5 (1/50)	0,75
1,0	0,8 (1/32)	1,10
1,4	1,2 (3/64)	0,75
1,8	1,6 (1/16)	0,60
8,0	3,2 (1/8)	0,70
15,0	4,8 (3/16)	0,80
24,0	6,0 (6/25)	0,80
50,0	8,0 (5/16)	0,80

DIMENSIONS



Panneau arrière

Repère	Description
1	USB 2.0
2	Connecteurs M12
3	Sélecteur de tension
4	Interrupteur d'alimentation
5	Fiche d'alimentation électrique

SPECIFICATIONS TECHNIQUES

Spécifications	Valeur
Température de fonctionnement	Utilisation en intérieur uniquement
Usage prévu	Environnements secs de production pour industries pharmaceutiques et de biotechnologie
Températures de fonctionnement	5 °C à 40 °C (41 °F à 104 °F)
Température de stockage	-40 °C à 70 °C (-40 °F à 158 °F)
Humidité (sans condensation)	80 % jusqu'à 31 °C (88 °F) et diminution <u>linéaire</u> jusqu'à 50 % à 40 °C (104 °F)
Altitude maximum	2 000 m (6 560 pieds)
Tension d'alimentation	100-120/200-240 V 50/60 Hz monophasés
Fluctuation maximum de tension	+/-10 % de tension nominale.
Consommation	140 VA
Courant de pleine charge	< 0,6 A à 230 V ; < 1,25 A à 115 V
Tension des fusibles	Céramique, 5x20 mm, 2,5 A, 250 V CA, Temporisation
Catégorie d'installation (catégorie de surtension)	II
Degré de pollution	2
Classe de protection	IP32 selon BS EN 60529
Niveau sonore	< 70 dB (A) à 1 m
Rapport de contrôle	30-400 tr/min (370:1)
Vitesse maximale	400 tr/min
Poids	10,9 kg
USB	2 x USB 2.0 haute vitesse type A, 500 mA par port

MATIÈRES

Pompe	
Composant	Matière
Clavier	Polycarbonate
Écran HM	Polycarbonate
Boîtier	Aluminium anodisé, EN AW-6060 T66
Parois latérales internes	ABS (Acrylonitrile butadiène styrène)
Parois latérales externes	Aluminium anodisé
Capots de connecteurs M12	HDPE
Pieds	Caoutchouc de silicone
Capots de prise Ethernet	Caoutchouc de silicone (blanc)
Capot de fiche USB 2.0	Caoutchouc de silicone (blanc)
Arbre de rotor	Acier inoxydable ANSI 304

Tête de pompe	
Composant	Matière
Boîtier de la tête de pompe	Aluminium anodisé EN AW-5754
Galets	Acier inoxydable ANSI 304
Cavalier de tube	Aluminium anodisé EN AW-5754
Doigts de cavalier de tube	Anodisation dure (évite les entailles et les rayures)
Verrouillage de tube	ABS (Acrylonitrile butadiène styrène)
Capot	ABS (Acrylonitrile butadiène styrène)
Interrupteurs de sécurité sur le cavalier de tube et le capot	Commutateur de proximité et aimant néodyme

RÉFÉRENCES DE COMMANDE

Codes produits des pompes	
Description	Référence
Unité de remplissage péristaltique PF7+	91-068-14X
PF7/QC14	91-060-14X
PF7/QC12	91-060-00X
Pédale PF7	88-010-040
Support de remplissage	81-100-200
Protocole IOQ PF7	74-156-443
Exécution IOQ PF7	74-156-444
FlexFeed 15	92-160-000
FlexFeed 20	92-170-000
FlexFeed 30	Contactez votre agence commerciale la plus proche
Pièces de rechange remplaçables sur site	
Ensemble de verrouillage pour tubes de tête de pompe QC14	87-068-000
Cavalier de tube QC14 pour pompe PF7+/PF7	87-068-047
Ensemble complet de verrouillage de tube QC14	87-068-500
Capot de rechange QC14	87-068-055

*La lettre X représente l'une des options de fiches indiquées dans le tableau ci-dessous :

Flexicon, tube catalysé platine - Codes de commande			
DI / mm (pouces)	Paroi / mm (pouces)	Longueur / m (pieds)	Nomenclature
0,5 (1/50)	1,6 (1/16)	10 (32,8)	84-103-005
		150 (492,1)	84-104-005
0,8 (1/32)	1,6 (1/16)	10 (32,8)	84-103-008
		150 (492,1)	84-104-008
1,2 (3/64)	1,6 (1/16)	10 (32,8)	84-103-012
		150 (492,1)	84-104-012
1,6 (1/16)	1,6 (1/16)	10 (32,8)	84-103-016
		150 (492,1)	84-104-016
3,2 (1/8)	1,8 (1/14)	10 (32,8)	84-103-032
		150 (492,1)	84-104-032
4,8 (3/16)	2,0 (10/127)	10 (32,8)	84-103-048
		125 (410,1)	84-104-048
6,0 (6/25)	2,1 (10/127)	10 (32,8)	84-103-060
		90 (290,3)	84-104-060
8,0 (5/16)	2,2 (2/23)	10 (32,8)	84-103-080
		85 (273,2)	84-104-080

Options de prises
<u>U</u> : Fiche Royaume-Uni
<u>E</u> : Fiche Europe
<u>A</u> : Fiche USA
<u>K</u> : Fiche Australie
<u>R</u> : Fiche Argentine
<u>C</u> : Fiche Suisse
<u>I</u> : Fiche Inde/Afrique du Sud
<u>B</u> : Fiche Brésil
<u>S</u> : Fiche Israël

Les informations contenues dans ce document sont réputées exactes. Watson-Marlow Corporation/AS décline toute responsabilité en cas d'éventuelle erreur et se réserve le droit de modifier ces informations sans préavis.
AVERTISSEMENT : Ces produits ne sont pas conçus pour des applications connectées à un patient et ne doivent pas être utilisés à cette fin. ~~Flexicon~~ et ~~Flexicon~~ sont des marques déposées.

Flexicon
Liquid Filling

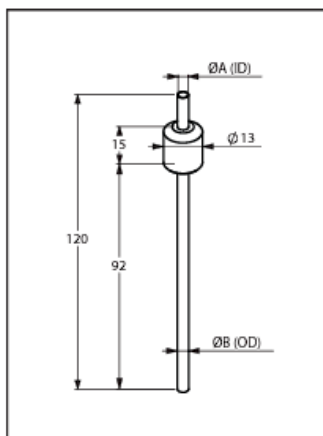
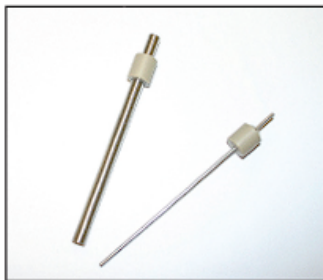
wmftg.com/000000
info@wmftg.com
+44 (0) 1326 370370

NEW

FILLING NOZZLE

STERILISABLE BY GAMMA IRRADIATION OR AUTOCLAVING

- Available in 7 different inner diameters from 0.6mm to 0.8mm
- Length allows for bottom-up filling applications
- Surface finish at tip of nozzle prevents dripping
- Meets FDA and USP Class VI requirements
- Supplied in Tyvek pouch ready for autoclaving or gamma irradiation
- Packed in ISO Class 8 clean room
- Full material traceability
- Batch specific quality certificate included
- Designed for fast change-over
- Fits standard nozzle holders on Flexicon Liquid Filling machines



SPECIFICATIONS				
Materials	AISI 316L and PEEK			
	A (ID)	B(OD)	Fill volumes from	Part number
Dimensions in mm	Ø0,6	Ø1,0	0.1 ml	30-040-006
	Ø1,0	Ø1,5	0.5 ml	30-040-010
	Ø1,6	Ø2,1	1.7 ml	30-040-016
	Ø3,2	Ø3,8	6.5 ml	30-040-032
	Ø4,5	Ø5,5	12.0 ml	30-040-045
	Ø6,0	Ø7,0	22.0 ml	30-040-060
	Ø8,0	Ø9,0	35.0 ml	30-040-080
Nozzle holder incl. arm for filling stand				30-700-053
Compliance	FDA USP, class VI FDA 21 CFR Part 177.2415, 2008 BSE/TSE free			
Quality certificate	Included			



Flexicon Liquid Filling cannot be held responsible for any errors or omissions. Subject to change without notice.

Rev. 23.10.2011

FIGURE 134 - FLEXICON FILLING NOZZLES TECHNICAL SPECIFICATIONS

10 Setting up the fluid path



If fluids hazardous to health are used applicable safety procedures must be followed to prevent injury.

10.1 Tube selection

Tubes must be selected according to the fluid and volume to be filled. Use the table shown below for choice of tubes according to minimum volume to be filled.

PF7/PF7+ has been designed to provide optimal performance when used in combination with Flexicon Accusil and Asepticsu. Use Accusil and Asepticsu to ensure the highest level of accuracy. Using tubing other than Accusil and Asepticsu can cause poor performance and undesirable results.

For best performance, choose Accusil tubing using "Table 3 - Tube sizes" below:

Table 3 - Tube sizes			
Volume (ml)	Filling Nozzle (mm i.d.)	Tubing (mm i.d.)	Y-Connector (mm i.d.)
0.01-0.50	0.6	0.5	1.2
>0.50 - 1.00	1.0	0.8	1.2
>1.00 - 1.70	1.0	1.2	1.8
>1.70 - 7.00	1.6	1.6	1.8
>7.00 - 12.0	3.2	3.2	3.6
>12.0 - 22.0	4.5	4.8	4.8
>22.0 - 35.0	6.0	6.0	4.8
> 35.0	8.0*	8.0	7.5

* use non-return valve

FIGURE 135 - FLUID PATH COMPONENTS ACCORDING TO PF7 MANUAL

<https://www.wmfts.com/siteassets/catalog/products/corporate/flexicon/table-top-filling-machines/7515/literature/wd-pf7plus-en-02.pdf>

7.4 E-beam tunnel and pulsed-light sanitization

As mentioned earlier in this project, one of the primary causes of bioburden in an aseptic process is the operators.

Although NTT¹¹ is working to address this issue, manual unpacking can result in turbulent airflow if not done properly. This increases the risk of cross-contamination. There are several solutions to this problem. The first is to automate the tub transfer process, which ensures smooth and continuous movements. Another solution, developed in this section, is an additional active decontamination step before the filling line. As this step can affect the throughput, it should be analysed.



FIGURE 136 - BEST PRACTICE FOR RTU HANDLING

[Enhancing sterility assurance in isolator-based aseptic filling - YouTube](#)

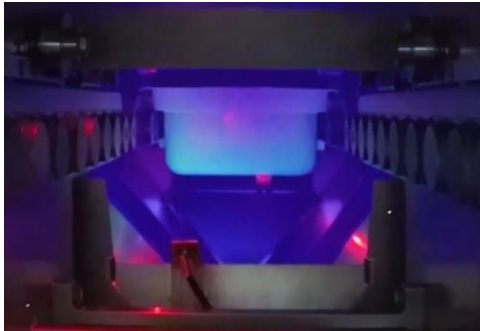
These active sanitizations are often carried out by radiation. Why? Because they are cold and fast methods, independent of chemicals. Not only can they be very effective (up to 6 log reduction in bioburden), but they are also flexible. They can sterilise products of varying shape, size, thickness and density at any pressure or temperature. As far as quality control is concerned, it is quite easy as only the time and intensity of exposure need to be controlled. In this section, two methods will be discussed. Namely electron-beam and pulsed-light sanitization.

Supply RTU in bulk: E-beam tunnel

[ebeam – innovative electron beam technology | SKAN](#)

Electron-beam tunnel, mainly commercialised by SKAN, allows a bulk supply of RTU. Indeed, it can reach **up to 6 tubs/min** making it especially suitable for high-capacity filling lines. How does it work?

¹¹ No touch transfer.



An electron beam achieves a 6 log reduction in bioburden through a radiation energy of 25 kGy. Hence, both surface and inside of the tub are sterilised through death of microorganism.

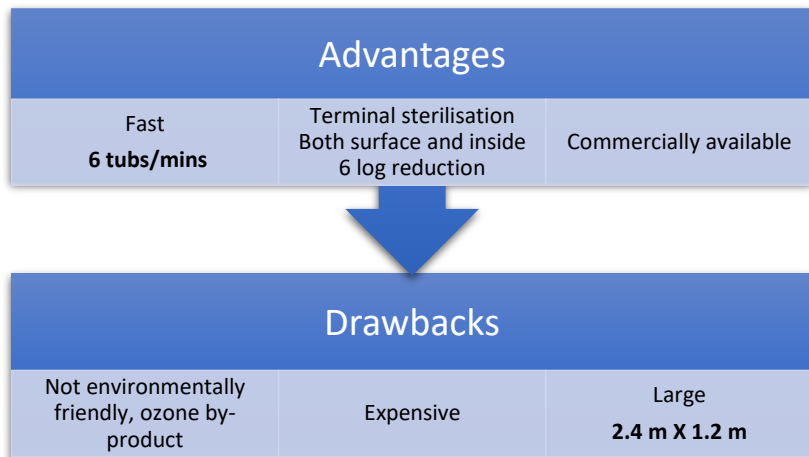
The tunnel can be located in a **Grade C or D** environment, **occupying 2.4 m X 1.2 m**.

FIGURE 137 – INSIDE OF SKAN E-BEAM TUNNEL [ebeam – innovative electron beam technology | SKAN](#)

[ebeam – innovative electron beam technology | SKAN](#)



FIGURE 138 - SKAN E-BEAM TUNNEL



[ebeam – innovative electron beam technology | SKAN](#)

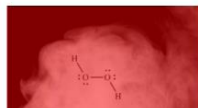
[Enhancing sterility assurance in isolator-based aseptic filling - YouTube](#)

Pulsed-light sanitization

This process, relatively new, have been developed by Steriline to compensate for the disadvantages of the already existing methods illustrated in *Figure 139*.

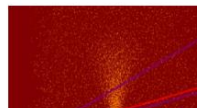
Addressing NTT Weaknesses

- Wanted a **positive sanitization** process for RTUs
- We considered:
 - Gaseous sterilants
 - Electron beam
 - Flash Steam
- Settled on a new technology: **Pulsed Light Technology**



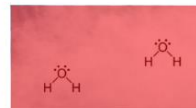
Problem:

- too slow – required aeration to remove sanitizers
- incompatible with some materials
- leaves residue



Problem:

- Large
- Very expensive to purchase & operate
- non-environmental



Problem:

- Not sufficiently mature/advanced commercially

7

FIGURE 139 - COMPARISON BETWEEN ACTIVE SANITIZATION ROCESS FOR RTU

[Pulsed Light Decontamination Slide Deck \(2022 PDA Conference\) - Berkshire Sterile Manufacturing](#)

Placed between the NTT and the filling line, the Steriline pulsed light process uses a Staübli robotic arm to process a maximum of **2 tubs/min** in a working space of **1.928 x 1.205 m**. This means that each surface of the tank can be exposed to the lamps and achieve a minimum reduction of 4 logs. The latter value is strongly affected by the complexity of the surface to be sanitised as shown in *Figure 140*. For pre-sterilised (RTU) tubs, a 6 log reduction in bioburden is often achieved.

[SYSTÈME ROBOTISÉ DE DÉCONTAMINATION DES TUBES RTDS2 - Steriline](#)

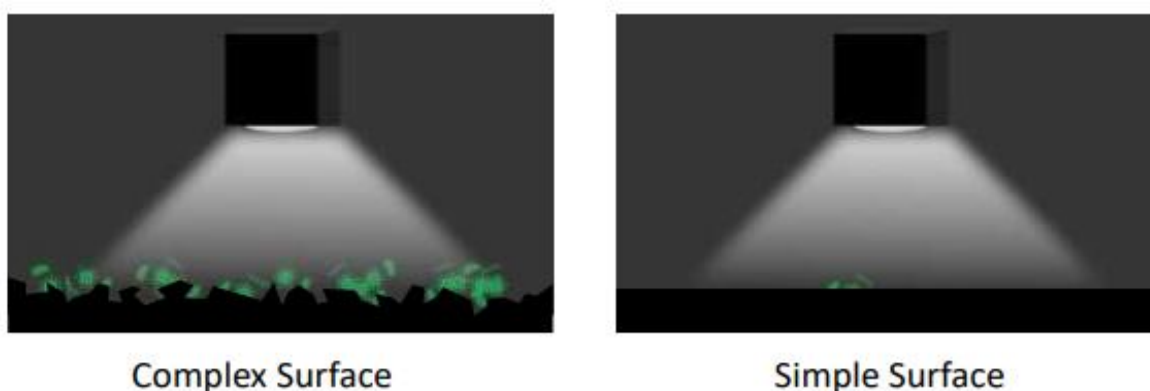
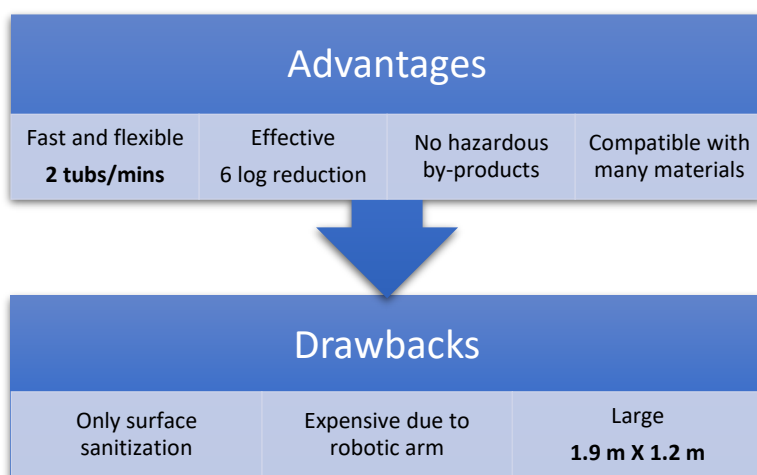


FIGURE 140 - PULSED-LIGHT SANITIZATION EFFICIENCY DEPENDING ON SURFACE COMPLEXITY

[2022 PDA Pharmaceutical Microbiology Conference \(berkshiresterilemanufacturing.com\)](#)



[Enhancing sterility assurance in isolator-based aseptic filling - YouTube](#)



FIGURE 141 - STERILINE RTDS2 PULSED-LIGHT STERILISATION STATION

[SYSTÈME ROBOTISÉ DE DÉCONTAMINATION DES TUBES RTDS2 - Steriline](#)

7.5 H2O2 compatibility charts

Material	Compatibility 10% H2O2	Compatibility 30% H2O2	Compatibility 50% H2O2	Compatibility 100% H2O2
304 stainless steel	B2 - Good	B2 - Good	B2 - Good	B2 - Good
316 stainless steel	B - Good	B - Good	A2 - Excellent	A2 - Excellent
ABS plastic	A - Excellent	N/A	N/A	A - Excellent
Acetal (Delrin®)	D - Severe Effect	D - Severe Effect	D - Severe Effect	D - Severe Effect
Aluminum	A - Excellent	A - Excellent	A - Excellent	A - Excellent
Brass	N/A	N/A	N/A	D - Severe Effect
Bronze	B1 - Good	B1 - Good	B1 - Good	B1 - Good
Buna N (Nitrile)	D - Severe Effect	D - Severe Effect	D - Severe Effect	D - Severe Effect
Carbon graphite	C - Fair	C - Fair	C - Fair	C - Fair
Carpenter 20	C - Fair	B - Good	B - Good	D - Severe Effect
Cast iron	C - Fair	B - Good	N/A	B - Good
Ceramic Al2O3	N/A	N/A	N/A	A - Excellent
Ceramic magnet	A - Excellent	A - Excellent	A - Excellent	A - Excellent
Copper	D - Severe Effect	D - Severe Effect	D - Severe Effect	D - Severe Effect
CPVC	A - Excellent	A - Excellent	A - Excellent	A - Excellent
EPDM	A - Excellent	B - Good	B - Good	D - Severe Effect
Epoxy	C1 - Fair	B - Good	N/A	A - Excellent
Hastelloy-C®	A - Excellent	A - Excellent	A - Excellent	A - Excellent
Hypalon®	D - Severe Effect	D - Severe Effect	D - Severe Effect	D - Severe Effect
Hytrel®	N/A	N/A	N/A	N/A
Kel-F®	A - Excellent	B - Good	A - Excellent	B - Good
LDPE	A - Excellent	C2 - Fair	C2 - Fair	C2 - Fair
Natural rubber	B - Good	C - Fair	C - Fair	C - Fair
Neoprene	D - Severe Effect	D - Severe Effect	D - Severe Effect	D - Severe Effect
NORYL®	A2 - Excellent	A2 - Excellent	N/A	A - Excellent
Nylon	C1 - Fair	D - Severe Effect	D - Severe Effect	D - Severe Effect
Polycarbonate	A2 - Excellent	A2 - Excellent	A2 - Excellent	A - Excellent
Polypropylene	A - Excellent	B1 - Good	B1 - Good	B1 - Good
PPS (Ryton®)	A - Excellent	A1 - Excellent	N/A	C - Fair
PTFE (Teflon®)	A - Excellent	A - Excellent	A - Excellent	A - Excellent
PVC	A1 - Excellent	A1 - Excellent	A1 - Excellent	A - Excellent
PVDF (Kynar®)	A - Excellent	A - Excellent	A1 - Excellent	A1 - Excellent
Silicone	A - Excellent	B - Good	B - Good	B - Good
Titanium	A - Excellent	B1 - Good	A - Excellent	B - Good
Tygon®	B - Good	B - Good	B - Good	B - Good
Viton®	A - Excellent	A - Excellent	A - Excellent	A - Excellent

Ratings -- Chemical Effect

A. **Excellent.**

B. **Good** -- Minor Effect, slight corrosion or discoloration.

C. **Fair** -- Moderate Effect, not recommended for continuous use. Softening, loss of strength, swelling may occur.

D. **Severe Effect**, not recommended for **ANY** use.

TABLE 11 – MATERIAL COMPATIBILITY CHART WITH H2O2 [100]

Hydrogen Peroxide Material Compatibility Chart

All wetted surfaces should be made of materials that are compatible with hydrogen peroxide. The wetted area or surface of a part, component, vessel or piping is a surface which is in permanent contact with or is permanently exposed to the process fluid (liquid or gas).

Less than 8% concentration H₂O₂ is considered a non-hazardous substance. Typically encountered versions are baking soda-peroxide toothpaste (0.5%), contact lens sterilizer (2%), over-the-counter drug store Hydrogen Peroxide (3%), liquid detergent non-chlorine bleach (5%) and hair bleach (7.5%).

At 8% to 28% H₂O₂ is rated as a Class 1 Oxidizer. At these concentrations H₂O₂ is usually encountered as a swimming pool chemical used for pool shock treatments.

In the range of 28.1% to 52% concentrations, H₂O₂ is rated as a Class 2 Oxidizer, a Corrosive and a Class 1 Unstable (reactive) substance. At these concentrations, H₂O₂ is considered industrial strength grade.

Concentrations from 52.1% to 91% are rated as Class 3 Oxidizers, Corrosive and Class 3 Unstable (reactive) substances. H₂O₂ at these concentrations are used for specialty chemical processes. At concentrations above 70%, H₂O₂ is usually designated as high-test peroxide (HTP).

Concentrations of H₂O₂ greater than 91% are currently used as rocket propellant. At these concentrations, H₂O₂ is rated as a Class 4 Oxidizer, Corrosive and a Class 3 Unstable (reactive) substance.

Material	Compatibility 10% H ₂ O ₂	Compatibility 30% H ₂ O ₂	Compatibility 50% H ₂ O ₂	Compatibility 100% H ₂ O ₂ (HTC)
<i>Chemical resistance data is based on 72° F (22° C) unless otherwise noted</i>				
<i>A - Suitable</i>				
<i>B - Good, minor effect, slight corrosion or discoloration</i>				
<i>F - Fair, moderate effect, not recommended for continuous use; softening, loss of strength, and/or swelling may occur</i>				
<i>X - Do Not Use - severe effect, not recommended for ANY use</i>				
<i>NA - Information Not Available</i>				
304 stainless steel	B ¹	B ¹	B ¹	B ¹
316 stainless steel	B	B	A ¹	A ¹
416 stainless steel	B	B	F	X
440C stainless steel	B	B	A	X
ABS plastic	A	A	A	A

It is the sole responsibility of the system designer and user to select products suitable for their specific application requirements and to ensure proper installation, operation, and maintenance of these products. Material compatibility, product ratings and application details should be considered in the selection. Improper selection or use of products described herein can cause personal injury or product damage. In applications where exposure to harmful chemicals is frequent, of long duration or in high concentrations, additional testing is recommended.



End your search, simplify your supply chain
ISO 9001:2015 Certified Companies

4091 S. Elliot St., Englewood, CO 80110-4396
Phone 303-781-8486 | Fax 303-761-7939
ismedspec.com

© Copyright 2020 IS MED Specialties

FIGURE 142 - MATERIALS COMPATIBLE WITH H₂O₂ CHEMICAL AGENT

<https://www.industrialspec.com/images/files/hydrogen-peroxide-material-compatibility-chart-from-ism.pdf>

Hydrogen Peroxide Material Compatibility Chart

ver 09-Jul-2020

Material	Compatibility 10% H ₂ O ₂	Compatibility 30% H ₂ O ₂	Compatibility 50% H ₂ O ₂	Compatibility 100% H ₂ O ₂ (HTC)
<i>Chemical resistance data is based on 72° F (22° C) unless otherwise noted</i>				
<i>A - Suitable</i>				
<i>B - Good, minor effect, slight corrosion or discoloration</i>				
<i>F - Fair, moderate effect, not recommended for continuous use;</i>				
<i>softening, loss of strength, and/or swelling may occur</i>				
<i>X - Do Not Use - severe effect, not recommended for ANY use</i>				
<i>NA - Information Not Available</i>				
Acetal (Delrin®)	X	X	X	X
Acrylic (PMMA)	B	F	NA	X
Alloy 20 (Carpenter 20)	F	B	B	X
Aluminum	A	A	A	A
Brass	X	X	X	X
Bronze	B	B	B	B
Buna N (Nitrile)	X	X	X	X
Carbon graphite	F	F	F	F
Carbon steel	X	X	X	X
Cast iron	F	X	X	X
Ceramic Al ₂ O ₃	A	A	A	A
Ceramic magnet	A	A	A	A
Copper	X	X	X	X
CPVC	A	A	A	A
EPDM	A	B	B	X
Epoxy (epoxide polymers)	F	B	B	X
FKM (fluoroelastomers, Viton®)	A	A	A	A
Hastelloy-C®	A	A	A	A
HDPE	A	A	A	X
Hypalon®	X	X	X	X
Hytrel® (polyester elastomer)	X	X	X	X
LDPE	A	F ¹	F ¹	F ¹
Natural rubber	B	F	F	F
Neoprene	X	X	X	X
NORYL®	A ¹	A ¹	A	A

It is the sole responsibility of the system designer and user to select products suitable for their specific application requirements and to ensure proper installation, operation, and maintenance of these products. Material compatibility, product ratings and application details should be considered in the selection. Improper selection or use of products described herein can cause personal injury or product damage. In applications where exposure to harmful chemicals is frequent, of long duration or in high concentrations, additional testing is recommended.



End your search, simplify your supply chain
ISO 9001:2015 Certified Companies

4091 S. Elliot St., Englewood, CO 80110-4396
Phone 303-781-8486 | Fax 303-761-7939
ismedspec.com

© Copyright 2020 IS MED Specialties

FIGURE 143 - MATERIALS COMPATIBLE WITH H₂O₂ CHEMICAL AGENT

<https://www.industrialspec.com/images/files/hydrogen-peroxide-material-compatibility-chart-from-ism.pdf>

Hydrogen Peroxide Material Compatibility Chart

ver 09-Jul-2020

Material	Compatibility 10% H ₂ O ₂	Compatibility 30% H ₂ O ₂	Compatibility 50% H ₂ O ₂	Compatibility 100% H ₂ O ₂ (HTC)
<i>Chemical resistance data is based on 72° F (22° C) unless otherwise noted</i>				
A - Suitable				
B - Good, minor effect, slight corrosion or discoloration				
F - Fair, moderate effect, not recommended for continuous use;				
softening, loss of strength, and/or swelling may occur				
X - Do Not Use - severe effect, not recommended for ANY use				
NA - Information Not Available				
Nylon (polyamides)	F	X	X	X
PCTFE (Kel-F® and Neoflon®)	A ¹	A ¹	A ¹	X
PFA (perfluoroalkoxy alkanes)	A	A	A	A
Polycarbonate	A ¹	A ¹	A ¹	A
Polypropylene	A	B	B	B
PP-363 (plasticized vinyl) ²	A	A	A	X
PPS (Ryton®)	A	A	F	F
PTFE (Garlock Glyon® 3500) ²	A	A	A	X
PTFE (Teflon®), virgin ²	A	A	A	A
PVC	A	A	A	A
PVDF (Hylar®)	A ¹	A ¹	X	X
PVDF (Kynar®)	A	A	A	A
PVDF (Solef®)	A ¹	A ¹	X	X
Silicone	A	B	B	B
SPR (styrene butadiene rubber)	X	X	X	X
Thiokol™ (polysulfide polymers)	X	X	X	X
Titanium ³	A	B	B	B
TPE (thermoplastic elastomers)	X	X	X	X
TPU (thermoplastic polyurethanes)	X	X	X	X
Tygon®	B	B	B	B
Tungsten carbide	X	X	X	X
Viton® A ²	A	A	A	A

It is the sole responsibility of the system designer and user to select products suitable for their specific application requirements and to ensure proper installation, operation, and maintenance of these products. Material compatibility, product ratings and application details should be considered in the selection. Improper selection or use of products described herein can cause personal injury or product damage. In applications where exposure to harmful chemicals is frequent, of long duration or in high concentrations, additional testing is recommended.



End your search, simplify your supply chain
ISO 9001:2015 Certified Companies

4091 S. Eliot St., Englewood, CO 80110-4396
Phone 303-781-8466 | Fax 303-761-7939
ismedspec.com

© Copyright 2020 IS MED Specialties

FIGURE 144 - MATERIALS COMPATIBLE WITH H₂O₂ CHEMICAL AGENT

<https://www.industrialspec.com/images/files/hydrogen-peroxide-material-compatibility-chart-from-ism.pdf>

- ✓ Commercial vacuum cleaners or similar unapproved cleaning equipment
- ✓ Wood products
- ✓ Bare aluminium
- ✓ Powders
- ✓ Cardboard
- ✓ Gum
- ✓ Unapproved plastic
- ✓ Books and book bags
- ✓ Open cell foam materials
- ✓ Plastic bags and bubble wrap
- ✓ Grease, oils and lubricants
- ✓ Leather
- ✓ Velcro
- ✓ Aerosol
- ✓ Food and drink
- ✓ Makeup
- ✓ Regular writing tools, paper or notebooks
- ✓ Silica gel
- ✓ Unapproved tape or adhesive

FIGURE 145 - MATERIALS PROHIBITED IN A GRADE A ASEPTIC ENVIRONMENT

[Materials | Modular Cleanrooms By Total Clean Air \(modular-cleanroom.net\)](https://modular-cleanroom.net)

7.6 Pressure cascade: guidance values justification

the US Aseptic Processing Guidance which requires a static pressure gradient of 12.5 Pa for both “controlled and critical” zone. They are also supported by **the European Community (EC) GMPs which gives a range of 10 to 15 pascals** ([Guidance for Industry \(fda.gov\)](#) pg 7, [Guideline on Sterile Drug Products Produced by Aseptic Processing, Center for Drugs and Biologics, Food and Drug Administration, Rockville, MD, June 1987.](#))

Hence a gradient of 12.5 Pa allows to satisfy both. Although those values seem magical, they are supported by several studies which have shown that a pressure gradient of 7.5 Pa to 12.5 Pa is effective in reducing contaminant infiltration. Beyond this threshold, no significant improvement was achieved and the increased energy cost was not worth it. When it comes to sterile compounding and aseptic filling, one can consider pressures of $+ \geq 37$ Pa in the working area and $+ \geq 25$ Pa in the adjacent chamber. In any cases, it is recommended to have a minimum overpressure of 5Pa between clean areas. Between a clean area and an unclean area, the recommended pressure is about 13 Pa.

[Air flow design: using the cascade approach \(cleanroomtechnology.com\)](#)

[What operating pressure to take during aseptic processing - European Pharmaceutical Manufacturer](#)

[USP 800 Cleanroom Design - Negative Pressure Changes and Requirements \(gotopac.com\)](#)

In general, for Grade A media, USP¹² 800 standard considers an area to be acceptable if it is under negative pressures between 0.01 and 0.03 in WC (2.5 Pa and 7.5 Pa). Indeed, too great a differential not only increases the risk of contamination from cracks, gaps and inlets, but also drastically increases the energy consumption from the air handling system.

¹² United States Pharmacopeia. They establish standards for medicines, food ingredient and drugs. [What Is a USP Standard? | Quality Matters | U.S. Pharmacopeia Blog](#)

7.7 ISO 8536-1:2011: Infusion vials

[ISO - ISO 8536-1:2011 - Infusion equipment for medical use — Part 1: Infusion glass bottles](#)

Interchangeability of infusion glass bottles is ensured through the standard ISO 8536-1:2011. This norm specifies the dimensions, performance and requirements for such bottles. However, it only applies to bib-reusable one.

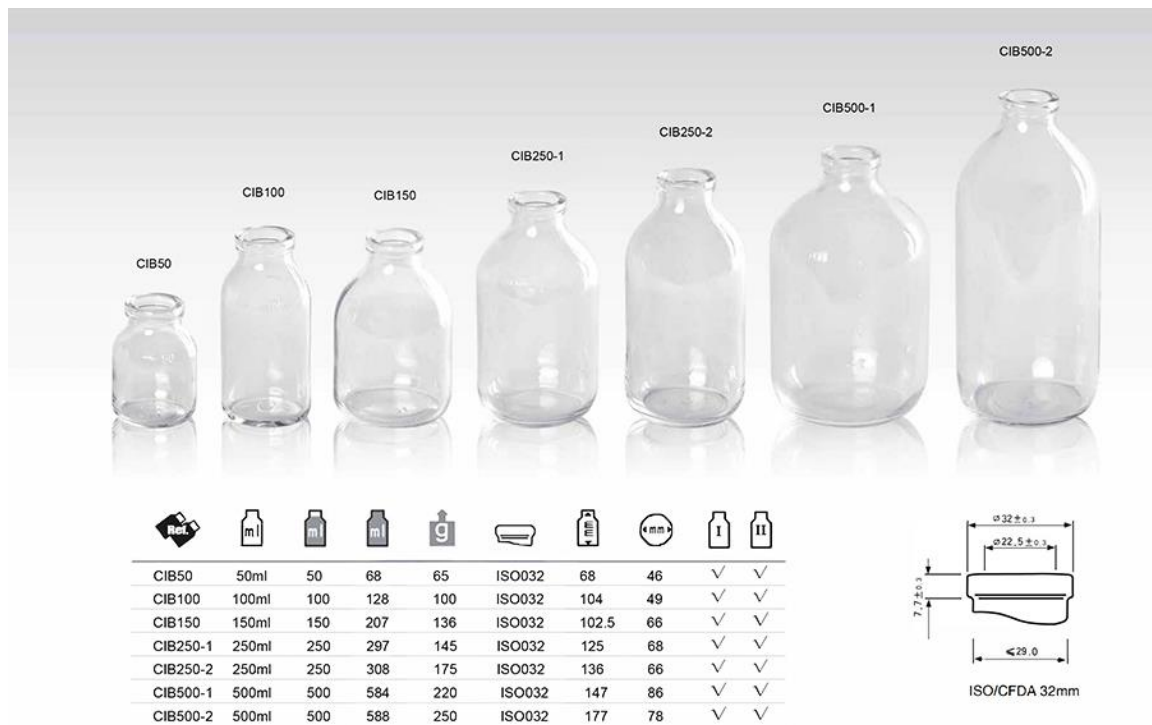


FIGURE 146 - ISO 8536-1 :2011 COLOURLESS COMPLIANT VIALS

[50ml to 500ml Infusion Glass Bottle for Pharmaceutical \(mpbottle.com\)](http://mpbottle.com)

7.8 ISO 8362-4:2011: Injection vials moulded from borosilicate



FIGURE 147 – ISO 8362-4:2011 COLOURLESS COMPLIANT VIALS

[Injection vials | Trading house "TD-PACK" LLC](#)

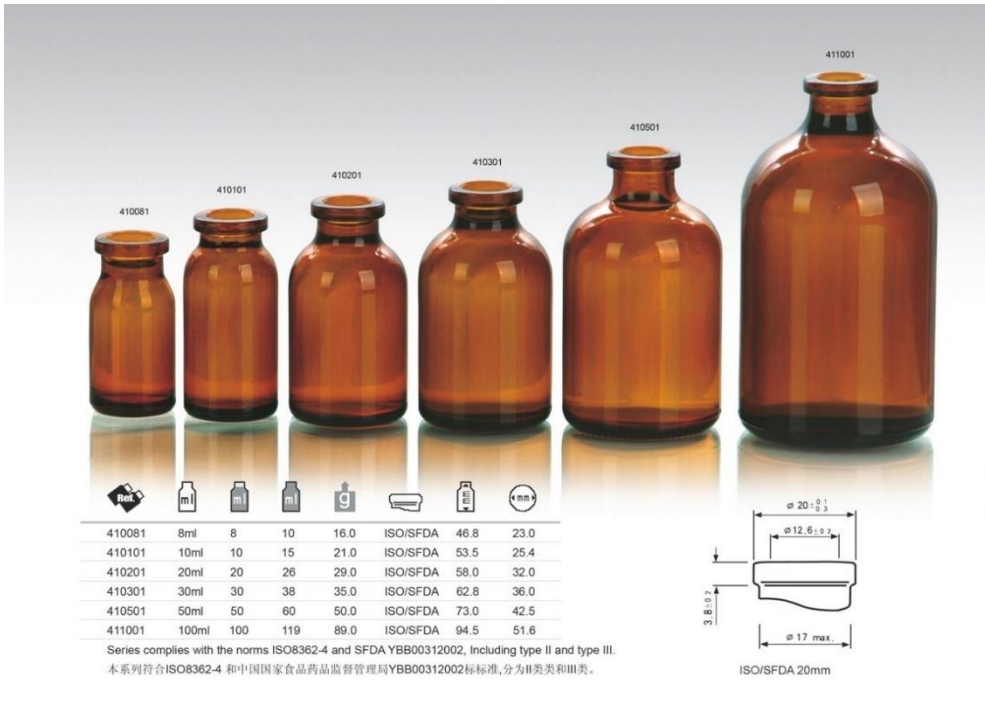


FIGURE 148 – ISO 8362-4:2011 AMBERED COMPLIANT VIALS

7.9 Debagging

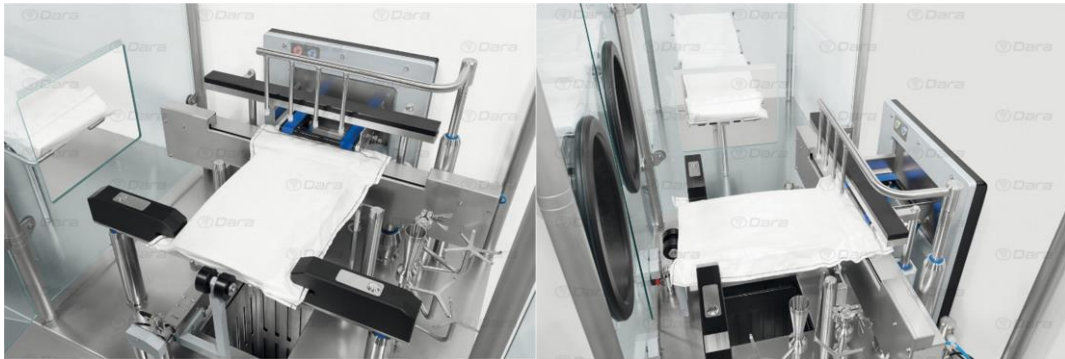


FIGURE 149 - AUTOMATED DE-BAGGING IN DARA RABS DB/A

[COMBO PHILL Filling line for ready-to-use vials and prefilled syringes \(comecer.com\)](http://comecer.com)

An alternative unpacking solution can be found at COMECER, which offers a manual operation. The operator himself has to open the mousehole of the RABS, insert the bag and then close it. Once this is done, the technician puts his arms in the gloves to interact with the RABS inside. With the push of a button, the bag is pressed to prevent unwanted movement and then cut by the front machine. Then the platform on which it rests tilts and slides the tub to the following mousehole. The mousehole then open automatically and the operator pushes the bag into the isolator.



FIGURE 150 - MANUAL DE-BAGGING EXAMPLE

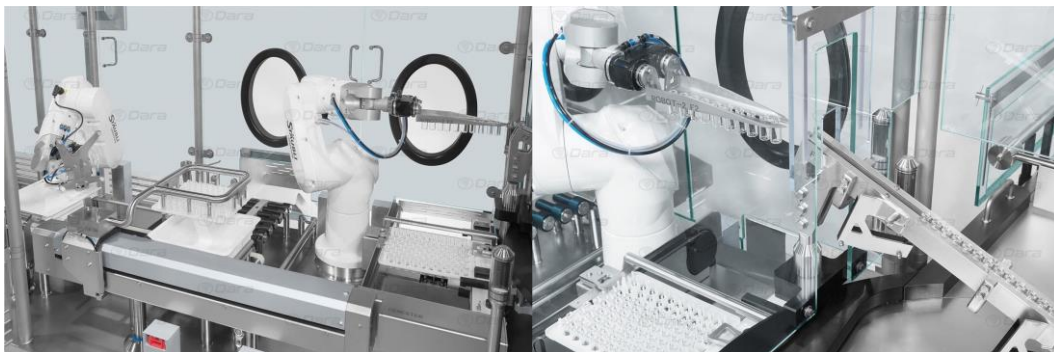
7.10 Denesting

his first example of Figure 152 features two different posts. The first one is responsible for removing the nest from its tub and discarding both. It uses a rail and a moving effector with suction cups for prehension. Once the nest is lift, it is placed under the second post, a robotic arm that picks a line of vial by squeezing their neck, and the place them on a gravity track.



FIGURE 151 - DARA VIAL DE-NESTING EXAMPLE 2

[DN / RN Vials. Loading and unloading for vials in nest - Dara \(dara-pharma.com\)](https://dara-pharma.com)



[DN / RN Vials. Loading and unloading for vials in nest - Dara \(dara-pharma.com\)](https://dara-pharma.com)

FIGURE 152 - DARA VIAL DENESTING DN/E

Mid-scale / high scale:

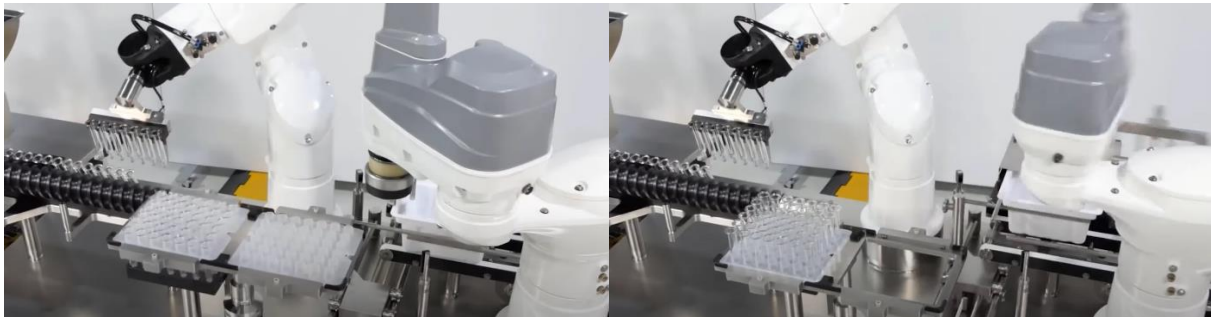
Advantages:

Good cadence, high level of automation

Drawbacks:

Cost of the installation, glass-to-glass contact due to the gravity rail, vials only, hard to remove vial if defects

In this next example of DARA denesting process, **containers can be multiformat**. A first robot places the nest on a rotating platform which places it on top of a mould. When the platform is lowered, the syringes are lifted out of the nest and can be picked up by line by the second robot. They are then placed on an endless-screw which ensures that no glass-to-glass contact is possible. While the first robot places the syringes, the second robot picks up the previous empty nest and places it back into the tub so that they exit the isolator together. **Output of 12000 UPH.**



[Dara Pharmaceutical Equipment in Virtual Pharma Expo 2022 together NJM packaging - YouTube](#)

Advantages:

No glass-to-glass contact

Drawbacks:

Expensive as two robots are needed

The Dara DN/N has a maximum output of 9000 UPH. It uses a conveyor to remove the vials from their housing. Two lines at a time are then picked by an x-z axis robot. The nesting stage involves placing the vials on a gravity track where their movement is initiated by a conveyor belt.



Advantages:

Simplicity, cheap

Drawbacks:

glass-to-glass contact due to the gravity rail, vials only, hard to remove vial from track if defect, upper prehension might disturb airflow.

7.11 Kinematic analysis of the box transport mechanism

Geometry and positions

In order to find an 8-linkage box transport mechanism with a stroke of 216 mm, a kinematics analysis of the system is first performed according to the scientific paper [Box Transport Mechanism \(mekanizmalar.com\)](#). First step is to impose denominations and axis which is performed on Figure 153.

First step is to impose denominations and axis which is performed on Figure 72.

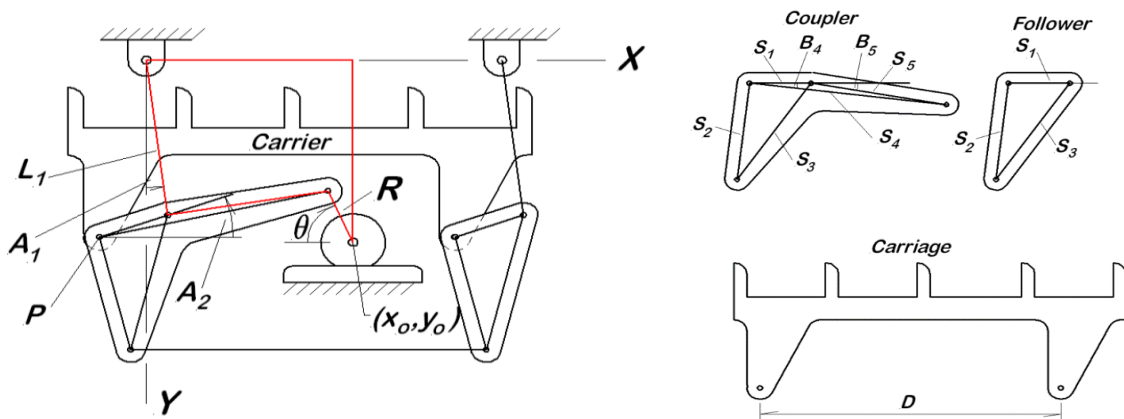


FIGURE 153 - 8-LINKAGE TRANSFER BOX: GEOMETRY AND DENOMINATIONS

[Box Transport Mechanism \(mekanizmalar.com\)](#)

To prevent any tilting of the carrier blades during the cycle, it is required for the follower triangle to be isometric to the coupler triangle S1-S2-S3. This assumption will thus be considered for the following calculations. Once this condition is met, it turns out that the size of this triangle have no impact on the kinematics of the system. It is therefore sufficient to limit the mathematical analysis to the triangle S1-S4-S5 for which following equations can be written:

$$S_1 + S_5 \cos(B_5) = S_4 \cos(B_4) \quad (31)$$

$$S_5 \sin(B_5) = S_4 \sin(B_4) \quad (32)$$

Recall the main goal, computing the motion of the carriage. As all its point follow the same displacement than point P, a kinematic study of the latter is sufficient. Using simple trigonometry, every point of the carriers can be described by the following equations:

$$x_p = L_1 \sin(A_1) - S_1 \cos(A_2) \quad (33)$$

$$y_p = L_1 \cos(A_1) + S_1 \sin(A_2) \quad (34)$$

Currently, the dependence of A_1 and A_2 on θ is not known. However, it can be easily obtained through a kinematic analysis of the red loop depicted in Figure 72. This loop is in fact a simple 4-bar crank rocker mechanism, governed by the following equations:

$$L_1 \sin(A_1) + S_5 \cos(A_2 - B_5) + R \cos(\theta) - x_0 = 0 \quad (35)$$

$$L_1 \cos(A_1) - S_5 \sin(A_2 - B_5) + R \sin(\theta) - y_0 = 0 \quad (36)$$

$$R = L_1 \left(\frac{3.14321}{180} \right) \sin^{-1} \left(\frac{Stroke}{2L_1} \right) \quad (37)$$

Velocity analysis

Velocity analysis is performed by differentiating equations **Error! Reference source not found.** and **Error! Reference source not found.** with respect to time.

$$\begin{bmatrix} L_1 \cos(A_1) & -S_5 \sin(A_2 - B_5) \\ -L_1 \sin(A_1) & -S_5 \cos(A_2 - B_5) \end{bmatrix} \begin{Bmatrix} \dot{A}_1 \\ \dot{A}_2 \end{Bmatrix} = \dot{\theta} R \begin{Bmatrix} \sin(\theta) \\ -\cos(\theta) \end{Bmatrix} \quad (38)$$

The left matrix is defined as the Jacobian matrix:

$$[J] = \begin{bmatrix} L_1 \cos(A_1) & -S_5 \sin(A_2 - B_5) \\ -L_1 \sin(A_1) & -S_5 \cos(A_2 - B_5) \end{bmatrix} \quad (39)$$

It can be inverted to find the angle rates equations:

$$\begin{Bmatrix} \dot{A}_1 \\ \dot{A}_2 \end{Bmatrix} = \dot{\theta} R \begin{bmatrix} L_1 \cos(A_1) & -S_5 \sin(A_2 - B_5) \\ -L_1 \sin(A_1) & -S_5 \cos(A_2 - B_5) \end{bmatrix}^{-1} \begin{Bmatrix} \sin(\theta) \\ -\cos(\theta) \end{Bmatrix} \quad (40)$$

By rewriting these equations to form ratios $\dot{A}_1 / \dot{\theta}$ and $\dot{A}_2 / \dot{\theta}$, it is possible to define dimensionless velocity coefficients $K_{A1}(\theta)$ and $K_{A2}(\theta)$:

$$\begin{Bmatrix} K_{A1} \\ K_{A2} \end{Bmatrix} = \begin{Bmatrix} \dot{A}_1 / \dot{\theta} \\ \dot{A}_2 / \dot{\theta} \end{Bmatrix} = R \begin{bmatrix} L_1 \cos(A_1) & -S_5 \sin(A_2 - B_5) \\ -L_1 \sin(A_1) & -S_5 \cos(A_2 - B_5) \end{bmatrix}^{-1} \begin{Bmatrix} \sin(\theta) \\ -\cos(\theta) \end{Bmatrix} \quad (41)$$

Knowledge of those functions lead to direct calculation of the angle rates through:

$$\dot{A}_1 = \dot{\theta} \cdot K_{A1}(\theta) \quad (42)$$

$$\dot{A}_2 = \dot{\theta} \cdot K_{A2}(\theta) \quad (43)$$

These last two equations directly show the strong dependence of the angle rates on the crank angle. Conclusions therefore that even at constant crank speed, they are very variable. Regarding velocity speed coefficients, they simply correspond to the derivative of the angle rates with respect to the crank speed.

$$K_{A1}(\theta) = \frac{dA_1}{d\theta} \quad (44)$$

$$K_{A2}(\theta) = \frac{dA_2}{d\theta} \quad (45)$$

Acceleration analysis

A fast way to perform an acceleration analysis is to start by derivation of equations **Error! Reference source not found.** and **Error! Reference source not found.** to introduce the so-called *velocity coefficient derivatives*:

$$L_{A1}(\theta) = \frac{dK_{A1}(\theta)}{d\theta} = \frac{d^2 A_1}{d\theta^2} \quad (46)$$

$$L_{A2}(\theta) = \frac{dK_{A2}(\theta)}{d\theta} = \frac{d^2 A_2}{d\theta^2} \quad (47)$$

Note: velocity coefficients derivatives should not be considered as **acceleration coefficients**. Indeed, in the following differentiation of equations **Error! Reference source not found.**(8) and (9), they never multiply the acceleration.

$$\ddot{A}_1 = \frac{d}{dt} (\dot{\theta} K_{A1}) = \ddot{\theta} K_{A1} + \dot{\theta}^2 L_{A1} \quad (48)$$

$$\ddot{A}_2 = \frac{d}{dt} (\dot{\theta} K_{A2}) = \ddot{\theta} K_{A2} + \dot{\theta}^2 L_{A2} \quad (49)$$

Goal is now to find a computable expression for the velocity coefficient derivative. To this end, equation **Error! Reference source not found.** can be rewritten:

$$\begin{bmatrix} L_1 \cos(A_1) & -S_5 \sin(A_2 - B_5) \\ -L_1 \sin(A_1) & -S_5 \cos(A_2 - B_5) \end{bmatrix} \begin{Bmatrix} K_{A1} \\ K_{A2} \end{Bmatrix} = R \begin{Bmatrix} \sin(\theta) \\ -\cos(\theta) \end{Bmatrix} \quad (50)$$

After differentiation with respect to θ , it gives:

$$\begin{aligned} R \begin{Bmatrix} \cos(\theta) \\ \sin(\theta) \end{Bmatrix} &= \begin{bmatrix} -K_{A1} L_1 \sin(A_1) & -K_{A2} S_5 \cos(A_2 - B_5) \\ -K_{A1} L_1 \cos(A_1) & -K_{A2} S_5 \sin(A_2 - B_5) \end{bmatrix} \begin{Bmatrix} K_{A1} \\ K_{A2} \end{Bmatrix} \\ &+ \begin{bmatrix} L_1 \cos(A_1) & -S_5 \sin(A_2 - B_5) \\ -L_1 \sin(A_1) & -S_5 \cos(A_2 - B_5) \end{bmatrix} \begin{Bmatrix} L_{A1} \\ L_{A2} \end{Bmatrix} \end{aligned} \quad (51)$$

Which can be solved to find velocity coefficient derivatives as follows:

$$\begin{Bmatrix} L_{A1} \\ L_{A2} \end{Bmatrix} = \begin{bmatrix} L_1 \cos(A_1) & -S_5 \sin(A_2 - B_5) \\ -L_1 \sin(A_1) & -S_5 \cos(A_2 - B_5) \end{bmatrix}^{-1} \left(\begin{matrix} R \\ R \end{matrix} \begin{Bmatrix} \cos(\theta) \\ \sin(\theta) \end{Bmatrix} + \begin{bmatrix} -K_{A1}L_1 \sin(A_1) & -K_{A2}S_5 \cos(A_2 - B_5) \\ -K_{A1}L_1 \cos(A_1) & -K_{A2}S_5 \sin(A_2 - B_5) \end{bmatrix} \begin{Bmatrix} K_{A1} \\ K_{A2} \end{Bmatrix} \right) \quad (52)$$

Now that all coefficients are known, they can be used to find the speed of point P at any crank angle:

$$K_{px}(\theta) = \frac{dx_p}{d\theta} = \frac{d}{d\theta} (L_1 \sin(A_1) - S_1 \cos(A_2)) = K_{A1}L_1 \cos(A_1) + K_{A2}S_1 \sin(A_2) \quad (53)$$

$$K_{py}(\theta) = \frac{dy_p}{d\theta} = \frac{d}{d\theta} (L_1 \cos(A_1) + S_1 \sin(A_2)) = -K_{A1}L_1 \sin(A_1) + K_{A2}S_1 \cos(A_2) \quad (54)$$

When it comes to acceleration of point P, its velocity derivative coefficient can be computed by differentiation of equations **Error! Reference source not found.** and **Error! Reference source not found.**:

$$\begin{aligned} L_{px}(\theta) &= \frac{dK_{px}(\theta)}{d\theta} = \frac{d}{d\theta} (K_{A1}L_1 \cos(A_1) + K_{A2}S_1 \sin(A_2)) \\ &= L_{A1}L_1 \cos(A_1) + L_{A2}S_1 \sin(A_2) - K_{A1}^2L_1 \sin(A_1) + K_{A2}^2S_1 \cos(A_2) \end{aligned} \quad (55)$$

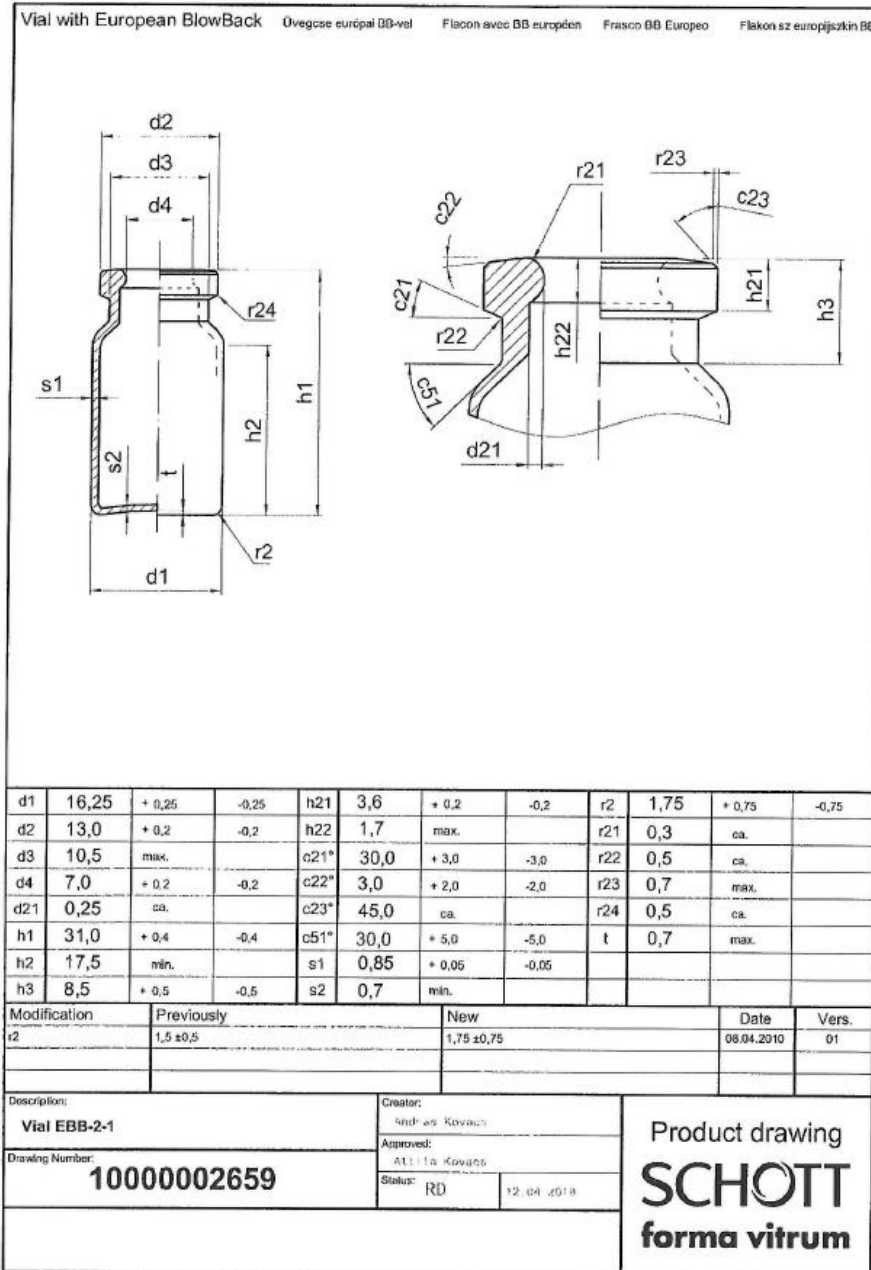
$$\begin{aligned} L_{py}(\theta) &= \frac{dK_{py}(\theta)}{d\theta} = \frac{d}{d\theta} (-K_{A1}L_1 \sin(A_1) + K_{A2}S_1 \cos(A_2)) \\ &= -L_{A1}L_1 \sin(A_1) + L_{A2}S_1 \cos(A_2) - K_{A1}^2L_1 \cos(A_1) + K_{A2}^2S_1 \sin(A_2) \end{aligned} \quad (56)$$

7.12 Pfizer vials datasheet

Status	Effective	Effective Date	01-Jun-2017	Version	8.0	Doc Name	SPEC-18047
Title	SCHOTT VIALS						
Doc Alias	G275			Site Code / Department	puu / Product Support		

5. Drawings

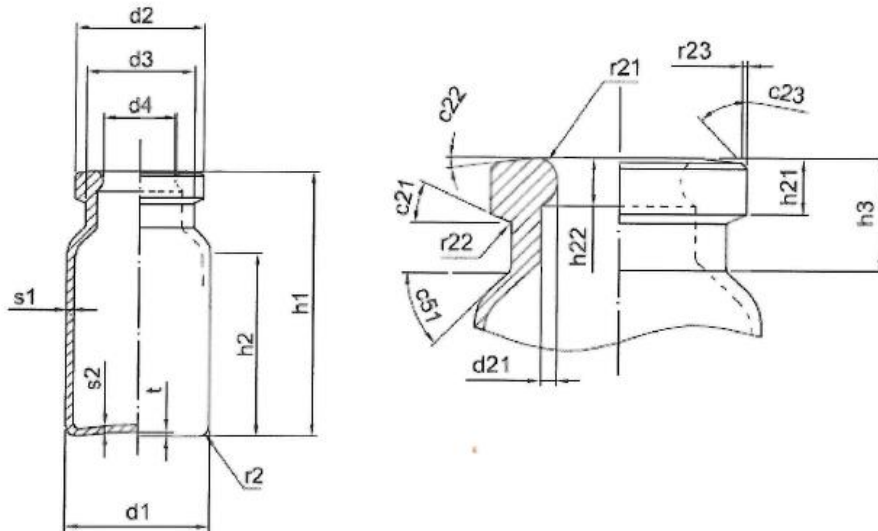
5.1. PPU5A0049, PPU5A0309, PPU5A0401



Status	Effective	Effective Date	01-Jun-2017	Version	8.0	Doc Name	SPEC-18047
Title	SCHOTT VIALS						
Doc Alias	G275			Site Code / Department	puu / Product Support		

5.11. PPU5A0211

Vial with European BlowBack Övegeze európai BB-vel Flacon avec BB européen Frasco BB Europeo Flakon sz. európai(k) BB



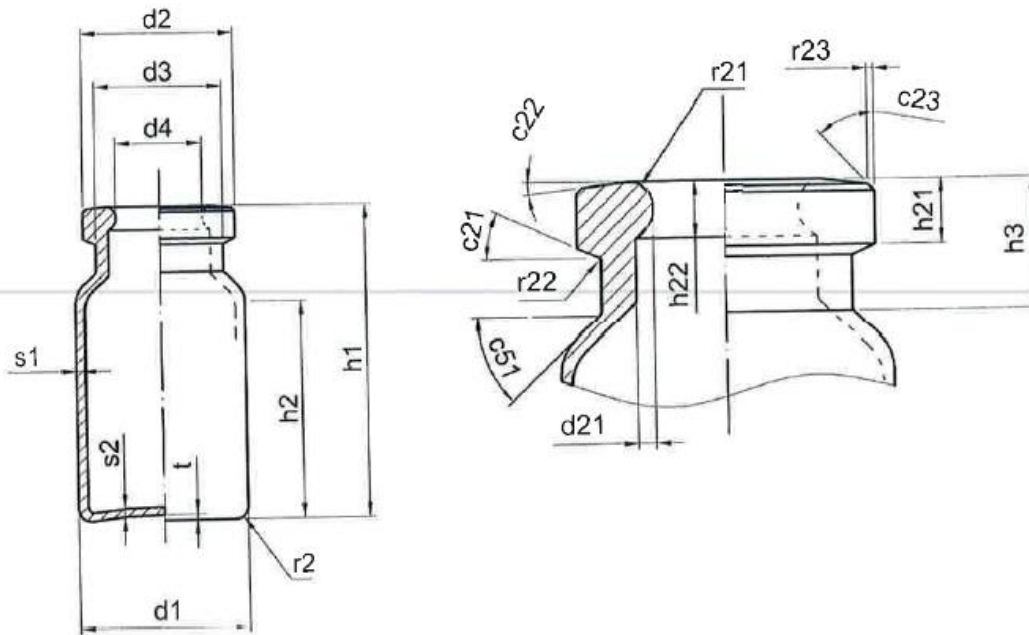
d1	25,25	- 0,25	-0,25	h3	9,0	+ 0,5	-0,5	s1	1,1	+ 0,05	-0,05
d2	19,95	+ 0,25	-0,25	h21	3,6	+ 0,2	-0,2	s2	0,7	min.	
d3	17,0	max.		h22	1,7	max.		r2	2,0	min.	
d4	12,6	+ 0,2	-0,2	c21*	30,0	+ 5,0	-5,0	r21	0,5	+ 0,2	-0,2
d21	0,2	min.		c22*	3,0	+ 2,0	-2,0	r22	0,5	ca.	
h1	53,5	+ 0,5	-0,5	c23*	45,0	ca.		r23	0,5	ca.	
h2	36,5	min.		c51*	30,0	+ 5,0	-5,0	t	0,7	max.	

Modification	Previously	New	Date	Vers.

Description: Vial 15 EBB		Creator: Bálint Szabados	Product drawing SCHOTT forma vitrum
Drawing Number: 10000008919		Approved: Hanspeter Kübler	
		Issue: RD 24.01.2011	

Drawing ID: 10000008919-009/09 NOTE: Linear dimensions are in millimeter

30 ml vial



d1	30,0	+ 0,3	-0,3	h3	9,0	+ 0,5	-0,5	s1	1,3	+ 0,05	-0,05
d2	19,95	+ 0,25	-0,25	h21	3,6	+ 0,2	-0,2	s2	0,7	min.	
d3	17,0	max.		h22	1,7	max.		r2	3,0	min.	
d4	12,6	+ 0,2	-0,2	c21°	30,0	+ 5,0	-5,0	r21	0,5	+ 0,2	-0,2
d21	0,2	min.		c22°	3,0	+ 2,0	-2,0	r22	0,5	ca.	
h1	76,0	+ 0,5	-0,5	c23°	45,0	ca.		r23	0,5	ca.	
h2	56,0	min.		c51°	30,0	+ 5,0	-5,0	t	0,7	max.	

Modification	Previously	New	Date	Vers.

Description: Vial 30 EBB	Creator: Baltazs Szabados	
	Approved: Gyorgy Kozma	
	Status: RD	2011.01.25.11

Drawing Number: **1000008917**

Product drawing
SCHOTT
forma vitrum

Drawing ID: 1000008917-000100

7.13 O'ring material compatibility with H2O2

O-Ring Materials Compatible with Hydrogen Peroxide (select a material to show its compatible chemicals)		
Aflas (4)	Buna-N (Nitrile) (1)	Butyl (0)
Chemraz (3)	Epichlorohydrin (3)	Ethylene-Propylene (0)
Fluorocarbon (0)	Fluorosilicone (3)	Hypalon (3)
Kalrez (4)	Natural Rubber (0)	Neoprene (3)
Nitrile, Hydrogenated (1)	Polyacrylate (0)	Polysulfide (0)
Polyurethane, Cast (0)	Polyurethane, Millable (0)	Silicone (1)
Styrene Butadiene (4)	Teflon, Virgin (0)	Vamac (0)

Key to O-Ring Material Compatibility Ratings	
(4)	Good, both for static and dynamic seals
(3)	Fair, usually OK for static seals
(2)	Sometimes OK for static seals; not OK for dynamic seals
(1)	Poor
(0)	No Data

FIGURE 154 - O-RING MATERIAL COMPATIBILITY WITH H2O2

https://www.efunda.com/designstandards/oring/oring_chemical.cfm?SM=none&SC=Hydrogen%20Peroxide

7.14 Python code

7.14.1 Filling configurations

Needles_locations:

```
import numpy as np

needles_a = np.zeros(4) + 1
stroke = np.zeros(10000)

number_a = 6
sol = 0

vial_dim_a = 16 + 1 # + 1 to take design tolerance into account
vial_dim_b = 24 + 1
vial_dim_c = 30 + 1
needles_dim = 13 + 1
min_tooth = 4 # Minimum dimension for teeth to be acceptable

for number_b in range(1, 7):
    # print("\nnumber_b =", number_b)
    for number_c in range(1, 7):
        for a in range(1, 70):
            # print("number_c =", number_c)
            needles_a = np.zeros(number_a)
            needles_b = np.zeros(number_b)
            needles_c = np.zeros(number_c)

            b = number_a/number_b * (a + vial_dim_a) - vial_dim_b #
            Condition to have same stroke
            c = number_a / number_c * (a + vial_dim_a) - vial_dim_c

            needles_a[0] = (a + vial_dim_a) / 2
            needles_b[0] = (b + vial_dim_b) / 2
            needles_c[0] = (c + vial_dim_c) / 2

            if b < 1 or c < 1:
                continue

            for i in range(number_a):
                needles_a[i] = needles_a[0] + i * (a + 17)
            for i in range(number_b):
                needles_b[i] = needles_b[0] + i * (b + 25)
            for i in range(number_c):
                needles_c[i] = needles_c[0] + i * (c + 31)

            dist_needles_a_b = np.zeros(len(needles_a) * len(needles_b))
            dist_needles_a_c = np.zeros(len(needles_a) * len(needles_c))
            dist_needles_b_c = np.zeros(len(needles_b) * len(needles_c))

            k = 0
            for i in needles_a:
                for j in needles_b:
                    dist_needles_a_b[k] = abs(i - j)
```

```

        k += 1
    k = 0
    for i in needles_a:
        for j in needles_c:
            dist_needles_a_c[k] = abs(i - j)
            k += 1
    k = 0
    for i in needles_b:
        for j in needles_c:
            dist_needles_b_c[k] = abs(i - j)
            k += 1

    # print("\ndist_needles_a_b", dist_needles_a_b)
    not_ok_1 = dist_needles_a_b[(dist_needles_a_b <= needles_dim) &
(dist_needles_a_b != 0)]
    # print("not_ok_1", not_ok_1)
    # print("\n")
    not_ok_2 = dist_needles_a_c[(dist_needles_a_c <= needles_dim) &
(dist_needles_a_c != 0)]
    not_ok_3 = dist_needles_b_c[(dist_needles_b_c <= needles_dim) &
(dist_needles_b_c != 0)]
    # print("\ndist_needles_b_c", dist_needles_b_c)

    if not_ok_1.size + not_ok_2.size + not_ok_3.size == 0\
        and min_tooth < c < 1.5 * vial_dim_c\
        and min_tooth < b < 1.5 * vial_dim_b\
        and min_tooth < a < 1.5 * vial_dim_a:

        entrax_a = vial_dim_a + a
        stroke[sol] = number_a * entrax_a
        print("\n")
        print("stroke = %.2f | Number of vials | Teeth width [mm]
|" % stroke[sol])
        print("-----")
        print("17 mm vials      |          %.f          |          %.f
|" % (number_a, a))
        print("-----")
        print("25 mm vials      |          %.f          |          %.f
|" % (number_b, b))
        print("-----")
        print("31 mm vials      |          %.f          |          %.f
|" % (number_c, c))
        print("-----")
        print("\nNeedles locations:")
        print("needles 17 =", needles_a)
        print("needles 25 =", needles_b)
        print("needles 31 =", needles_c)
        length = max(5 * number_a * (a + 17) + a, int(5 * number_b
* (b + 25) + b),
                    int(5 * number_c * (c + 31) + c))
        print("\nTotal length of box transfer mechanism : %.2f
[cm]" % (length/10))
        print("-----")
        sol += 1
        break

```

7.14.2 Kinematics of the box transport mechanism

Box_transfer_class:

```
__author__ = 'Cédric Keutgen'
import numpy as np
import Plot_data_class as Data
import Solve_equations as Equations
import Straight_line_class as Find
import math
import Speed_planning_class as Speed
from Straight_line_class import TrioLineClass, LineClass, EngagingClass

"""
Design criteria
"""

min_stroke = 216 # Minimum linear conveying sought
tol_stroke = 5 # Max trajectory deviation from the straight line

engage_length = 15 # Minimum length of straight lines allowing to engage
the linear trajectory
tol_engage = 5 # Max trajectory deviation from the straight line

cosine_tol = math.cos(math.radians(0)) # Maximum cosine between engagement
lines and stroke (goal: near 90°)

#

---

"""
Motion profile parameters
"""

v_start = [0, 0]
v_end = [0, 0]
v_max = [300, 300]
a_max = [6000, 6000]
j_max = [5000, 5000]
cycle_time = [1.2, 1.2] # [sTime for stroke, Time for retreat]
t = np.transpose(np.linspace(0, cycle_time, 100))

#

---

"""
Information display parameters
"""

line_width = 2 # Line width used for plot display
only_stroke = False # Put to True for only stroke lines and no care about
interceptions
info = True # Put to False for no information display on results found
```

```

#
---
"""
Beginning of the main code
"""
a = 1.322 # Coefficient to scale the parameters, and so the trajectory.

# Units for plot based on the principle that parameters are mm. In
practice, parameters have arbitrary units
a_unit = {1: "[mm]", 0.1: "[cm]", 0.01: "[dm]", 0.001: "[m]", -np.inf:
"[?]" }

unit = next(v for k, v in a_unit.items() if a >= k)

x0 = 198*a # [mm]
y0 = 240*a
L1 = 234.75*a # [mm]

theta_range = np.linspace(0, 2 * np.pi, 100) # Define the range of theta
values [rad]
theta_degree = theta_range * 360 / (2 * np.pi) # Define the range of theta
values [°]

xp = np.zeros(len(theta_range)) # Trajectory vector
yp = np.zeros(len(theta_range)) # Trajectory vector

S1 = 108
S4 = 319
S5 = 217
R = 91.95*a # [mm]
s = {k: a * v for k, v in {'S1': S1, 'S4': S4, 'S5': S5}.items()} # Scale
the triangle
print(R)

def box_transfer(s, x0, y0, L1, min_stroke, engage_length, tol_stroke,
tol_engage, cosine_tol):

    assembly, grashoft = Equations.assembly_and_grashoft(s, R, L1, x0, y0)

    # If they are respected, solve equations
    if assembly and grashoft:
        # Newton-Raphson to find B4 and B5
        [B4, B5] = Equations.solve_b4_b5(s)

        # fsolve to find A1 and A2
        [A1, A2] = Equations.solve_a1_a2(theta_range, [B4, B5], s, L1, R,
x0, y0)

        # Compute velocity and velocity derivative coefficients for each
value of theta
        [K_A1, K_A2, L_A1, L_A2] = Equations.solve_ka_la(theta_range, [A1,
A2], [B4, B5], s, L1, R)
        [K_px, K_py, L_px, L_py, K_p, L_p] = Equations.solve_kp_lp([K_A1,
K_A2], [L_A1, L_A2], [A1, A2], s, L1)
        print(max(abs(np.diff(K_p))))

```

```

    [xp, yp] = Equations.solve_trajectory(L1, [A1, A2], s)

    lines, intersected_lines, trio_lines, line_stroke, closer_line =
Find.conveying_path(
    xp, yp, min_stroke, engage_length, tol_stroke, tol_engage,
cosine_tol, unit, info=info,
    only_stroke=only_stroke)
    Data.anim_trajectory(xp, yp, s, lines, x0, y0, L1, R, line_width,
unit) # Trajectory + lines
    Data.plot_all_coefficients(theta_degree, K_px, K_py, L_A1, L_px,
L_A2, L_py, K_p, L_p)

    if max(abs(np.diff(K_p))) <= 20:

        # Trajectory of point P
        if closer_line:
            # Compute an S-curve speed profile for the stroke
            Data.anim_trajectory(xp, yp, s, lines, x0, y0, L1, R,
line_width, unit)

            # stroke_speed, motion_profiles =
Speed.s_curve(closer_line, v_start, v_end, v_max, a_max, j_max, cycle_time)

            # Convert the previous profile in terms of motor
requirements
            # motor_speed, motor_acc, motor_speed_rpm, motor_acc_rpm =
Equations.theta_speed_acc(motion_profiles, K_p, L_p, info)

            # Compute other speed profiles for comparisons
            trapz_profile = Speed.trapz(-4.6, -16.8, t[0])
            quintic_profile = Speed.quintic(-4.6, -16.8, t[0])
            lspb_profile = Speed.lspb(closer_line, t[1])

            # max_motor_speed = max(max(abs(motor_speed[0])),
max(abs(lspb_profile.qd))) # rad/s
            max_motor_speed = max(abs(lspb_profile.qd)) # rad/s

        else:
            max_motor_speed = []
    else:
        closer_line = []
        max_motor_speed = []

    """
    Data.plot_s_profile(stroke_speed, cycle_time, unit, line_width)
    Data.plot_motor(motor_speed, motor_acc, t)
    Data.plot_speed_profile(['Trapezoidal', 'Quintic', 'linear segment
with parabolic blend'],
                            [trapz_profile, quintic_profile,
lspb_profile],
                            line_width,
via_points=lspb_profile.viapoints)
    Data.plot_all_coefficients(theta_degree, K_px, K_py, L_A1, L_px,
L_A2, L_py, K_p, L_p)
    """
    """
    [x_p_speed, y_p_speed, p_speed] =
Equations.solve_p_speed(theta_speed, [K_px, K_py])

```

```

        [x_p_acc, y_p_acc, p_acc] = Equations.solve_p_acc(theta_acc,
theta_speed, [K_px, K_py], [L_px, L_py])

        """

    else:
        closer_line = []
        max_motor_speed = []

    return closer_line, max_motor_speed

closer_line, motor_speed = box_transfer(s, x0, y0, L1, min_stroke,
engage_length, tol_stroke, tol_engage, cosine_tol)

```

Solve_equations:

```

import math
import numpy as np
from numpy.linalg import inv
from scipy.optimize import fsolve

def solve_b4_b5(s):

    s1 = s['S1']
    s4 = s['S4']
    s5 = s['S5']

    b4 = -2
    b5 = -2.5
    b4_test = 0
    b5_test = 0

    iter_max = 50 # To get out of the loop if no convergence
    iter_count = 0

    while (abs(b4 - b4_test) > 1E-4 or abs(b5 - b5_test > 1E-4)) and
iter_count < iter_max:

        f = np.matrix([[s1 + s5 * math.cos(b5) - s4 * math.cos(b4)],
                        [s5 * math.sin(b5) - s4 * math.sin(b4)]])

        jacobian = np.matrix([[s4 * math.sin(b4), -s5 * math.sin(b5)],
                               [-s4 * math.cos(b4), s5 * math.cos(b5)]])

        j_inv = inv(jacobian)

        b4_test = b4 # Required to test tolerance with previous solution
        b5_test = b5

        xn = np.array([[b4], [b5]])
        xn_1 = xn - (j_inv * f)

        b4 = xn_1[0, 0]
        b5 = xn_1[1, 0]

```

```

        iter_count = iter_count + 1

    return b4, b5

def equations_a1_a2(variables, angle, l1, x_0, y_0, s, b, r):

    b5 = b[1]
    s5 = s['S5']

    a1, a2 = variables
    eq1 = l1 * np.sin(a1) + s5 * np.cos(a2 - b5) + r * np.cos(angle) - x_0
    eq2 = l1 * np.cos(a1) - s5 * np.sin(a2 - b5) + r * np.sin(angle) - y_0

    return [eq1, eq2]

def solve_a1_a2(theta_range, b, s, l1, r, x_0, y_0):
    # Define the function that returns the residuals to be minimized

    # Initialize arrays to store the solutions and x_p, y_p
    a1 = np.zeros_like(theta_range)
    a2 = np.zeros_like(theta_range)

    # Solve the system for each value of theta
    for i, theta in enumerate(theta_range):
        # Initial guesses for A1 and A2
        initial_guess = np.array([1, 1])

        # Solve the system using fsolve
        sol = fsolve(equations_a1_a2, initial_guess, args=(theta, l1, x_0,
y_0, s, b, r))
        a1[i] = sol[0]
        a2[i] = sol[1]

    return a1, a2

def solve_trajectory(l1, a, s):

    a1 = a[0]
    a2 = a[1]
    s1 = s['S1']

    x_p = l1 * np.sin(a1) - s1 * np.cos(a2)
    y_p = l1 * np.cos(a1) + s1 * np.sin(a2)

    return x_p, y_p

def solve_ka_la(theta_range, a, b, s, l1, r):
    b5 = b[1]
    s5 = s['S5']

    a1 = a[0]
    a2 = a[1]

    k_a1 = np.zeros_like(theta_range)
    k_a2 = np.zeros_like(theta_range)

```



```

l_a1 = np.zeros_like(theta_range)
l_a2 = np.zeros_like(theta_range)

for i, theta in enumerate(theta_range):
    # Compute K_A
    jacobian = np.matrix([[l1 * np.cos(a1[i]), -s5 * np.sin(a2[i] -
b5)],
                          [-l1 * np.sin(a1[i]), -s5 * np.cos(a2[i] -
b5]])
    j_inv = inv(jacobian)

    x_dot = np.array([[r * np.sin(theta)], [-r * np.cos(theta)]]
    x_dot_dot = np.array([[r * np.cos(theta)], [r * np.sin(theta)]]

    k_a = j_inv @ x_dot

    k_a1[i] = k_a[0, 0]
    k_a2[i] = k_a[1, 0]

    matrix4 = np.matrix([[k_a1[i] * l1 * np.sin(a1[i]), k_a2[i] * s5 *
np.cos(a2[i] - b5)],
                          [k_a1[i] * l1 * np.cos(a1[i]), -k_a2[i] * s5 *
np.sin(a2[i] - b5)]]
    matrix5 = np.matrix([[k_a1[i]], [k_a2[i]]])

    l_a = j_inv @ (x_dot_dot + matrix4 @ matrix5)

    l_a1[i] = l_a[0, 0]
    l_a2[i] = l_a[1, 0]

return k_a1, k_a2, l_a1, l_a2

def solve_kp_lp(k_a, l_a, a, s, l1):

    k_a1 = k_a[0]
    k_a2 = k_a[1]

    l_a1 = l_a[0]
    l_a2 = l_a[1]

    a1 = a[0]
    a2 = a[1]

    s1 = s['S1']

    k_px = k_a1 * l1 * np.cos(a1) + k_a2 * s1 * np.sin(a2)
    k_py = -k_a1 * l1 * np.sin(a1) + k_a2 * s1 * np.cos(a2)

    l_px = l_a1 * l1 * np.cos(a1) + l_a2 * s1 * np.sin(a2)
    - (k_a1 ** 2) * l1 * np.sin(a1) + (k_a2 ** 2) * s1 * np.cos(a2)

    l_py = -l_a1 * l1 * np.sin(a1) + l_a2 * s1 * np.cos(a2)
    - (k_a1 ** 2) * l1 * np.cos(a1) - (k_a2 ** 2) * s1 * np.sin(a2)

    k_p = (k_px ** 2 + k_py ** 2) ** 0.5
    l_p = (l_px**2 + l_py**2)**0.5

return k_px, k_py, l_px, l_py, k_p, l_p

```

```

def solve_p_speed(theta_speed, k_p): # Speed of point P

    k_px = k_p[0]
    k_py = k_p[1]

    x_p_speed = theta_speed * k_px
    y_p_speed = theta_speed * k_py

    p_speed = np.sqrt(x_p_speed ** 2 + y_p_speed ** 2)

    return x_p_speed, y_p_speed, p_speed

def solve_p_acc(theta_acc, theta_speed, k_p, l_p): # Acceleration of point
P

    k_px = k_p[0]
    k_py = k_p[1]

    l_px = l_p[0]
    l_py = l_p[1]

    x_p_acc = theta_acc * k_px + (theta_speed ** 2) * l_px
    y_p_acc = theta_acc * k_py + (theta_speed ** 2) * l_py
    p_acc = np.sqrt(x_p_acc ** 2 + y_p_acc ** 2)

    return x_p_acc, y_p_acc, p_acc

def assembly_and_grashoft(s, r, l1, x0, y0):

    l_min = min(float(s['S5']), r, l1, math.sqrt(x0 ** 2 + y0 ** 2))
    l_max = max(float(s['S5']), r, l1, math.sqrt(x0 ** 2 + y0 ** 2))
    sum_l = s['S5'] + r + l1 + math.sqrt(x0 ** 2 + y0 ** 2)

    if 2 * l_max < sum_l:
        print("\nAssembly ok")
        assembly = 1
    else:
        print("\n !\ ASSEMBLY DISRESPECTED: 2 l_max should be < sum")
        print("S5 = %.3f, R = %.3f, L1 = %.3f, L2 = %.3f, sum = %.3f"
              % (s['S5'], r, l1, math.sqrt(x0 ** 2 + y0 ** 2), sum_l))
        assembly = 0

    if 2 * (l_min + l_max) < sum_l:
        print("Grashoft ok")
        grashoft = 1
    else:
        print("\n !\ GRASHOFT DISRESPECTED: 2 (l_min + l_max) should be <
sum")
        print("S5 = %.3f, R = %.3f, L1 = %.3f, L2 = %.3f, sum = %.3f"
              % (s['S5'], r, l1, math.sqrt(x0 ** 2 + y0 ** 2), sum_l))
        grashoft = 0

    return assembly, grashoft

def theta_speed_acc(motion_profiles, k_p, l_p, info=True):

```

```

theta_speed = []
theta_acc = []

theta_speed_rpm = []
theta_acc_rpm = []

for i in range(len(motion_profiles)):

    theta_speed.append(motion_profiles[i][1]/k_p)

    theta_acc.append((motion_profiles[i][0] - (theta_speed[i]**2)*l_p)
/ k_p)

    theta_speed_rpm.append((theta_speed[i] * 60) / (2 * np.pi))
    theta_acc_rpm.append((theta_acc[i]*60**2/(2*np.pi)))

    if info:
        print('\nMax \u03b8 speed = %.2f [rad/s] = %.2f RPM'
              % (max(abs(theta_speed[i])),
(max(abs(theta_speed[i]*60/(2*np.pi))))))

        print('Max \u03b8 acc = %.2f [rad/s^2] = %.2f RPM^2\n'
              % (max(abs(theta_acc[i])),
(max(abs(theta_acc[i]*60**2/(2*np.pi))))))

    return theta_speed, theta_acc, theta_speed_rpm, theta_acc_rpm

def theta_speed_acc2(motion_profiles, k_p, l_p, info=True):

    speed = np.transpose(motion_profiles.qd)[0]
    acc = np.transpose(motion_profiles.qdd)[0]

    theta_speed = speed/k_p
    theta_acc = (acc - (theta_speed**2)*l_p) / k_p

    theta_speed_rpm = (theta_speed * 60) / (2 * np.pi)
    theta_acc_rpm = ((theta_acc*60**2)/(2*np.pi))

    if info:
        print('\nMax \u03b8 speed = %.2f [rad/s] = %.2f RPM'
              % (max(abs(theta_speed)),
(max(abs(theta_speed*60/(2*np.pi))))))

        print('Max \u03b8 acc = %.2f [rad/s^2] = %.2f RPM^2\n'
              % (max(abs(theta_acc)),
(max(abs(theta_acc*60**2/(2*np.pi))))))

    return theta_speed, theta_acc, theta_speed_rpm, theta_acc_rpm

#

```

Plot_data_class:

```

__author__ = 'Cédric Keutgen'
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from Speed_planning import get_s_traj_profile
import roboticstoolbox # Makes figures looking good

def on_key_press(event):
    if event.key == 'up':
        plt.close()

#

def plt_display(show=False):

    if show:
        plt.tight_layout()
        plt.show()

    else:
        plt.draw()
        plt.waitforbuttonpress(0)
        plt.close()

#

def plot_coefficient(theta, function, label, derivative=False):

    print(f"Max {label} = %.2f" % (max(abs(function))))

    fig = plt.figure()
    fig.canvas.mpl_connect('key_press_event', on_key_press)
    plt.plot(theta, function, linewidth=2)
    plt.xlabel('theta [°]')
    plt.ylabel(f"${label}$")

    if derivative:
        plt.title(f'VeLOCITY coefficient derivative ${label}$ regarding
theta')

    else:
        plt.title(f'VeLOCITY coefficient ${label}$ regarding theta')

#

def plot_all_coefficients(theta_degree, k_px, k_py, l_a1, l_px, l_a2, l_py,
k_p, l_p):

    plot_coefficient(theta_degree, k_p, 'K_{p}')
    plot_coefficient(theta_degree, l_p, 'L_{p}', True)
    plot_coefficient(theta_degree, k_px, 'K_{px}')

```

```

plot_coefficient(theta_degree, k_py, 'K_{py}')
plot_coefficient(theta_degree, l_a1, 'L_{A1}', True)
plot_coefficient(theta_degree, l_px, 'L_{px}', True)
plot_coefficient(theta_degree, l_a2, 'L_{A2}', True)
plot_coefficient(theta_degree, l_py, 'L_{py}', True)
plt_display(True)

#
-----

def anim_trajectory(xp, yp, s, all_lines, x0, y0, l1, r, lw, unit,
plot_line=True, plot_ani=True):

    s1 = s['S1']
    s4 = s['S4']
    s5 = s['S5']
    fraction1 = r'\frac{S_4}{S_1}'
    fraction2 = r'\frac{S_5}{S_1}'
    fraction3 = r'\frac{S_5}{S_4}'
    fig, ax = plt.subplots()
    ax.invert_yaxis()
    ax.set_xlabel(f'$x_p$ {unit}')
    ax.set_ylabel(f'$y_p$ {unit}')
    ax.set_title('Trajectory of point P \n'
                 f'S1 = %.1f; S4 = %.1f; S5 = %.1f; x0 = %.1f; y0 = %.1f;
L1 = %.1f; R = %.1f'
                 % (s1, s4, s5, x0, y0, l1, r))

    if plot_ani:
        plt.plot(xp, yp, linewidth=lw)
        line, = ax.plot([], [], color='b', linewidth=lw)

        # Define the update function for the animation
        def update(frame):
            line.set_data(xp[:frame], yp[:frame])
            return line,

        # Create the animation object
        ani = FuncAnimation(fig, update, frames=len(xp), interval=20,
blit=True)
        plt.gcf().canvas.mpl_connect('key_press_event', on_key_press)

    if plot_line:
        plot_lines(all_lines, ax, lw+1)
        # plot_lines(engagement_lines, ax, xp, yp, lw+2)

    plt_display(True)

#
-----

def plot_lines(lines, ax, lw):

    for line in lines:
        x = [line.start[0], line.end[0]]
        y = [line.start[1], line.end[1]]

```

```

ax.plot(x, y, linewidth=lw)

#

```

```

def plot_s_profile(tr, t, unit, lw):

    string = ['S velocity profile for the stroke: comb point of view',
              'S velocity profile for the retreat: \u03b8(t) point of
view']
    unit = [unit, '[rad]']

    for i in range(len(t)):
        fig = plot_t_trajectory(tr[i], unit[i], lw)
        fig.suptitle(string[i])

    plt_display(True)

#

```

```

def plot_t_trajectory(traj, unit, lw):

    acceleration_id = 0
    speed_id = 1
    position_id = 2
    r_profiles, dof, time = get_s_traj_profile(traj)
    fig1 = plt.figure()
    fig1.canvas.mpl_connect('key_press_event', on_key_press)
    # fig.suptitle("DOF profiles")

    for i, profile in zip(range(dof), r_profiles):
        plt.subplot(300 + dof*10 + (i+1))
        plt.plot(time, profile[acceleration_id][:], linewidth=lw)
        plt.xlim()
        plt.ylim()
        plt.ylabel(f'Acceleration {unit}/s^2')

        plt.subplot(300 + dof*10 + (i+1)+dof)
        plt.plot(time, profile[speed_id][:], linewidth=lw)
        plt.xlim()
        plt.ylim()
        plt.ylabel(f'Speed {unit}/s')

        plt.subplot(300 + dof*10 + (i+1)+dof*2)
        plt.plot(time, profile[position_id][:], linewidth=lw)
        plt.xlim()
        plt.ylim()
        plt.xlabel('Time [s]')
        plt.ylabel(f'Position {unit}')

    return fig1

#

```

```

def plot_motor(motor_speed, motor_acc, t):
    title = ['S-profile for stroke: motor view', 'S-profile for retreat:
motor view']

    for i in range(len(motor_speed)):
        fig = plt.figure()
        fig.canvas.mpl_connect('key_press_event', on_key_press)
        fig.suptitle(title[i])

        plt.subplot(211)
        plt.title("Acceleration profile")
        plt.plot(t[i], motor_acc[i], linewidth=2)
        plt.xlim()
        plt.ylim()
        plt.ylabel(f'rad/s^2')

        plt.subplot(212)
        plt.title("Speed profile")
        plt.plot(t[i], motor_speed[i], linewidth=2)
        plt.xlim()
        plt.ylim()
        plt.ylabel(f'rad/s')
        plt.xlabel('Time [s]')

    plt_display(True)

#

```

```

def plot_speed_profile(function, profile, lw, via_points=None):

    for i in range(len(function)):
        fig = plt.figure()
        fig.canvas.mpl_connect('key_press_event', on_key_press)
        plt.suptitle(f'{function[i]} motion profile for stroke: comb point
of view')

        plt.subplot(313)
        plt.plot(profile[i].t, profile[i].q, linewidth=lw)

        if function[i] == 'linear segment with parabolic blend':
            plt.scatter(via_points.passing_times,
via_points.passing_points, s=50, marker='x', color='b', linewidths=1)

        plt.xlabel("Time (s)")
        plt.ylabel("Position")

        plt.subplot(312)
        plt.plot(profile[i].t, profile[i].qd, linewidth=lw)
        plt.ylabel("Speed")

        plt.subplot(311)
        plt.plot(profile[i].t, profile[i].qdd, linewidth=lw)
        plt.ylabel("Acceleration")

    plt_display(True)

```

```

#
---

def plot_motor2(motor_speed, motor_acc, t):
    title = ['Trapezoidal profile for stroke: motor view', 'Quintic profile
for stroke: motor view', 'Lspb profile for retreat: motor view']

    for i in range(len(motor_speed)):

        fig = plt.figure()
        fig.canvas.mpl_connect('key_press_event', on_key_press)
        fig.suptitle(title[i])

        plt.subplot(211)
        plt.title("Acceleration profile")
        plt.plot(t[i], motor_acc[i], linewidth=2)
        plt.xlim()
        plt.ylim()
        plt.ylabel(f'rad/s^2')

        plt.subplot(212)
        plt.title("Speed profile")
        plt.plot(t[i], motor_speed[i], linewidth=2)
        plt.xlim()
        plt.ylim()
        plt.ylabel(f'rad/s')
        plt.xlabel('Time [s]')

plt_display(True)

```

7.14.3 Trajectory analysis

Straight_line_class:

```

__author__ = 'Cédric Keutgen'
import numpy as np
import math

class LineClass:
    def __init__(self, length=None, start=None, end=None, deviation=None):
        self.length = length
        self.start = start
        self.end = end
        self.deviation = deviation

#
---

class EngagingClass:

```



```

    def __init__(self, length=None, start=None, end=None, deviation=None,
cosine=None):
        self.length = length
        self.start = start
        self.end = end
        self.deviation = deviation
        self.cosine = cosine

#
-----

class TrioLineClass:
    def __init__(self, line=None, engaging=None, disengaging=None):
        self.line = line
        self.engaging = engaging
        self.disengaging = disengaging

#
-----

def conveying_path(xp, yp, stroke, engage_length, tol_stroke, tol_engage,
        cosine_tol, unit, info=True, only_stroke=False):
    """Find straight lines in a trajectory.

    xp and yp are arrays of x and y coordinates, respectively.
    tolerance is the maximum perpendicular deviation distance between the
    actual trajectory and the final straight line.
    min_length is the minimum length of a straight line.

    Returns a list of dictionaries with keys 'length', 'start', and 'end'.
    """
    trajectory = np.column_stack((xp, yp))
    line_stroke = compliant_stroke(trajectory, stroke, tol_stroke)
    all_lines = line_stroke

    if all_lines:
        max_length = max(line.length for line in all_lines)

        if info:
            print(f"\n%.f straight line(s) satisfying the stroke
requirement have been found,"
                f" longest line = %.2f {unit}"
                f" % (len(all_lines), max_length))
        else:
            if info:
                print("No straight line satisfying the stroke have been
found\n")
            return [], [], [], [], []

    intersected_lines = []
    trio_lines = []
    closer_line = []

    if not only_stroke and all_lines:
        """Find if stroke lines can be engaged and disengaged according to
tolerance"""

```

```

    all_lines = []

    for line in line_stroke:
        disengage_line = {}
        # We start by searching for existence of engaging_lines
        engage_line = engage(line, trajectory, engage_length,
tol_engage, cosine_tol)
        # If we found an engaging line, search for a disengaging one

        if engage_line:
            disengage_line = disengage(line, trajectory, engage_length,
tol_engage, cosine_tol)

        # If we found both engaging and disengaging line, then those
lines can be considered and append
        if engage_line and disengage_line:
            trio = TrioLineClass()
            trio.line = line
            trio.engaging = engage_line
            trio.disengaging = disengage_line
            trio_lines.append(trio)
            all_lines.append(line)
            all_lines.append(engage_line)
            all_lines.append(disengage_line)
            intersected_lines.append(line)

    if trio_lines:
        closer_stroke = np.inf

        for path in trio_lines:
            if abs(path.line.length - stroke) < closer_stroke:
                closer_line = path
                closer_stroke = abs(path.line.length - stroke)

    if info:
        print_info(closer_line, intersected_lines, unit)

    return all_lines, intersected_lines, trio_lines, line_stroke,
closer_line

#

```

```

def compliant_stroke(trajectory, stroke, tol_stroke):

    lines = []

    for start in range(len(trajectory)):
        x_start, y_start = trajectory[start]

        for end in range(1, len(trajectory) - 1):
            x_end, y_end = trajectory[(start + end) % len(trajectory)]
            # Go further in the trajectory until the straight line is
greater than min_length
            line_length = np.hypot(x_end - x_start, y_end - y_start)

            if line_length < stroke:
                continue

```

```

        else: # Start looking deviations once the straight line has
reached the minimum stroke required
            # Computation of perpendicular deviations
            u = [(x_end - x_start) / line_length, (y_end - y_start) /
line_length]

            if end + start < len(trajectory):
                # Calculate deviations for range from start to end
                deviations = [abs((x - x_start) * u[1] - (y - y_start)
* u[0])
                            for x, y in trajectory[start + 1:start +
end]]

            else: # Case when the trajectory begins in a straight line
array
                # Calculate deviations for range from start to end of
array
                deviations1 = [abs((x - x_start) * u[1] - (y - y_start)
* u[0]) for x, y in trajectory[start + 1:]]
                # Calculate deviations for range from beginning of
array to end
                deviations2 = [abs((x - x_start) * u[1] - (y - y_start)
* u[0])
                              for x, y in trajectory[::(start + end) %
(len(trajectory) - 1)]]
                # Combine deviations from both ranges
                deviations = deviations1 + deviations2

            # Check if all deviations are below tolerance
            if len(deviations) and max(deviations) <= tol_stroke:
                line = LineClass()
                line.length = line_length
                line.start = [x_start, y_start]
                line.end = [x_end, y_end]
                line.deviation = max(deviations)

                if line not in lines:
                    lines.append(line)
            else:
                break
    return lines

#

def engage(line, trajectory, engage_length, tol_engage, cosine_tol):

    # End of the engaging line is the beginning of the straight line
    x_end, y_end = line.start
    # Find index in the trajectory to later move away from it and find the
longest engaging_line
    index = np.where((trajectory[:, 0] == x_end) & (trajectory[:, 1] ==
y_end))[0][0]
    engage_line = {}

    # Go as far as we can from straight_line to find the longest line
    for start in range(1, len(trajectory) + 1):
        # Find current length of the considered engaging_line

```

```

    x_start, y_start = trajectory[index + (-start)][0],
trajectory[index + (- start)][1]
    line_length = np.hypot(x_end - x_start, y_end - y_start)

    # Go further in trajectory until its length is greater than
tolerance
    if line_length < engage_length:
        continue

    # Once a line greater than tolerance is found, u is used to compute
its deviations from the trajectory
    # Later, max(deviation) < tolerance is going to be checked
    u = [(x_end - x_start) / line_length, (y_end - y_start) /
line_length]

    # Case when engaging line indices does not cross the index 0
    if index - start + 1 > 0:
        # Calculate deviations for range from start to end
        deviations = [abs((x - x_start) * u[1] - (y - y_start) * u[0])
            for x, y in trajectory[index - start + 1: index]]

    # Case when start is negative index = straight line begin at end of
trajectory
    else:
        # Calculate deviations for range from index - start to end of
array
        deviations1 = [abs((x - x_start) * u[1] - (y - y_start) * u[0])
            for x, y in trajectory[index - start + 1:]]
        # Calculate deviations for range from beginning of array to end
        deviations2 = [abs((x - x_start) * u[1] - (y - y_start) * u[0])
            for x, y in trajectory[:index]]
        # Combine deviations from both ranges
        deviations = deviations1 + deviations2

    # Check if all deviations are below tolerance
    if len(deviations) and max(deviations) <= tol_engage:
        current_line = EngagingClass()
        current_line.length = line_length
        current_line.start = [x_start, y_start]
        current_line.end = [x_end, y_end]
        current_line.deviation = max(deviations)
        cosine_angle = compute_cosine_angle(line, current_line)

        # If maximum deviation < tolerance, check if tolerance on
cosine satisfied.
        if abs(cosine_angle) < cosine_tol:
            engage_line = current_line
            engage_line.cosine = cosine_angle

            return engage_line
        else:
            break
    else:
        break

return engage_line

#

```

```

def disengage(line, trajectory, engage_length, tol_engage, cosine_tol):

    # Find existence of engaging line
    x_start, y_start = line.end
    # Find index in the trajectory
    index = np.where((trajectory[:, 0] == x_start) & (trajectory[:, 1] ==
y_start))[0][0]
    disengage_line = {}

    for start in range(1, len(trajectory) + 1):
        x_end, y_end = [trajectory[(index + start) % len(trajectory)][0],
            trajectory[(index + start) % len(trajectory)][1]]
        line_length = np.hypot(x_end - x_start, y_end - y_start)

        if line_length < engage_length:
            continue

        u = [(x_end - x_start) / line_length, (y_end - y_start) /
line_length]

        if index + start < len(trajectory):
            # Calculate deviations for range from start to end
            deviations = [abs((x - x_start) * u[1] - (y - y_start) * u[0])
                for x, y in trajectory[index + 1: index + start]]

        else: # Case when start is negative index = straight line begin at
end of trajectory
            # Calculate deviations for range from start to end of array
            deviations1 = [abs((x - x_start) * u[1] - (y - y_start) * u[0])
                for x, y in trajectory[index:]]
            # Calculate deviations for range from beginning of array to end
            deviations2 = [abs((x - x_start) * u[1] - (y - y_start) * u[0])
                for x, y in trajectory[:index + start -
len(trajectory)]]
            # Combine deviations from both ranges
            deviations = deviations1 + deviations2

        # Check if all deviations are below tolerance
        if len(deviations) and max(deviations) <= tol_engage:
            current_line = EngagingClass()
            current_line.length = line_length
            current_line.start = [x_start, y_start]
            current_line.end = [x_end, y_end]
            current_line.deviation = max(deviations)
            cosine_angle = compute_cosine_angle(line, current_line)

            if abs(cosine_angle) < cosine_tol:
                disengage_line = current_line
                disengage_line.cosine = cosine_angle

                return disengage_line
            else:
                break
        else:
            break

    return disengage_line

```

```

#


---


def compute_cosine_angle(line, can_engage):
    # Compute the vectors of the two segments
    x1, x2, x3, x4 = line.start[0], line.end[0], can_engage.start[0],
can_engage.end[0]
    y1, y2, y3, y4 = line.start[1], line.end[1], can_engage.start[1],
can_engage.end[1]
    v1 = [x2 - x1, y2 - y1]
    v2 = [x4 - x3, y4 - y3]
    # Compute the magnitudes of the vectors
    magnitude1 = math.sqrt(v1[0] ** 2 + v1[1] ** 2)
    magnitude2 = math.sqrt(v2[0] ** 2 + v2[1] ** 2)
    # Compute the dot product of the vectors
    dot_prod = v1[0] * v2[0] + v1[1] * v2[1]
    # Compute the cosine of the angle
    if magnitude1 * magnitude2:
        cosine_angle = - dot_prod / (magnitude1 * magnitude2) # Minus sign
is due to the vector directions
    else:
        cosine_angle = 2

    return cosine_angle

#


---


def print_info(closer_line, intersected_lines, unit):
    if not intersected_lines:
        print("No stroke being intersected twice and respecting tolerance
have been found")

    else:
        max_length = max(line.length for line in intersected_lines)
        print(f"%f straight line(s) that can be engaged while respecting
tolerance have been found, "
            f"longest line = %.2f {unit} \n" % (len(intersected_lines),
max_length))

        if closer_line:
            print(f"Closest line from stroke:\n length:
{closer_line.line.length:.2f} {unit}, "
                f"start = ({closer_line.line.start[0]:.2f},
{closer_line.line.start[1]:.2f}), "
                f"end = ({closer_line.line.end[0]:.2f},
{closer_line.line.end[1]:.2f}),"
                f" deviation max = {closer_line.line.deviation:.2f}
{unit}\n")

            print(f"Engagement line:\n length:
{closer_line.engaging.length:.2f} {unit}, "
                f"cosine: {closer_line.engaging.cosine:.2f}, Start
(x, y):")

```

```

        f" ({closer_line.engaging.start[0]:.2f},
{closer_line.engaging.start[1]:.2f}), "
        f" End (x,y):"
        f" ({closer_line.engaging.end[0]:.2f},
{closer_line.engaging.end[1]:.2f})\n")

    print(f"Disengagement line:\n length:
{closer_line.disengaging.length:.2f} {unit}, "
        f"cosine: {closer_line.disengaging.cosine:.2f}, Start
(x,y):"
        f" ({closer_line.disengaging.start[0]:.2f},
{closer_line.disengaging.start[1]:.2f}), "
        f" End (x,y):"
        f" ({closer_line.disengaging.end[0]:.2f},
{closer_line.disengaging.end[1]:.2f})\n")

    for line in intersected_lines: # Write information about lines
length and coordinates
        print(f"Length: {line.length:.2f} {unit}, Start (x,y):
({line.start[0]:.2f} , "
            f"{line.start[1]:.2f} ) {unit}, End (x,y):
({line.end[0]:.2f}, {line.end[1]:.2f}) "
            f"{unit}")

```

7.14.4 Motion profiles analysis

Speed_planning_class:

```

__author__ = 'Cédric Keutgen'
from pyscurve import ScurvePlanner
import roboticstoolbox as robot
import numpy as np

"""Point to point trajectory"""

def s_curve(closer_path, v_start, v_end, v_max, a_max, j_max, t):
    tr = []
    motion_profile = []

    for i in range(len(v_max)):
        x = [closer_path.line.start[i]]
        y = [closer_path.line.end[i]]

        p = ScurvePlanner()
        tr.append(p.plan_trajectory(x, y, [v_start[i]], [v_end[i]],
v_max[i], a_max[i], j_max[i], t=t[i]))

```

```

        motion_profile.append(get_s_traj_profile(tr[i])[0][0])

    return tr, motion_profile

#

```

```

def get_s_traj_profile(traj, timesteps=100):

    # acceleration_id = 0
    # speed_id = 1
    # position_id = 2
    # plt.plot(time, profile[acceleration_id][:])

    dof = traj.dof
    # timesteps = int(max(traj.time) / dt)
    time = np.linspace(0, max(traj.time), timesteps)

    # NOW
    # profiles[t]          --- profiles for each DOF at time x[t]
    # profiles[t][d]      --- profile for d DOF at time x[t]
    # profiles[t][d][k]   --- accel/vel/pos profile for d DOF at time
x[t]
    p_list = [traj(t) for t in time]
    profiles = np.asarray(p_list)

    # NEED
    # profiles[d]          --- profiles for each DOF 0 <= d <= DOF number
    # profiles[d][k]      --- accel/vel/pos profile for DOF d where j
    # profiles[d][k][t]   --- accel/vel/pos at time x[k] for DOF i
    # profiles = np.reshape(profiles, (dof, 3, timesteps))

    r_profiles = np.zeros((dof, 3, timesteps))
    for d in range(dof):
        for p in range(3):
            r_profiles[d, p, :] = profiles[:, d, p]

    return r_profiles, dof, time

#

```

```

def trapz(q0, q1, t):
    return robot.mtraj(robot.trapezoidal, q0, q1, t)

#

```

```

def quintic(q0, q1, t):
    return robot.mtraj(robot.quintic, q0, q1, t)

#

```



```

"""Trajectory with waypoints"""

class ViaPoints:
    def __init__(self, coordinates=None, passing_point=None,
passing_time=None):
        self.coordinates = coordinates
        self.passing_point = passing_point
        self.passing_time = passing_time

#

class LspbTraj:
    def __init__(self, q=None, qd=None, qdd=None, viapoints=None, t=None):
        self.q = q
        self.qd = qd
        self.qdd = qdd
        self.viapoints = viapoints
        self.t = t

#

def lspb(closer_line, t):

    """Waypoints parameters"""

    ts = [0.4*t[-1], (1-2*0.4)*t[-1], 0.4*t[-1]] # 40% of the total time
is taken from engagement and disengagement
    tacc = 0.1 # Acceleration time to round edged.

    dt = 1/len(t)

    qd0 = np.array([0])
    qdf = np.array([0])
    viapoint = get_viapoints(closer_line, ts)
    position = robot.mstraj(viapoint.coordinates, dt, tacc, tsegment=ts,
qd0=qd0, qdf=qdf)

    traj = LspbTraj()
    traj.q = np.transpose(position.q)[0]
    traj.qd = np.gradient(np.transpose(traj.q), position.t)
    traj.qdd = np.gradient(traj.qd, position.t)
    traj.viapoints = viapoint
    traj.t = np.linspace(0, t[-1], len(traj.q))

    return traj

#

```

```

def get_viapoints(closer_line, ts):

    # initial and final positions
    q0 =
np.array([np.sqrt((closer_line.line.start[0])**2+closer_line.line.start[1]*
*2)])
    qf =
np.array([np.sqrt((closer_line.line.end[0])**2+closer_line.line.end[1]**2)]
)
    # Via points
    q1 =
np.array([np.sqrt((closer_line.disengaging.end[0])**2+closer_line.disengagi
ng.end[1]**2)])
    q2 =
np.array([np.sqrt((closer_line.engaging.start[0])**2+closer_line.engaging.s
tart[1]**2)])
    viapoints = ViaPoints()
    viapoints.coordinates = np.array([q0, q1, q2, qf])
    viapoints.passing_points = np.array([q1, q2])
    viapoints.passing_times = [ts[0], ts[0]+ts[1]]

    return viapoints

```

Box_transfer_class_with_velociti es:

```

__author__ = 'Cédric Keutgen'
import numpy as np
import Plot_data_class as Data
import Solve_equations as Equations
import Straight_line_class as Find
import math
import Speed_planning_class as Speed
from Straight_line_class import TrioLineClass, LineClass, EngagingClass

"""
Design criteria
"""

min_stroke = 216 # Minimum linear conveying sought
tol_stroke = 5 # Max trajectory deviation from the straight line

engage_length = 15 # Minimum length of straight lines allowing to engage
the linear trajectory
tol_engage = 5 # Max trajectory deviation from the straight line

cosine_tol = math.cos(math.radians(0)) # Maximum cosine between engagement
lines and stroke (goal: near 90°)

#

```

```

"""
Motion profile parameters
"""

v_start = [0, 0]
v_end = [0, 0]
v_max = [300, 300]
a_max = [6000, 6000]
j_max = [10000, 10000]
cycle_time = [1.2, 1.2] # [Time for stroke, Time for retreat]
t = np.transpose(np.linspace(0, cycle_time, 100))

#

"""
Information display parameters
"""

line_width = 2 # Line width used for plot display
only_stroke = False # Put to True for only stroke lines and no care about
interceptions
info = True # Put to False for no information display on results found

#

"""
Beginning of the main code
"""

a = 1.322 # Coefficient to scale the parameters, and so the trajectory.

# Units for plot based on the principle that parameters are mm. In
practice, parameters have arbitrary units
a_unit = {1: "[mm]", 0.1: "[cm]", 0.01: "[dm]", 0.001: "[m]", -np.inf:
"[?]" }

unit = next(v for k, v in a_unit.items() if a >= k)

x0 = 198*a # [mm]
y0 = 240*a
L1 = 234.75*a # [mm]

theta_range = np.linspace(0, 2 * np.pi, 100) # Define the range of theta
values [rad]
theta_degree = theta_range * 360 / (2 * np.pi) # Define the range of theta
values [°]

xp = np.zeros(len(theta_range)) # Trajectory vector
yp = np.zeros(len(theta_range)) # Trajectory vector

S1 = 108
S4 = 319
S5 = 217
R = 91.95*a # [mm]
s = {k: a * v for k, v in {'S1': S1, 'S4': S4, 'S5': S5}.items()} # Scale

```

```

the triangle
print(R)

def box_transfer(s, x0, y0, L1, min_stroke, engage_length, tol_stroke,
tol_engage, cosine_tol):

    assembly, grashoft = Equations.assembly_and_grashoft(s, R, L1, x0, y0)

    # If they are respected, solve equations
    if assembly and grashoft:
        # Newton-Raphson to find B4 and B5
        [B4, B5] = Equations.solve_b4_b5(s)

        # fsolve to find A1 and A2
        [A1, A2] = Equations.solve_a1_a2(theta_range, [B4, B5], s, L1, R,
x0, y0)

        # Compute velocity and velocity derivative coefficients for each
value of theta
        [K_A1, K_A2, L_A1, L_A2] = Equations.solve_ka_la(theta_range, [A1,
A2], [B4, B5], s, L1, R)
        [K_px, K_py, L_px, L_py, K_p, L_p] = Equations.solve_kp_lp([K_A1,
K_A2], [L_A1, L_A2], [A1, A2], s, L1)
        print(max(abs(np.diff(K_p))))

        [xp, yp] = Equations.solve_trajectory(L1, [A1, A2], s)

        lines, intersected_lines, trio_lines, line_stroke, closer_line =
Find.conveying_path(
            xp, yp, min_stroke, engage_length, tol_stroke, tol_engage,
cosine_tol, unit, info=info,
            only_stroke=only_stroke)
        Data.anim_trajectory(xp, yp, s, lines, x0, y0, L1, R, line_width,
unit) # Trajectory + lines
        # Data.plot_all_coefficients(theta_degree, K_px, K_py, L_A1, L_px,
L_A2, L_py, K_p, L_p)

        if max(abs(np.diff(K_p))) <= 20:

            # Trajectory of point P
            if closer_line:
                # Compute an S-curve speed profile for the stroke
                stroke_speed, motion_profiles = Speed.s_curve(closer_line,
v_start, v_end, v_max, a_max, j_max,
                                                                    cycle_time)

                # Convert the previous profile in terms of motor
requirements
                print('\nS-curve for stroke:')
                motor_speed, motor_acc, motor_speed_rpm, motor_acc_rpm =
Equations.theta_speed_acc(motion_profiles,
K_p, L_p, info)
                print('Max comb speed = %.2f [mm/s]' %
(max(abs(motion_profiles[0][1][:])))
                print('Max comb acc = %.2f [mm/s^2]\n' %
(max(abs(motion_profiles[0][0][:])))
                Data.plot_s_profile(stroke_speed, cycle_time, unit,
line_width)

```

```

        Data.plot_motor(motor_speed, motor_acc, t)
        # Compute other speed profiles for comparisons

        # Motion_profile = Speed.motiontype(p_engage = 0,
p_disengage = closest_stroke_found, t = [Filling time, retreat time]
        trapz_profile = Speed.trapz(0, closer_line.line.length,
t[0]) # p_disengage = closest_stroke_found
        quintic_profile = Speed.quintic(0, closer_line.line.length,
t[0]) #
        = closer_line.line.length

        print('Trapezoidal for stroke:')
        motor_trapz_speed, motor_trapz_acc, motor_trapz_speed_rpm,
motor_trapz_acc_rpm = Equations.theta_speed_acc2(trapz_profile,
K_p, L_p, info)
        print('Max comb speed = %.2f [mm/s]' %
(max(abs(np.transpose(trapz_profile.qd)[0])))
        print('Max comb acc = %.2f [mm/s^2]\n' %
(max(abs(np.transpose(trapz_profile.qdd)[0])))

        print('Quintic for stroke:')
        motor_quintic_speed, motor_quintic_acc,
motor_quintic_speed_rpm, motor_quintic_acc_rpm =
Equations.theta_speed_acc2(quintic_profile,
K_p, L_p, info)
        print('Max comb speed = %.2f [mm/s]' %
(max(abs(np.transpose(quintic_profile.qd)[0])))
        print('Max comb acc = %.2f [mm/s^2]\n' %
(max(abs(np.transpose(quintic_profile.qdd)[0])))

        # max_motor_speed = max(max(abs(motor_speed[0])),
max(abs(lspb_profile.qd))) # rad/s

        print('lspb profile for retreat:')

        lspb_profile = Speed.lspb(closer_line, t[1])
        motor_lspb_speed, motor_lspb_acc, motor_lspb_speed_rpm,
motor_lspb_acc_rpm = Equations.theta_speed_acc2(
        lspb_profile,
        K_p, L_p, info)
        print('Max comb speed = %.2f [mm/s]' %
(max(abs(lspb_profile.qd)))
        print('Max comb acc = %.2f [mm/s^2]\n' %
(max(abs(lspb_profile.qdd)))

        Data.plot_speed_profile(['Trapezoidal', 'Quintic', 'linear
segment with parabolic blend'],
                                [trapz_profile, quintic_profile,
lspb_profile],
                                line_width,
via_points=lspb_profile.viapoints)
        Data.plot_motor2([motor_trapz_speed, motor_quintic_speed,
motor_lspb_speed],
                        [motor_trapz_acc, motor_quintic_acc,
motor_lspb_acc],
                        [t[0], t[0], t[1]])

        max_motor_speed = max(abs(lspb_profile.qd)) # rad/s
    else:

```

```

        max_motor_speed = []
    else:
        closer_line = []
        max_motor_speed = []

        """
        Data.plot_s_profile(stroke_speed, cycle_time, unit, line_width)
        Data.plot_motor(motor_speed, motor_acc, t)
        Data.plot_speed_profile(['Trapezoidal', 'Quintic', 'linear segment
with parabolic blend'],
                                [trapz_profile, quintic_profile,
lspb_profile],
                                line_width,
via_points=lspb_profile.viapoints)
        Data.plot_all_coefficients(theta_degree, K_px, K_py, L_A1, L_px,
L_A2, L_py, K_p, L_p)
        """
        """
        [x_p_speed, y_p_speed, p_speed] =
Equations.solve_p_speed(theta_speed, [K_px, K_py])
        [x_p_acc, y_p_acc, p_acc] = Equations.solve_p_acc(theta_acc,
theta_speed, [K_px, K_py], [L_px, L_py])
        """

    else:
        closer_line = []
        max_motor_speed = []

    return closer_line, max_motor_speed

closer_line, motor_speed = box_transfer(s, x0, y0, L1, min_stroke,
engage_length, tol_stroke, tol_engage, cosine_tol)

```

7.14.5 Multi-start genetic algorithm applied to the box transport mechanism

Box_optimize_main:

```

import numpy as np
import math
from Box_optimize import msga_constrained_main as msga
import matplotlib.pyplot as plt

"""
Design criteria
"""

min_stroke = 216 # Minimum linear conveying sought [mm]
tol_stroke = 5 # Max trajectory deviation from the straight line [mm]

```

```

engage_length = 15 # Minimum length of straight lines allowing to engage
the linear trajectory
tol_engage = 5 # Max trajectory deviation from the straight line [mm]

cosine_tol = math.cos(math.radians(10)) # Maximum cosine between
engagement lines and stroke (goal: near 90°)

#
-----

"""
Motion profile parameters
"""

v_start = [0, 0]
v_end = [0, 0]
v_max = [3, 3]
a_max = [160, 200]
j_max = [5000, 1000]
cycle_time = [0.6, 1] # [s]
t = np.transpose(np.linspace(0, cycle_time, 100))

#
-----

"""
Information display parameters
"""

line_width = 2 # Line width used for plot display
only_stroke = False # Put to True for only stroke lines and no care about
interceptions
info = False # Put to False for no information display on results found

#
-----

a = 0.1 # Coefficient to scale the parameters, and so the trajectory.

# Units for plot based on the principle that parameters are mm. In
practice, parameters have arbitrary units
a_unit = {1: "[mm]", 0.1: "[cm]", 0.01: "[dm]", 0.001: "[m]", -np.inf:
"[?]" }

unit = next(v for k, v in a_unit.items() if a >= k)

#
-----

"""
Design parameters: bound
"""

nv = 7 # Number of variables (S1, S4, S5, x0, y0, L1, v)
lb = [100, 100, 100, 0, 0, min_stroke/2, -40] # Lower bounds
ub = [200, 400, 400, 400, 400, 900, 40] # Upper bounds

```

```

ot = -1 # ot = 1 for maximisation, otherwise minimization

"""
Optimization parameters
"""

pop_size = 20
cr = 4 # Crossover_rate
mr = 6 # Mutation_rate
sr = 5 # Local_search_rate
a = 6 # adaptive restart

computing_time = 1000
no_gen = 3000000 # because we use computing time as termination criterion

A = 0

for i in range(1):
    A = msga(pop_size, cr, mr, sr, nv, lb, ub, ot, a, no_gen,
             computing_time, min_stroke, engage_length,
             tol_stroke, tol_engage, cosine_tol)

fig = plt.figure()
ax = fig.add_subplot()
fig.show()
plt.title('Evolutionary process of the objective function value')
plt.xlabel("Iteration")
plt.ylabel("Objective function value")
plt.plot(A, '*', markersize=2, color='red')
plt.show()

```

Box_optimize:

```

import numpy as np
import math
from Box_optimize import msga_constrained_main as msga
import matplotlib.pyplot as plt

"""
Design criteria
"""

min_stroke = 216 # Minimum linear conveying sought [mm]
tol_stroke = 5 # Max trajectory deviation from the straight line [mm]

engage_length = 15 # Minimum length of straight lines allowing to engage
the linear trajectory
tol_engage = 5 # Max trajectory deviation from the straight line [mm]

cosine_tol = math.cos(math.radians(10)) # Maximum cosine between
engagement lines and stroke (goal: near 90°)

#

```

```

"""

```



```

Motion profile parameters
"""

v_start = [0, 0]
v_end = [0, 0]
v_max = [3, 3]
a_max = [160, 200]
j_max = [5000, 1000]
cycle_time = [0.6, 1] # [s]
t = np.transpose(np.linspace(0, cycle_time, 100))

#

---

"""
Information display parameters
"""

line_width = 2 # Line width used for plot display
only_stroke = False # Put to True for only stroke lines and no care about
interceptions
info = False # Put to False for no information display on results found

#

---

a = 0.1 # Coefficient to scale the parameters, and so the trajectory.

# Units for plot based on the principle that parameters are mm. In
practice, parameters have arbitrary units
a_unit = {1: "[mm]", 0.1: "[cm]", 0.01: "[dm]", 0.001: "[m]", -np.inf:
"[?]" }

unit = next(v for k, v in a_unit.items() if a >= k)

#

---

"""
Design parameters: bound
"""

nv = 7 # Number of variables (S1, S4, S5, x0, y0, L1, v)
lb = [100, 100, 100, 0, 0, min_stroke/2, -40] # Lower bounds
ub = [200, 400, 400, 400, 400, 900, 40] # Upper bounds
ot = -1 # ot = 1 for maximisation, otherwise minimization

"""
Optimization parameters
"""

pop_size = 20
cr = 4 # Crossover_rate
mr = 6 # Mutation_rate
sr = 5 # Local_search_rate
a = 6 # adaptive restart

```

```

computing_time = 1000
no_gen = 3000000 # because we use computing time as termination criterion

A = 0

for i in range(1):
    A = msga(pop_size, cr, mr, sr, nv, lb, ub, ot, a, no_gen,
             computing_time, min_stroke, engage_length,
             tol_stroke, tol_engage, cosine_tol)

fig = plt.figure()
ax = fig.add_subplot()
fig.show()
plt.title('Evolutionary process of the objective function value')
plt.xlabel("Iteration")
plt.ylabel("Objective function value")
plt.plot(A, '*', markersize=2, color='red')
plt.show()

```

Box_transfer_class_without_constraints:

```

__author__ = 'Cédric Keutgen'
import numpy as np
import Plot_data_class as Data
import Solve_equations as Equations
import Straight_line_class as Find
import math
import Speed_planning_class as Speed
from Straight_line_class import TrioLineClass, LineClass, EngagingClass

"""
Design parameters
"""

min_stroke = 120 # Minimum linear conveying sought [cm]
tol_stroke = 5 # Max trajectory deviation from the straight line [cm]

engage_length = 1 # Minimum length of straight lines allowing to engage
the linear trajectory
tol_engage = 3 # Max trajectory deviation from the straight line [cm]

cosine_tol = 0.3 # Maximum cosine between engagement lines and stroke
(goal: near 90°)

#

"""
Motion profile parameters
"""

v_start = [0, 0]

```

```

v_end = [0, 0]
v_max = [300, 300]
a_max = [6000, 6000]
j_max = [5000, 5000]
cycle_time = [0.6, 1] # [s]
t = np.transpose(np.linspace(0, cycle_time, 100))

#
---
"""
Information display parameters
"""

line_width = 2 # Line width used for plot display
only_stroke = False # Put to True for only stroke lines and no care about
interceptions
info = True # Put to False for no information display on results found

#
---
"""
Beginning of the main code
"""

a = 1 # Coefficient to scale the parameters, and so the trajectory.

# Units for plot based on the principle that parameters are mm. In
practice, parameters have arbitrary units
a_unit = {1: "[mm]", 0.1: "[cm]", 0.01: "[dm]", 0.001: "[m]", -np.inf:
"[?]" }

unit = next(v for k, v in a_unit.items() if a >= k)

x0 = 210 * a # [mm]
y0 = x0 * 0.95
L1 = 200 / 15 * min_stroke * a # [mm]

theta_range = np.linspace(0, 2 * np.pi, 100) # Define the range of theta
values [rad]
theta_degree = theta_range * 360 / (2 * np.pi) # Define the range of theta
values [°]

xp = np.zeros(len(theta_range)) # Trajectory vector
yp = np.zeros(len(theta_range)) # Trajectory vector

S1 = 108
S4 = 319
S5 = 217

s = {k: a * v for k, v in {'S1': S1, 'S4': S4, 'S5': S5}.items()} # Scale
the triangle

def box_transfer(s, x0, y0, L1, min_stroke, engage_length, tol_stroke,
tol_engage, cosine_tol):

```

```

R = (L1 * 3.14321 / 180) * math.degrees(math.asin(min_stroke / (2 *
L1))) # [mm]
# Test conditions for crank rocker mechanism

# Newton-Raphson to find B4 and B5
[B4, B5] = Equations.solve_b4_b5(s)

# fsolve to find A1 and A2
[A1, A2] = Equations.solve_a1_a2(theta_range, [B4, B5], s, L1, R, x0,
y0)

# Compute velocity and velocity derivative coefficients for each value
of theta
[K_A1, K_A2, L_A1, L_A2] = Equations.solve_ka_la(theta_range, [A1, A2],
[B4, B5], s, L1, R)
[K_px, K_py, L_px, L_py, K_p, L_p] = Equations.solve_kp_lp([K_A1,
K_A2], [L_A1, L_A2], [A1, A2], s, L1)

if max(abs(np.diff(K_p))) <= 10:

    # Trajectory of point P
    [xp, yp] = Equations.solve_trajectory(L1, [A1, A2], s)

    lines, intersected_lines, trio_lines, line_stroke, closer_line =
Find.conveying_path(
        xp, yp, min_stroke, engage_length, tol_stroke, tol_engage,
cosine_tol, unit, info=info, only_stroke=only_stroke)

    if closer_line:
        # Compute an S-curve speed profile for the stroke
        Data.anim_trajectory(xp, yp, s, lines, x0, y0, L1, R,
line_width, unit)

        # stroke_speed, motion_profiles = Speed.s_curve(closer_line,
v_start, v_end, v_max, a_max, j_max, cycle_time)

        # Convert the previous profile in terms of motor requirements
        # motor_speed, motor_acc, motor_speed_rpm, motor_acc_rpm =
Equations.theta_speed_acc(motion_profiles, K_p, L_p, info)

        # Compute other speed profiles for comparisons
        trapz_profile = Speed.trapz(-4.6, -16.8, t[0])
        quintic_profile = Speed.quintic(-4.6, -16.8, t[0])
        lspb_profile = Speed.lspb(closer_line, t[1])

        # max_motor_speed = max(max(abs(motor_speed[0])),
max(abs(lspb_profile.qd))) # rad/s
        max_motor_speed = max(abs(lspb_profile.qd)) # rad/s

    else:
        max_motor_speed = []

    """
    Data.anim_trajectory(xp, yp, s, lines, line_width, unit) #
Trajectory + lines
    Data.plot_s_profile(stroke_speed, cycle_time, unit, line_width)
    Data.plot_motor(motor_speed, motor_acc, t)
    Data.plot_speed_profile(['Trapezoidal', 'Quintic', 'linear segment
with parabolic blend'],

```

```

                                [trapz_profile, quintic_profile,
lspb_profile],
                                line_width,
via_points=lspb_profile.viapoints)
    Data.plot_all_coefficients(theta_degree, K_px, K_py, L_A1, L_px,
L_A2, L_py, K_p, L_p)
    """
    """
    [x_p_speed, y_p_speed, p_speed] =
Equations.solve_p_speed(theta_speed, [K_px, K_py])
    [x_p_acc, y_p_acc, p_acc] = Equations.solve_p_acc(theta_acc,
theta_speed, [K_px, K_py], [L_px, L_py])
    """

    else:
        closer_line = []
        max_motor_speed = []

    return closer_line, max_motor_speed

closer_line, motor_speed = box_transfer(s, x0, y0, L1, min_stroke,
engage_length, tol_stroke, tol_engage, cosine_tol)

```

