Optimal Trajectory Planning for a Robotic Manipulator Palletizing Tasks*

Fabio Parisi  
Electrical and Information Engineering  
Polytechnic University of Bari  
Bari, Italy  
fabio.parisi@poliba.it

Agostino Marcello Mangini  
Electrical and Information Engineering  
Polytechnic University of Bari  
Member IEEE  
Bari, Italy  
agostinomarcello.mangini@poliba.it

Maria Pia Fanti  
Electrical and Information Engineering  
Polytechnic University of Bari  
Fellow IEEE  
Bari, Italy  
mariapia.fanti@poliba.it

Abstract—In recent years, the employment of robots has become a value-added entity in industries in gaining their competitive advantages. Moreover, thanks to Industry 4.0 paradigm, many production tasks have grown in terms of dimensionality, complexity and higher precision and need to be performed by robots. Among them, the palletizing task is still highly dependent on the particular problem to solve, and its optimization needs to be performed basing on the ground condition. In this paper a palletizing task problem performed by a robotic manipulator is studied. More in detail, some objects have to be transported from a pre-determined storage area to a delivery area. In the storage area the objects are stacked one on the other in columns, while in the delivery area the robotic manipulator poses the objects in horizontal levels, one over another. The process is optimized by minimizing the total distance travelled by the robotic manipulator to transport all the objects from the storage area to the delivery area. An Integer Linear Programming (ILP) problem is formalized and tested by simulations and experimental results.

Index Terms—Optimization, robotics, trajectory planning

I. INTRODUCTION

Robotics is one of the Industry 4.0 enabling technologies [3], adopted and declined under different perspectives in different countries [11]. The Industry 4.0 paradigm is pushing towards massive use of automated process in production.

The industrial tasks can be also performed by humans, but the use of robots assumes a key role when high levels of performance are required. The following benefits are observed in companies equipped with a large number of robots: decreased costs of labour, increased flexibility and versatility, higher precision and productivity, better human working conditions and displace human working in hazardous and impractical environment [9], [18].

Robotic simulation covers the visualization and analysis of how the robot moves through its environment and it allows processes and tasks to be studied, designed and highly rationalized, in order to achieve the maximum benefits in terms of time and cost [14], [15].

Robotics manipulator are used in industries for many different tasks like machining [7], handling and pick-and-place, palletizing, welding, painting and cooperation with humans in specific tasks. Some papers investigate and study the kinematics model of new robotic manipulators. For example, [17] analyses the kinematic characteristic of the dual-beam laser cooperative welding robot with multiple manipulators. The authors in [8] propose a computationally efficient and robust kinematic calibration methodology applied to industrial robots.

However, other researchers addresses the issue of improving the performance of a robotic manipulator in a work cell. Paper [6] proposes a simulation study on a pick and place application, where a palletizer robot moves bags at the end of the production line. The authors focus their attention on the analysis of the performance of the robotic work cell. In particular, the cycle time is improved by simulations. Papers [4] and [10] optimize servo system trajectory minimizing energy. The authors in [12] present an engineering method for reducing the total energy consumption of pick-and-place manipulators for a given end-effector trajectory.

In many industrial applications, the robots are used to perform a set of tasks and the tasks sequencing and how the robot moves between them greatly influence the overall performance of the application [13]. In this context, paper [2] states that is difficult to compare existing approaches because they address various case studies in various industrial applications and various types of robots. For example, Alatartsev et al. [1] propose a new heuristic and effective way to solve such problems by TSPN (Traveling Salesman Problem with Neighbourhoods) method, a combination of TSP (Traveling Salesman Problem) and TPP (Touring-a-sequence-of-Polygons Problem) based on minimizing the Euclidean distance with different characteristics and constraints. Naturally, the characteristics and the constraints are linked to the analysed case study.

*This work was supported by the EU Adriatic-Ionian Programme INTER-REG V-B FUTURE 4.0 - ManuFactUring educatioN and training governance model for IndUstry 4.0 in the Adriatic-Ionian aReA.
This paper deals with the problem of palletizing objects from a pre-determined storage area to a delivery area. In the storage area the objects are stacked in columns, while in the delivery area the robotic manipulator has to pose the objects in horizontal levels, one over another.

The aim is minimizing the length of the travelled path performed by the robotic manipulator to transport the objects from the storage area to the delivery area. The optimal solution is obtained by formalizing the palletizing task problem in an Integer Linear Programming (ILP) problem. In particular, the ILP problem finds the optimal sequence of the moves of the robotic manipulator, by determining the succession of the operations for each object allocated in the storage area. Hence, this work focuses on the scheduling of the tasks that the robotic manipulator has to perform.

Finally, the proposed ILP problem is applied to a real case study involving a limited number of objects and positions. The obtained solution is tested in two experimental frameworks: i) an off-line Grasshopper environmental [16] simulating a KUKA.prc [5]; ii) a real-time test by a KUKA KR3 R540 robot.

The structure of the work is outlined as follows. In Section II the problem description is presented. Section III introduces the notation and formalizes the ILP problem that minimizes the distance travelled by the robotic manipulator. In Section IV the process simulation is performed and an experimental test validates the solution of the ILP problem. Finally, Section V draws the conclusions and enlightens how the paper can be improved.

II. PROBLEM DESCRIPTION

A robotic manipulator has to move a finite number of objects from a starting pre-determined geometric configuration (storage area) to an ending different pre-determined configuration (delivery area).

The starting geometry is configured by disposing the objects to be moved in a set of columns, where the objects are stacked one on the other. The distances between the columns has to take into account also the geometry of the gripper on the end-effector; the manipulator can grip every object only in a single fixed configuration (see for instance Fig. 2).

The ending object configuration is obtained by disposing the objects in a certain number of levels, one over another. At every level, the objects position is shifted of an half of the length referring to the previous one and the next one. The final configuration is similar to the construction of a wall made of bricks.

All the objects have the same geometry, and the robotic manipulator can handle only one object for each transfer operation, starting from the initial position and ending to the final position. The manipulator will perform this global task in a number of sub-tasks depending on the geometric configuration of the problem.

The problem we want to solve is to compute the optimal sequence of actions for moving objects from the starting to the ending configurations in order to minimize the length of the travelled paths.

III. THE MATHEMATICAL PROGRAMMING FORMULATION

In this section we propose a time-indexed formulation of the problem where the decision variables are the pick-up and
delivery operations in the discrete-time instants, in which the operations are scheduled. In order to schedule the operations in discrete time periods, we assume that the handling times are integer parameters in the time-indexed formulation. Moreover, we assume that each operation is performed in one time interval, even if not all the operations have the same durations.

A. Notation

In this subsection the notation necessary for the ILP problem formulation is defined. First, let us define the following sets of elements:

- $n$ is the number of objects to be palletized from the storage area to the delivery area;
- $S = \{S_1, ..., S_i, ..., S_n\}$ is the set of objects in the pre-determined starting positions, with $i = 1, ..., n$; each object is represented by its center of gravity in the geometrical space;
- $E = \{E_1, ..., E_j, ..., E_n\}$ is the set of the pre-determined possible ending positions, with $j = 1, ..., n$; each position is represented by a point in the geometrical space that is the ending destination of the $S_i$ object’s center of gravity after a movement;
- the starting positions are described by $c$ columns where the objects are stacked before being moved;
- the ending positions are configured in a number of $f$ levels;
- $L = \{l_1, ..., l_v, ..., l_f\}$ is the set of the lengths of the $f$ levels, where $l_v$ denotes the length (measured as number of objects) of level $v$ with $v = 1, ..., f$;
- $d_{i,j}$, for $i, j = 1, ..., n$ denotes the length of the sub-task trajectories performed by the end-effector of the robotic manipulator during a pick-up and delivery operation, depending on the starting position $S_i$ and ending position $E_j$. Hence, the end-effector trajectory is given by the union of the $n$ sub-task trajectories.

Moreover, the following parameters and variables describe the generic positions and configurations of the objects:

- $l_i(w) = \sum_{v=1}^{w-1} l_v + 1$ for $w = 2, ..., f$ denotes the first time interval in which the object can be positioned in level $w$;
- $l_i(w) = \sum_{v=1}^{w-1} l_v$ for $w = 1, ..., f$ denotes the last time interval in which the object can be positioned in level $w$.

The time-indexed formulation of the problem considers a planning horizon that is discretized into $n$ time units (t.u.). In addition, a time-indexed binary variable is defined as follows:

$$ y_{i,j}^t = \begin{cases} 1 & \text{if the object is moved from } S_i \text{ to } E_j \text{ at time } t \\ 0 & \text{otherwise.} \end{cases} $$

B. The Integer Programming Problem Formulation

The aim is finding the sequence of the moves of the manipulator that minimizes the length of the travelled path.

Hence, the objective function can be defined as follows:

$$ D_{tot} = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{t=1}^{n} d_{i,j} \cdot y_{i,j}^t $$  \hspace{1cm} (1)

The ILP is the following:

$$ \begin{align*} \text{min } D_{tot} \quad & (2) \\ \text{subject to:} \quad & (3) \\ & \sum_{i=1}^{n} \sum_{j=1}^{n} y_{i,j}^t = 1 \text{ for } t = 1, ..., n \\ & \sum_{i=1}^{n} \sum_{t=1}^{l_1} y_{i,j}^t = 1 \text{ for } j = 1, ..., l_1 \\ & \sum_{i=1}^{n} \sum_{t=1}^{l_i(w)} y_{i,j}^t = 1 \text{ for } j = l_i(w), ..., l_i(w), \ w = 2, ..., f \\ & \sum_{j=1}^{n} \sum_{i=1}^{n} y_{i,j}^t = 1 \text{ for } i = 1, ..., n \\ & \sum_{k=1}^{n} \sum_{j=1}^{n} y_{i,j}^t - \sum_{k=1}^{n} \sum_{j=1}^{n} y_{i,j}^{t+1} = 1 \text{ for } k = 1, ..., n \\ & y_{i,j}^t \in \{0, 1\} \text{ for } i, j, t = 1, ..., n, \end{align*} $$  \hspace{1cm} (8)

- The cost function (1) is the sum of all distances travelled by the end-effector of the robotic manipulator.
- Constraints (3) take into account the unitary robot’s load capacity and imposes that in each time interval the robot moves only one brick at a time.
- Constraints (4) impose that during the first $l_1$ time intervals the first level of the final positions is filled.
- Constraints (5) take into account the configuration of the ending positions staring from the second level $(w = 2)$: the bricks are structured in $f$ levels, each one with its own number of elements. These constraints impose to fill the previous level before starting the next one.
- Constraints (6) impose that each object has to be posed in only one final destination.
- Constraints (7) impose that the robot picks one of the objects on the top of one of the columns. This constraint is imposed by verifying that every previous object in the column has already been picked in one of the previous time intervals.
- Constraints (8) force the decision variables $y_{i,j}^t$ to be binary.

IV. Experimental Results

This section presents an experimental system composed of a robotic manipulator and a limited set of objects to be palletized. The solution of the ILP problem (2) -(8) is tested by two strategies: i) an off-line simulation framework is specified; ii) a real case study is performed.
A. Experimental System Description and ILP Solution

In order to solve the problem, a geometric configuration is set as it is shown in Fig. 1. To define the configuration, the parameters and variables introduced in section III.B are specified as follow:

- the objects to be moved are \( n = 14 \);
- \( S = \{ S_1, ..., S_i, ..., S_{14} \} \) is the set of objects in the initial positions;
- \( E = \{ E_1, ..., E_j, ..., E_{14} \} \) is the set of the ending positions;
- the moving process is discretized in \( t = 14 \) time intervals;
- the starting positions are configured in \( c = 3 \) columns where the bricks are stored;
- the ending positions are configured as in Fig. 1, where the objects are positioned in \( f = 3 \) levels, with \( l_1 = 5 \), \( l_2 = 5 \) and \( l_3 = 4 \);
- \( d_{ij} \) are the elements of matrix \( D \) shown in Tab. I. To compute the elements of the matrix, each sub-task trajectory is structured so that collisions are avoided. Each sub-task trajectory is divided into straight movements by introducing the following fixed point:
  - \( P_{cz} = \{ P_{c1}, ..., P_{cz}, ..., P_{c3} \} \) is a set of fixed point over the columns from where the end-effector starts the vertical movement to pick up the object to be moved;
  - \( Pl_j = \{ Pl_1, ..., Pl_j, ..., Pl_{14} \} \) is a set of fixed point over each ending position from where the end-effector starts the vertical movement to pose the object at the ending position;
  - \( P_0 \) is a fixed point that represents the initial point reached by the end-effector to start each new transport task.

Example 1: Let us consider Fig. 2 that shows the example of the sub-task from \( S_{14} \) to \( E_4 \): the red lines with red numbers represent the discretized trajectory components, while the blue symbols represent the fixed points introduced to discretize the trajectory. Moreover, Fig. 2 shows that each trajectory is composed by the following components:

- a straight component from the initial point \( P_0 \) to the fixed point \( P_{cz} \) over the column of the object to pick up (i.e., from \( P_0 \) to \( P_{c3} \));
- a vertical component from \( P_{cz} \) to the position \( S_i \) of the object to move (i.e., from \( P_{c3} \) to \( S_4 \));
- a vertical component back to the fixed point \( P_{cz} \) from the position \( S_i \) of the object to move (i.e., from \( S_4 \) to \( P_{c3} \));
- a straight component from \( P_{cz} \) to the position over the ending position \( Pl_j \) (i.e., from \( P_{c3} \) to \( Pl_4 \));
- a vertical component from the point \( Pl_j \) to the position \( E_j \) of the object to pose (i.e., from \( Pl_4 \) to \( E_4 \));
- a vertical component back to the point \( Pl_j \) from the position of the delivered object \( E_j \) (i.e., from \( E_4 \) to \( Pl_4 \));
- a straight component from the point \( Pl_j \) to the initial point \( P_0 \) for the next sub-task (i.e., from \( Pl_4 \) to \( P_0 \)).

The solution of the ILP problem is obtained in few seconds in the Matlab framework by a PC equipped with 530 3.40-GHz AMD A4-5300, 16 GB of memory. The outputs of the solution are described by the decision binary variables that identify the sequence of the manipulator moves in each interval time. Then, the result is reported in Table II and the minimum path to perform all the movements is given by the ILP objective function \( D_{tot} = 19.222m \).

B. Simulation Results

Once obtained the optimal solution, the system has been modeled in order to simulate the tasks in an offline environment. The simulation aims at:

- visually inspecting all the process in order to eventually observe critical issues not considered during the mathematical modeling phase;
- estimating the real experimentation duration;
- eventually performing an offline programming to produce the code to directly use on robot environment, depending on the offline simulation software.

The simulation is performed in Grasshopper [16] environment, by modeling the robotic manipulator and motions with KUKA-prc [5]. In particular, Grasshopper is a visual programming language and environment that runs within the Rhinoceros 3D computer-aided design, primarily used to build generative algorithms (see for instance Fig. 3). KUKA-prc also gives the possibility to export a program in KUKA Robot Language (krl) that is directly usable within the robot programming environment.

The simulation is performed by choosing a KUKA KR3 R450, that is the same robotic manipulator used for the real application. In Fig. 2 a screen-shot of the simulation phase depicts the fifth time interval of the solution shown in Table II, where the end-effector moves the objects from \( S_{14} \) to \( E_4 \).
TABLE I
Distances Matrix (mm)

|   |  e1  |  e2  |  e3  |  e4  |  e5  |  e6  |  e7  |  e8  |  e9  |  e10 |  e11 |  e12 |  e13 |  e14 |
|---|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| s1 | 1219 | 1222 | 1242 | 1342 | 1481 | 1205 | 1202 | 1266 | 1404 | 1199 | 1198 | 1184 | 1321 | 1467 |
| s2 | 1269 | 1272 | 1292 | 1392 | 1531 | 1255 | 1252 | 1316 | 1454 | 1249 | 1248 | 1234 | 1371 | 1517 |
| s3 | 1319 | 1322 | 1342 | 1442 | 1581 | 1305 | 1302 | 1366 | 1504 | 1299 | 1298 | 1284 | 1421 | 1567 |
| s4 | 1369 | 1372 | 1392 | 1492 | 1631 | 1335 | 1332 | 1396 | 1534 | 1329 | 1328 | 1314 | 1447 | 1592 |
| s5 | 1419 | 1422 | 1442 | 1542 | 1681 | 1405 | 1402 | 1466 | 1604 | 1399 | 1398 | 1384 | 1517 | 1667 |
| s6 | 1092 | 1085 | 1094 | 1185 | 1220 | 1071 | 1078 | 1142 | 1280 | 1175 | 1174 | 1160 | 1293 | 1447 |
| s7 | 1142 | 1135 | 1144 | 1235 | 1370 | 1171 | 1178 | 1242 | 1380 | 1275 | 1274 | 1260 | 1393 | 1547 |
| s8 | 1292 | 1285 | 1294 | 1385 | 1520 | 1271 | 1278 | 1342 | 1480 | 1375 | 1374 | 1360 | 1493 | 1647 |
| s9 | 1438 | 1431 | 1437 | 1522 | 1650 | 1421 | 1428 | 1492 | 1630 | 1525 | 1524 | 1510 | 1643 | 1793 |
| s10| 1488 | 1481 | 1487 | 1572 | 1700 | 1471 | 1478 | 1550 | 1688 | 1583 | 1582 | 1568 | 1708 | 1858 |
| s11| 1538 | 1531 | 1537 | 1622 | 1750 | 1521 | 1518 | 1594 | 1732 | 1627 | 1626 | 1612 | 1747 | 1897 |
| s12| 1588 | 1581 | 1587 | 1672 | 1800 | 1571 | 1568 | 1651 | 1789 | 1684 | 1683 | 1669 | 1808 | 1958 |

TABLE II
Optimal Solution

| Time intervals | Starting position | Ending position |
|----------------|------------------|-----------------|
| 1              | 11               | 5               |
| 2              | 12               | 3               |
| 3              | 13               | 2               |
| 4              | 1                | 1               |
| 5              | 14               | 4               |
| 6              | 6                | 8               |
| 7              | 2                | 7               |
| 8              | 3                | 6               |
| 9              | 7                | 9               |
| 10             | 8                | 12              |
| 11             | 9                | 14              |
| 12             | 10               | 13              |
| 13             | 4                | 10              |
| 14             | 5                | 11              |

Fig. 3. Grasshopper programming environment and KUKA-prc

Note that the end-effector geometric and inertial description is set coherently with geometric and inertial characteristic of the real end-effector.

C. Real Experimentation

The programming phase is performed by using the online approach directly on the workcell, to take into account eventual inaccuracies in the modeling phase. With the KUKA teach pendant (SmartPAD), the exact geometry and travel set modeled in the simulation environment is replicated, in order to compare the simulation with the real experimentation.

The theoretical solution of the problem is tested in an experimental setup involving a KUKA KR3 R540 robot, i.e., a robotic manipulator with a reach of 541 mm and a payload of 3 kg, with main applications in painting/glueing, palletizing/packaging, measuring/inspection, assembling and handling.

The objects to be moved are 3D printed parallelepipeds with dimensions of $25 \times 25 \times 75$ mm and a height of 30 g (see Fig. 4). Fig. 4, 5 and 6 show some phases of the experimentation: starting, intermediate and final phase, respectively.

The speed used for every movement is $sp = 0.4 m/s$. This setting, together with the low height of the bricks, make neglect-able the robot dynamics.

The obtained results reported in Table III take into account also the time needed to grip and release the objects. However, the experimental results are consistent with the theoretical solution obtained by the ILP problem.

V. Conclusions

This research aims at formalizing the problem of tasks sequencing of a robotic manipulator that has to palletize...
objects. In particular, the tasks consist of gathering objects from a pre-determined starting geometrical configuration and collecting them in a specific pre-determined ending geometrical configuration.

The problem is modelled as an Integer Linear Programming (ILP) problem by a time-indexed formulations that considers each operations performed in a time interval and minimizes the length of the end effector travelled paths. The ILP problem solution is tested firstly by simulating the system in a Grasshopper/KUKA-prc environment. Then, the solution is applied by controlling the KUKA KR3 R540 robotic manipulator. The results show how the proposed approach can be easily applied in real environments with limited computational effort.

Future works will focus on the generalization of the movements without using fixed points over the load columns to avoid collisions. Moreover, systems for palletizing large number of tasks will be solved and the optimal solutions will be compared with heuristic and reinforcement learning ones.

Acknowledgment
We thank the Fablab POLIBA laboratory of Polytechnic University of Bari and the Fablab Bitonto Association for the cooperation in this works and the possibility of experimenta-

tion on the KUKA KR3 R450 robotic manipulator.

REFERENCES

[1] S. Alatartsev, V. Mersheeva, M. Augustine, and F. Ortmeier. On optimizing a sequence of robotic tasks. In 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 217–223, 2013.
[2] S. Alatartsev, S. Stellmacher, and F. Ortmeier. Robotic task sequencing problem: A survey. Journal of Intelligent & Robotic Systems, 80:279–298, 2015.
[3] V. Alcacer and Virgilio Cruz-Machado. Scanning the industry 4.0: A literature review on technologies for manufacturing systems. Engineering Science and Technology, an International Journal, 22, 06 2019.
[4] G. Berselli, F. Balugani, M. Pellicciari, and M. Gadaleta. Energy-optimal motions for servo-systems: A comparison of spline interpolants and performance indexes using a cad-based approach. Robotics and Computer-Integrated Manufacturing, 40:55 – 65, 2016.
[5] Association for Robot in Architecture. Kuka-prc. https://www.robotsinarchitecture.org/kuka-prc.
[6] N. A. M. Johari, H. Haron, and A. S. M. Jaya. Robotic modeling and simulation of palletizer robot using workspace5. In Computer Graphics, Imaging and Visualisation (CGIV 2007), pages 217–222, 2007.
[7] Alexandre Klimchik, Alexandre Ambiehl, Sebastien Garnier, Benoit Furet, and Anatol Pashkevich. Efficiency evaluation of robots in machining applications using industrial performance measure. Robotics and Computer-Integrated Manufacturing, 48:12–29, 12 2017.
[8] T. Messay, R. Ordóñez, and E. Marcil. Computationally efficient and robust kinematic calibration methodologies and their application to industrial robots. Robotics and Computer-Integrated Manufacturing, 37:33 – 48, 2016.
[9] C. Mineo, S. Gareth Pierce, P. I. Nicholson, and I. Cooper. Robotic path planning for non-destructive testing – a custom matlab toolbox approach. Robotics and Computer-Integrated Manufacturing, 37:1 – 12, 2016.
[10] E. Oliva, G. Berselli, M. Pellicciari, and A. O. Andrisano. An engineering method for the power flow assessment in servo-actuated automated machinery: Mechatronic modeling and experimental evaluation. Robotics and Computer-Integrated Manufacturing, 38:31 – 41, 2016.
[11] Ercan Oztемel and Samet Gursev. Literature review of industry 4.0 and related technologies. Journal of Intelligent Manufacturing, 07 2018.
[12] M. Pellicciari, G. Berselli, F. Leali, and A. Vergnano. A minimal touch approach for optimizing energy efficiency in pick-and-place manipulators. In 2011 15th International Conference on Advanced Robotics (ICAR), pages 100–105, 2011.
[13] Riani, M. and M. A. Prichichi. Industrial robot trajectory optimization—a review. MATEC Web Conference, 126:02005, 2017.
[14] F. Rubio, C. Llopis-Albert, F. Valero, and J. L. Sufer. Assembly line productivity assessment by comparing optimization-simulation algorithms of trajectory planning for industrial robots. Mathematical Problems in Engineering, June 2015, 2015.
[15] O. M. Ulgen, G. Coffman, M. Rohrer, J. Shore, D. Sly, and D. Wood. Increasing the power and value of manufacturing simulation via collaboration with other analytical tools: a panel discussion. In WSC’99. 1999 Winter Simulation Conference Proceedings. Simulation - A Bridge to the Future’ (Cat. No.99CH37038), volume 1, pages 749–753 vol.1, 1999.
[16] Wikipedia. Grasshopper. https://en.wikipedia.org/wiki/Grasshopper_3D.
[17] Q. Zeng, X. Liu, C. Qiu, and A. Li. Kinematic characteristics analysis of cooperative welding robot with multiple manipulators. In 2019 4th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS), pages 1–5, 2019.
[18] Albert Y. Zomaya. Modelling and Simulation of Robot Manipulators: A Parallel Processing Approach. World Scientific Publishing Co., Inc., USA, 1993.