Proceedings of ASME 2019
International Mechanical Engineering Congress and Exposition
IMECE2019
November 11-14, 2019, Salt Lake City, UT, USA

IMECE2019-11337

AN APPLICATION OF A DIGITAL TWIN TO ROBOTIC SYSTEM DESIGN FOR AN UNSTRUCTURED ENVIRONMENT

Matthew Q. Marshall*, Cameron Redovian
Department of Mechatronics Engineering
Kennesaw State University
Kennesaw, Georgia 30144
mqm@kennesaw.edu

ABSTRACT

An experimentable digital twin is created to aid in a design decision (beginning of life stage) for a robotic system. This product is meant to automate a material-feed system. The robot comprises a six-axis manipulator mounted on a mobile base. Due to variability in the dimensions of the material-feed system and positioning error of the mobile base, the material-placement routine is considered to take place in an unstructured environment. Working therein requires exteroceptive sensors, in this instance taking the form of computer vision. Data from this subsystem are used to match the geometry of the digital twin to the physical environment. This close correspondence between physical and virtual embodiments allows for significant design decisions to be reached from simulated experiments. In this case, two motion-planning approaches are compared and it is determined that the costs associated with implementing the dynamic one in the lab for testing are merited by its ease of use and reliability, since simulation-based control employs all current information.

1 INTRODUCTION

Automation of one step in a certain legacy production process is desired. A digital twin is created to aid in the design (beginning-of-life, or BOL, stage) of the robotic system facilitating this automation. The concept of a digital twin has many industrial applications and the term is interpreted in many ways, but digital twins fundamentally comprise physical products, virtual products, and the connections tying them together [1].

Here, the product is a robotic system. Virtual representations exist in both the Gazebo simulation environment [2] and the rviz 3D visualization tool [3]. Computer vision connects the physical and virtual embodiments. Digital rendering of the physical system and computer vision to update that representation are examples of visual computing, the use of which is key to maximizing the potential of digital twin technology for design, decision making, and control [4].

The greatest returns from a digital twin are expected in a product's middle-of-life (MOL) stage, but there is significant literature showing that it can be leveraged in BOL, as here, or that it's benefits are lifecycle-neutral [5]. An example of using a digital twin (DT) for MOL in a robotic product is the mobile-robot navigation system developed by Gonzales et al. using computer models of an industrial environment, a planning structure, and a robot [6]. Besides the lifecycle stage, another difference between that work and the present paper is that Gonzales et al. assume a static factory environment and here any pertinent changes to the environment are measured using computer vision and passed to the digital twin.

In this work, the DT is utilized during BOL for choosing the best motion-planning method for a robotic manipulator but it is expected to be useful for other design decisions and also in MOL. It is important to create a digital twin early in the lifecycle so that "the information created in each stage … is seamlessly made available to subsequent stages." Extending the usefulness

*Address all correspondence to this author.

FIGURE 1. MATERIAL FRAME AND ROBOT MANIPULATOR.

of a simulation past BOL is predicted to be a significant area of research in the near future [7].

The present work employs a digital twin to inform the decision of which of two path-planning approaches to use for the robotic system under development. Experiments are carried out with the physical system using one method and the task is repeated with the digital twin using the second approach. Using a DT is ideal here because the latter approach (A) requires equipment not currently available in the lab, and (B) circumvents some of the safety features provided by the robot manipulator used in the lab. Visual computing provides cyber-physical equivalence in the DT so that experiments can be performed in realistic scenarios. This "tight integration of virtual and real worlds" is the hallmark of an *experimentable digital twin* (EDT) and "greatly enhances the state of the art in simulation technology" [8].

The paper is organized as follows. The legacy system is described in Section 2, which also provides sketch of the proposed robotic system and motivation for the digital twin. Section 3 contains information about the modeling and physical/virtual connection used in the digital twin, a description of the trials performed to compare the two path-planning methods, and durations for the physical and virtual trajectories executed. Finally, Section 4 gives conclusions from the results and topics for future work.

2 BACKGROUND AND MOTIVATION

A material-feeder system, currently without instrumentation, is to be automated. The system supplies material to a machine in a production process. Currently, each material package is manually loaded onto the frame of the feeder system. A package is placed on a horizontal peg that is one of a dozen arranged in two columns inside a bay; the system contains numerous such bays. A robotic system for loading the material is the product being designed in this work. The system is to be a robotic manipulator mounted on a mobile robot.

A manipulator is pictured in front of the feeder-system frame

FIGURE 2. VARIATION IN PEG POSE.

in Fig. 1. Six black horizontal pegs are visible on the left side of the image. These are half of the pegs in one bay. More pegs in an adjacent bay are seen to the right of these six, though they are partially occluded by the robot.

Owing to the material package's large size relative to the clearance provided in the bay and the fact that the central axis of each peg intersects a vertical member of the frame, careful path planning is required. Two major impediments exist to offline design of paths for loading of the packages on the twelve pegs of a bay. The first is that the current frame dimensions can deviate from those of its design, either due to low fabrication precision or deformation during use. (Consider, for example, the middle peg shown in Fig. 2; its orientation clearly differs from that of the neighboring pegs.) The second obstacle is the positioning accuracy of the mobile-robot base. Due to these two considerations, when the robot parks in front of a bay, there can be greater than ± 5 cm variability in the location of the pegs and frame members relative to the robot coordinate system. The robot can therefore not be pre-programmed with paths for placing packages on the twelve pegs. The proposed solution to this problem is to use exteroceptive sensing to shape the paths.

Once the robot parks in front of a bay, computer vision is used to estimate the pose of a target peg relative to the robot base. A fiducial is placed on each peg to yield a 4×4 homogeneous transformation matrix to the robot base frame from the peg frame, which is defined as the center of the fiducial. See the black and white marker on the peg in Fig. 3. This transformation matrix is generated with the AprilTag tool [9] and an eye-in-hand camera.

Two approaches to path generation are tested. The first re-

FIGURE 3. APRILTAG FIDUCIAL ON PEG.

lies on twelve pre-programmed poses that are close to each of the pegs and are guaranteed not to bring the robot into a collision with the frame even with a ± 5 cm positioning error of the mobile base. From that point, the path comprises waypoints pre-defined relative to the peg so that once the peg pose has been estimated, the values of the waypoints can be calculated relative to the robot base. This is therefore referred to as a manual/hybrid approach in the present paper. The manufacturer's robot controller uses its inverse kinematics solver to guide the manipulator along a linear path in Cartesian space between these waypoints because the package must travel in a straight line to be placed on a peg. For a peg located at or near its nominal dimensions, these waypoints yield a collision-free path for the robot to hang the material, but for one far from normal it is conceivable that a robot link could impinge upon the material frame when executing the motion. A dynamic obstacle-avoidance system is therefore desirable, which provides motivation for the second approach: using the MoveIt! package for the Robot Operating System (ROS) with an RGB-D camera [10].

As mentioned in Section 1, there are two concerns that must be dealt with before testing MoveIt! on the physical product. First, there is no RGB-D camera currently available in the lab. Second, using the ROS driver for the manipulator of Fig. 1 means that the robot's velocity- and force-control features are disabled. For these reasons, simulation is ideal for testing. An experimentable digital twin is created in order to compare MoveIt! to the first method and thereby determine if its benefits potentially outweigh its costs and judge whether it should be implemented in the lab.

There are two criteria used for assessing whether or not MoveIt! ought to be implemented on the physical system for further testing: if it can execute collision-free paths to all pegs and if it is fast enough compared to the manual/automatic hybrid approach. The latter is a concern because it is well known that path planning in MoveIt! often yields circuitous routes. Another reason duration is a concern is that controlling speed of Cartesian movements is not possible with MoveIt!

Placing a package can fail in one of three ways: failure to acquire accurate fiducial pose, attempt to move the robot through a singularity, and collision of the robot or package with the material-storage frame. In practice it is found that adequate lighting can easily be provided to ensure the fiducial is recognized by the machine vision system and that, once the RGB camera is calibrated, the AprilTag tool provides a pose estimate of precision sufficient to hang a package on the peg. The manual/hybrid approach used here avoids the other two failure modes through painstaking selection of the twelve pre-programmed poses mentioned above. The planning library used by MoveIt! ensures that neither singularities nor collisions with the known environment will be problems.

The experimentable digital twin is created by first rendering the fabrication drawings of the structure as a 3D model. Next, a 3D model of the robotic system is added to the simulation and computer-vision data are used to update the model of the current system, specifically the pose of the target peg relative to the robot coordinate system. The only difference between the actual and simulated robot systems is the addition of a robot-mounted RGB-D camera (the Microsoft Kinect) model. The DT is then used to test whether MoveIt! is able to generate a collision-free path to each peg in a bay and to measure how much time execution of each trajectory takes.

3 RESULTS

Preparation for the trials is begun by selecting a start configuration for the robot that is common to all trials. This is shown in Fig. 1. Next, the known-safe pose (discussed in Section 2) for each peg is manually determined in the lab. The final step before commencing trials is to estimate the pose of each peg relative to the robot base using computer vision. For example, the transformation matrix of the fiducial in Fig. 3 (which also appears in Fig. 1) is

$$T = \begin{bmatrix} .940 & .0434 & .338 & .945 \\ -.335 & -.0725 & .939 & .436 \\ .0653 & -.996 & -.0536 & -.0237 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

This translation value (that is, the first three elements of the fourth column) is as expected since the robot x-axis is parallel to the horizontal link shown in Fig. 1 and its z-axis points up. The rotation matrix (the upper-left 3×3 sub-matrix) value also makes sense because the peg coordinate frame is coincident with that of the fiducial, which has its x-axis pointing to the right in Fig. 3 and its y-axis pointing down. The twelve transformation matrices are used for placing the frame model in the digital twin, as well as for path planning in both the physical and the virtual systems.

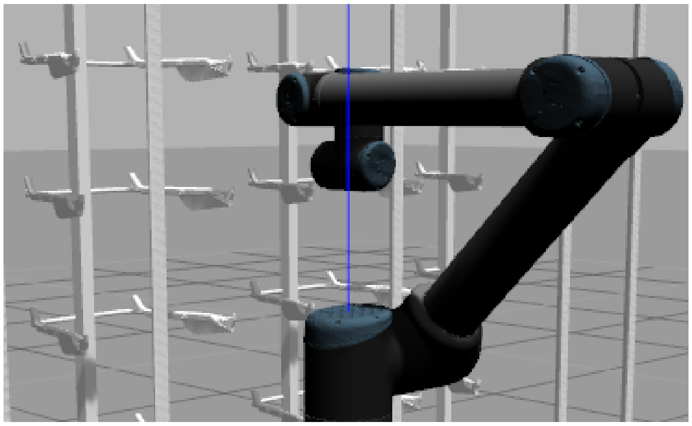

FIGURE 4. FRAME AND MANIPULATOR IN GAZEBO.

3.1 Digital Twin

The DT is restarted for every peg transformation in the interest of closely matching the physical and virtual worlds. Using the parlance of Gazebo, the material frame is spawned so that the target peg is at the same pose relative to the robot coordinate system as the one returned by the eye-in-hand camera and AprilTag. An example of the resulting model is shown in Fig. 4.

The only significant difference between the physical manipulator and its virtual counterpart is that a Microsoft Kinect RGB-D camera is mounted on the latter. As the simulated manipulator moves into the common start pose, point-cloud data from this camera are used by MoveIt! to update the *planning scene*, which models the current state of both the robot and its environment. Figure 5 shows the planning scene visualization (viewed from the opposite side of the frame as Fig. 1) in rviz when representing placement of a package (shown in red) on a particular peg. The voxels in that image depict the the 3D occupancy map that is part of the planning scene generated using the simulated Kinect point-cloud data. The green and blue voxels represent the material frame and the purple voxels the floor.

3.2 Trajectory Execution

The physical robot executed its material-placing routine for each of the pegs in a bay. In this routine, the robot starts at the common configuration, moves in joint space to the appropriate known-safe pose, thence executes a series of Cartesian motions to place the package on the peg using the estimated fiducial pose and the pre-programmed waypoints described in Section 2, finally, this series of motions is reversed to return the manipulator to start.

The simulated robot begins its material-placement routine at the same start configuration as the physical manipulator does. The path used by MoveIt! comes by way of the Open Motion Planning Library (OMPL), which makes more than two dozen planners available. The default in MoveIt! is the RRT Connect, a variation of the Rapidly-exploring Random Trees planner. From the start pose, the simulated robot executes the trajectory determined by the Open Motion Planning Library and MoveIt! After placing the package on a peg, MoveIt! returns the simulated manipulator to the start. MoveIt! successfully generated a path to each of these pegs. This is one important result sought from the simulation. The other is a comparison of trajectory durations.

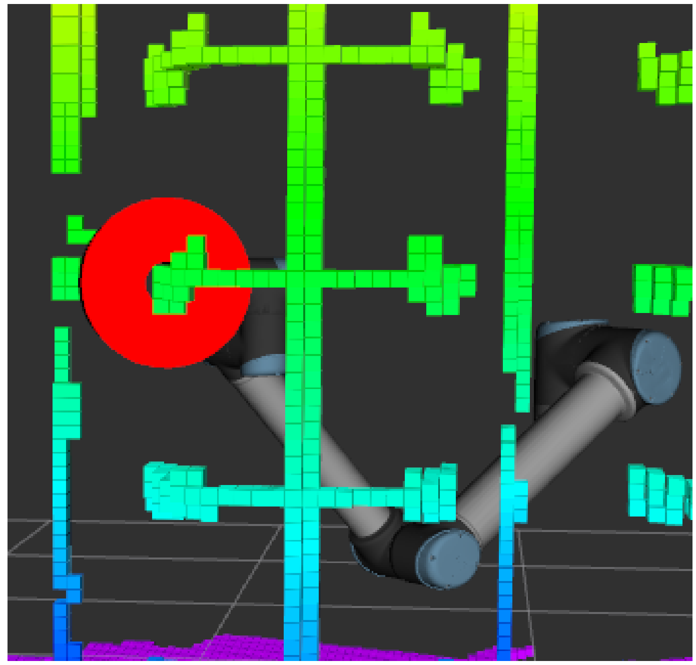

FIGURE 5. PLANNING SCENE SHOWN IN RVIZ.

The time required for placing a package on each peg is shown in Fig. 6 for both the physical and simulated robots. It is apparent that the average duration is shorter for the manual/automatic hybrid approach used for the physical robot than for the OMPL-generated path used in the simulation.

The manual/automatic hybrid approach used with the physical robot yields an average time to place a package of 16 seconds. The average execution time for its trajectories is 28 seconds in the simulation. Though this average is higher than that of the physical system using the manual/automatic hybrid approach, it is better than those obtained when using motion planners other than the default (RRT Connect). Alternative planners tested are an optimal version of the Probabilistic Roadmap Method (PRM), which is called PRM* and Lazy Bi-directional Kinematic Planning by Interior-Exterior Cell Exploration (LBKPIECE).

It can also be seen in Fig. 6 that the variation of duration is less for the hybrid approach than for the random search method of the default planner in MoveIt! The standard deviation for the former is 1.8 seconds and it is 4.5 seconds for the latter.

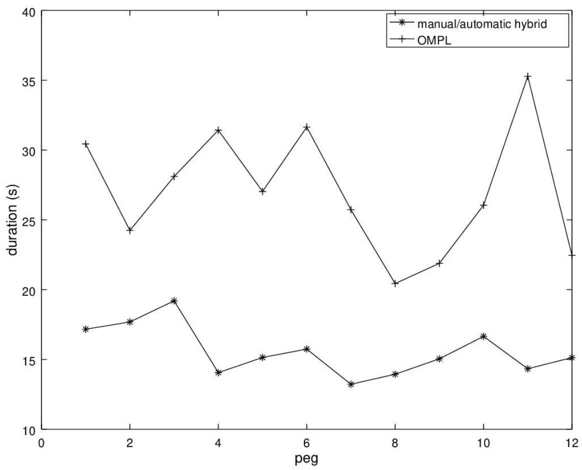

FIGURE 6. TRAJECTORY DURATION FOR EACH PEG WITH PHYSICAL AND VIRTUAL ROBOTS.

4 DISCUSSION

Though the placing trajectory is faster when using custom tailored poses and the robot controller for linear motions, the time spent (which was not measured in this experiment) coming up with a viable set of known-safe intermediate poses for the routines is significant. Using point-cloud data and MoveIt! to generate a path obviates the need for this laborious process. Coupling this with the fact that reliability is of paramount importance in the system means that MoveIt! is as attractive as the manual/automatic hybrid approach; it's certainly worth the cost of an RGB-D camera to be able to test the method in the lab. Thus, the experimentable digital twin has been helpful in the design process.

There are other expected applications of the digital twin for this robotic system. One beginning-of-life use is to find the optimal positioning of the manipulator on the mobile robot in order to design the mount. A likely middle-of-life application is for dispatching of a fleet of mobile robots for loading material at a large scale. According to the results of this work, simulation-based control is a third plausible use of the digital twin.

An experimentable digital twin was created for a robotic system as an aid in a design decision. The digital twin shed light on the subject: it was determined that the path-planning method used in simulation warrants implementation in the physical system. Numerous future applications for this digital twin to the robotic system have been identified.

REFERENCES

[1] Grieves, M., 2014. Digital twin: Manufacturing excellence through virtual factory replication (white paper). Tech. rep., Center for Advanced Manufacturing and Innovative Design, Florida Institute of Technology, Melbourne, Florida.

[2] Gazebo simulator. `http://http://gazebosim.org`. Accessed: 2019-04-29.

[3] rviz 3d visualization tool for ros. `http://http://wiki.ros.org/rviz`. Accessed: 2019-04-29.

[4] Stork, A., 2015. "Visual computing challenges of advanced manufacturing and industrie 4.0 [guest editors' introduction]". *IEEE Computer Graphics and Applications,* **35**(2), pp. 21–25.

[5] Holler, M., Uebernickel, F., and Brenner, W., 2016. "Digital twin concepts in manufacturing industries - a literature review and avenues for further research". In Proceedings of the 18th International Conference on Industrial Engineering (IJIE), Korean Institute of Industrial Engineers.

[6] Gonzalez, A. G. C., Alves, M. V. S., Viana, G. S., Carvalho, L. K., and Basilio, J. C., 2018. "Supervisory control-based navigation architecture: A new framework for autonomous robots in industry 4.0 environments". *IEEE Transactions on Industrial Informatics,* **14**(4), pp. 1732–1743.

[7] Rosen, R., von Wichert, G., Lo, G., and Bettenhausen, K. D., 2015. "About the importance of autonomy and digital twins for the future of manufacturing". *IFAC-PapersOnLine,* **48**(3), pp. 567–572. 15th IFAC Symposium onInformation Control Problems inManufacturing.

[8] Schluse, M., Priggemeyer, M., Atorf, L., and Rossmann, J., 2018. "Experimentable digital twinsstreamlining simulation-based systems engineering for industry 4.0". *IEEE Transactions on Industrial Informatics,* **14**(4), pp. 1722–1731.

[9] Olson, E., 2011. "AprilTag: A robust and flexible visual fiducial system". In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 3400–3407.

[10] Chitta, S., 2016. *MoveIt!: An Introduction*. Springer International Publishing, Cham, pp. 3–27.