

# IMPsys: An Intelligent Mold Processing System for Smart Factory

Xueyi Zhou<sup>1</sup>, Yohan Na<sup>1</sup>, Minju Bang<sup>2</sup> and Dong-Kyu Chae<sup>1\*</sup>

<sup>1</sup>Department of Computer Science, Hanyang University, Republic of Korea

<sup>2</sup>Department of Artificial Intelligence, Hanyang University, Republic of Korea

{hokyeejau, nayohan, minj0329, dongkyu}@hanyang.ac.kr

## Abstract

The explosive popularity of *smart manufacturing* has caught the attention of researchers in terms of intelligent mold processing and management. *Mold component machining* is a crucial step in the mold production process for many industries, which produces the individual parts (e.g., core pins, ejector pins, cavities, slides, and lifters) that make up a mold used in manufacturing. We present IMPsys, an AI-based system that automatically explores machining jobs, infers their processing time and schedules them on machines, given numerous 3D modelling files of mold components. Our demo video can be found at: <http://bit.ly/3EeKnyL>

## 1 Introduction

Smart manufacturing, an interdisciplinary paradigm linking Artificial Intelligence (AI) and manufacturing, facilitates the innovation of traditional production environment in recent years. Under this guiding trend, mold manufacturing factories are attempting to advance manufacturing technologies by applying the advanced smart manufacturing technologies targeted at improving mold manufacturing efficiency, reducing human labors and production costs, and enhancing market competitiveness.

Previous researches on intelligent mold manufacturing focus on assistance of accomplishing 3D mold development [Kim *et al.*, 2003; Jong *et al.*, 2009], draft verification [Yeon *et al.*, 2005], quality control [Sadeghi, 2000; Choi *et al.*, 2010], smart injection molding [Lee *et al.*, 2017; Kumar *et al.*, 2020], smart manufacturing workshop framework [Zheng and Ming, 2017] and reuse of standard parts [Mok *et al.*, 2011]. Different from the aforementioned works that focus on the molding injection process, we look into the overall streamline of *machining mold components* that refers to the processes of using machines to shape and fabricate various components of a mold.

Towards developing an integrated and automated mold machining management system for factories, we identify three primary functions which ever have been the workloads of human experts and implement them with recent AI techniques.

- **Job Explorer (JE).** The machining process involves cutting, drilling, and shaping metal. Given a 3D model of a mold component (typically designed using CAD software), the system is able to indicate which part of the model should be processed by a specific machining type (grinding, EDM, etc.).
- **Process Time Inferrer (PTI).** The system should forecast the machining time consumed by each equipment to process the machining parts identified by JE. The system needs to be capable of storing, retrieving and modifying the predicted time for the subsequent job scheduling.
- **Job Scheduler (JS).** The system should schedule the machining procedures of mold components according to available devices and the pre-determined machining orders (for example, grinding must be performed before EDM<sup>1</sup>).

In this demo, we present **IMPsys: Intelligent Mold Processing System**, integrating all the abovementioned functions for smart factories to easily manage the entire mold component machining streamlines. We extend YOLOv5 for JE to discover the machining parts of 3D models and indicate corresponding machining types. We devise a multivariate regression model for PTI, which is delicately designed for inferring process time of each machining procedures. Finally, a greedy algorithm is employed for job scheduling. IMPsys has been deployed in UJU Electronics, a factory located in the Republic of Korea, innovating its production environment that had been heavily relying on human labor.

## 2 System Overview

IMPsys is developed under an environment that is consisted with Bootstrap 4.6, jQuery 3.3, Django 3.2 and MySQL 8.0. RESTful APIs are particularly applied to support complicated data processing across frontend and backend. To balance the network load, manage the processes, and supervise the gateway of the system, we additionally utilize uWSGI and NginX in terms of high concurrency. Figure 1 demonstrates the system overview of IMPsys with key components including but not limited to *Job explorer (JE)*, *processing time inferrer (PTI)*, and *job scheduler (JS)*. The technical details for each

<sup>1</sup>The order might vary depending on the specific design and requirements of the mold components.

\*Corresponding author.

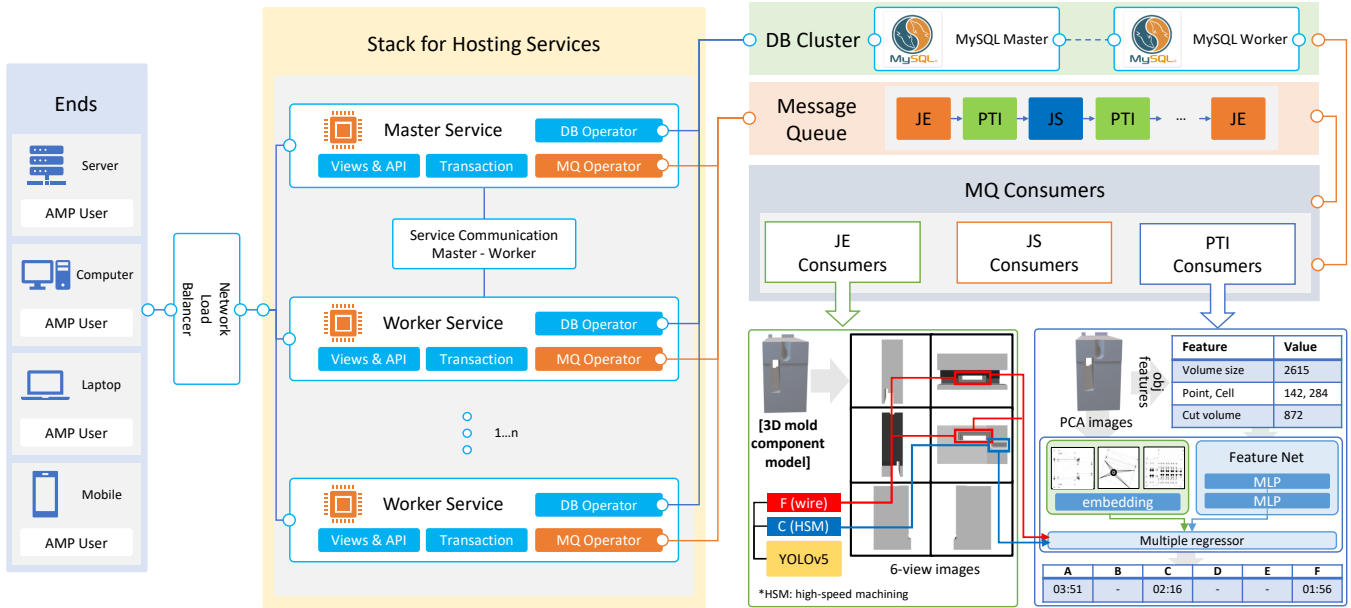


Figure 1: System Overview of IMPsys. There are six types of machining: *grinding electrodes* (A), *grinding* (B), *electrical discharge machining (EDM)* (C), *EDM electrodes* (D), *high-speed machining* (E), and *wire EDM (WEDM)* (F). Given 3D modelling files of mold components, JE tries to label specific areas with their machining types; PTI infers the machining duration of every job (in hours); then the JS module schedules the jobs on equipments. When users interact with the system front, each network request goes through the firewall and network load balancer, and is assigned to an arbitrary service by the balancer. Either master service or worker service(s) connects with three components: one node of the database (DB) cluster, message queue, and message queue consumers (i.e., threads for holding algorithms). A message queue is employed to contain time-series molds and jobs, and its consumers are customized for dealing for tasks JE, PTI, and JS.

key component are summarized in the following subsections respectively.

### 2.1 Job Exploration

Given 3D modeling files of mold components corresponding to client orders, this component aims at figuring out which type of machining (e.g., EDM, WEDM, and grinding) should be performed on each given model. Since directly labeling on 3D models is a tough task, we made a transformation from 3D model labelling to object detection on 2D images. To this end, we constructed a dataset including 3,108 2D images originated from 518 3D mold files (6 side views in 6 directions for each 3D model). Each image may include bounding boxes around the machining locations together with their identified machining types, annotated by human experts. We also reflected depth (concavity) in the 3D model obtained from each viewpoint, since the feature plays an important role in figuring out machining types.

As JE is deemed as an object detection task, we then opted to tailor and train a state-of-the-art model—YOLOv5 [Jocher et al., 2022]—with the dataset constructed. We observe that the trained model achieved a perfect accuracy in terms of machining part detection as well as machining type identification, which results in a successful implementation of JE.

### 2.2 Process Time Inference

After figuring out which types of machining should be performed, this component predicts process time for each machining type of each mold. To solve this regression problem,

we typically defined several useful features that can be extracted from 3D models as follows: *volume size*, *the number of points and cells*, and *cut volume size* (we measured this by computing the difference between the rectangular volume that is surrounding the model and its actual volume.). In addition, the 3D model is represented by generating the 2D  $24 \times 24$  using Principal Component Analysis (PCA), from each view of the model. Through this, the key parts of the model are highlighted in the images. Finally the embeddings of these images are used as additional features.

Based on the aforementioned features, we constructed a dataset where a data instance, generated from a 3D model, consists of the extracted features as model inputs and the type-specific process time provided by factory workers as the target outputs. We trained a multivariate regression model on our dataset, and observe that it achieved a prediction time error of less than 0.5 hours.

### 2.3 Job Scheduling

Job scheduling brings forward the timelines of machining jobs under conditions prescribed by devices, mold components, and prior knowledge of machining. Here, the procedure standing for a triplet  $\langle$ mold component, machining type, process time $\rangle$  is regarded as a *job* in the Job-Shop Scheduling Problem (JSSP) [Juvin et al., 2022]. The constraints, in this case, include a device’s latest working time, a mold component’s latest acquirable time, machining orders of mold components, unappointed devices, pre-defined processing sequences of machining types (for example, A mold should be processed by following the sequence  $B \rightarrow C \rightarrow F$ )

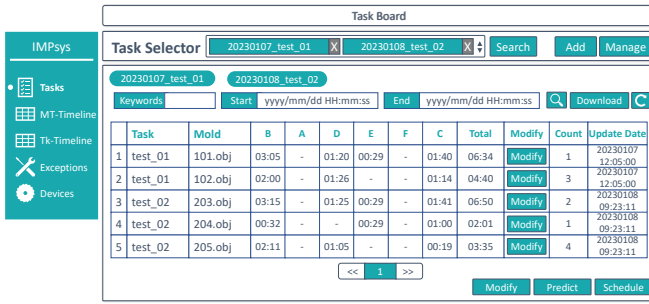


Figure 2: Task board for showing task details.

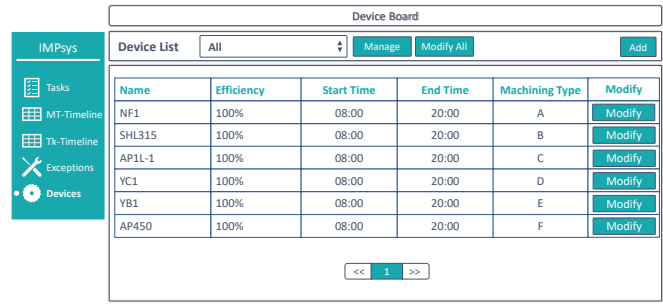


Figure 3: Device board for showing device details.

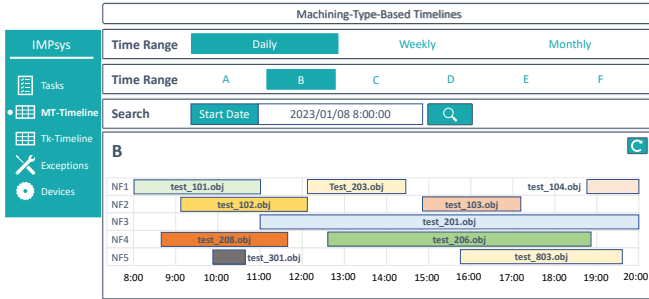


Figure 4: Timelines for showing machining-type-based schedules.

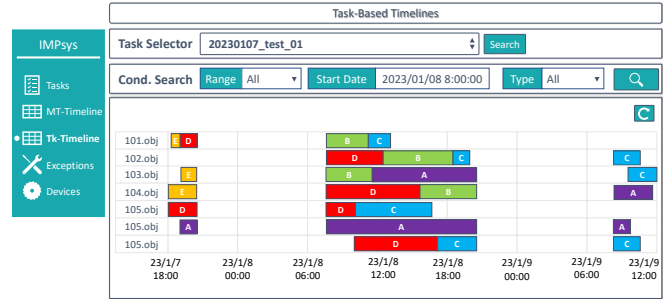


Figure 5: Timelines for showing task-based schedules.

and device efficiency. We followed the concept of exhausting greediness in local optimum: assigning jobs to the latest available devices depending on processing sequences of machining types where the jobs are ordered by mold privileges.

Our initial step is to generate a machine idleness timetable and a mold component idleness timetable based on the current timelines. The corresponding jobs are distinctly ordered according to the fixed sequences of machining types and the idleness timetables. The system then derives the finish time according to the constraints of the devices and mold components themselves. Since the finish time is fetched, it is also considered as the latest idle time of both the corresponding device and the mold component, and thus timetables are updated. By iterating the jobs and assigning them to devices, all the jobs can be retrieved and scheduled recursively.

### 3 Use Cases

Figures 2–5 display four main web pages in our system. All the pages in IMPsys are comprised of two modules: a navigational bar on the left and a content block on the right. We illustrate the contents in each web page as follows:

- **Task board.** The page shown in Figure 2 is tailored to show digital information of an arbitrary task or a list of tasks, including task ID, task title, corresponding mold components and their process time (if the task is processed by JE and PTI), the status of whether the task is predicted and whether it is scheduled, etc. We also provide several buttons to support various functions: adding new task, managing existed tasks, downloading task table as an Excel file, modifying current task information, processing current task with JE and PTI, and scheduling. Specifically, the button ‘**Predict**’ is designed for asynchronously processing JE and PTI on

current client order (task); the button ‘**Schedule**’ is customized to schedule the corresponding jobs instantly.

- **Device board.** Device board as shown in Figure 3 is typically developed for retrieving and managing equipment information. Users can add new machines by clicking ‘**Add**’ and then navigating to the form page. Meanwhile, it is also possible for factory workers to update the information of a single device, or update the work time of all devices by the ‘**Modify**’ and ‘**Modify All**’ buttons respectively.
- **Timelines.** Under consideration of the fact that different staffs focus on different types of the job schedules, IMPsys provides two kinds of timelines: machining-type-based Gantt chart (Figure 4) and task-based Gantt chart (Figure 5). The former can filter scheduled jobs with time ranges, start date, and machining types while the latter is capable of drawing a set of scheduled jobs corresponding to a task restricted by time range, start time for filtering and machining types.

### 4 Conclusion

This demo paper presents IMPsys, an AI-based system that smartly manages the overall streamlines of mold component machining, from 3D modeling files to job scheduling. Our system equips with three core components, aiming at exploring jobs, inferring process time, and scheduling the jobs, respectively. IMPsys is completely novel and innovative in the mold manufacturing domain, replacing what were traditionally conducted by human labor. IMPsys has been successfully deployed in the factory in South Korea, UJU Electronics, and has been significantly improving mold processing efficiency as well as reducing human labors and production costs.

## Acknowledgements

First of all, we would like to thank **UJU Electronics** for providing all the data and experimental environments and supervising the development of IMPsys.

This work was partially supported by (1) Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2020-0-01373, Artificial Intelligence Graduate School Program(Hanyang University)) and (2) the DGIST R&D program of the Ministry of Science and ICT of KOREA (**23-DPIC-08**).

## References

- [Choi *et al.*, 2010] Youngjae Choi, Jeonghoon Shin, Honzong Choi, and Seokwoo Lee. Quality management system for web-based collaboration in mold & die industry. In *The 40th International Conference on Computers & Industrial Engineering*, pages 1–6. IEEE, 2010.
- [Jocher *et al.*, 2022] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, 曾逸夫(Zeng Yifu), Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation, November 2022.
- [Jong *et al.*, 2009] Wen-Ren Jong, Chun-Hsien Wu, Hung Hsun Liu, and Ming-Yan Li. A collaborative navigation system for concurrent mold design. *The International Journal of Advanced Manufacturing Technology*, 40(3):215–225, 2009.
- [Juvin *et al.*, 2022] Carla Juvin, Laurent Houssin, and Pierre Lopez. Logic-based benders decomposition for preemptive flexible job-shop scheduling. In *18th International Conference on Project Management and Scheduling (PMS 2022)*, 2022.
- [Kim *et al.*, 2003] Jong-Hun Kim, Jong-Soo Lee, and Sung-Woon Cha. The optimization of injection molding system using axiomatic approach. *Transactions of the Korean Society of Mechanical Engineers A*, 27(6):1020–1027, 2003.
- [Kumar *et al.*, 2020] Saurabh Kumar, Hong Seok Park, and Chang Myung Lee. Data-driven smart control of injection molding process. *CIRP Journal of Manufacturing Science and Technology*, 31:439–449, 2020.
- [Lee *et al.*, 2017] Hwaseop Lee, Kwangyeol Ryu, and Youngju Cho. A framework of a smart injection molding system based on real-time data. *Procedia Manufacturing*, 11:1004–1011, 2017.
- [Mok *et al.*, 2011] Hak-Soo Mok, Chang-Ho Kim, and Chang-Bong Kim. Automation of mold designs with the reuse of standard parts. *Expert Systems with Applications*, 38(10):12537–12547, 2011.
- [Sadeghi, 2000] BHM Sadeghi. A bp-neural network predictor model for plastic injection molding process. *Journal of materials processing technology*, 103(3):411–416, 2000.
- [Yeon *et al.*, 2005] Kwang-Heum Yeon, In-Ho Song, and Sung-Chong Chung. Web-based draft verification system for injection mold design. *Transactions of the Korean Society of Mechanical Engineers A*, 29(10):1353–1360, 2005.
- [Zheng and Ming, 2017] Maokuan Zheng and Xinguo Ming. Construction of cyber-physical system-integrated smart manufacturing workshops: A case study in automobile industry. *Advances in Mechanical Engineering*, 9(10):1687814017733246, 2017.