A Python and Java software approach for 2.5 axes, self-adaptive stretch forming process and IoT solution

C C Grigoras, B Chirita and G Brabie
“Vasile Alecsandri” University of Bacau, Faculty of Engineering, Calea Marasesti 157, 600115, Bacau, Romania

E-mail: cosmin.grigoras@ub.ro

Abstract. Stretch forming is a metal forming process that implies bending and stretching a sheet of metal over a die, with the sheet being plastically deformed into the desired shape. The purpose is to obtain large parts. One of the main aspects is that the metal sheet is locked into position by gripping jaws. A hydraulic ram is raised into the metal sheet, therefore increasing the tensile forces. This process is used to draw into shape materials like aluminium, magnesium, titanium alloys, stainless steel, Inconel. These materials have in common the fact that they have poor formability or the elastic spring-back has unacceptable values. Taking this into consideration, the nature of the materials along with the fact that the metal sheet can crack, due to excessive strain, we propose a self-adaptive stretch forming process. Underlining this process is the stress-strain curve, that has three inputs: force, sample section area and strain. For each of these factors, our method uses a particular approach, as in a Python-based software and Android-based IoT solution, that uses stress and strain data. Furthermore, it controls, in real-time the hydraulic press to the point at which the material is stretch close to its ultimate yield strength.

1. Introduction
While not so popular among the research topics, the stretch forming process is an essential part of the aeronautics industry, as it allows the making of large contoured panels in one piece. Investigations in this field are mainly conducted in R&D projects for the military sector or by major corporations like Stork Fokker or Dassault Aviation [1, 2] and access to this work is restricted. For other sectors, like automotive, stretch forming can be used for components like door panel or hoods but the process is less efficient and very cost-intensive [3].

From the analyzed researched papers, the key aspect of stretch forming is strain control [4-6]. Furthermore, a considerable number of finite element analysis were conducted to understand the material flow [2, 7-9]. Approaches of the stretch forming process include stretch forming under fluid pressure [10], electromagnetic incremental forming [11], axisymmetric stretch forming combined with deep drawing [12], hybrid incremental sheet forming combined with stretch forming [13] adaptive manufacturing using 3D printed dies, fuzzy logic fracture investigation [14-16] and determination of friction coefficient [17]. In present days concepts like digital strategy, process tracking, IoT platform, artificial intelligence, automation, process digitalization, data analytics, smart logistics, IT infrastructure, cloud & connectivity, data transparency, sustainability, augmented & virtual reality, expert networks or predictive maintenance are implemented with success by industrial and IT groups like Schaeffler, Trumpf, NEONEX, Siemens, BASF, Pia Automation, IoT One, Mercedes, own space, Bosch & Bosch Rexroth, MUNSCH, Caterpillar, Honeywell, Boeing, Airbus and many others. The
integration of these smart solutions leads to improvements in manufacturing costs, operations agility, product innovation and improved client service [18].

Taking any industrial process to the next level of automation is one essential step in implementing the Industry 4.0 standard [19]. Therefore, this paper proposes a novel approach, using a complex Python-based algorithm. This takes advantage of the stress and strain developed in the material during the stretch forming process. Furthermore, an Android Java-based application was developed as an auxiliary tool for real-time data display and in-process control, thus closing one small gap between smart process implementation and stretch forming.

2. The methodology of the adaptive stretch forming process

The ASFP (Adaptive Stretch Forming Process) consists of implementing image processing, object detection and tracking algorithms along with dedicated scripting to retrieve force data from a dynamometer. The algorithms search the video feed for white on black 3/5 [mm] reference markers (figure 1, camera view), locking them into position and measuring the initial distance between them “L₀”, as XY coordinates. The software tracks the markers and returns their position “L₇”, calculating in real-time the elongation of the material. Predefined level of stresses is used as a threshold so that the algorithm switches between elastic and plastic deformation modes. The combination of these two input factors is necessary when working with materials like magnesium alloys, as previous research [14] revealed that the fracture limit is very hard to predict, using classic techniques.

2.1. Industrial equipment and hardware

The schematic representation of the industrial equipment is presented in figure 1. The hydraulic press (1) has an intermediary plate (2) that is software-controlled in the vertical direction, over which a Kistler 9272 type dynamometer is mounted. On top of it, a 150x150 [mm] steel alloy die is mounted, with a radius of 150 [mm]. The image acquisition hardware is provided by a USB camera (5), that is mounted on a light diffusion system (6), for shadow cancellation. A sheet of metal (7) is placed between the gripping jaws (8). The system is designed so that the hydraulic pistons (9) stretch the material in the horizontal plane, while the hydraulic pistons (10) assures that gripping jaws are on the same direction as the metal sheet.

![Figure 1. Schematic representation of the industrial equipment and camera view.](image)

2.2. Software architecture

The software approach consists of two open-source applications, as shown in figure 2. A universal platform (Windows, Linux, OSX) Python-based software, which ensures data acquisition, data interpretation, decision making, and an IoT Android-based solution that returns real-time data from the stretch forming process and offers the possibility of in-process manual control. The software logic is based on the stress and strain input data. The input parameters are user defined (section area and modulus of elasticity) and measured (stress and strain). For this version of the algorithm some
assumptions were made: a two-camera system is ideal for 3D measuring displacement, but it causes more problems than it solves in this actual stage, by means of camera calibration and intricate algorithms for 3D space motion, thus an approximation of the strain is made by a planar measurement and analytical approximation; this is made by knowing at each step the distance between the markers and the camera, this also allows to dynamically calculate the radius of the part, therefore knowing the distance between the markers as they on the radius of a circle; for the elastic region the stress is being calculated by force on section area while for the plastic region the stress is calculated using equation (1), where K and n are constant for each alloy, with n being the strain hardening exponent. The force is exported as a resultant of the normal and axial directions. The main goal is avoiding material failure. In the following, the paper presents how these values are obtained.

\[ \sigma = Ke^n \]  

Figure 2. Simplified Adaptive Stretch Forming Process software architecture.

2.2.1. Python-based adaptive stretch forming software.

The ASFP application consist of eight Python-based individual modules joined together under a GUI (Graphical User Interface) highlighted in figure 3. The first two modules “Video calibration” and “Test video calibration” open the video feed, place a mesh over it, tracing the ideal position of the markers and indicating if the self-adaptive stretch forming process can start with no errors.

Module number three is a complex environment, composed of multiple scripts. The force values are returned from the dynamometer’s official software (DEWESoft X3 SP10) via two python scripts.
and represent the resulting force developed on the vertical and horizontal directions. The processing speed is 200 Hz or 1 reading per 5 milliseconds. The data is timestamped and stored in a text file. The scripts were developed by DEWESoft technical team and adapted to this specific application; the communication, with the main application, is done by sockets and multiprocessing methods. The image processing is made in an OpenCV environment, that is represented by an open-source library. This includes several hundred computer vision algorithms, from which we have used image processing, video analysis, video capturing, camera calibration, object detection and object tracking. The script manages to detect the markers using the Shi-Thomas corner detector [20] and Hough Circle transformation [21] algorithms. The first method is rather a way of cancelling any additional point if they are found. For the tracking & tracing the Lucas-Kanade optical flow method is used [22]. It detects the outer shape of the black circles by RGB to GRAY colour conversion; it converts the shape to a circle and self-correction the radius size; after this, it locks the circles centre with XY coordinates, as shown in figure 4. When the stretch forming process starts the algorithm traces those points, returning their position, thus the distance between them “Lf”. Having these values, the algorithm calculates in real-time the elongation, by using equation (2). It does this at a maximum speed of 100 Hz or 1 frame per 10 milliseconds. If GPU processing is used, the algorithm can handle video and force data to speeds of 1000 Hz, thus a slow-motion camera can be connected. As the force and elongation data are recorded at different rates, the plot script uses timestamps and positions the data from the same timeframe in one line, that represent the real strain obtained at a specific stress. Furthermore, in this script, the decision making takes place via an “if-else goal achieve” algorithm. This controls an Arduino Mega, that is connected to the hydraulic press main panel. This way the intermediary plate (2) speed, the gripping jaws angle and stretching force in the hydraulic pistons (9 and 10) are controlled.

\[ \varepsilon = \frac{\Delta L}{L_0} \Rightarrow \varepsilon = \frac{L_f - L_0}{L_0} \]  

(2)

The video feed can be provided by USB camera, IP camera, smartphone camera or video file. The last one was needed as the software records the process from figure 4, and can be analysed afterwards. The “Decision making” module highlights the command given from module 3 to the Arduino Mega board. Connecting a 3D-connection space mouse allows working in module 6 in manual mode, useful for in-process user intervention or setup purpose.

![Figure 4. ASFP video feed analyser GUI.](image)

2.2.2. Android IoT application.

This IoT solution was designed as an auxiliary tool and comes as an important step in linking this process to the Industry 4.0 standard. Having Bluetooth capabilities, it can connect wirelessly to the Arduino Mega board. The application is divided into the MainScreen.java (left side), MainScan.java (not shown) and the MainApp.java (right side) modules, as shown in figure 5. The main screen shows a list of the Bluetooth devices recently connected and can access the network scan module if the
device is not listed. This interface also handles Bluetooth on-off switch and user permissions. After selecting the equipment, the main application screen appears. In the upper part of the screen information regarding the process elapsed time is displayed along with a “stop the connection” button. The next widget can be customized to show the video feed or it can display images taken from figure 4. The two seeker bars (stretch forming force and ram speed) can override the command given by the python ASFP algorithm and set the values as the user requested. To have full control of the two hydraulic pumps four-button were placed; the green ones turn on the hydraulic system for the ram and gripping system while the red one stops them. A scroll view widget is placed to return process information from the Arduino Board as text messages. In the lower part of the screen a manual input and send button was added, for custom control.

![Figure 5. Adaptive Stretch Forming Process Android IoT software graphical user interface.](image)

3. Conclusions
The applications were developed by using two frameworks with more than 40 libraries. The precision of the camera measurements is within ±20 [μm]. The speed from data acquisition to decision making to process control if very fast. The software can handle as much as 50 complete set of instructions per second, but it’s limited to 10 due to the Arduino board. Although permanent software development is made, the application, at its actual version, proves that this solution is a useful resource as it can control the stretch forming process in real-time.

4. References
[1] Wisselink H H and van den Boogaard AH 2005 Finite Element Simulation of the Stretch-forming of Aircraft Skins *Numisheet 2005* ed. Smith L M et al. (Detroit: American Institute of Physics) pp 60-65
[2] Kurukuri S et al. 2011 Simulation of stretch forming with intermediate heat treatments of aircraft skins *Int J Mater Form* 4 pp 129–140
[3] Vlahovic D and Liewald M 2008 Benchmarking Methods for Short Cycle Stretch-Forming *International Journal of Material Forming* 1(1) pp 193-196
[4] *** 1990 Scientific Technical Information Program NASA Center for AeroSpace Information Scientific and Technical Aerospace Reports. NASA, Office of Scientific and Technical Information 28 p 2573
[5] Cubberly W H and Bakerjian R 1989 Society of Manufacturing Engineers, Tool and Manufacturing Engineers Handbook Desk Edition *Society of Manufacturing Engineers* pp 33-46
[6] Hardt D E, Norfleet W A, Valentin V M and Parris A 2000 In Process Control of Strain in a Stretch Forming Process Journal of Engineering Materials and Technology 123(4) pp 496-503
[7] Tasan C C, Diehl M, Yan D, Zambaldi C, Shanthraj P, Roters F and Raabe D 2014 Integrated experimental–simulation analysis of stress and strain partitioning in multiphase alloys Acta Materialia 81 pp 386-400
[8] Yan A M and Klappka I 2008 Springback in stretch forming process of aeronautic panel production by finite element simulation International Journal of Material Forming 1(1) pp 201-204
[9] Zhang H, Diehl M, Roters F, Raabe D 2016 A virtual laboratory using high resolution crystal plasticity simulations to determine the initial yield surface for sheet metal forming operations International Journal of Plasticity 80 pp 111-138
[10] Mellor P B 1956 Stretch forming under fluid pressure Journal of the Mechanics and Physics of Solids 5(1) pp 41-56
[11] Cui X, Mob J, Lib J, Xiaoa X, Zhoub B and Fangb J 2016 Large-scale sheet deformation process by electromagnetic incremental forming combined with stretch forming Journal of Materials Processing Technology 237 pp 139-154
[12] Lim TC, Ramakrishna S and Shang H M 1999 Axisymmetric sheet forming of knitted fabric composite by combined stretch forming and deep drawing. Composites Part B Engineering 30(5) pp 495-502
[13] Araghi B T, Manco G L, Bambach M and Hirt G 2009 Investigation into a new hybrid forming process: Incremental sheet forming combined with stretch forming CIRP Annals 58(1) pp 225-228
[14] Ciofu C, Grigoraș C C, Chiriță B, Iancu C and Brabie G 2020 Fracture Investigation in Draw Bending of AZ31B Sheets using Fuzzy Logic Prediction Procedia Manufacturing 47 pp 1462-67
[15] Grigoraș C, Chiriță B and Brabie G 2019 Additive manufacturing of a stretch forming die using 3D printing technology IOP Conference Series: Materials Science and Engineering 564 p 012017
[16] Grigoras C, Chiriță B, Brabie G and Ciofu C 2019 Experimental analysis of AZ31B magnesium alloy sheet failure using punch stretching IOP Conference Series: Materials Science and Engineering 682 p 012009
[17] Kafıanoğlu B 1973 Determination of coefficient of friction under conditions of deep-drawing and stretch forming Wear 25(2) pp 177-188
[18] Commission E 2017 Digital Transformation Monitor. Industry 4.0 in Aeronautics: IoT applications Available from: https://ec.europa.eu/growth/tools-databases/dem/monitor/sites/default/files/DTM_Aeronautics%20-%20IoT%20Applications%20v1.pdf.
[19] Saurabh V, Prashant A and Santosh B 2018 Industry 4.0 – A Glimpse Procedia Manufacturing 20 pp 233-238
[20] Mordvintsev A, Abid K and Tomasi S 2013 Corner Detector & Good Features to Track Available from: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_shi_tomasi/py_shi_tomasi.html.
[21] Mordvintsev A and Abid K 2013 Hough Circle Transform Available from: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghcircles/py_houghcircles.html
[22] Mordvintsev A and Abid K 2013 Optical Flow Available from: https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_video/py_lucas_kanade/py_lucaskanade.html

Acknowledgements
This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-20170446 / 82PCCDI / 2018, within PNCDI III.