VISUAL SERVOING CONTROL FOR LINE AND OBJECT DETECTION AND FOLLOWING USING A ROBOTIC ARM MANIPULATOR MOUNTED REAL TIME CAMERA SYSTEM

James Ross Buckle [*,1,2] **Andrew Jason Tickle** [*,1,3]
Fan Wu [*,4] **Paul Kenneth Harvey** [*,5]
Jeremy Simon Smith [*,6]

[*] *Intelligence Engineering and Industrial Automation Research Group, Department of Electrical Engineering and Electronics, The University of Liverpool, Liverpool, L69 3GJ, U.K.*

Abstract: This paper presents an investigation into compliant robotic systems, focusing on the use of high speed visual servoing without modeling to correct manipulator elasticity when working either above the manufacturers recommended velocities or load levels. This work is based on a PUMA 500 series manipulator in SLAVE control, interfaced with a PC running RTAI Real-time Linux and a PixeLINK CMOS camera. This paper addresses how the system was designed and the testing mechanism which consists of the robot tracking the centre of a target line and how it responds to target step-change input. The control response graphs and the accuracy of the system are discussed in detail. Also included here is a possible system implementation using Altera's DSP Builder graphical block interface that could see the systems control mechanism improve in performance and cost.

Keywords: Image Processing, PUMA 560 Robot, Real-Time Control, Field Programmable Gate Array (FPGA), Two Timescale, RTAI, Eye In Hand

1. INTRODUCTION

Most rigid manipulators have very high accuracy and repeatability, for example the PUMA 560 has a quoted repeatability of $\pm 0.1mm$ and a maximum end effector speed of $1ms^{-1}$. These are only achieved by way of the high precision of the gearing and links, careful control over the gear backlash and play in the joint bearings. If even a small amount of pivot tolerance or flexibility were allowed to be present, the rigid body assumption collapses. This leads to process errors and possibly manipulator or workpiece damage in extreme situations. This flexibility and backlash can occur due to wear, such as becomes present

[1] This work was supported in part by the EPSRC (Engineering and Physical Science Research Council).
[2] James R. Buckle was a Ph.D. student with the Univ. of Liverpool and is now a Research Assistant at the Univ. of Glasgow - Email: james.buckle@gmail.com
[3] Andrew J. Tickle is a Ph.D. student with the Univ. of Liverpool - Corresponding author - Tel: 0044 1517944602; Fax: 0044 1517944540; Email: a.j.tickle@liv.ac.uk
[4] Wu Fan was a Ph.D. student with the Univ. of Liverpool and now is a Digital Systems Design Engineer at RF Engines, Isle of Wight - Email: wufanliang@gmail.com
[5] Paul K. Harvey was an M.Sc student and is now a Ph.D. student with the Univ. of Liverpool - Email: pkh@liv.ac.uk
[6] Jeremy S. Smith is a Professor of Electrical Engineering with the Univ. of Liverpool - Email: j.s.smith@liv.ac.uk

in an industrial manipulator after many hours of operation, or could be due to faults or cost savings in the original construction.

In specialist circumstances, such as in the space environment and other high value applications, the lightweight and slender nature of the manipulators is overcome with detailed modeling and amalgamation of an array of sensor data to produce a close to perfect model of the system as a whole. This method is expensive, both computationally and financially, due to the array of sensors required. Visual servoing is simply the use of visual information in the control feedback loop and can be used as part of the control system to provide actual end effector error measurements. It is widely researched, though industrially is generally used on assumed-rigid robots to find targets, not to correct for model error. It has been used recently by Bascetta (2004 and 2006) as part of the feedback loop, along with an array of other sensors, to refine the positioning accuracy of a flexible link robot. Solutions relying on a model suffer from inaccuracy when the model is incorrect due to unknown, changing payload or unexpected robot flexibility. Here we present a low-cost system where an Eye In Hand camera is used as the sole position and pose error input and a compliance model is not used to calculate expected position, based on (Buckle, 2007).

2. REAL-TIME CONTROL SYSTEM

The system is based primarily on standard personal computer architecture, using an AMD Athlon XP 2500+ processor and 512Mb of RAM. A range of operating systems was investigated to the aim of realtime control, in agreement with Aarno (2004) regarding linux derivatives for realtime control, the final choice being Linux (Fedora Core 5) with RTAI patch (Real-Time Application Interface) to form a modular control system. This modular system consisted of an RS422 link to the PUMA hardware controller, with respective hardware driver for the SLAVE control interface, a TCP/IP connection to a camera subsystem with image processing, a shared memory area and control loops executed at user selectable frequency.

2.1 RTAI Subsystems

RTAI is an open source operating system patch, providing guaranteed system latency, ideal for control, while allowing full functionality of the normal operating system for user interface. RTAI runs realtime processes as tasks, either periodically or a-periodically, or in response to hardware interrupt. In this case, the RTAI system deals with duplex serial communication to the robot hardware with approximately 28ms period, over the SLAVE interface. This interface is a method of retrieving and demanding joint positions from the manipulator controller, bypassing the standard robot kinematic equations, allowing the RTAI system to assume full positional control over the joints. Joint position data is stored with a timestamp in a shared memory area for use by the control loops.

The camera subsystem, including image processing, is carried out on a separate computer system. The values of the parameters extracted from the frames are transmitted over a dedicated 100Mbit Ethernet connection to the RTAI machine. The RTAI machine has a realtime server process, into which the camera subsystem connects and delivers the data. The camera system used was a PixeLINK A641 monochrome CMOS machine vision camera, capable of windowing. Windowing allows a small section of the CMOS sensor to be scanned at high frame rates, in this case 350 frames per second at 320x16 pixels. As the camera subsystem is a separate entity and is built around modular software, it can easily be replaced by dedicated video processing hardware, as is discussed later. At the speeds discussed here, the RTAI control system possessed sufficient power to make the modular approach unnecessary, however this modularity removes CPU load from the main control system, allowing future increases in vision capabilities without compromising control system speed.

2.2 Flexible Link Design

In order to degrade the performance of the rigid PUMA 560, an attachment in the form of a slender link with compliance in one plane was bolted in place of the gripper, with the camera mounted in Eye-In-Hand fashion at its tip, with a variable mass. A selection of masses were used to adjust the oscillation frequency of the beam/mass combination. The oscillatory response of the beam was determined by assuming the link to be a cantilever beam, determining its effective spring constant using beam bending equations and substituting this into the mass-spring dynamic system equation:

$$\omega_n = \sqrt{\frac{k}{m}} \tag{1}$$

Where k is the value of spring constant, determined by beam dimensions and material, m is the mass at the tip and ω_n is the natural frequency of the combination. According to these calculations, with 3.5Kg mass, the selected beam had a natural frequency of 4.21Hz. Practical experimentation returned a value of 5Hz, acceptable error relating to material quality and approximations in the model of the link.

2.3 Image Processing

The initial image processing carried out on each successive frame is thresholding, taking a mean value from initial frames. As the camera is capable of auto-exposure control and combined with a high contrast target, this provides good noise removal and simplification for the secondary step, as in Figure 4. The second processing layer is line-centre determination, carried out by means of edge detection using the gradient presented by the thresholded image of the target. This is followed by identification of the target centre by simple mathematics, then subsequent relation of target centre to centre of frame, in the form of integer pixel values. Further information extraction is possible, such as range, with simple modifications of the target and vision algorithms. This provides an 8-bit output that is transmitted to the RTAI control machine. This is sufficient to provide visual servoing feedback along a single axis of movement as presented here, but is easily expandable to further degrees of freedom.

2.4 Control Algorithms

The control system consisted of a periodic real-time task, calculating the rigid-body kinematics and inverse kinematics of the manipulator using the shared data from the sensor collection tasks, capable of operating at around 60kHz but actually operating at 1kHz. When executed with sufficiently short period, errors between frames are minimised, and actuation of the arm can be used to directly counter the link compliance, within reasonable bounds set primarily by the joint controller hardware. Testing consisted of a step-change in target position, a worst case scenario for a robotic welding seam following task. The rigid body calculations were applied without joint acceleration or speed calculations, relying only on joint position control, with a two-timescale PID feedback loop determining controller response to the image data. The controller has been split into two PID controllers, slow CO^s and fast CO^f, CO^s operates on less frequent data updates at frame time t (can include filtered or averaged data), the fast controller on full frame rate z. The slow and fast controllers also employed their own gains for each term, separating these into ${}^sk_p$, ${}^sk_i$, ${}^sk_d$, ${}^fk_p$, ${}^fk_i$ and ${}^fk_d$.

$$CO^s_t = {}^s k_p C^s_t + {}^s k_i \sum_{t-n}^{t} C^s_t + {}^s k_d (C^s_t - C^s_{t-1}) \quad (2)$$

and the fast controller

$$CO^f_z = {}^f k_p C^f_z + {}^f k_i \sum_{z-n}^{z} C^f_z + {}^f k_d (C^f_z - C^f_{z-1}) \quad (3)$$

but CO^f uses the last C recorded by the CO^s routine as the reference by which it determines its own error input C^f_z

$$C^f_z = (C_z - C^s_t) \quad (4)$$

therefore the overall controller output CO is given by

$$P_c = E + (CO^s_t + CO^f_z) \quad (5)$$

where E is the position estimated from rigid kinematics. The slow controller calculates its output based on images sampled at a lower frequency, while the fast controller utilises all of the image data that is captured. Both the slow and fast controllers are individually online-tunable PID controllers, with correctly selected frequency the slow controller responds to general motion of the manipulator tip and provides a non-oscillatory position correction response. The high speed controller compares the current tip position to the position demanded by the slow controller output and attempts to reduce oscillation about this point, it does not provide steady-state correction to tip position. This forms a control combination that attempts to maintain overall motion towards the target, while reducing oscillations due to manipulator compliance. Figure 1 shows the ideal controller response, the solid line representing actual camera motion, the dotted line representing the slow controller, following an overall output towards zero error. The fast controller provides feedback determined by the speed, extent and polarity of the oscillatory error about the slow controller output - depicted by the light and dark shaded areas. Tuning of the slow controller requires an

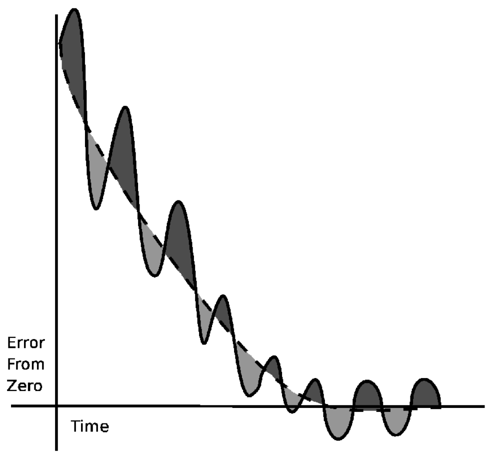

Fig. 1. Ideal controller response

emphasis on proportional gains, with a small component of differential to reduce the initial “kick” from a step change input, this reduces end effector oscillation slightly. The fast controller emphasis is on differential gains, attempting to damp the speed of the manipulator tip oscillation. This provides a response similar to a human moving a glass of water - overall motion in one direction while finely adjusting hand velocity to reduce water oscillation to prevent a spill. Controller gains were tuned as described in section 4.1.

3. DEVELOPMENT OF A POSSIBLE FIELD PROGRAMMABLE GATE ARRAY CONTROL INTERFACE

3.1 Image Capture

The image capture capacities, would be based around a dedicated CameraLink video capture interface for real-time video acquisition, which can then be processed (Wu, 2008). This system core would include a systhesised NIOS processor, along with custom IP (Intellectual Properties) components, all integrated by the Altera SOPC Builder. They would communicate with each other via the Altera specific internal bus protocol - Avalon bus. As this operation would require fast processing of the images, the video capture clock runs at 20 MHz, which is fed into the video capture controller and provides one frame approximately every 16.36ms at 640x480 resolution, therefore giving a frame rate of 61 f/s. The SDRAM buffers the incoming video stream captured from the CameraLink camera and the processed video data, which are ready to be displayed by the graphical block circuitry. This memory has the capacity to store 4 full size video frames at any time to ensure lossless real time video processing. For higher frame rate the capture window size can be reduced which increases the overall processing speed as there are fewer pixels to be captured and processed. For example the capture frame rate could be increased to 3355 f/s if the capture window is reduced to 320x16.

3.2 Image Processing and Line Tracking

The system in Figure 5, shows the gate level of abstraction for the visual servoing control mechanism, based around the graphical block methodology of the Altera DSP Builder software. The gate level of abstraction is the lowest of the three possible levels (Tickle et al., 2007). The data produced by the capture stage, is now turned into a binary in order to enable faster processing. The 8-bit greyscale (GS) image is converted by binary thresholding the image at greyscale value 240. The image is then processed by an openclose morphological filter, similar to the cascaded circuitry in (Tickle et al., 2007), before being passed to the tracking circuitry.

The solid lined is a set of three counters, that count the x and y locations of the pixel currently being considered, and a third depth counter, z. This latter counter is used to activate the signals when the image data reaches the tracking circuitry, so it does not try to find the distance that the line is from the center when there is no line to detect. The outputs from these blocks are then sent to the other parts of the circuitry. These cascaded counters work by the system clk, activating the x counter and the output is sent into a comparator with a constant value of ‘319’ sent into the other input. This triggers when it reaches the end of the line, and the output is fed into the y counter which has a similar arrangement, but now the constant tag is set to ‘2’. This is due to that fact that the openclose morphological filter requires two full scans of the image, so after this is completed, the z counter can then activate the tracking circuitry to analyse the incoming data.

The dotted area is used to detect where the line is located in the incoming data, the morphological filter will have cleared up the image so just the line remains with very little noise. The system looks for specific data strings to determine when the line is reached in the image. A binary string of ‘1100’ (when the line is detected), or ‘0011’ (when the end of the line is reached) are used. This data is produced from a string of three $\frac{1}{z}$ delays that are then fed into a comparator (Tickle et al., 2007), to compare them with the value associated for each delay. The result is passed into a 4 input AND gate, that feeds another comparator connected to an OR gate in an algebraic loop. Once triggered, the value is retained on the output line due to how the output is connected in a feedback loop to the other input. Flip flops are too unreliable during the triggering process to be used, thus the latter configuration was created. Connected to this is a 2 input AND gate, which also receives the incoming data, and determines if its a ‘1’ or a ‘0’, again via a comparator with a value tag. These comparators are used to drive the counters in the un-boxed section. These count the distance to the line and the depth of the line that are then used to calculate the distance back to the center. These counters need to be stopped when the end of the line is reached and that is what these AND gates do, so before triggering, the OR gates are used to make sure that a ‘1’ is always sent to one input. After the binary string is detected the loop sends out a signal now, that changes this input to a ‘0’ so nothing comes out of the AND gate, and thus the counters can not be triggered anymore.

The other part of the un-boxed area has the input data connected to a product block, this also is connected to a two input AND gate, that has two comparators feeding its inputs. One comparator is from the y counter, to look at the line which has the y coordinate ‘8’, the line selected to determine the distance. The other is the ‘z’ counter and must also be ‘2’ for the reasons stated previously. When both these conditions are met, the AND gate will be triggered and allow the data to be sent out for consideration by the system.

The dashed area performs the actual calculations, this sub-system is split into two situations, case 1 which is if the line is positioned to the left of the center, and case 2 if the line is positioned to the right of the center. A comparator looking at the x counter result, producing two signals in the form of either '01' for case 1, or '10' for case 2. These are fed into two separate arrangements, each with two comparators connected to a 2 input AND gate set to detect this. For case 1, it adds the count1 (distance to the line), and the count0 (depth of line), together, and this is subtracted from '160' to determine the distance to center. For case 2, it simply subtracts '160' from count1. These are sent to product blocks, which are triggered by the case value decider, and passed to a 2 input multi-bit OR gate, since only one output will be valid at any one time there will be no signal clashes in the data so the output will be correct. The last part of circuitry is to lock this answer in place, this uses the count data from the y and z counters so that when the values of '9' and '2' are reached respectively, it activates another algebraic loop connected to a NOT gate, connected to a product block, with a value of '8191' going into it. The latter corresponds to a 13 bit long binary string of 1's, the product block then goes into a multi-bit AND gate, which is connected to the output of the system. When the y and z parameters are met, the output from the NOT gate connected to loop changes from '1' to '0' so that the product block does not output the binary string to the multi-bit AND gate anymore. This is then connected to another algebraic multi-bit, OR gate so that the locked value is continuously output until reset.

4. RESULTS

4.1 Control System Responses

The RTAI controller datalogs the camera output, this output was used to analyse system response with selected gains. The response of the system with a large payload, presenting high deformation, was compared to open-loop control (Table 1, A) using the manipulators VAL commands to make the same size step change in position. The results were also compared against single timescale PID control (Table 1, B) across a range of frequencies around the natural frequency of the flexible link, from 5Hz to 36Hz. The step change selected was 200 pixels, translating to a 10mm step change, with the payload and speed requested, this is outside the manufacturers operating envelope even for the rigid robot. Analysis of the response of the

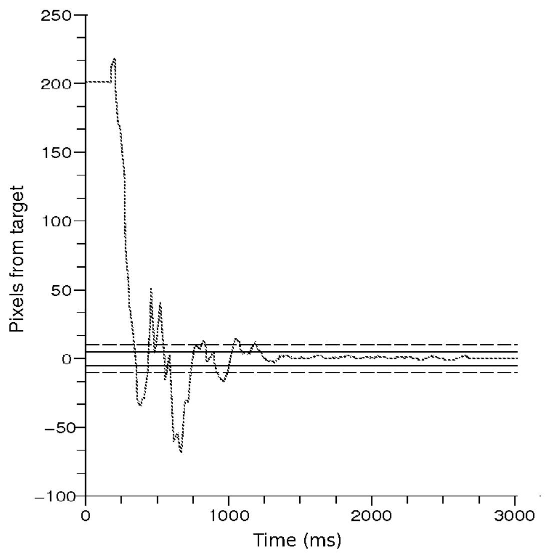

Fig. 2. Overall best controller output, PID Gains Slow (10Hz - 0.6,0,-0.2), Fast(0.3,0,-1)

system was carried out using a primary and secondary settling time, as well as determination of the quality of oscillation reduction after settling. Primary settling time was judged to be to within $\pm 0.5mm$ and secondary, $\pm 0.25mm$. The gains on the two-timescale controller (Table 1, C) were systematically tested and set by first achieving the best result possible using the slow controller alone, then taking the same approach to the fast controller. Further testing proved that, using the improvements gained by the fast controller, the pair could be iteratively improved significantly (Table 1, D). The open-loop control caused wild oscillations in the tip, leading to joint overload and divergence from the target, even on a VAL speed setting of only 30.

Table 1. Summary of control architecture performance with 3.5kg mass.

Control	$\pm 0.5mm$(ms)	$\pm 0.25mm$(ms)	Osc.
A	2750	2750	Very Poor
B	1800	2500	Poor
C	1400	1900	Good
D	1210	1260	Best

Figure 2 shows the optimum result obtained with a 3.5Kg mass and iteratively tuned two timescale control.

4.2 FPGA Simulation Results

The FPGA system produced some very good results, the two data lines had the correct data on them and was tested for four cases, two case 1 situations and two case 2 situations. Each time the right case signal was sent out along with the number of pixels to the center position. The latter did have a small error on it, the worst case error was ± 7 pixels. Figures 3 and 4 show how an image

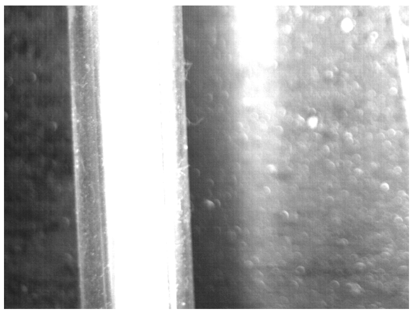

Fig. 3. Original image that is input into DSP Builder

Fig. 4. Processed image ready to be tracked after the morphological filter

enters the FPGA system and that exiting (along with the associated results).

5. CONCLUSIONS

The results presented show good improvement in control over a compliant system is possible using direct, model-free, visual servoing with no extra sensor data, based on low-cost realtime control hardware and software. With two timescale control, a degraded manipulator was operated well outside the manufacturer specification and performed well. A process of systematic tuning of the two timescale controller was employed in a fashion that could be automated online using the visual feedback for self-tuning of a system in operation. Also presented, a possible FPGA image processing solution, providing the possibility of replacing the camera subsystem PC with a low cost, higher performance alternative. Although, with current image resolution, the system is capable of extremely high frame rates, use of the system for control with higher degrees of freedom would require larger resolution. This solution can be easily accomplished via a modification of the line buffer circuitry, though this requires the system to be run on a faster FPGA in order to meet the above requirements. Future work will consist of further developing the mechanism by which the target pose is identified, while investigating better model-free control strategies and online calibration methods. Operation of the FPGA image processing on faster hardware is also proposed.

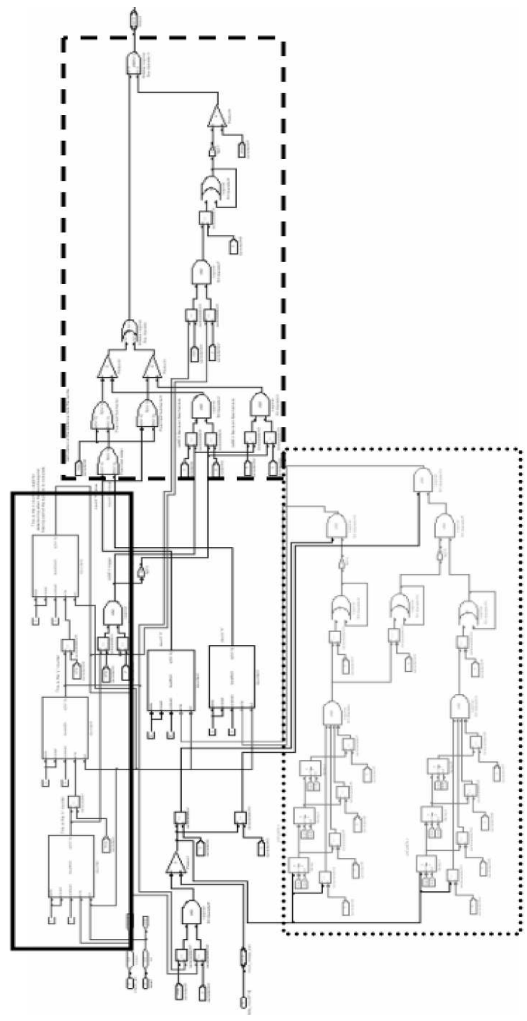

Fig. 5. Gate level of the FPGA based control mechanism

REFERENCES

Aarno D 2004 *Autonomous path planning and real-time control, a solution to the narrow passage problem for path planners and an evaluation of real-time linux derivatives for use in robotic control*, Royal Institute of Technology, Sweden, Masters Thesis

Bascetta L 2004 *Visual Servoing of Flexible Manipulators* Politechnico di Milano, PhD Thesis

Bascetta L 2006 *Two-time scale visual servoing of eye-in-hand flexible manipulators*, IEEE Transactions on Robotics, vol. 22, pp. 818830.

Buckle J 2007 *Visual Servoing of Compliant Welding Manipulators* University of Liverpool EEE, PhD Thesis

Tickle A J, Smith J S and Wu Q H 2007 *Development of Morphological Operators for FPGAs* Proceedings of the IOP Electronic Journal of Physics: Conferences Series Volume 76: 'Sensors and their Applications XIV' Liverpool, (012028)

Wu F 2008 *SOPC Based Image Processing System* University of Liverpool EEE, PhD Thesis