# A VCA Protocol Based Multi-level Flexible Architecture on Embedded PLCs for Visual Servo Control

**摘要—**

***Index Terms*—motion control, visual servo control, embedded PLC, multi-level architecture**

## I. Introduction

Integration technologies 驱动着工业自动化的发展 [1]。对传感器，控制器，机器人，生产线以及软件工具的不断整合，催生了自省设备，智能工厂，CPS，等概念 [2], [3]。最近的一些研究， [4]–[6]介绍了融合技术的发展。在工业自动化中，逻辑控制系统，运动控制系统和视觉系统具有重要作用，且密不可分 [7]–[9]。另一方面，边缘计算，雾计算，边缘人工智能等新技术的发展 [10]–[12]，也对逻辑逻辑控制系统，运动控制系统和视觉系统等相关边缘设备提出了新的要求。

### A. Motivations

Nowdays, visual system has been applied in various fields. The logic control on PLC has wide applications. Motion control [13] is a typical case that describes how the three parts collaborate. The visual system analyzes the context and get error put into the motion system. Simultaneously the logic program are judging the information, such as the position limitation of the every axis. Hence, how to pose a flexible structure to the integration of logic control system, servo system and visual system, guides us.

### B. Related Works

视觉控制系统均是由专用控制系统和视觉系统组成。例如搬运 [14],，电路检测 [15],分拣系统，welding [13]，装配 [16], [17]，教育,robot [18], [19]，unmanned aerial vehicles [20], [21]。这些研究都很好的解决了各自领域的问题，但是这些方案都是基于视觉系统和专用控制系统完成。

例如 [18], [19]中均使用机器人专用控制系统和视觉系统完成,在 [20], [21]中，使用无人机专用控制系统和视觉系统完成。 [15]中，通过专用运动控制卡和视觉系统实现电路检测。

On the other hand. The integration of logic control and motion control has variously deep researches [22]–[25]. [22], [25] realize the motion control directly in PLC. [26]–[28] use motion control module collaborated with PLC to implement their applications. However, the development method in these papers is disordered. Therefore, in 2005, PLCopen organization has released a related standard [29] which standardizes the motion control in PLC. Based on this standardization, [30] provides an advanced implementation in distributed automation system and companies, such as 3S [31], provide some tools. [32] poses a customized real-time compilation method to reduce the development complexity.

Above works provide impressive integrations on visual servo control system and PLC with motion control functions, however there are few papers discussing the integration of visual system, motion control and PLC. Most of applications are focusing on its application with three individual system, such as [13]. Hence, an integration structure of logic control system, servo system and visual system should be provided to reduce the complexity and expand the application fields.

### C. Our Contributions

我们提出了一种基于VCA协议的ePLC和视觉系统的柔性架构整合方案，来实现视觉私服控制。VCA协议描述了从视觉系统中获取数据后的组帧和在ePLC中

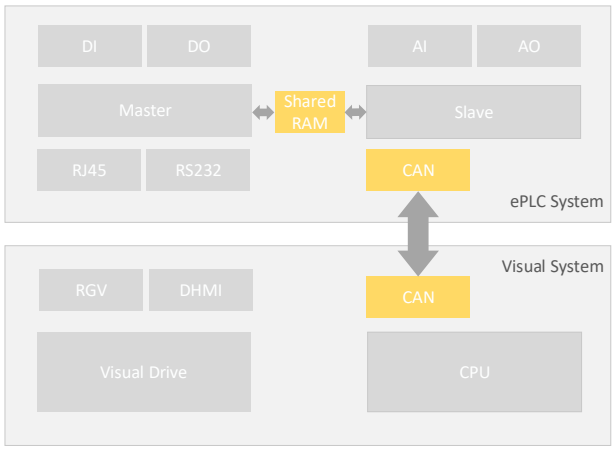


图 1. ePLC 和视觉系统融合的硬件架构，ePLC采用双处理器的主从结构并和视觉系统通过CAN总线通信。

数据帧在各层中的解析，从而实现视觉系统和ePLC的整合。ePLC硬件上的一种定制化设计，软件架构上的三层设计以及融合其中的VCA协议使得整个视觉私服系统变成一种柔性结构。

在接下来的部分中，II会介绍系统组成，包括硬件结构，软件结构，线程结构和内存设计。III会介绍ePLC如何与视觉系统融合，包括VCA协议，PLC接口和视觉接口介绍。IV介绍系统执行过程，包括视觉层执行过程，逻辑层执行过程，算法层执行过程和多线程执行。V中介绍了两个例子，带视觉检测的绕线机和双目接球机器人。

## II. 系统结构

### A. 硬件结构

硬件架构由嵌入式PLC系统和视觉系统组成。嵌入式PLC采用定制化结构，处理器以及外部资源例如数字量输入输出，模拟量输入输出和私服系统控制数量可根据需要定制等。图1显示了一种典型的ePLC系统，采用双处理器架构，主从处理器之间通过共享内存通信。ePLC系统和视觉系统之间可以采用合适的已有通信方式，例如TCP/IP，Modbus,CAN等。图示采用了CAN通信。

### B. 软件结构

通常我们要实现视觉控制系统的典型软件架构如图2左侧所示，通过逻辑程序连接各控制算法组成模块，在通过逻辑关系根据视觉系统的获取信息来连接各模

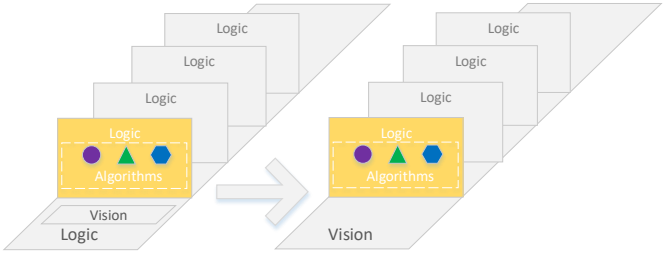


图 2. The three-layer architecture of ePLC

块执行。这种设计通常会将视觉处理，逻辑处理和控制算法混在一起编程，造成混乱。我们通过VCA协议实现算法层，逻辑层和视觉层的三层架构。软件结构设计成图2右侧所示结构。算法程序完成相关算法。逻辑程序处理算法间逻辑关系并调用相关算法。视觉程序解析视觉数据，调用模块执行。

### C. 线程结构

我们采用了 [32]中类似的线程结构。我们特别强调以下三个线程:

1) Visual Thread. 视觉线程，该线程主要完成视觉层协议解析，和逻辑层数据交互和模块调用。
2) LC Thread. 逻辑线程，该线程主要完成逻辑程序处理，改成协议解析，和逻辑层和算法层之间数据交互和算法调用。
3) Algorithm Thread. 算法线程，主要完成和逻辑层之间的数据交互和算法执行。

### D. 内存设计

在内存的PLC专用存储空间主要由位数据区（M区）和字节数据区(D区)组成。整个系统的PLC专用区分布在三个内存中：主控制器内存，主从共享内存和从控制器内存中。在我们的系统中规定了集合的两个属性如下：

**Definition 1** 集合 $S$ 具有 $\mathcal{B}$ 属性：$\exists(S = \{s_1, s_2, ..., s_i, ...s_n\}) \subset M$ and $(\forall s_i \in S) \in \{\mathcal{F}(s_i) = 0, \mathcal{O}(s_i) = 1\}$

**Definition 2** 集合 $S$ 具有 $\mathcal{D}$ 属性：$\exists(S = \{s_1, s_2, ..., s_i, ...s_n\}) \subset D$ and $\forall s_i \in S$ has 4 Byte。

其中$\mathcal{F}(s_i)$表示$s_i$中数据为0，$\mathcal{O}(s_i)$表示$s_i$中数据为1。

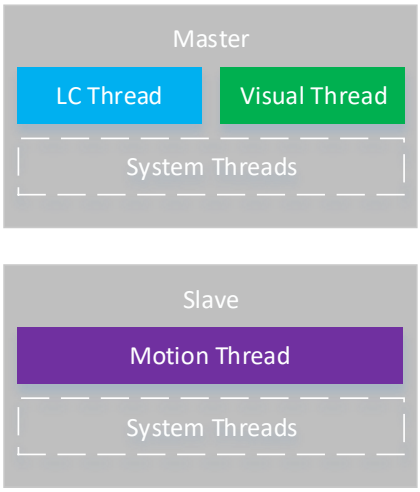


图 3. The process of the three type threads

内存结构如图4所示。主要分三个部分，主处理器内存分配，共享内存分配，从处理器内存分配。各个分区作用介绍如下:

1) LCF。该区域表示逻辑控制区。
2) MPF。主到从数据传递标志区。
3) LCD。逻辑数据存放区。
4) VD。视觉数据存放区。存放从视觉系统传输过来的数据。
5) MtoS。主到从数据存放区。
6) StoM。从到主数据存放区。
7) AF。算法标志区。
8) AD。算法数据区。
9) SPF。从到主传递标志区。

我们定义了主从处理器之间的数据交互方法为$\mathcal{P}$，如下：

$$\begin{cases} \mathcal{P}_{mts} = \mathcal{I}(ms_i, mc_i, mf_i, sa_i, sf_i, sd_i) \\ \mathcal{P}_{stm} = \mathcal{I}(ss_i, sd_i, sf_i, ma_i, mf_i, md_i) \end{cases} \tag{1}$$

$\mathcal{P}_{mts}$和$\mathcal{P}_{mts}$具有相同执行函数$\mathcal{I}$，且执行过程如下: $\mathcal{O}(ms_i) \rightarrow \mathcal{O}(mf_i) \rightarrow send(sd_i) \rightarrow \mathcal{O}(sf_i) \rightarrow check(sd_i) \rightarrow \mathcal{O}(sa_i) \rightarrow \mathcal{O}(mc_i) \rightarrow \mathcal{F}(ms_i) \rightarrow \mathcal{F}(mf_i) \rightarrow \mathcal{F}(mc_i) \rightarrow \mathcal{F}(sf_i) \rightarrow \mathcal{F}(sa_i)$

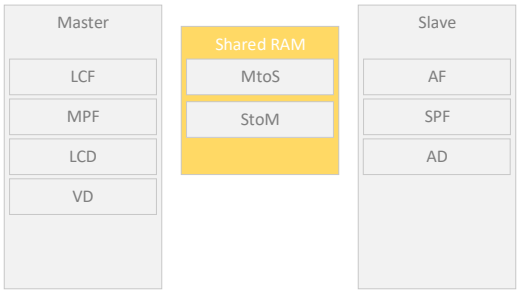


图 4. The process of the three type threads

## III. 视觉控制系统融合

### A. VCA Protocol Template

In cater to most applications, we propose a VCA-based flexible architecture. As shown in Fig. , a protocol template ($PT$) is adopted to support various types of implementations and a $PT$ uniquely corresponds to a type of application. In the flexible architecture, users only need to redesign and reload the $PT$ and then they can reuse the visual servo system again. The $PT$ could be loaded into a stationary address of visual system and ePLC system. After restarting both systems, it will be put into a fixed area of RAM. The parsing modules form both systems will read it when parsing the protocol data.

The $PT$ is defined below:

$$\begin{cases} PT = \{HPT, VLT, CLT, ALT\} \\ HPT = \{CID, TDA\} \\ VLT = \{MID, MAN, MSF, MDA, MAS, CDATA\} \\ CLT = \{AID, APN, AF, ADA, APS, ADATA\} \\ ALT = \{PID, PDATA\} \end{cases} \tag{2}$$

The $PT$ contains four parts: head of protocol template ($HPT$), visual layer template ($VLT$), control layer template ($CLT$) and algorithm layer template ($ALT$). Every part explains as follows:

1) $HPT$: this part includes communication unique ID ($CID$), template data storage address ($TDA$). Every $PT$ only has one $HPT$.
2) $VLT$: it consists of module unique ID ($MID$), module contained algorithm number ($MAN$), module start flag ($MSF$), module data start address ($MDA$), module contained algorithm IDs ($MAS$) and control

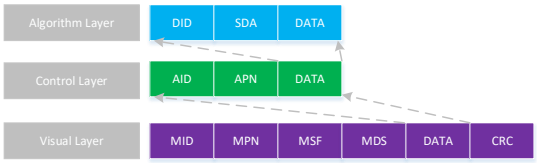


图 5. VCA Protocol.

data ($CDATA$). $VLT$ is not $\emptyset$. Every $MAS$ include $MAN$ algorithm IDs.

3) $CLT$: it is comprised of algorithm unique ID ($AID$), algorithm contained parameter number ($APN$), $AF$, algorithm data start address $ADA$, algorithm contained parameter IDs ($APS$) and algorithm data ($ADATA$). $CLT$ is not $\emptyset$.Every $APS$ include $APN$ parameter IDs.
4) $ALT$: it contains parameter unique ID ($PID$), parameter data ($PDATA$). $CLT$ is not $\emptyset$.

### B. VCA Protocol Frame

The VCA protocol frame $PF$ consists of visual frame, control frame and algorithm frame as shown in 5. We use the same name if the $PT$ and $PF$ both have the items.

1) Visual frame: it consists items of $MID$, $MSF$, $MDA$, visual frame length $VFL$, $CDATA$ and cyclic redundancy check $CRC$.The $CDATA$ contains several control frames.
2) Control frame: it is comprised of $AID$, $AF$, $ADA$, control frame length ($CFL$) and $ADATA$. The $ADATA$ includes several algorithm frames.
3) Algorithm frame: it contains $PID$, $PDATA$. $PID$ is also the address of $PDATA$.

### C. PLC Interface

The PLC interface is designed to transfer VCA protocol frame to PLC in $VS$. It has five steps: create data, $VL$ framing, $CL$ framing, $AL$ framing and transmission.

**Create date**: after image processing, the $VS$ extracts the useful data and storages into visual extracted data ($VED$) in which the parameters could be indexed by the $PID$.

***VL* Framing**: the process is realized as the Algorithm 1. It searches the $PT$ with $mid$ to find the relevant $vlt_x$. Through it, the $VS$ can obtain the $msf$ and $mda$. According to $man$, call the next step ($VL$ Framing) to gain the $cdata$. Then calculate the length ($vfl$) and $crc$ to finish the $VL$ framing.

***CL* Framing**: Algorithm 2 illustrates the process. The $VS$ seeks the $FT$ to gain the $clt_y$ which contains the $af$ and $ada$. According to $apn$, call the next step ($AL$ Framing) to obtain the $cdata$ and then calculate the length ($cfl$) to finish the $CL$ framing.

***AL* Framing**: Algorithm 3 shows the process. The $VS$ obtains the $alt_z$ frome $FT$ and then combines the $pid$ and $pdata$.

**Transmission**: transfer the $VCA$ protocol frame to $ePLC$ with the relevant communication protocol in $CID$.

**Algorithm 1:** $VLFraming$

**Input:** $mid$, $VED$
**Output:** $vlf$
Search the $PT$ with $mid$ and get $vlt_x$;
Create $vlf$ according to $vlt_x$;
**if** *create successfully* **then**
  $vlf.mid$ = $vlt_x.mid$;
  $vlf.msf$ = $vlt_x.msf$;
  $vlf.mda$ = $vlt_x.mda$;
**end**
**for** $i = 0; i < vlt_x.man; i++$ **do**
  ALFraming($vlt_x.mas$[i], $VED$,$cdata$);
**end**
$vlf.cdata$ = $cdata$;
$vlf.vfl$ = Length($vfl$)+4;
$vlf.crc$ = CRC16($vlf$);

### D. Vision Interface

Vision interface is in the ePLC designed to interact with the $VS$. It includes five parts: receiving $VCA$ protocol frame, saving it, $VL$ deframing, $CL$ deframing and $AL$ deframing.

**Receiving *VCA* protocol frame**: according to the $CID$ in $PT$, the $ES$ chooses the communication protocol receives the frame form $VS$.

**Algorithm 2:** $CLFraming$

**Input:** $aid$, $VED$, $cdata$
**Output:** $cdata$
Search the $PT$ with $aid$ and get $clt_y$;
Create $clf$ according to $clt_y$;
**if** *create successfully* **then**
  $clf.aid$ = $clt_y.aid$;
  $clf.af$ = $clt_y.af$;
  $clf.ada$ = $clt_y.ada$;
**end**
**for** $i = 0; i < clt_y.apn; i++$ **do**
  ALFraming($clt_y.aps$[i], $visualData$, $pdata$);
**end**
$clf.cfl$ = Length($clf$);
$cdata$ += $clf$;

**Algorithm 3:** $ALFraming$

**Input:** $pid$, $VED$,$pdata$
**Output:** $pdata$
Search the $PT$ with $pid$ and get $alt_z$;
Create $alf$ according to $alt_z$;
**if** *create successfully* **then**
  $alf.pid$ = $alt_z.pid$;
  $alf.pdata$ = $VED.pid$;
**end**
$pdata$ += $alf$;

**Savint *VCA* protocol frame**: ePLC中需要接收从视觉系统获得的数据。该接口主要包括两部分:数据接收和视觉层数据解析。

**数据接收**：包括双方系统规定，使用已有协议（TCP/IP,MODBUS,CAN等）传递数据。

**视觉层数据解析**：先对数据进行CRC校验，如果数据错误则丢弃数据并通知视觉系统。如果数据正确，根据VLA协议对接收到的数据进行视觉层协议解析，获取数据后通知。

## IV. 系统运行原理

### A. 视觉层的执行

视觉层通过视觉接口和视觉执行模块两部分实现。通过视觉接口，从视觉系统中获取数据。视觉执行模

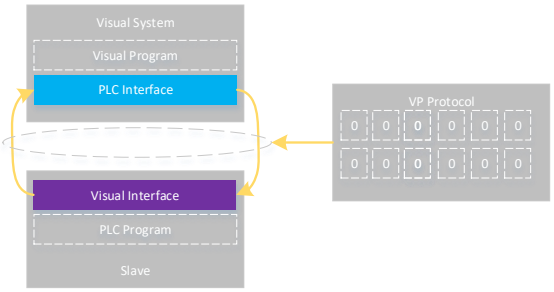


图 6. Data Interaction Based on VP Protocol.

**Algorithm 4:** $VLDeframing$

**Input:** $PDF$
$mid \leftarrow$ four bytes start form $PDF$;
$msn \leftarrow$ four bytes start form $PDF + 4$;
$mda \leftarrow$ four bytes start form $PDF + 8$;
$cdata \leftarrow msn - 12$ bytes start form $PDF + 12$;
$CassMemM[mda] \leftarrow cdata$;
$PDF \leftarrow PDF + msn$;

**Algorithm 5:** $CLDeframing$

**Input:** $mid$
searchPT($mid$, $vlt_x$);
$mda \leftarrow vlt_x.mda$;
**for** $i = 0; i < vlt_x.man; i++$ **do**
  searchPT($vlt_x.mas[i]$, $clt_y$);
  $CassMemS[clt_y.apa] \leftarrow clt_y.apn \times 8$ bytes start form $mda$;
  $mda \leftarrow mda + clt_y.apn \times 8$;
**end**
clean;

**Algorithm 6:** $ALDeframing$

**Input:** $aid$
searchPT($aid$, $clt_y$);
$ada \leftarrow vlt_x.ada$;
**for** $i = 0; i < clt_y.apn; i++$ **do**
  $CassMemA[CassMemA[ada]] \leftarrow CassMemA[ada + 4]$;
  $ada \leftarrow ada + 8$;
**end**
clean;

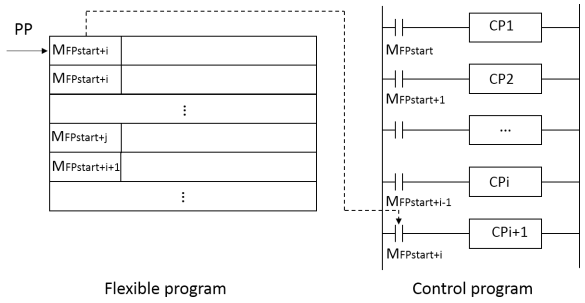


图 7. The execution OF flexible program

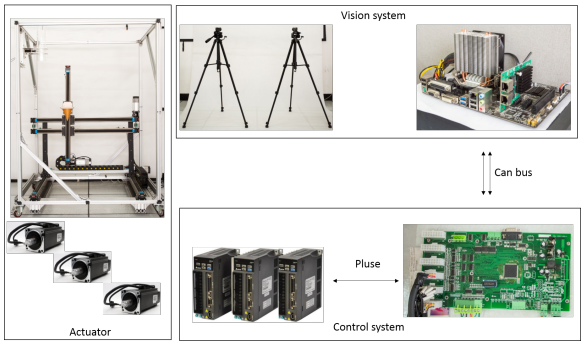


图 8. Visual servo control system

块，通过执行顺序依次执行模块，并将视觉系统的数据不断反馈给逻辑层。

### B. 逻辑层执行

逻辑层由模块数据初始化，逻辑程序，协议解析程序和数据交互程序组成。

当视觉层控制模块调用逻辑层模块执行时，先启动初始化程序初始化模块，运行逻辑程序，通过对视觉层给出的数据解析协议后，通过$\mathcal{P}_{mts}$方法将数据传递给引擎层，并一直获取从引擎层反馈的数据，如果算法执行完毕则清除所有标志位和状态。

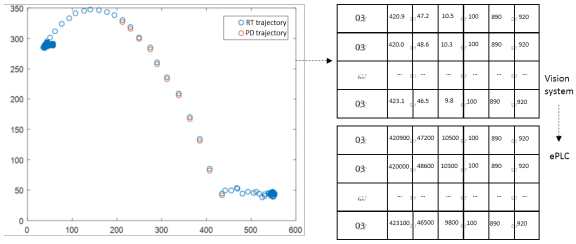


图 9. The execution OF flexible program

### C. 算法层执行

算法层包含算法层协议解析模块，不断执行解析从控制层传递数据，选择更新算法模块或者执行新的模块。

### D. 线程运行机制

通过驱动模块，线程间交互和数据交互实现整个系统的运行。

## V. Case Analysis

### A. *Case 1* 双目抛球机器人

*1)* 协议设计*:*

*2)* 程序设计*:*

*3)* 运行效果*:*

### B. *Case 2* 带视觉的绕线机

*1)* 协议设计*:*

*2)* 程序设计*:*

*3)* 运行效果*:*

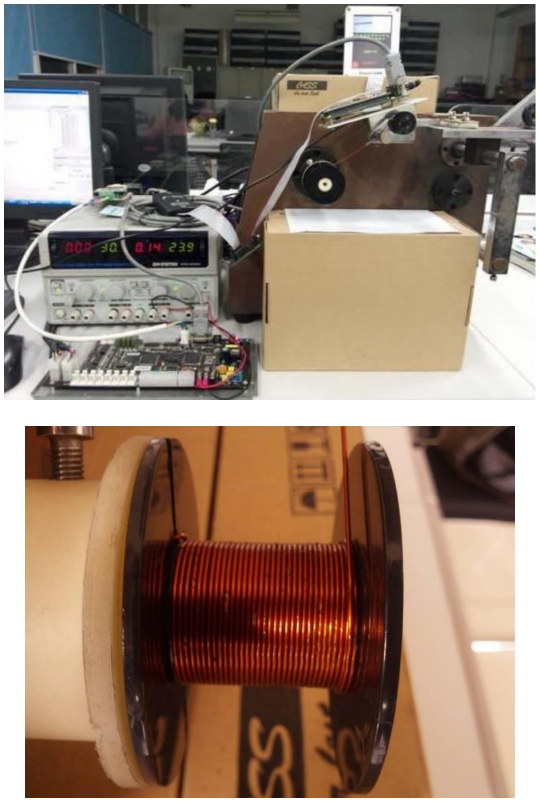

图 10. Winding machine Based on Visual System

## VI. Conclusion

下一步我们将实现在PLC开发平台中对逻辑，视觉，运动控制的统一编程，实现支持视觉运动控制的嵌入式PLC。

## 参考文献


[1] M. P. Kazmierkowski, "Integration technologies for industrial automated systems (zurawski, r., ed.; 2006) [book reviews]," *IEEE Industrial Electronics Magazine*, vol. 1, no. 1, pp. 51–52, 2007.

[2] J. Wan, B. Yin, D. Li, A. Celesti, F. Tao, and Q. Hua, "An ontology-based resource reconfiguration method for manufacturing cyber-physical systems," *IEEE/ASME Transactions on Mechatronics*, vol. PP, no. 99, pp. 1–1, 2018.

[3] D. A. Chekired, L. Khoukhi, and H. T. Mouftah, "Industrial iot data scheduling based on hierarchical fog computing: A key for enabling smart factory," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2018.

[4] A. W. Colombo, R. Schoop, and R. Neubert, "An agent-based intelligent control platform for industrial holonic manufacturing systems," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 1, pp. 322–337, 2006.

[5] A. Vaccaro, M. Popov, D. Villacci, and V. Terzija, "An integrated framework for smart microgrids modeling, monitoring, control, communication, and verification," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 119–132, 2010.

[6] E. Dean, K. R. Amaro, F. Bergner, I. Dianov, and G. Cheng, "Integration of robotic technologies for rapidly deployable robots," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2017.

[7] X. Feng, S. A. Velinsky, and D. Hong, "Integrating embedded pc and internet technologies for real-time control and imaging," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 1, pp. 52–60, 2002.

[8] T. N. Chang, B. Cheng, and P. Sriwilaijaroen, "Motion control firmware for high-speed robotic systems," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1713–1722, 2006.

[9] X. Feng, R. Mathurin, and S. A. Velinsky, "Practical, interactive, and object-oriented machine vision for highway crack sealing," *Journal of Transportation Engineering*, vol. 131, no. 6, pp. 451–459, 2005.

[10] P. Hu, H. Ning, T. Qiu, Y. Zhang, and X. Luo, "Fog computing based face identification and resolution scheme in internet of things," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1910–1920, 2017.

[11] W. Hou, Z. Ning, and L. Guo, "Green survivable collaborative edge computing in smart cities," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1, 2018.

[12] P. Pace, G. Aloi, R. Gravina, G. Caliciuri, G. Fortino, and A. Liotta, "An edge-based architecture to support efficient applications for healthcare industry 4.0," *IEEE Transactions on Industrial Informatics*, vol. PP, no. 99, pp. 1–1.

[13] H. Chen, K. Liu, G. Xing, Y. Dong, H. Sun, and W. Lin, "A robust visual servo control system for narrow seam double head welding robot," *International Journal of Advanced Manufacturing Technology*, vol. 71, no. 9-12, pp. 1849–1860, 2014.

[14] W. Xing, P. Lou, J. Yu, X. Qian, and D. Tang, "Intersection recognition and guide-path selection for a vision-based agv in a bidirectional flow network," *International Journal of Advanced Robotic Systems*, vol. 11, no. 1, p. 1, 2014.

[15] C. Y. Nian and Y. S. Tarng, "An auto-alignment vision system with three-axis motion control mechanism," *International Journal of Advanced Manufacturing Technology*, vol. 26, no. 9-10, pp. 1121–1131, 2005.

[16] Y. Wang, H. Lang, and C. W. D. Silva, "Visual servo control and parameter calibration for mobile multi-robot cooperative assembly tasks," in *IEEE International Conference on Automation and Logistics*, 2008, pp. 635–639.

[17] S. Xiao and Y. Li, "Visual servo feedback control of a novel large working range micro manipulation system for microassembly," *Journal of Microelectromechanical Systems*, vol. 23, no. 1, pp. 181–190, 2014.

[18] H. Wu, L. Lou, C. C. Chen, S. Hirche, and K. Kuhnlenz, "Cloud-based networked visual servo control," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 2, pp. 554–566, 2013.

[19] C. Y. Tsai, C. C. Wong, C. J. Yu, C. C. Liu, and T. Y. Liu, "A hybrid switched reactive-based visual servo control of 5-dof robot manipulators for pick-and-place tasks," *IEEE Systems Journal*, vol. 9, no. 1, pp. 119–130, 2017.

[20] N. Guenard, T. Hamel, and R. Mahony, "A practical visual servo control for an unmanned aerial vehicle," *IEEE Transactions on Robotics*, vol. 24, no. 2, pp. 331–340, 2010.

[21] P. Serra, R. Cunha, T. Hamel, D. Cabecinhas, and C. Silvestre, "Landing of a quadrotor on a moving target using dynamic image-based visual servo control," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1524–1535, 2016.

[22] M. G. Ioannides, "Design and implementation of plc-based monitoring control system for induction motor," *IEEE Transactions on Energy Conversion*, vol. 19, no. 3, pp. 469–476, 2004.

[23] X. M. Shi, W. J. Fei, and S. P. Deng, "The research of circular interpolation motion control based on rectangular coordinate robot," *Key Engineering Materials*, vol. 693, pp. 1792–1798, 2016.

[24] N. Fang, "Design and research of multi axis motion control system based on plc," *Academic Journal of Manufacturing Engineering*, vol. 15, no. 1, pp. 17–23, 2017.

[25] A. Syaichu-Rohman and R. Sirius, "Model predictive control implementation on a programmable logic controller for dc motor speed control," in *Electrical Engineering and Informatics (ICEEI), 2011 International Conference on*. IEEE, 2011, pp. 1–4.

[26] W. F. Peng, G. H. Li, P. Wu, and G. Y. Tan, "Linear motor velocity and acceleration motion control study based on pid+velocity and acceleration feedforward parameters adjustment," *Materials Science Forum*, vol. 697-698, pp. 239–243, 2011.

[27] J. Qian, H. B. Zhu, S. W. Wang, and Y. S. Zeng, "A 5-dof combined robot platform for automatic 3d measurement," *Key Engineering Materials*, vol. 579-580, pp. 641–644, 2014.

[28] O. Co.Ltd, "Cs1w-mc221(-v1)/mc421(-v1) motion control units." *Operation Mannual*, 2004.

[29] P. T. C. 2., *Function blocks for motion control version 1.1*, 2005.

[30] C. Sünder, A. Zoitl, F. Mehofer, and B. Favre-Bulle, "Advanced use of plcopen motion control library for autonomous servo drives in iec 61499 based automation and control systems," *E & I Elektrotechnik Und Informationstechnik*, vol. 123, no. 5, pp. 191–196, 2006.

[31] S. S. S. GmbH, "Logic and motion control integrated in one iec 61131-3 system:development kit for convenient engineering of motion, cnc and robot applications." 2017.

[32] H. Wu, Y. Yan, D. Sun, and S. Rene, "A customized real-time compilation for motion control in embedded plcs," *IEEE Transactions on Industrial Informatics*, 2018.