Auto Disturbances Refection Visual Servoing Control for Excavator Robot Based on Particle Swarm Optimization

Tongtong Lou

College of Information and Computer Engineering,
Northeast Forestry University, Harbin 150040, China

Shihui Jiang, Jiye Han

Material Science and Engineering College,
Northeast Forestry University, Harbin 150040, China

Hanxue Sun*

College of Engineering & Technology, Northeast Forestry University,
Harbin, China, 150040
**Corresponding author, E-mail: 542748979@qq.com*

Abstract
The article researched on typical joint type robot visual servo system structure and designed visual servo controller based on image moments features. A typical robot visual servo simulation system model is built. The article introduced particle swarm optimization algorithm on controller parameter for optimization and compared it with niching particle swarm optimization algorithm. The article also designed suitable auto-disturbance-rejection visual servoing controller for excavator robot and set tuning parameters for auto-disturbance rejection controller by particle swarm optimization algorithm.
Key words: AUTO DISTURBANCES REFECTION, VISUAL SERVOING CONTROL, PARTICLE SWARM OPTIMIZATION, EXCAVATOR ROBOT

1. Introduction

In 1995, Kennedy and Eberhart proposed a novel optimization algorithm which is called Particle Swarm Optimization algorithm (PSO). The algorithm and Ant Colony algorithm (ACO) is similar which is a optimization algorithm based on swarm intelligence (SI). It simulates the process of foraging flocks, while its functional and genetic algorithm (GA) is very similar [8]. Particle Swarm Optimization algorithm originates from simulations of simple social systems which is a good optimization tool [4]. Due to its advantages of being simple and easy to realize, it is increasingly being applied to function optimization, neural networks, pattern recognition, and applications of traditional optimization algorithms [5].

Visual was people's most main senses to get information, and it also was robot's most important perception capacity. We can use visual sensor to get figure to feedback information, and this can construct location closed ring control of robot which is called visual servoing. It is different to machine vision. Machine vision is defined for analyzing figure automatically to get data of description of a scene or controlling of a species action. But visual sensor is defined for image acquisition and analysis due to servoing control of robot. The principle is that feedback information is given in the shortest possible time and constituting a closed loop position control of robots from the direct image feedback information [6].

Robot visual servo control technology was born in the 1980 of the 20th century that is a key technology in the intelligent robot research and development. It is the important technology of multisensor information fusion and integration in robotics applications. Compared with the non-vision sensor for robot control based on traditional technologies, it has obvious advantages: greater flexibility, higher precision, higher robustness against robot calibration. It makes traditional robots to giant step towards the intelligent, humanoid robots, strong environmental adaptability [3]. The outside information of work environment of robot is got by vision sensor to the controller in robot visual servo system to adjust its position in real time and conduct precise tracking or positioning and accomplish the work. Their research related to image processing, machine vision, control theory, robotics, kinematics, dynamics, and many other disciplines [2]. Currently the main content of a visual servo robot includes visual servo control system, fast and accurate image processing algorithms and visual servo controller design [7]. The results can be directly used for robot hand-eye system, automatic obstacle avoidance for mobile robot and adaptive to the environment, trajectory tracking problems [8]. The real-time robot visual servo system involves image processing, control theory, kinematics, dynamics, real-time computing, and many other respects [1].

The rest of the paper is organized as follows. In Section 2, particle swarm optimization is summarized briefly. In Section 3, research on visual servo system is described. In Section 4, experiments are presented and the results are discussed. Finally, a conclusion is provided in Section 5.

Overview of particle swarm optimization

Algorithm theory

The basic concept of particle swarm optimization stems from research on bird flock predator behavior. People get revelation from bird folck predator model and use for solution optimization problem. In particle swarm optimization, each optimization solution is a bird in search space which is called particle. All of particles have a fitness value decided by optimization function and they also have a velocity to determine the direction and distance. Particle swarm optimization initials into group random particles and then all particles follow the current optimal particles to search optimal solutions in solutions space. In each iteration, particles update themselves through tracking two "extreme value". The first extreme value is the optimal solutions that are found by particles himself, and it is called personal best. The another is the optimal solutions that are found by whole particles group currently, and it is called global best.

We assume that $X_i = (\mathrm{x}_{i1}, \mathrm{x}_{i2}, \cdots, \mathrm{x}_{id})$ indicates the i-th particle, d indicates the dimensions of the particle. The best position passed by the particle is indicated by $p_b = (\mathrm{p}_{i1}, \mathrm{p}_{i2}, \cdots, \mathrm{p}_{id})$, and the best position passed by whole particles group is indicated by $g_b = (\mathrm{p}_{g1}, \mathrm{p}_{g2}, \cdots, \mathrm{p}_{gd})$. The speed in *i*-th particle is indicated by $V_i = (v_{i1}, v_{i2}, \cdots, v_{id})$. For each generation, particles update their speed and location based on a formula such as the following

$$v_{id} = w \times v_{id} + c_1 \times random() \times (p_{id} - x_{id}) + \mathrm{c}_2 \times random() \times (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

w indicates the inertia weights. $random()$ is a random number between zero and one. C1 and C2 indicates a learning factor. In addition, every dimension speed of the particle has a maximum velocity which is $V\max$.

Algorithm steps

There are six basic steps in particle swarm optimization. 1) initialize each particle's initial position and velocity. 2) calculate the fitness value of each particle. 3) update the best location with current fitness values as the best fitness value superior to its own position for each particle. 4) update the overall best new locations with the particle that has the best fitness value in the entire paricles group if there is such an individual whose fitness value is better than the history best location for the whole particles group. 5) recalculate the velocity of the particles according to equation (1), and then recalculate the particle's position according to equation (2). 6) If it reaches the maximum number of iterations or minimum standards, it finishes; otherwise, it goes to step 2).

Algorithm characteristics

Although particle swarm optimization and GA are very similar in function, but its implementation has the following five distinct advantage. (1) no crossover and mutation operations. Searching relies on velocity of particle. (2) with memory. The history best location of particles can be passed to other particles through memory. (3) requiring less adjustment of parameters. The structure is simple and easy to implement. (4) using real code. It is decided by the solution of the problem directly. The number of variable can be used as the dimensions of the particle directly. (5) the faster convergence. Only the best information is passed to other particles in the iterative evolution, and this is a one-way flow of information.

Research on Visual servo system

From the perspective of feedback information, robot vision system can be classified into position-based visual servo system and image-based visual servo control. Feedback deviation of the former is calculated in a 3D cartesian space and feedback deviation of the latter is calculated in a 2D image plane.

Position-based Visual servo system

Visual servo errors are defined in 3D cartesian space. Visual or characteristics information is used to estimate the relative position between the end of manipulator and the object. The frame of the program as shown in Figure 1.

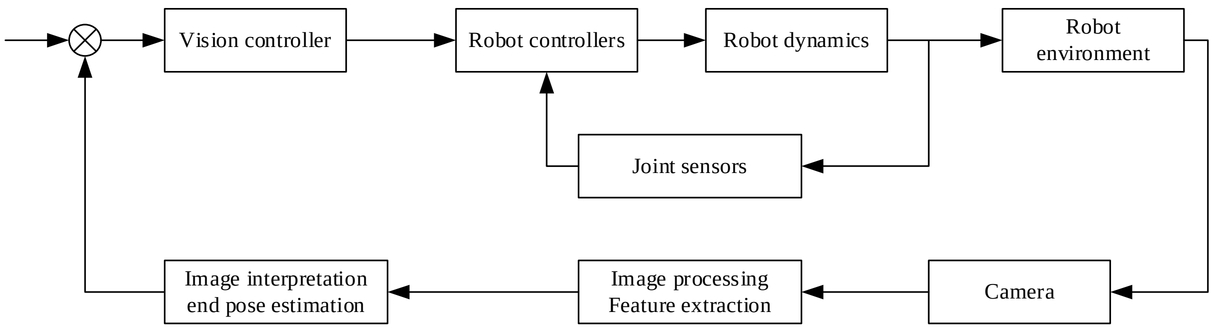

Figure 1. Position-Based Visual Servo System

Image-based Visual servo system

Visual servo errors are defined in image feature space directly. Feature information observed by the cameras is directly used for feedback and we don't need to estimate the three-dimensional attitude. The frame of the program as shown in Figure 2.

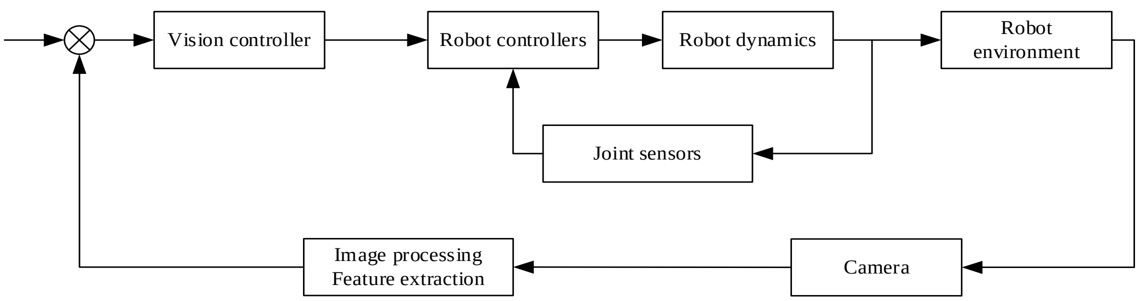

Figure 2. Image-Based Visual Servo System

Vision controller

Most robot visual servo system currently used dynamic structure way of look-and-move. We can use coordinates transform and track generated links due to low sampling rate of visual sensor and big operation volume of figure operation, and this makes visual sampling cycle T to be greater than robot joint servo control sampling cycle TC. The visual controller is designed according to performance of manipulator and visual system and complexity of the task, we can select different design method.

(1) PID control technique. The robot is used as a controlled object. We entered the location or speed to cartesian space or robot joint space regardless of its dynamic nature, and then the visual servo errors can establish control law.

$$\mu = K_{p^e}(\mathrm{k}) + \mathrm{k}_1 \sum_{1}^{k} e(k) + K_D(e(\mathrm{k}) - \mathrm{e}(\mathrm{k}-1)) \quad (3)$$

(2) Task function approach. Espian introduced the concept of task functions in sensor space and used to design controllers.

$$e(r(t)) = C(S(r(t)) - S^*) \quad (4)$$

(3) Vision controller design based on state-space description. It assumes that track and manipulator control performance based on joint sensor control is idealized.

$$x(k) = x_d(\mathrm{k}) = \mu(\mathrm{k}) \quad (5)$$

If we assume that state variable changes continuous and visual control cycle T is enough small, we can have following state space description:

$$\xi(k+1) = A\xi(k) + \mathrm{B}\,\mu(\mathrm{k}) + \mathrm{E}\,\mathrm{d}(\mathrm{k}) + \mathrm{H}\,\mathrm{v}(\mathrm{k}) \quad (6)$$

$$\varphi(k) = C\,\mathrm{q}^{-d}\,\xi(k) + D\,\mathrm{w}(\mathrm{k}) \quad (7)$$

Computational results and comparisons

After analysis, the TD parameter is opposite and independent. So the controller parameters located between the ESO and NLSEF. In order to coordinate influence between the two parameters, we use particle swarm optimization algorithm to optimize the main performance parameters. Base coordinate is selected in turret turning center. The bucket moved on the z coordinate plane motion which digging. Scoop is equivalent to manipulators grippers at the end, and bucket target above can be used to identify shovel and bucket tilt can be used to identify bucket attitude. We can control buckets to fit the attitude to do sand mining while working. The bucket moved on the x-z coordinate plane motion which mining. Position error is showed in table 1.

Table 1. Static Target Positioning Error in Visual Servo System

Time	X(mm)	Y(mm)	Z(mm)	$w_z(°)$
1	20	30	100	60
5	16.67	21.71	38.81	45.44
10	12.04	15	0.08	32.44
15	8.73	10.22	-6.43	23.25

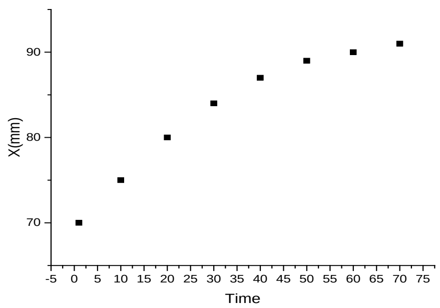

Figure 3. Change in X Direction

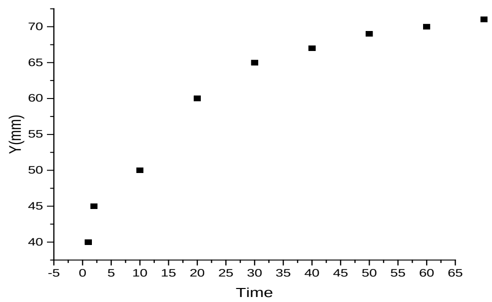

Figure 4. Change in Y Direction

Static object location

We conduct the simulation of location in a binary image. The position of the target object in the camera coordinate system changes showed in Figure 3 and Figure 4..

Dynamic object positioning

We conduct the simulation of location in a binary image. The position of the target object in the camera coordinate system changes showed in following figures.

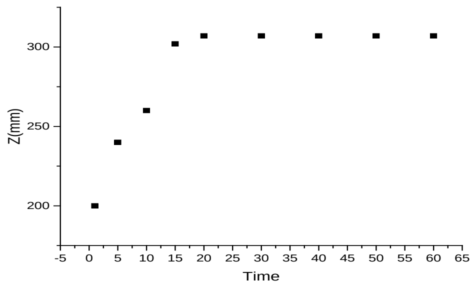

Figure 5. Change in Z Direction

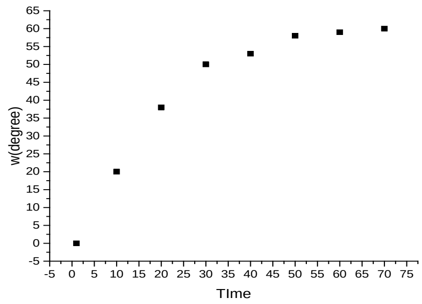

Figure 6. Change in *w* Direction

Conclusion

In this paper, a particle swarm optimization algorithm is developed for excavator robot with auto disturbances refection visual servoing control. (1) The article proposed auto disturbance-rejection visual servoing control strategy for excavator robot and used particle swarm optimization algorithm to optimize the parameters of auto-disturbance-rejection controller. (2) Due to the simple structure of article swarm optimization algorithm, less parameters need to be adjusted and this can simplify the process of auto-disturbance-rejection controller parameter optimization. It is desirable to further apply particle swarm optimization algorithm to solving those more complex real-world optimization problems and it will be our further work.

Acknowledgements

This work is supported by the Fundamental Research Funds for the Central Universities (No.2572015BC19).

References

1. Feddema J T and Mitchell O R (1989) Vision-guided servoing with feature based trajectory generation. *IEEE Trans. on Robotics and Automation*, 5(5), p.p.691-700.
2. Hill J and Park W T (1979) Real time control of robot with a mobile camera. *Proc. 9th Int. Sympo. on Industrial Robots* , Washington, USA, 233-246.
3. Kim Y H, Lee S W, Yang H S, et al. (2012). Toward autonomous robotic containment booms: visual servoing for robust inter-vehicle docking of surface vehicles. *Intelligent Service Robotics*, 5(1), p.p.1-18.
4. Kennedy J, Eberhart R C (1995) Particle swarm optimization. *Proc. of IEEE International Conference on Neural Networks*, p.p.1942-1948.
5. Liu X L, Xiong L P (2013) Robot Visual Servo Based on Particle Swarm Optimized ADRC. *Advanced Materials Research*, Vol. 756, p.p.669-672.
6. Ma Hongyu, Su Jianbo (2003) Uncal ibrated robotic hand-eye coordination based on theory of coupled ADRC. *Robot*, 25(1), p.p.39-43.
7. Qiu B, Wan J, Wang J (2012) Active Disturbance Rejection Controller of the Aerial Photography Stabilized Platform. *Opto-Electronic Engineering*, 39(4), 21-26.
8. Shi Y, Eberhart R C (1998) Parameter selection in particle swarm optimization. *Lecture Notes in Computer Science*, Vol. 1447, p.p.591-600.