4. Uncalibrated Eye-in-hand Visual Servoing

This chapter develops a convex characterization for the control of robot manipulator in an uncalibrated environment. First, the control strategies consisting of PD and CTC are explored for the control of robot manipulators [82], [83]. Secondly, an LMI based optimization scheme is derived for the computation of composite Jacobian [84]. By invoking suitable system representation and Lyapunov analysis, the stability conditions are described in terms of linear matrix inequalities.

4.1. Control Strategies

Over the years, many type of control schemes such as robust control [85], adaptive control [86], learning control [87], and many others are proposed for the control of robot manipulator. These techniques usually require high level of mathematical
understanding in order to apply them successfully. Contrarily, PD-control and PD-CTC strategies provide simple and effective solution; hence, they are the most commonly used schemes in the robot industry today. They actually facilitate well to conveniently derive robot controllers in an effective manner, while providing a framework to bring together classical independent joint control and some modern design techniques.

4.1.1. PD-Type

An independent joint PD-control scheme for \( n \)-link serial manipulator is formulated in this section. The PD-type compensator is mostly used due to its simplicity and effectiveness. An independent joint PD-control scheme can be written in vector form as

\[
\tau = K_p e + K_v \dot{e}
\]  

(4.1)

where \( e \) denotes the error between the desired joint displacements \( q_d \) and the actual joint displacements \( q \). \( K_p \) and \( K_v \) are p.d (diagonal) Proportional and Derivative gain matrices. The gravitational torque is also included in the control law in order to nullify the effect produced by \( g(q) \) in (3.1). The inclusion of this gravity improves the tracking performance, as well as, helps in removing the steady state error. Therefore, the PD control can be modified as follows

\[
\tau = K_p e + K_v \dot{e} + g(q)
\]  

(4.2)
The resulting control scheme appears in Figure 4.1. The PD control using some reasonable estimates of the system parameters is used to stabilize the robot system in the sense that all internal signals remain bounded [82].

Figure 4.1: PD control scheme
4.1.2. PD Computed-Torque Control (CTC)

The classical control algorithms usually overlook the inherent nonlinearities involving dynamical behavior of the manipulator. In effect, classical control laws deteriorate for high speed operation when nonlinear effects become important. For such scenario, an advanced control algorithms such as Computed-torque control is able to achieve high speed and precision.

4.1.2.1 Derivation of Inner Feed-forward Loop

In order to derive the CTC strategy, it is required to compute desired trajectory $q_d$ using some interpolation technique. To ensure that the controller behaves well in the joint space, the tracking error is also defined in the joint space as

$$ e(t) = q_d(t) - q(t) \quad (4.3) $$

To demonstrate the influence of the input $\tau$ on the tracking error, differentiate twice to obtain

$$ \dot{e} = \dot{q}_d - \dot{q} $$
$$ \ddot{e} = \ddot{q}_d - \ddot{q} $$

Solving now for $\ddot{q}$ in (3.1), we have

$$ \ddot{q} = M^{-1}(\tau - C\dot{q} - g) \quad (4.4) $$

Substituting this into the error differential equation yields

$$ \ddot{e} = \ddot{q}_d + M^{-1}(C\dot{q} + g - \tau) \quad (4.5) $$
Defining the control input function as

\[ u = \ddot{q}_d + M^{-1}(C \dot{q} + g - \tau) \]  \hspace{1cm} (4.6)

A state \( x(t) \in \mathbb{R}^{2n} \) can be defined by the following vector

\[ x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \]  \hspace{1cm} (4.7)

And write the tracking error dynamics as

\[ \frac{d}{dt} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} 0 & I \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} u \]  \hspace{1cm} (4.8)

This is a linear error system in Brunovsky canonical form and is driven by the control input \( u \). The feedback linearizing transformation (4.6) may be inverted to yield

\[ \tau = M(\ddot{q}_d - u) + C \dot{q} + g \]  \hspace{1cm} (4.9)

The control law of this form is famously known as Computed-torque control law. The CTC facilitates well that there is no state-space transformation involved between (3.1) and (4.8). Therefore, if such a control \( u \) is selected that stabilizes (4.8), so that \( e \) goes to zero, then the nonlinear control input \( \tau \) given by (4.9) will cause trajectory following in the robot arm (3.1).

4.1.2.2 PD Outer Loop Design

CTC is known to ascertain asymptotic stability using PD (diagonal) gain matrices. One way to select the auxiliary control signal \( u \) is as the proportional-derivative feedback,
\[ u = -K_p e - K_v \dot{e} \quad (4.10) \]

Then the PD-CTC law with gain matrices has the final form using (4.9)

\[ \tau = M(q)\dot{q}_d + K_p e + K_v \dot{e} + C \dot{q} + g \quad (4.11) \]

The objective is to design an input torque \( \tau \), so that the tracking error \( e \to 0 \) as \( t \to \infty \) with the following assumptions:

**Assumption 4.1:** The robot manipulator is \( n \)-link serial manipulator having all revolute joints.

**Assumption 4.2:** The control law in (4.11) is a simplified version from the family of computed-torque-like controllers.

The closed loop error dynamics are

\[ \ddot{e} + K_p e + K_v \dot{e} = 0 \tag{4.12} \]

or in state-space form

\[
\begin{bmatrix}
\frac{de}{dt}
\
\frac{d\dot{e}}{dt}
\end{bmatrix} =
\begin{bmatrix}
0 & I \\
-K_p & -K_v
\end{bmatrix}
\begin{bmatrix}
e
\
\dot{e}
\end{bmatrix}
\tag{4.13}
\]

It is usual to take the \( n \times n \) gain matrices diagonal so that

\[ K_v = \text{diag}\{k_v\}, \quad K_p = \text{diag}\{k_p\} \tag{4.14} \]

Then the error system is asymptotically stable as long as the \( k_v \) and \( k_p \) are all positive. The overall closed-loop system with PD-CTC is shown in Figure 4.2. The computed-torque technique [88], [89] provides a mathematically well-defined, basic control algorithm for the study of the effects of dynamics on real-time robot control.
4.2. Design Considerations

Before moving to the visual servo control, several issues related to visual servoing scheme and camera placement need to be discussed. These questions should be answered first in order to design a vision-based control system. How many cameras should be used? Where the camera should be placed? Whether the 2D image features or 3D coordinates can be used to derive a robot manipulator? The answer to these questions can be found in the sections below.
4.2.1. Camera Configuration

Probably the most important decision to be made when constructing a vision-based control system is where to place the camera. The camera is configured in two possible ways: it can be fixed somewhere in the workspace or can be mounted on the robot manipulator. These are often referred to as eye-to-hand and eye-in-hand camera configurations, respectively.

With a fixed camera configuration, the camera is placed in such a fashion as it can observe both the manipulator and the target object. There are several advantages to this approach. Since the camera is not allowed to make any movement, camera fov does not change as the manipulator moves. The geometrical properties between the camera and the workspace are also fixed. The main advantage of this configuration which makes it less preferable to others is as the manipulator moves through the workspace, it can occlude the camera’s fov. This can be particularly important for tasks that require high precision.

In second type of configuration, the camera is often attached to the manipulator above the wrist so that the motion of the wrist does not affect the camera motion. In this way, the camera can observe the motion of the end effector at a fixed resolution and without occlusion as the manipulator moves through the workspace. One difficulty that confronts the eye-in-hand configuration is that the geometric relationship between the camera and the workspace changes as the manipulator moves. The fov can change drastically for even small motions of the manipulator,
particularly if the link to which the camera is attached experiences a change in orientation.

4.2.2. Position-based vs. Image-based Approaches

Based upon the way in which the data provided by the vision system is used, the visual servoing approaches are mainly categorized in two respects.

The first approach is commonly known as Position-based visual servo scheme. With this approach, the visual data is employed to build a partial 3D representation of the world. The main problem with position-based scheme is related to the difficulty of building the 3D representation in real time. In particular, this method tends not to be robust with respect to errors in camera calibration. Furthermore, with position-based scheme, there is no direct control over the image itself. Therefore, a common problem with this approach is that camera motion can cause the object of interest to leave the camera fov.

An Image-based visual servo scheme is the other scheme that uses image features directly to control the robot motion. An error function is directly defined in an image and a control law is constructed that maps this error directly to robot motion. To date, the most commonly adopted approach is to choose easily detected points on an object as visual features. The error function is then defined as the difference between the desired and measured locations of these points in the image.
In the light of above mentioned facts, the proposed schemes are designed in a way that all the desired features must remain inside the camera fov, otherwise the servoing would fail. Therefore the camera is configured in an eye-in-hand fashion to avoid occlusion of desired objects. To get rid of the tiresome calibration procedure, and to eradicate the effect of camera calibration errors, all the proposed schemes are based upon IBVS model.

4.3. Uncalibrated Visual Servoing

The uncalibrated visual servo control is derived for the case, where no prior information of camera, kinematic and object model is available. The uncalibrated visual servoing scheme is derived based upon these basic assumptions:

Assumption 4.3: The monocular vision system is rigidly attached to the manipulator's end-effector.

Assumption 4.4: The camera has always in sight all of the desired features.

In order to steer the end-effector to the desired position, it is required that error between the initial and desired features i.e. $\xi$ converges towards zero at each time step. The objective function to be minimized is the norm of the error i.e.

$$ F = \frac{1}{2} \xi^T \xi $$

The assumption 4.3 models the circumstances where camera has an eye only on the observed object, whereas assumption 4.4 confines the use of the proposed scheme for
stationary targets only. If, for example, the target is moving with respect to time, then the assumption would not be valid.

4.3.1. Simple Visual Servo Controller

In IBVS, feature's position can only be determined through camera; thus, direct knowledge of $q_d$ is not available. Usually, $q_d$ can be obtained by solving the inverse image and kinematics problem. However, the proposed scheme implements direct visual servo control that computes joint torques; hence, using vision alone to stabilize the mechanism. The control input $\tau$ which is fed to the manipulator is calculated using proposed control law

$$\tau = M (\dot{q}_d - u) + g$$  \hspace{1cm} (4.16)

This controller is from a class of computed-torque-like control law. If we develop relationship between desired joint velocity and desired feature velocity using (3.11), it will be

$$\dot{s}_d = J_c^T \dot{q}_d$$  \hspace{1cm} (4.17)

From assumption 3.1, we have $s_d = 0$; hence, $\dot{q}_d, \ddot{q}_d = 0$. Subsequently, using (4.16) we obtain

$$\tau = M (-u) + g$$  \hspace{1cm} (4.18)

The control input $u$ chosen here is the Proportional feedback which includes transpose Jacobian, and is desired to stabilize the overall scheme, so that $\xi$ goes zero.
\[ \mathbf{u} = J_e^{\top} K_p \lambda \xi \]  

(4.19)

where \( K_p \in \mathbb{R}^{k \times k} \) is a diagonal p.d proportional matrix, which means that each axis is controlled separately. \( \lambda \) is a positive gain and has value between 0 and 1. \( J_e^* \) is calculated using the LMI optimization algorithm. By plugging control input \( \mathbf{u} \) from (4.19) into (4.18), the overall robot arm input becomes

\[ \mathbf{\tau} = -MJ_e^{\top} K_p \lambda \xi + \mathbf{g} \]  

(4.20)

It is worth noticing that the controller directly uses the feature error vector \( \xi \). This means that the manipulator input is directly computed based upon image feature error. The controller also requires knowledge of the composite Jacobian \( J_e^{\top} \) and gravitational torque \( \mathbf{g} \). Figure 4.3 depicts a closed-loop block diagram of the system.

The overall closed-loop system is obtained by substituting the control action \( \mathbf{\tau} \) from (4.20) into the robot dynamics (3.1).

\[ M \ddot{\mathbf{q}} + C \dot{\mathbf{q}} = -MJ_e^{\top} K_p \lambda \xi \]  

(4.21)

The system behavior can be written in terms of state vector \( \mathbf{q}^{\top} \mathbf{q}^{\top} \in \mathbb{R}^{2n} \) as [90]

\[ \frac{d}{dt} \begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} \dot{\mathbf{q}} \\ M^{-1}(-MJ_e^{\top} K_p \lambda \xi - C \dot{\mathbf{q}}) \end{bmatrix} \]  

(4.22)

Notice that the closed-loop system is described by an autonomous nonlinear differential equation. From assumption 3.2, we have \( \begin{bmatrix} \mathbf{q}^{\top} \\ \dot{\mathbf{q}}^{\top} \end{bmatrix} = \begin{bmatrix} \mathbf{q}_d^{\top} \\ 0 \end{bmatrix} \) as an equilibrium point.
Figure 4.3: The closed-loop block diagram of the system

**Discussion**

Most visual servo controllers are velocity based controllers, means that the kinematic screw velocity will be fed to the robot controller as an input, and the controller most commonly is of the form

$$v_c = -\lambda L_s^+ \xi$$

where $L_s^+ \in \mathbb{R}^{6xk}$ is known as the Moore-Penrose pseudo-inverse of $L_s$, i.e.,

$$L_s^+ = (L_s^T L_s)^{-1} L_s^T.$$ When $k=6$, if $\det L_s \neq 0$ it is possible to invert $L_s$, giving the control $v_c = -\lambda L_s^{-1} \xi$.

In real-time visual servo systems, it is impossible to determine perfectly either $L_s$ or $L_s^+$. Therefore, an approximation of one of these two matrices must be realized and is
denoted by $\hat{L}_s$ or $\hat{L}_s^+$. Using this notation, the control law is in fact:

$$v_e = -\lambda \hat{L}_s^+ \xi$$

This is the basic design implemented by most visual servo controllers. The issue that remains is how to approximate the interaction matrix. [91]

**Approximating the Interaction matrix**

Several instances can be found in literature for constructing the estimate $\hat{L}_s^+$ to be used in the control law. One popular scheme is, of course, to choose $\hat{L}_s^+ = L_s^+$ [1]. Another popular approach is to choose $\hat{L}_s^+ = L_{s_d}^+$, where $L_{s_d}$ is the value of $L_s$ for the desired position $s = s_d = 0$ [92]. In this case $\hat{L}_s^+$ is constant, and only the desired depth of each point has to be set, which means no varying 3D parameters has to be estimated during the visual servo. Finally the choice $\hat{L}_s^+ = 1/2 \left( L_s + L_{s_d} \right)^+$ has recently been proposed in [93]. Since $L_s$ is involved in this method, the current depth of each point must also be available.

These schemes work well for the environment where information of camera and object model is known. The problem arises where we do not have exact information of any of these. In order to determine the optimization and stability of the vision-based controller described in (4.20), following theorem is proposed.

Before moving to the theorem, following are the system properties that will help in deriving the theorem.
Property 4.1: Any square matrix $Q$ can be broken up into a symmetric part and a skew-symmetric part, namely [72]:

$$Q = \left(\frac{Q + Q^T}{2}\right) + \left(\frac{Q - Q^T}{2}\right)$$  \hspace{1cm} (4.23)

Property 4.2: Any skew-symmetric matrix when written in the form of quadratic function is always zeros i.e. [103],

$$x^T (Q - Q^T) x = 0$$  \hspace{1cm} (4.24)

As, skew symmetric part is zero, we are only left with

$$x^T Q x = x^T \left(\frac{Q + Q^T}{2}\right) x$$  \hspace{1cm} (4.25)

Theorem 4.1: Consider a robot-camera system satisfying assumptions 4.3-4.4. Let $K_p$ and $\lambda$ are defined as a diagonal p.d matrix and convergence factor. Suppose that there exists matrix $J^*_c$ satisfying the following LMI

$$\begin{bmatrix}
-\lambda K_p & -0.5 \lambda K_p J^*_c M \\
-0.5 M J^*_c^T K_p \lambda & 0_{n \times n}
\end{bmatrix} < 0, \forall x \neq 0$$  \hspace{1cm} (4.26)

Thus $V < 0$ as long as $x \neq 0$, so that $x = 0$ is a globally stable equilibrium point.

Proof: From assumption 3.1 and 3.4, $J^*_c$ is a full rank matrix i.e. $\text{rank}(J^*_c) = n$. Using this fact and considering that $M$ and $K_p$ are square nonsingular matrix ($\text{rank}(M) = n$) and ($\text{rank}(K_p) = k$), $\text{rank}(MJ^*_c K_p) = n$ is acquired. Consider
a positive definite function \( V > 0, \forall x \neq 0 \). It is to prove that the time derivative of \( V \) is strictly negative along the trajectories of the closed-loop system for all admissible \( q \).

To derive the LMI and ensure stability, following Lyapunov function candidate is chosen

\[
V = \frac{1}{2} \dot{q}^T M \dot{q} + \frac{1}{2} \left( s(q) - s(q_d) \right)^T K_p \left( s(q) - s(q_d) \right) \tag{4.27}
\]

The time derivative of \( V \) is given by

\[
\dot{V} = \frac{1}{2} \left( \dot{q}^T M \dot{q} + \dot{q}^T M \dot{q} + \dot{q}^T M \dot{q} \right) + \frac{1}{2} \xi^T K_p \dot{\xi} + \frac{1}{2} \dot{\xi}^T K_p \xi \tag{4.28}
\]

which then simplifies to

\[
\dot{V} = \frac{1}{2} \left( \dot{q}^T M \dot{q} + \dot{q}^T M \dot{q} + \dot{q}^T M \dot{q} \right) + \frac{1}{2} (2 \xi^T K_p \dot{\xi}) \tag{4.29}
\]

By employing closed-loop dynamics (4.21), it follows that

\[
V = \frac{1}{2} \left( -MJ_c^T K_p \lambda \xi \right)^T \dot{q} + \frac{1}{2} \dot{q}^T M \dot{\dot{q}} - \frac{1}{2} \dot{q}^T C \dot{q} - \frac{1}{2} \dot{q}^T C^T \dot{q} \tag{4.30}
\]

\[
+ \frac{1}{2} \dot{q}^T \left( -MJ_c^T K_p \lambda \xi \right) + \xi^T K_p \dot{\xi}
\]

combining the alike terms results into

\[
V = \frac{1}{2} \left( -MJ_c^T K_p \lambda \xi \right)^T \dot{q} + \frac{1}{2} \dot{q}^T M \dot{\dot{q}} - \frac{1}{2} \dot{q}^T \left( C + C^T \right) \dot{q} \tag{4.31}
\]

\[
+ \frac{1}{2} \dot{q}^T \left( -MJ_c^T K_p \lambda \xi \right) + \xi^T K_p \dot{\xi}
\]
By Property 4.1, symmetric part of $C$ is replaced by its original and its skew-symmetric part. Since, $\frac{1}{2}\dot{q}^T(C-C^T)\dot{q} = 0$ using skew-symmetric property, we have

$$V' = \frac{1}{2}\left(-MJ_c^{\ast T}K_p\lambda\xi\right)^T\dot{q} + \frac{1}{2}\dot{q}^TM\dot{q} - q^TCq + \frac{1}{2}\dot{q}^T\left(-MJ_c^{\ast T}K_p\lambda\xi\right) + \xi^TK_p\dot{\xi}$$

(4.32)

which is then re-written in this form

$$V' = \frac{1}{2}\left(-MJ_c^{\ast T}K_p\lambda\xi\right)^T\dot{q} + \dot{q}^T\left(\frac{1}{2}M - C\right)\dot{q} + \frac{1}{2}\dot{q}^T\left(-MJ_c^{\ast T}K_p\lambda\xi\right) + \xi^TK_p\dot{\xi}$$

(4.33)

Making use of the Property 3.1, some of the terms can be dropped out. Also, if we would like to ensure exponential decoupled decrease of error, we use $\dot{\xi} = -\lambda\xi$

$$V' = \frac{1}{2}\xi^T\left(-\lambda K_p J_c^* M\right)\dot{q} + \frac{1}{2}\dot{q}^T\left(-MJ_c^{\ast T}K_p\lambda\right)\xi + \xi^T\left(-\lambda K_p\right)\xi$$

(4.34)

In order to assess global asymptotic stability, Theorem 3.2 is employed, which states that $V'$ must be n.d. Thus, the asymptotic stability of the system is ensured if we have

$$V' = \frac{1}{2}\xi^T\left(-\lambda K_p J_c^* M\right)\dot{q} + \frac{1}{2}\dot{q}^T\left(-MJ_c^{\ast T}K_p\lambda\right)\xi + \xi^T\left(-\lambda K_p\right)\xi < 0$$

(4.35)

Writing this in LMI form, the above equation is transformed into

$$\begin{bmatrix} \xi \\ \dot{q} \end{bmatrix}^T \begin{bmatrix} -\lambda K_p & -0.5\lambda K_p J_c^* M \\ -0.5MJ_c^{\ast T}K_p\lambda & 0_{n\times n} \end{bmatrix} \begin{bmatrix} \xi \\ \dot{q} \end{bmatrix} < 0$$

(4.36)

The above LMI is re-written by leaving the outer factors as

$$\begin{bmatrix} -\lambda K_p & -0.5\lambda K_p J_c^* M \\ -0.5MJ_c^{\ast T}K_p\lambda & 0_{n\times n} \end{bmatrix} < 0$$

(4.37)
The negative definiteness of the above equation implies the negative definiteness of $\dot{V}$. Therefore according to Theorem 3.2, the equilibrium state at the origin is globally asymptotically stable.

$J^*_c$ is the varying parameter and is update with each iteration. According to assumption 3.4, composite Jacobian has rank $n$. The only possibility that the Jacobian matrix loses its rank is at $s(t) = s_d$.

$J^*_c$ is the on-line update parameter in this case, where $K_p$ is the gain matrix chosen by the designer. The decaying factor $\lambda$ has a value $0 < \lambda \leq 1$ also chosen by the designer, and its choice determines the rate of convergence. The controller based upon these unknowns ensures that the system exhibits convergence as well as stability by minimizing the error norm.

$$\text{Minimize} \quad \frac{1}{2} \xi^T \xi \quad (4.38)$$

The optimal solution of $J_c$ has its effect on the asymptotic stability of the system. One possible choice is to calculate its optimal value during an off-line step. But, this seems inappropriate where depth of points and camera focal length are varying. This can also cause system to reach local minima. Therefore, to ascertain global asymptotic stability $J_c$ must be updated at each iteration of the control scheme.

Since, the LMI introduced in (4.37) is such that it always ensures negative definiteness for (4.36); hence, $\dot{V}$ is globally n.d function. Therefore, by invoking the
Lyapunov's direct method, it can be concluded that $\begin{bmatrix} q^T & q^T \end{bmatrix} = \begin{bmatrix} q_d^T & 0 \end{bmatrix}$ is a stable equilibrium. The whole process needs no prior information about camera or object model. The LMI established in (4.37) is solved using convex optimization.

Regarding the stability analysis described above, the following comments can be made.

- In case of redundant robots, the proposition made above that we have $\begin{bmatrix} q^T & q^T \end{bmatrix} = \begin{bmatrix} q_d^T & 0 \end{bmatrix}$ as an equilibrium point is no longer valid. If the Jacobian matrix satisfies $\text{rank}(J(q)) = r < n$ (i.e. redundant robot), it may not exist a unique equilibrium point and a space of dimension $n - r$ may be arbitrarily assigned. In case of a redundant robots, the control objective $\lim_{t \to \infty} \xi(t) = 0$ can be satisfied, but a space of joint coordinates of dimension $n - r$ may be freely set. This means that phenomena such as self-motion may be present.

- $K_p$ is a diagonal p.d proportional matrix and $\lambda$ is a scaling factor. Both of these parameters have their own significance as $K_p$ is used for controlling each axis independently and $\lambda$ is used to control an exponential decrease of error. Therefore, these are not combined together and known as separate tuning factors.
• It is well known that Interaction matrix is singular if \( s \) is composed by the image of three points such that they are collinear, or belong to a cylinder containing the camera optical center. Using more than three points generally allows us to avoid such singularities.

**4.4. Summary**

This chapter identifies an LMI based optimization method for vision-based control of robot manipulator in an uncalibrated environment. Two types of formulations are derived in this chapter. First one devises an independent joint controller to control \( n \)-link serial manipulator. The controller uses the joint angles error vector to drive the robot. The next scheme uses vision-based controller to drive the robot and utilizes LMI optimization to determine composite Jacobian without using any prior information of system parameters. This scheme can get unstable, where features leave camera fov. To encounter this, extension of second algorithm is proposed in the next chapter which incorporates visibility as well as kinematic constraints.