Image-based coordinates for rigid motion

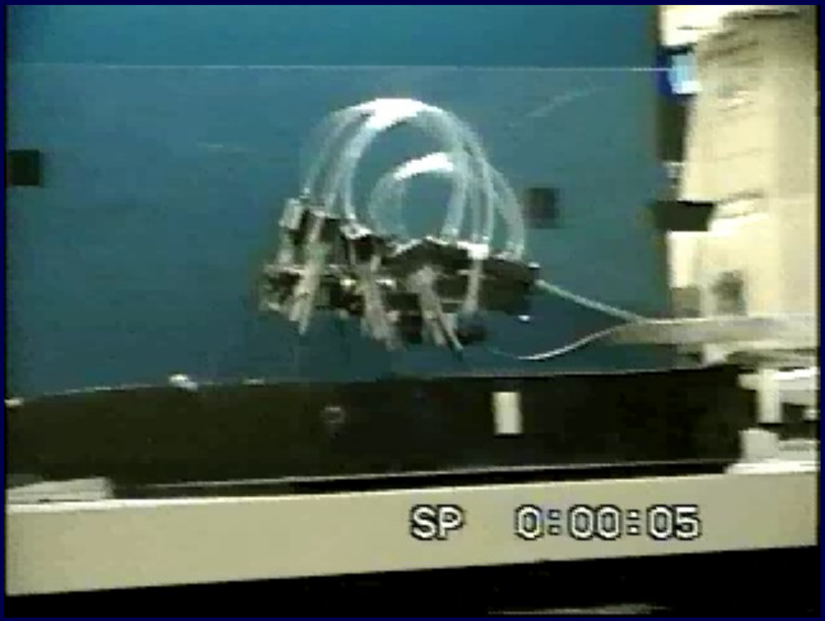
Noah Cowan
Johns Hopkins University
Department of Mechanical Engineering
ncowan@jhu.edu

14–16 July 2003

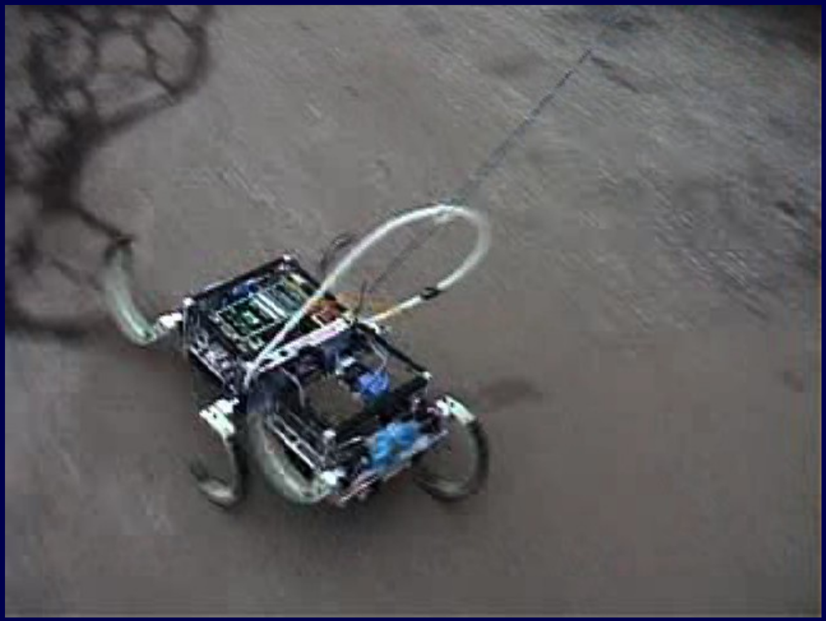
Abstract

A collection of novel coordinate transformations from a robot's task space (e.g. $\mathrm{SE}(3)$) to a sensor-space enables the construction of globally convergent, fully sensor-based, dynamic servoing systems that keep measurements within full "view" of the sensor suite.

Motivation: Two Dynamical Legged Robots



Sprawl running on a treadmill.



RHex running outdoors.

Controlling Dynamic Robots with Sensor Feedback

Controlling Dynamic Robots with Sensor Feedback

- Key factors

Controlling Dynamic Robots with Sensor Feedback

- Key factors
  - ⋆ dynamics: most visual servoing algorithms are *kinematic*
  - ⋆ convergence: only a handful of algorithms with provable convergence
  - ⋆ sensor limits: features may exceed sensor range (e.g. FOV), especially when *inertia* is at work!
  - ⋆ robots are underactuated and / or exhibit nonholonomic constraints

Controlling Dynamic Robots with Sensor Feedback

- Key factors
  - ⋆ dynamics: most visual servoing algorithms are *kinematic*
  - ⋆ convergence: only a handful of algorithms with provable convergence
  - ⋆ sensor limits: features may exceed sensor range (e.g. FOV), especially when *inertia* is at work!
  - ⋆ robots are underactuated and / or exhibit nonholonomic constraints

- RHex and Sprawl

Controlling Dynamic Robots with Sensor Feedback

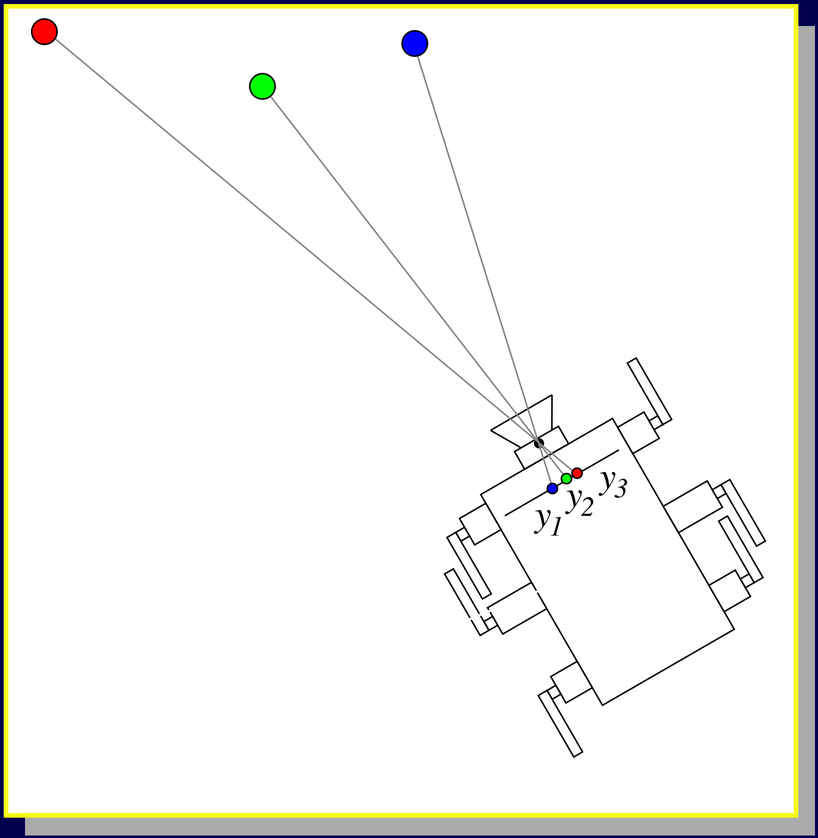
- Key factors
  - ⋆ dynamics: most visual servoing algorithms are *kinematic*
  - ⋆ convergence: only a handful of algorithms with provable convergence
  - ⋆ sensor limits: features may exceed sensor range (e.g. FOV), especially when *inertia* is at work!
  - ⋆ robots are underactuated and / or exhibit nonholonomic constraints

- RHex and Sprawl
  - ⋆ are highly stable, fast, dynamical platforms
  - ⋆ exhibit stable locomotion highly dynamical running
  - ⋆ can generate “second order” turns while running (not kinematic turns)
  - ⋆ cannot run “sideways,” i.e. they are nonholonomically constrained

Visual Servoing: Basic idea

Given

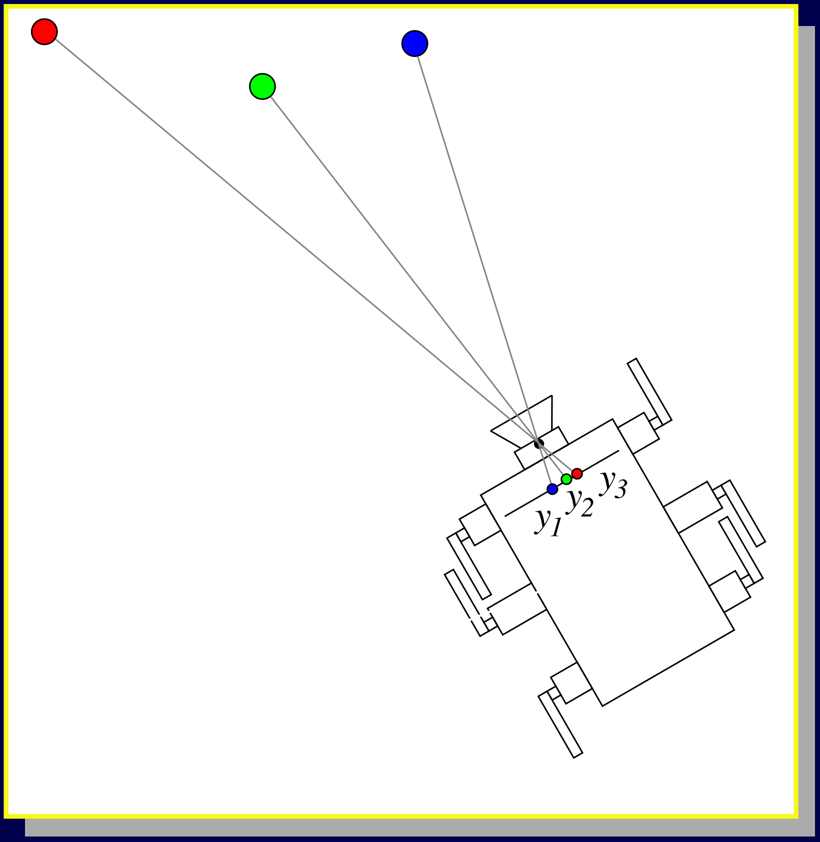
- robot: $q \in \mathbb{R}^n$



Visual Servoing: Basic idea

Given

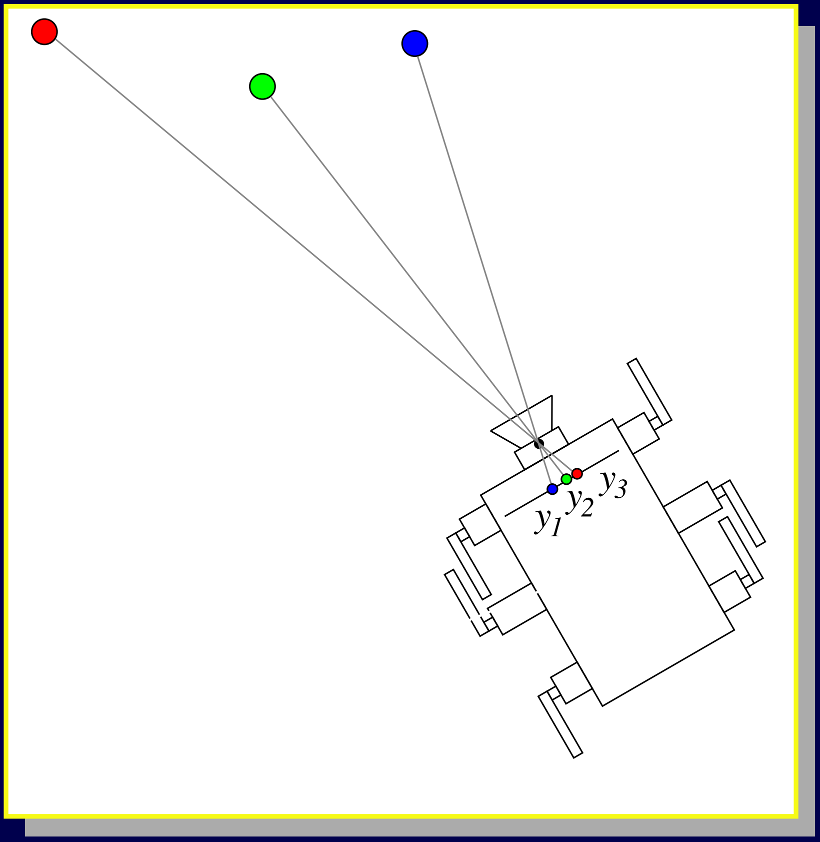
- robot: $q \in \mathbb{R}^n$
- features: $p_1,\ p_2,\ p_3$



Visual Servoing: Basic idea

Given

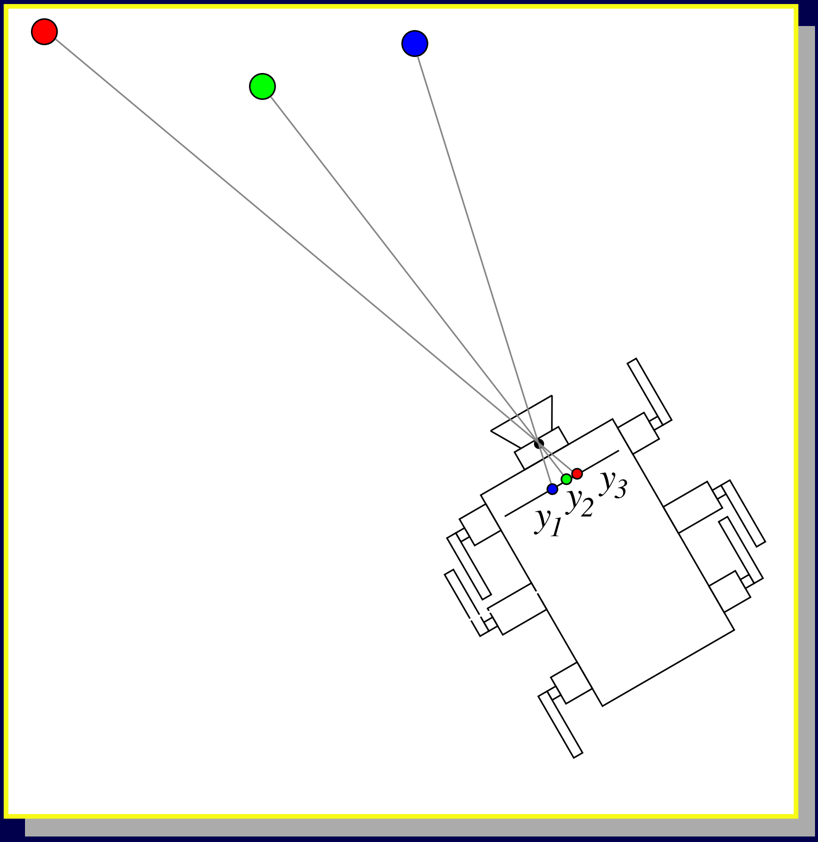
- robot: $q \in \mathbb{R}^n$
- features: $p_1,\ p_2,\ p_3$
- camera image: $y = c(q)$



Visual Servoing: Basic idea

Given

- robot: $q \in \mathbb{R}^n$
- features: p_1, p_2, p_3
- camera image: $y = c(q)$



Objective

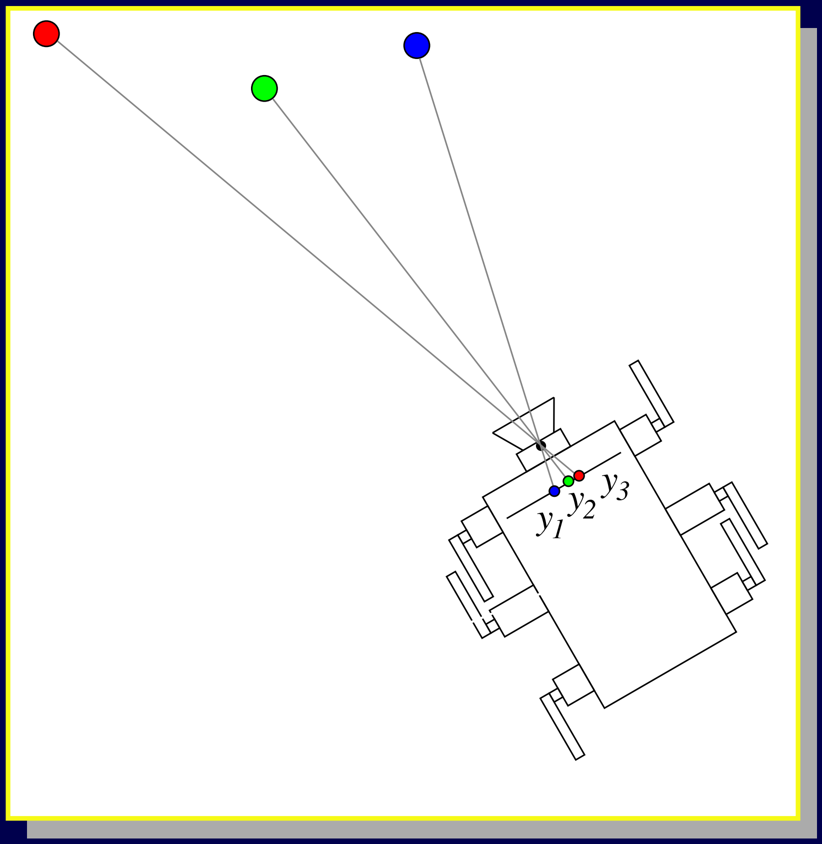
Visual Servoing: Basic idea

Given

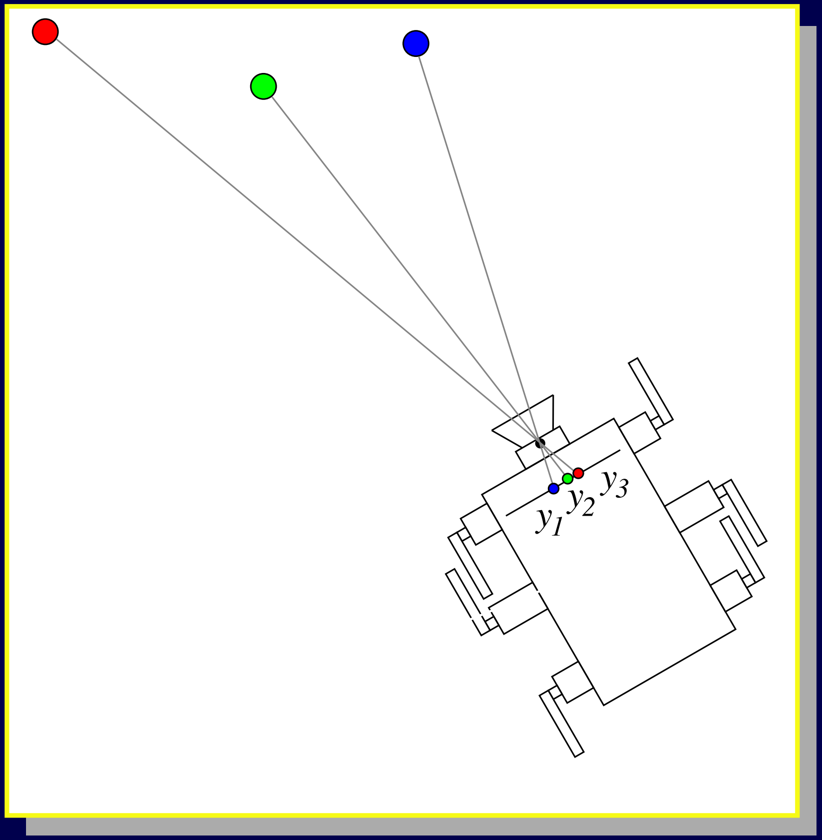
- robot: $q \in \mathbb{R}^n$
- features: $p_1,\ p_2,\ p_3$
- camera image: $y = c(q)$

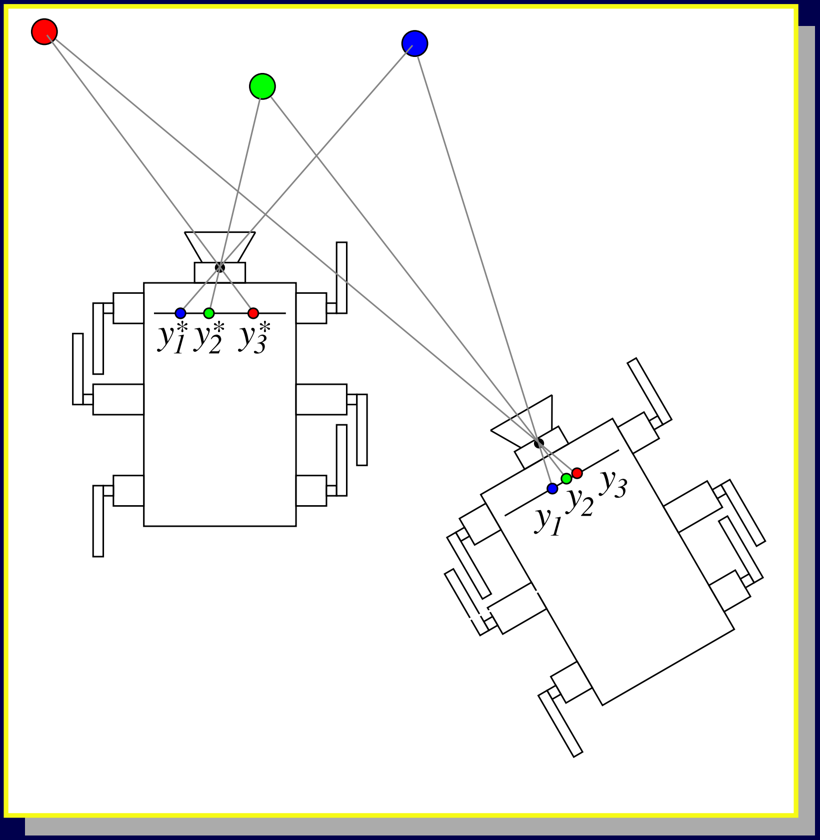
Objective

- move robot to align current view with a “goal” image, namely $y \to y^*$ as $t \to \infty$



Visual Servoing: Basic idea



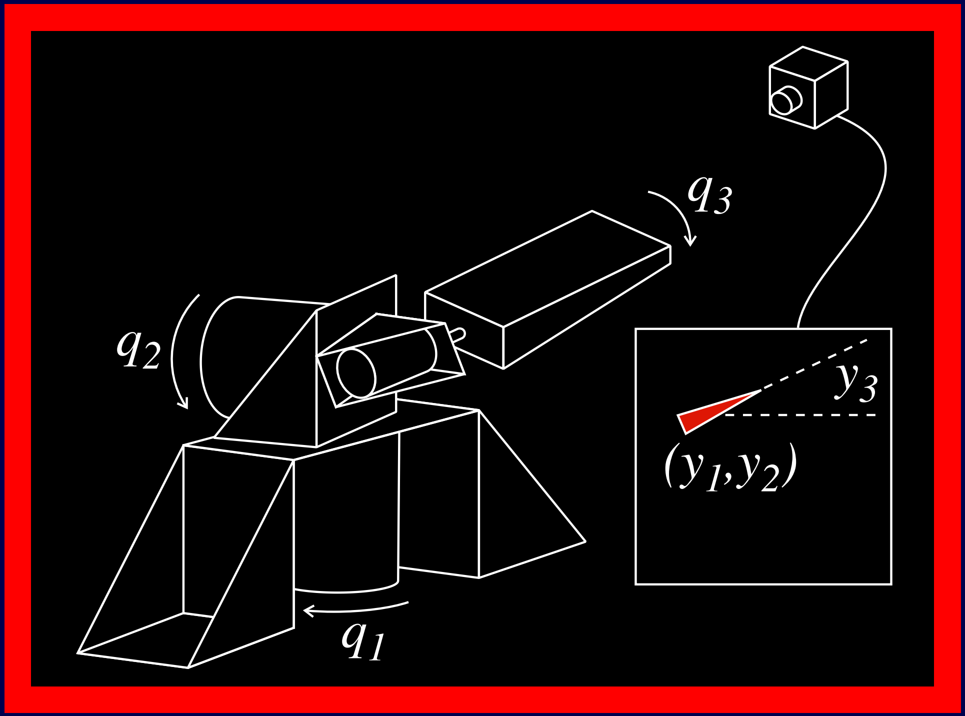


Visual Servoing: Eye-in-Hand vs. Fixed Camera

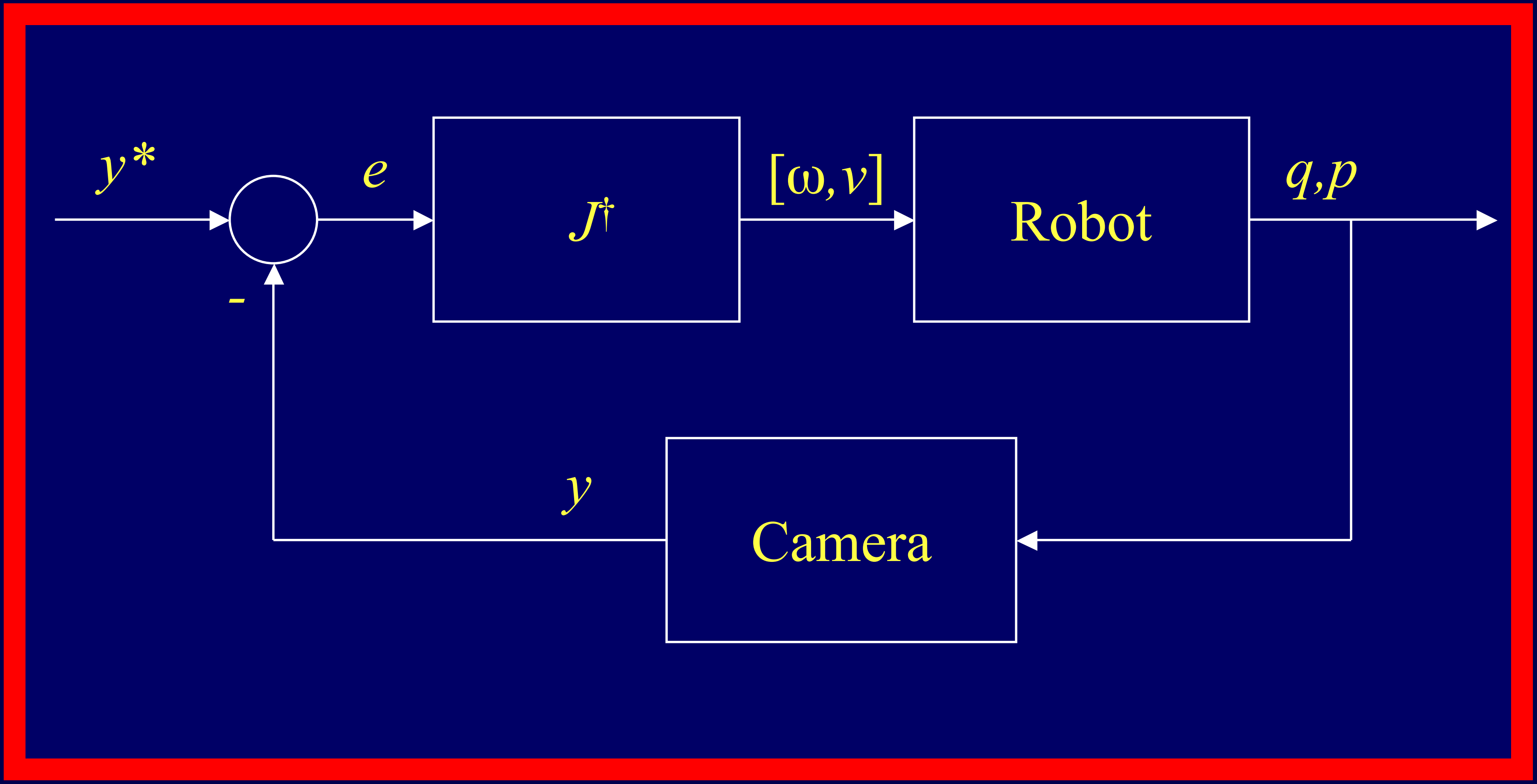
- For RHex and other *eye-in-hand* setups, the camera is attached to the moving robot

Visual Servoing: Eye-in-Hand vs. Fixed Camera

- For RHex and other *eye-in-hand* setups, the camera is attached to the moving robot
- In the *fixed camera* setup, the robot moves in front of stationary camera



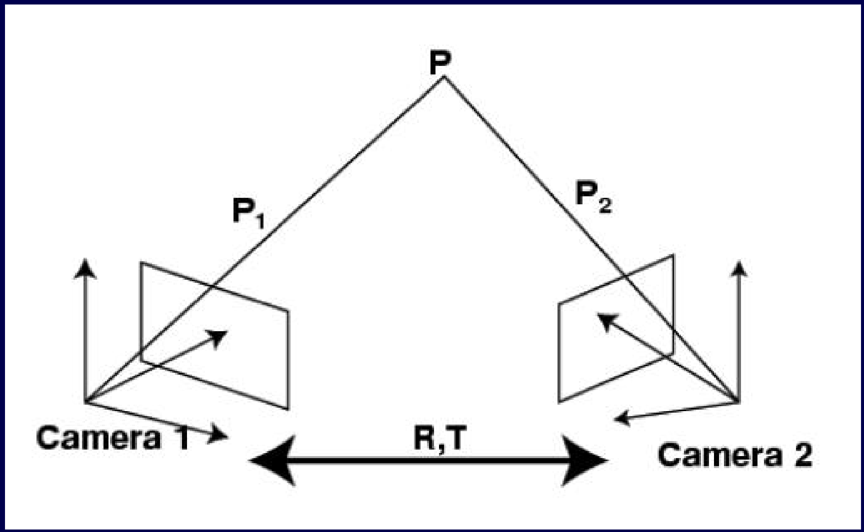
Visual Servoing: Traditional Approach



[Hutchinson, Hager, Corke 96] [Hill & Park 79]

Visual Servoing: State-of-the-Art

- dynamical blimp servo [Zhang & Ostrowski, 99]
- global partial pose servo [Taylor & Ostrowski, 00]
- 2 1/2 D servo [Malis, et. al., 99]
- partitioned visual servo [Hutchinson & Corke, 01]
- epipolar servoing [Prattichizzo et. al, ’02]



Taking a step back...

- Q: What is servoing?

Taking a step back...

- Q: What is servoing?
  $\Longrightarrow$ Controlling the mechanical degrees of freedom, q_i, $i = 1, \ldots n$, of a system to some desired "configuration," q_i^*

Taking a step back...

- Q: What is servoing?
  $\Longrightarrow$ Controlling the mechanical degrees of freedom, q_i, $i = 1, \ldots n$, of a system to some desired “configuration,” q_i^*

- Q: What is **visual** servoing?

Taking a step back...

- Q: What is servoing?
  $\Longrightarrow$ Controlling the mechanical degrees of freedom, q_i, $i = 1, \ldots n$, of a system to some desired "configuration," q_i^*

- Q: What is **visual** servoing?
  $\Longrightarrow$ Controlling the "visual degrees of freedom," y_i, $i = 1, \ldots m$, to some "desired image", y_i^*

Taking a step back...

- Q: What is servoing?
  $\Longrightarrow$ Controlling the mechanical degrees of freedom, q_i, $i = 1, \ldots n$, of a system to some desired "configuration," q_i^*

- Q: What is **visual** servoing?
  $\Longrightarrow$ Controlling the "visual degrees of freedom," y_i, $i = 1, \ldots m$, to some "desired image", y_i^*

- More questions:

Taking a step back...

- Q: What is servoing?
  $\Longrightarrow$ Controlling the mechanical degrees of freedom, q_i, $i = 1, \ldots n$, of a system to some desired "configuration," q_i^*

- Q: What is **visual** servoing?
  $\Longrightarrow$ Controlling the "visual degrees of freedom," y_i, $i = 1, \ldots m$, to some "desired image", y_i^*

- More questions:
  - ⋆ What are degrees of freedom?

Taking a step back...

- Q: What is servoing?
  $\Longrightarrow$ Controlling the mechanical degrees of freedom, q_i, $i = 1, \ldots n$, of a system to some desired "configuration," q_i^*

- Q: What is **visual** servoing?
  $\Longrightarrow$ Controlling the "visual degrees of freedom," y_i, $i = 1, \ldots m$, to some "desired image", y_i^*

- More questions:
  - ⋆ What are degrees of freedom?
  - ⋆ How are the mechanical degrees of freedom controlled?

Taking a step back...

- Q: What is servoing?
  $\Longrightarrow$ Controlling the mechanical degrees of freedom, q_i, $i = 1, \ldots n$, of a system to some desired "configuration," q_i^*

- Q: What is **visual** servoing?
  $\Longrightarrow$ Controlling the "visual degrees of freedom," y_i, $i = 1, \ldots m$, to some "desired image", y_i^*

- More questions:
  - ⋆ What are degrees of freedom?
  - ⋆ How are the mechanical degrees of freedom controlled?
  - ⋆ What are "visual" degrees of freedom?

Generalized coordinates & Lagrange's equations

- the *generalized coordinates*, q_i, $i = 1, \ldots n$, form a *local chart* for the configuration manifold $\mathcal{Q}$

Generalized coordinates & Lagrange's equations

- the *generalized coordinates*, q_i, $i = 1, \ldots n$, form a *local chart* for the configuration manifold $\mathcal{Q}$
- there is one coordinate for each degree-of-freedom

Generalized coordinates & Lagrange's equations

- the *generalized coordinates*, q_i, $i = 1, \ldots n$, form a *local chart* for the configuration manifold $\mathcal{Q}$
- there is one coordinate for each degree-of-freedom
- often the coordinates are hand picked, e.g. joint angles of a robot, or the position and orientation (x, y, θ) of a body in the plane

Generalized coordinates & Lagrange's equations

- the *generalized coordinates*, q_i, $i = 1, \ldots n$, form a *local chart* for the configuration manifold $\mathcal{Q}$
- there is one coordinate for each degree-of-freedom
- often the coordinates are hand picked, e.g. joint angles of a robot, or the position and orientation (x, y, θ) of a body in the plane
- equations of motion, i.e. *Lagrange's Equations*, are expressed

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = \tau + F_{\text{ext}} \qquad \Longrightarrow \qquad M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau + F_{\text{ext}}(q, \dot{q})$$

where $L(q, \dot{q}) = \text{KE} - \text{PE}$ is the *Lagrangian*, where $\text{KE} = \dot{q}^T M(q)\dot{q}$ is the kinetic energy and PE is the potential, and $G(q) = \nabla_q \text{PE}$. F_{ext} represents "non energetic" forces (e.g. friction), and τ is a torque input vector.

Example Lagrangian systems

- mass-spring-damper;
  - $q = x$ is the position of the mass
  - $\mathrm{KE} = \frac{1}{2} M \dot{x}^2 \implies C = 0$
  - $\mathrm{PE} = \frac{1}{2} K x^2 \implies G(x) = Kx$
  - $F_{\mathrm{ext}} = -B\dot{x}$

Example Lagrangian systems

- mass-spring-damper;
  - ⋆ $q = x$ is the position of the mass
  - ⋆ $\mathrm{KE} = \frac{1}{2}M\dot{x}^2 \implies C = 0$
  - ⋆ $\mathrm{PE} = \frac{1}{2}Kx^2 \implies G(x) = Kx$
  - ⋆ $F_{\mathrm{ext}} = -B\dot{x}$
- dynamical equations

$$M\ddot{x} + B\dot{x} + Kx = 0$$

A simplified view of robot control

- dynamical equations & control input

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + F_{\mathrm{ext}}(q,\dot{q})$$

A simplified view of robot control

- dynamical equations & control input

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + F_{\text{ext}}(q,\dot{q})$$

$$\tau = u - F_{\text{ext}}(q,\dot{q}) + G(q)$$

A simplified view of robot control

- dynamical equations & control input

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + F_{\text{ext}}(q,\dot{q})$$

$$\tau = u - F_{\text{ext}}(q,\dot{q}) + G(q) \quad \Longrightarrow \quad M(q)\ddot{q} + C(q,\dot{q})\dot{q} = u$$

A simplified view of robot control

- dynamical equations & control input

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + F_{\mathrm{ext}}(q,\dot{q})$$

$$\tau = u - F_{\mathrm{ext}}(q,\dot{q}) + G(q) \qquad \Longrightarrow \quad M(q)\ddot{q} + C(q,\dot{q})\dot{q} = u$$

- the virtual control u can shape the energy landscape, e.g.

$$u = -\nabla\varphi(q) - B(q,\dot{q})$$

where $\varphi(q)$ is an *artificial potential function* and B is a damping term.

A simplified view of robot control

- dynamical equations & control input

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + F_{\text{ext}}(q,\dot{q})$$

$$\tau = u - F_{\text{ext}}(q,\dot{q}) + G(q) \quad \Longrightarrow \quad M(q)\ddot{q} + C(q,\dot{q})\dot{q} = u$$

- the virtual control u can shape the energy landscape, e.g.

$$u = -\nabla\varphi(q) - B(q,\dot{q})$$

  where $\varphi(q)$ is an *artificial potential function* and B is a damping term.

- this is just like the mass-spring-damper but with our own virtual, high-dimensional spring-damper!

A simplified view of robot control

- dynamical equations & control input

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau + F_{\mathrm{ext}}(q,\dot{q})$$

$$\tau = u - F_{\mathrm{ext}}(q,\dot{q}) + G(q) \qquad \Longrightarrow \quad M(q)\ddot{q} + C(q,\dot{q})\dot{q} = u$$

- the virtual control u can shape the energy landscape, e.g.

$$u = -\nabla\varphi(q) - B(q,\dot{q})$$

  where $\varphi(q)$ is an *artificial potential function* and B is a damping term.

- this is just like the mass-spring-damper but with our own virtual, high-dimensional spring-damper!

- In general, the system will converge to the minima of φ

Changes of coordinates

- A map $g : \mathcal{Q} \rightarrow \mathcal{Y}$ is a *diffeomorphism* if it is a smooth and smoothly invertible function, i.e.

Changes of coordinates

- A map $g : \mathcal{Q} \to \mathcal{Y}$ is a *diffeomorphism* if it is a smooth and smoothly invertible function, i.e.
  - ⋆ g has an inverse, g^{-1} on the image $g(\mathcal{Q})$
  - ⋆ the matrix $G = \partial g / \partial q$, called the *Jacobian*, is well defined on $\mathcal{Q}$
  - ⋆ the Jacobian is nonsingular, i.e. $|G| = 0$

Changes of coordinates

- A map $g : \mathcal{Q} \to \mathcal{Y}$ is a *diffeomorphism* if it is a smooth and smoothly invertible function, i.e.
  - ⋆ g has an inverse, g^{-1} on the image $g(\mathcal{Q})$
  - ⋆ the matrix $G = \partial g / \partial q$, called the *Jacobian*, is well defined on $\mathcal{Q}$
  - ⋆ the Jacobian is nonsingular, i.e. $|G| = 0$
- Note that the Jacobian of g^{-1} is just given by the *matrix inverse* G^{-1}

Changes of coordinates

- A map $g : \mathcal{Q} \rightarrow \mathcal{Y}$ is a *diffeomorphism* if it is a smooth and smoothly invertible function, i.e.
  - ⋆ g has an inverse, g^{-1} on the image $g(\mathcal{Q})$
  - ⋆ the matrix $G = \partial g / \partial q$, called the *Jacobian*, is well defined on $\mathcal{Q}$
  - ⋆ the Jacobian is nonsingular, i.e. $|G| = 0$
- Note that the Jacobian of g^{-1} is just given by the *matrix inverse* G^{-1}
- If $y = g(q)$, then there are as many DOF's of y as there are of q

Examples of a changes of coordinates

- Common transformations of $\mathbb{R}^3$ that are local diffeomorphisms

Examples of a changes of coordinates

- Common transformations of $\mathbb{R}^3$ that are local diffeomorphisms
  - ⋆ Cylindrical coordinates: if $q = (r, \theta, z)$ then

$$[\,x,\ y,\ z\,]^T = g(q) = (r\cos\theta, r\sin\theta, z)$$

Examples of a changes of coordinates

- Common transformations of $\mathbb{R}^3$ that are local diffeomorphisms
  - ⋆ Cylindrical coordinates: if $q = (r, \theta, z)$ then

$$[\, x,\ y,\ z\,]^T = g(q) = (r\cos\theta, r\sin\theta, z)$$

  - ⋆ Spherical coordinates: if $q = (r, \phi, \theta)$ then

$$[\, x,\ y,\ z\,]^T = g(q) = (r\cos\phi\sin\theta, r\sin\phi\sin\theta, r\cos\phi)$$

Examples of a changes of coordinates

- Common transformations of $\mathbb{R}^3$ that are local diffeomorphisms
  - ⋆ Cylindrical coordinates: if $q = (r, \theta, z)$ then

$$[\ x,\ y,\ z\ ]^T = g(q) = (r\cos\theta, r\sin\theta, z)$$

  - ⋆ Spherical coordinates: if $q = (r, \phi, \theta)$ then

$$[\ x,\ y,\ z\ ]^T = g(q) = (r\cos\phi\sin\theta, r\sin\phi\sin\theta, r\cos\phi)$$

- Linear change of coordinates
  - ⋆ Let $g : \mathbb{R}^n \longrightarrow \mathbb{R}^n$

Examples of a changes of coordinates

- Common transformations of $\mathbb{R}^3$ that are local diffeomorphisms
  - ⋆ Cylindrical coordinates: if $q = (r, \theta, z)$ then

$$[\ x,\ y,\ z\ ]^T = g(q) = (r\cos\theta, r\sin\theta, z)$$

  - ⋆ Spherical coordinates: if $q = (r, \phi, \theta)$ then

$$[\ x,\ y,\ z\ ]^T = g(q) = (r\cos\phi\sin\theta, r\sin\phi\sin\theta, r\cos\phi)$$

- Linear change of coordinates
  - ⋆ Let $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$
  - ⋆ $g(q) = Aq$, where A is a square, nonsingular matrix

Examples of a changes of coordinates

- Common transformations of $\mathbb{R}^3$ that are local diffeomorphisms
  - ⋆ Cylindrical coordinates: if $q = (r, \theta, z)$ then

$$[\,x,\ y,\ z\,]^T = g(q) = (r\cos\theta, r\sin\theta, z)$$

  - ⋆ Spherical coordinates: if $q = (r, \phi, \theta)$ then

$$[\,x,\ y,\ z\,]^T = g(q) = (r\cos\phi\sin\theta, r\sin\phi\sin\theta, r\cos\phi)$$

- Linear change of coordinates
  - ⋆ Let $g : \mathbb{R}^n \longrightarrow \mathbb{R}^n$
  - ⋆ $g(q) = Aq$, where A is a square, nonsingular matrix
  - ⋆ $G = \partial g/\partial q = A$ is nonsingular

Examples of a changes of coordinates

- Common transformations of $\mathbb{R}^3$ that are local diffeomorphisms
  - ⋆ Cylindrical coordinates: if $q = (r, \theta, z)$ then

$$[\ x,\ y,\ z\ ]^T = g(q) = (r\cos\theta, r\sin\theta, z)$$

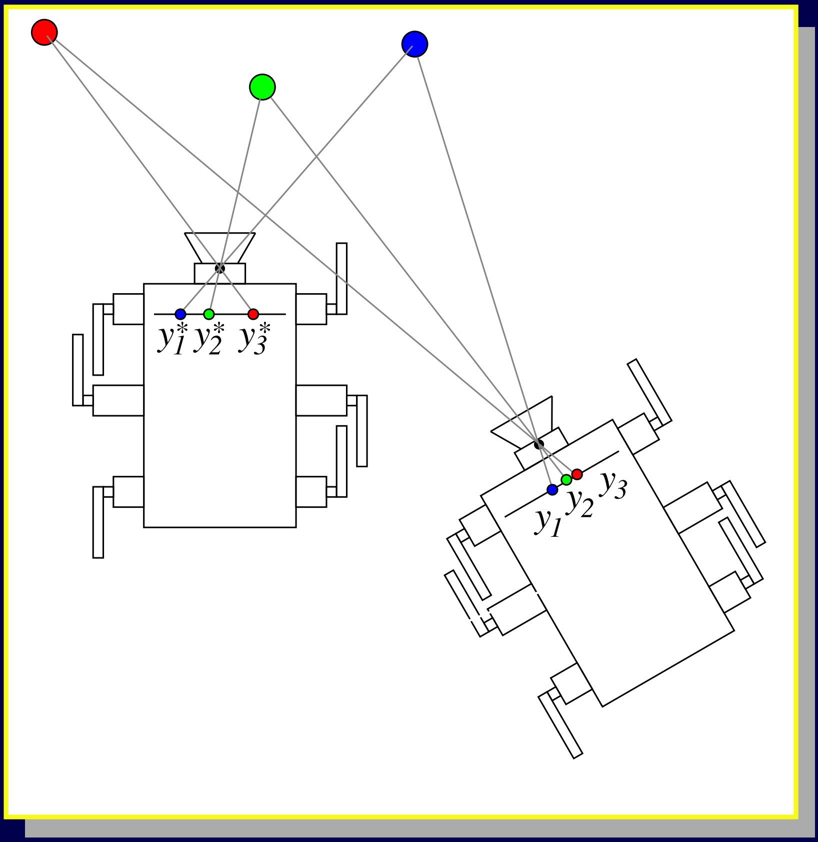
  - ⋆ Spherical coordinates: if $q = (r, \phi, \theta)$ then

$$[\ x,\ y,\ z\ ]^T = g(q) = (r\cos\phi\sin\theta, r\sin\phi\sin\theta, r\cos\phi)$$

- Linear change of coordinates
  - ⋆ Let $g : \mathbb{R}^n \longrightarrow \mathbb{R}^n$
  - ⋆ $g(q) = Aq$, where A is a square, nonsingular matrix
  - ⋆ $G = \partial g/\partial q = A$ is nonsingular
  - ⋆ g is just a similarity transform!

Back to Visual Servoing

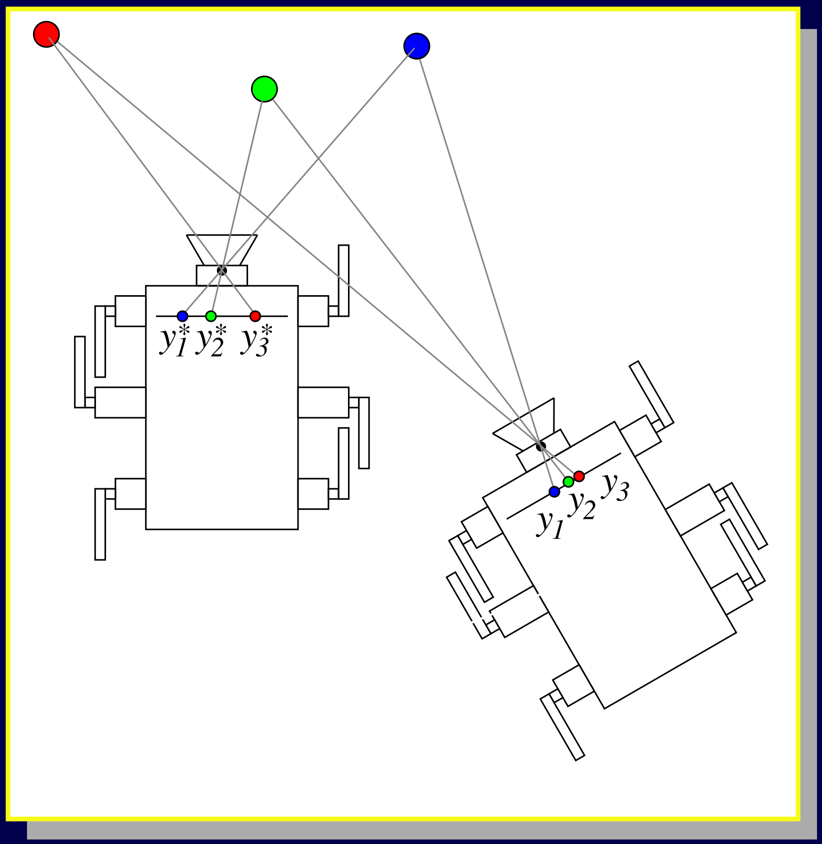
Given: a robot with configuration $q \in \mathcal{Q}$, with projection to an image plane $y = c(q)$.



Back to Visual Servoing

Given: a robot with configuration $q \in \mathcal{Q}$, with projection to an image plane $y = c(q)$.

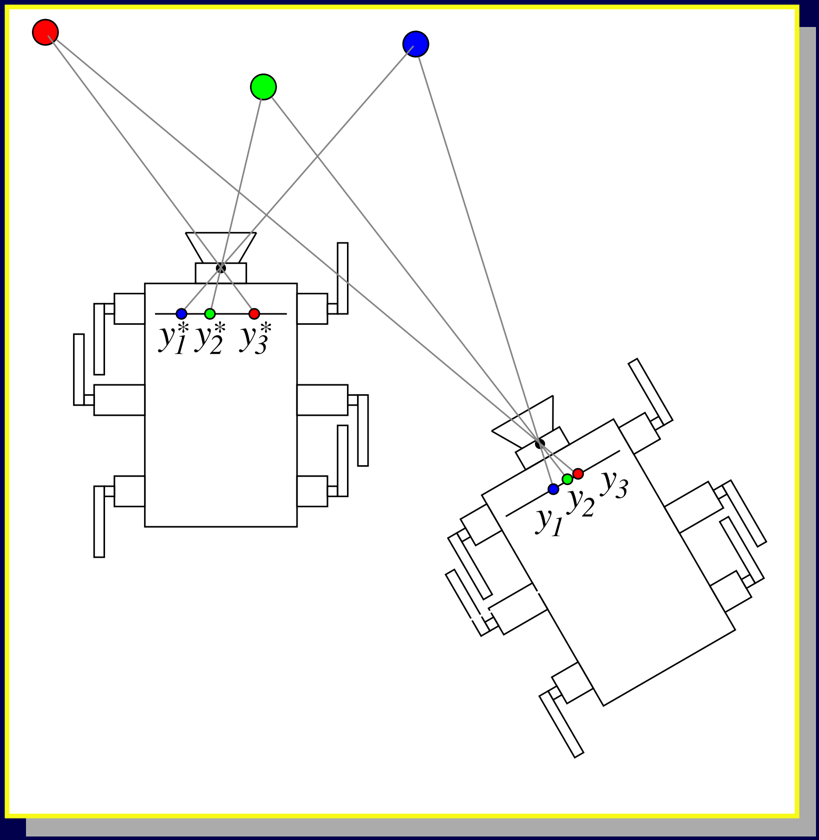
Objective: to move robot such that $y \rightarrow y^*$ as $t \rightarrow \infty$.



Back to Visual Servoing

Given: a robot with configuration $q \in \mathcal{Q}$, with projection to an image plane $y = c(q)$.

Objective: to move robot such that $y \to y^*$ as $t \to \infty$. The goal image, $y^* = c(q^*)$ was recorded in the goal state q^*.

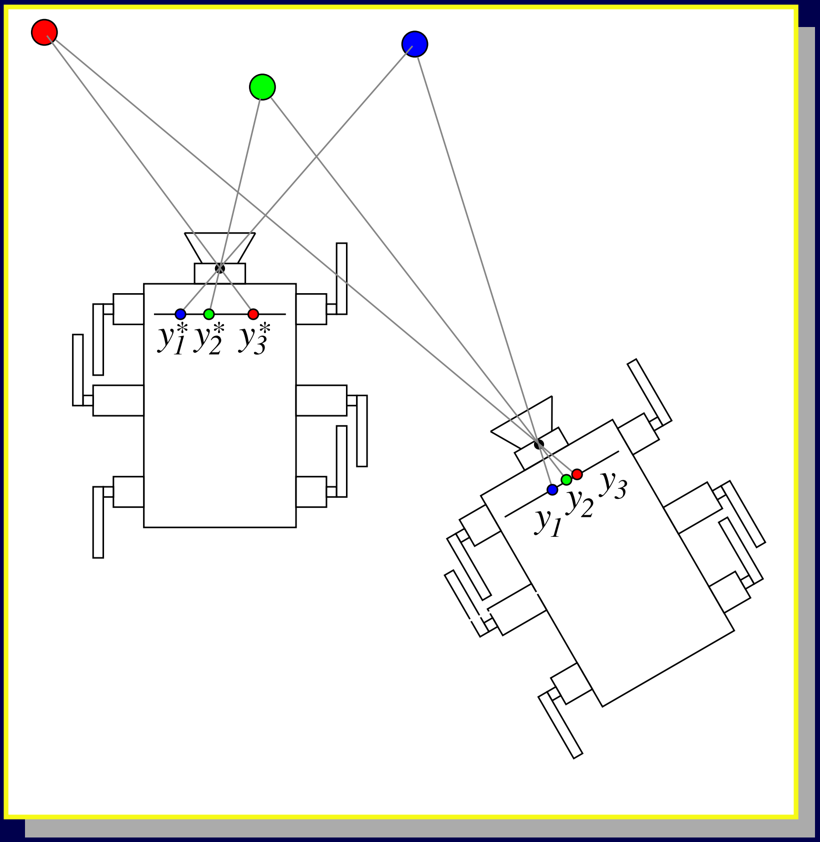


Back to Visual Servoing

Given: a robot with configuration $q \in \mathcal{Q}$, with projection to an image plane $y = c(q)$.

Objective: to move robot such that $y \to y^*$ as $t \to \infty$. The goal image, $y^* = c(q^*)$ was recorded in the goal state q^*.

Assumption: c is *into* (one-to-one) in some neighborhood of q^*.

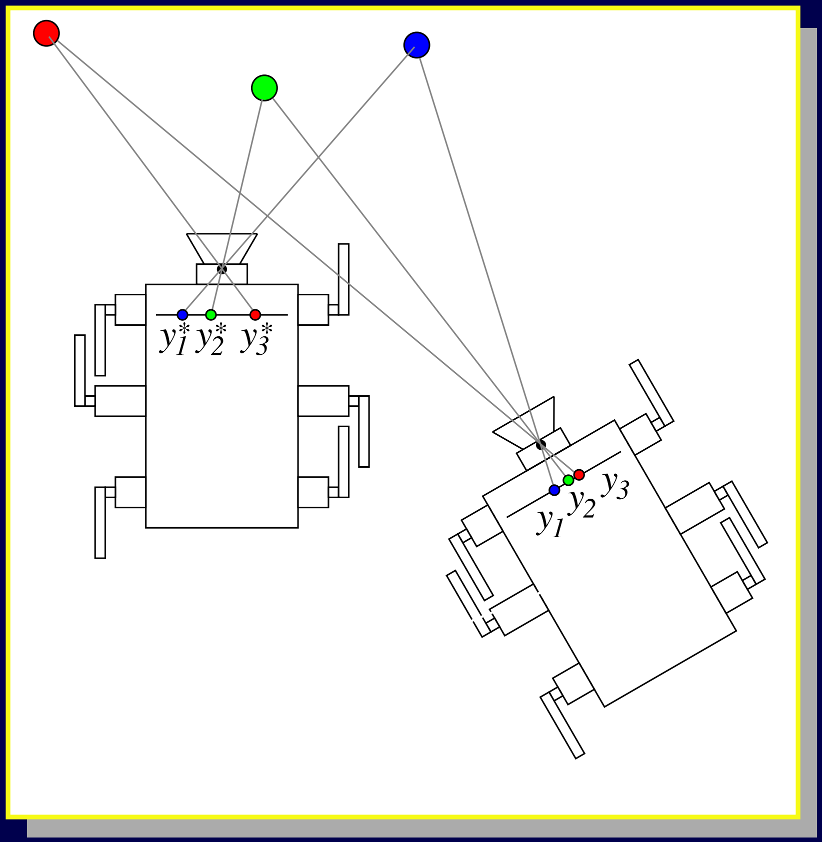


Back to Visual Servoing

Given: a robot with configuration $q \in \mathcal{Q}$, with projection to an image plane $y = c(q)$.

Objective: to move robot such that $y \rightarrow y^*$ as $t \rightarrow \infty$. The goal image, $y^* = c(q^*)$ was recorded in the goal state q^*.

Assumption: c is *into* (one-to-one) in some neighborhood of q^*.This implies (locally) that if $y \rightarrow y^*$, then $q \rightarrow q^*$.

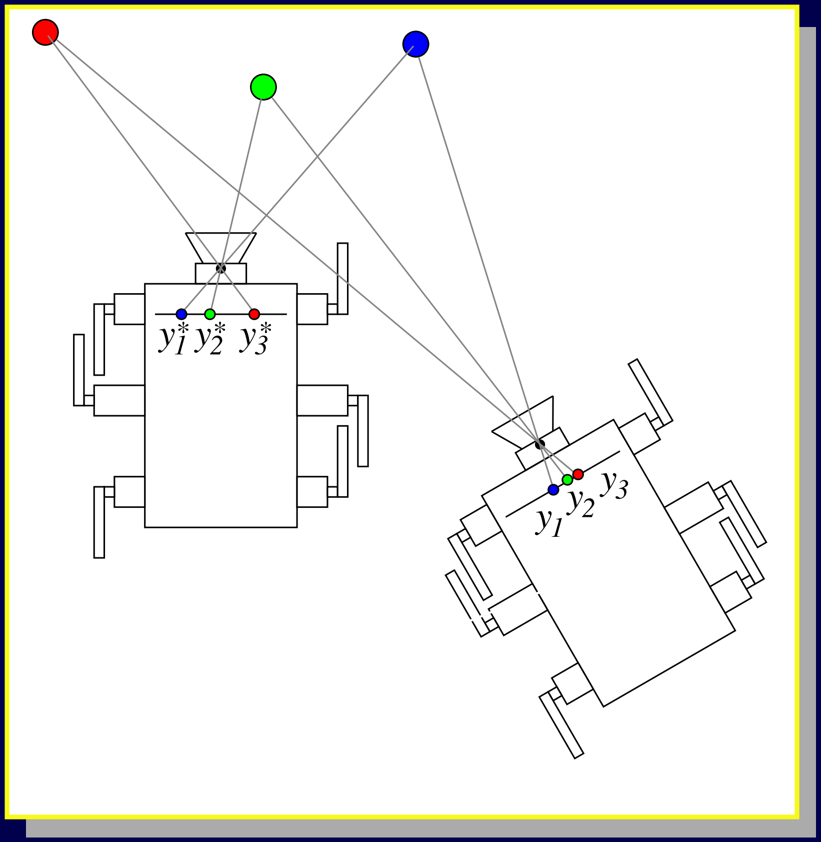


Back to Visual Servoing

Given: a robot with configuration $q \in \mathcal{Q}$, with projection to an image plane $y = c(q)$.

Objective: to move robot such that $y \to y^*$ as $t \to \infty$. The goal image, $y^* = c(q^*)$ was recorded in the goal state q^*.

Assumption: c is *into* (one-to-one) in some neighborhood of q^*.This implies (locally) that if $y \to y^*$, then $q \to q^*$.

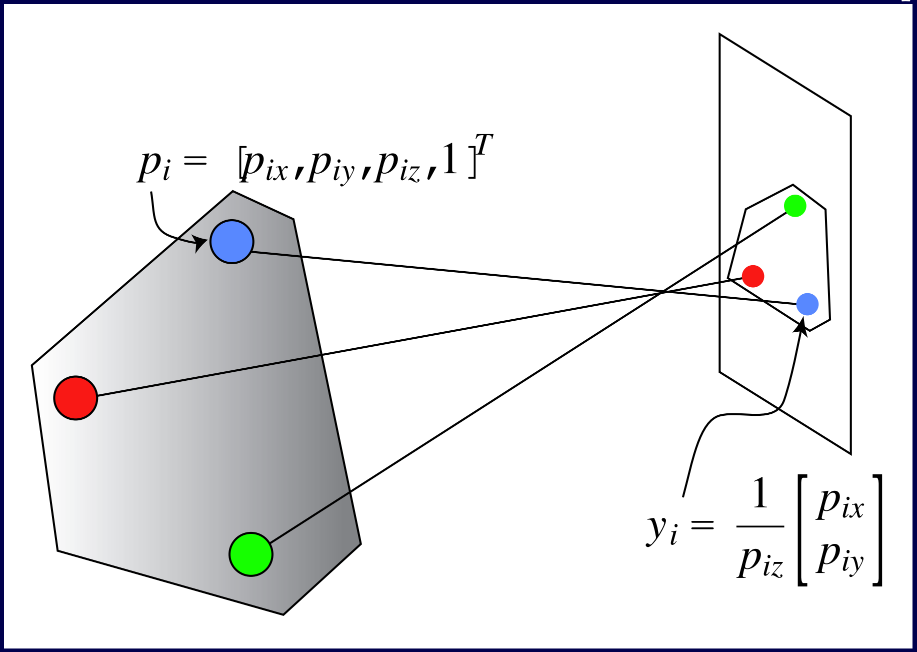


When is c *just a change of coordinates?*

The usual definition of c

- c maps rigid transformations to image feature locations. It is composed from the action of $\mathrm{SE}(3)$ on $\mathbb{R}^3$ and perspective projection, as follows.

- The camera is a perspective projection camera, and thus if $p = [\ p_x,\ p_y,\ p_z,\ 1\ ]^T$ is a point in space then $y = \pi(p) := \frac{1}{p_z}[\ p_x,\ p_y\ ]^T$



- Rigid body motion given by

$$H = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

- Rigid body motion given by

$$H = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

- Let $B = [b_1, b_2, \ldots, b_n]$ be n feature points relative to the body frame where $b_i = [\ b_{xi},\ b_{yi},\ b_{zi},\ 1\ ]^T$.

- Rigid body motion given by

$$H = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

- Let $B = [b_1, b_2, \ldots, b_n]$ be n feature points relative to the body frame where $b_i = [\ b_{xi},\ b_{yi},\ b_{zi},\ 1\ ]^T$.

- In the camera frame, $P = HB$, where $P = [p_1, p_2, \ldots, p_n]$ and $p_i = [\ p_{xi},\ p_{yi},\ p_{zi},\ 1\ ]^T$.

- Rigid body motion given by

$$H = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

- Let $B = [b_1, b_2, \ldots, b_n]$ be n feature points relative to the body frame where $b_i = [\ b_{xi},\ b_{yi},\ b_{zi},\ 1\ ]^T$.

- In the camera frame, $P = HB$, where $P = [p_1, p_2, \ldots, p_n]$ and $p_i = [\ p_{xi},\ p_{yi},\ p_{zi},\ 1\ ]^T$.

- The image of all the points is $Y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}$ where $y_i = \pi(p_i)$.

- Rigid body motion given by

$$H = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

- Let $B = [b_1, b_2, \ldots, b_n]$ be n feature points relative to the body frame where $b_i = [\ b_{xi},\ b_{yi},\ b_{zi},\ 1\ ]^T$.

- In the camera frame, $P = HB$, where $P = [p_1, p_2, \ldots, p_n]$ and $p_i = [\ p_{xi},\ p_{yi},\ p_{zi},\ 1\ ]^T$.

- The image of all the points is $Y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}$ where $y_i = \pi(p_i)$.

- The camera map, $c : \mathrm{SE}(3) \to \mathbb{R}^{2n}$ is the given by $c : H \mapsto Y$.

- Rigid body motion given by

$$H = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

- Let $B = [b_1, b_2, \ldots, b_n]$ be n feature points relative to the body frame where $b_i = [\ b_{xi},\ b_{yi},\ b_{zi},\ 1\ ]^T$.

- In the camera frame, $P = HB$, where $P = [p_1, p_2, \ldots, p_n]$ and $p_i = [\ p_{xi},\ p_{yi},\ p_{zi},\ 1\ ]^T$.

- The image of all the points is $Y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}$ where $y_i = \pi(p_i)$.

- The camera map, $c : \mathrm{SE}(3) \to \mathbb{R}^{2n}$ is the given by $c : H \mapsto Y$.

- Note that, in general $\dim \mathrm{SE}(3) \neq 2n$ except when $n = 3$.

- Rigid body motion given by

$$H = \begin{bmatrix} R & T \\ 0^T & 1 \end{bmatrix} \in \mathrm{SE}(3)$$

- Let $B = [b_1, b_2, \ldots, b_n]$ be n feature points relative to the body frame where $b_i = [\ b_{xi},\ b_{yi},\ b_{zi},\ 1\ ]^T$.

- In the camera frame, $P = HB$, where $P = [p_1, p_2, \ldots, p_n]$ and $p_i = [\ p_{xi},\ p_{yi},\ p_{zi},\ 1\ ]^T$.

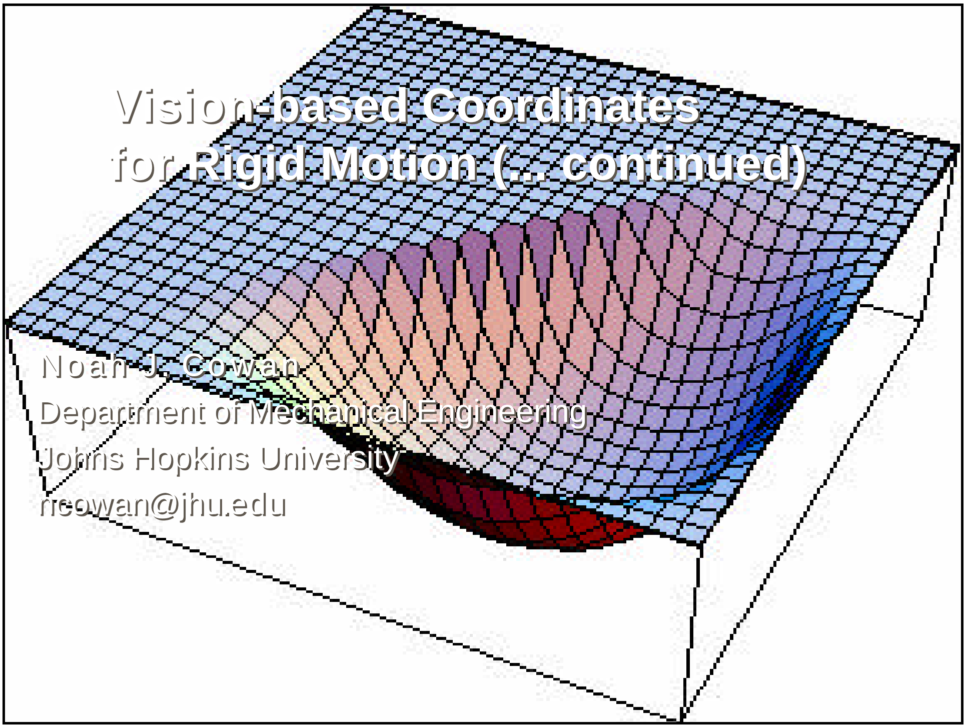
- The image of all the points is $Y = \begin{bmatrix} y_1 & \cdots & y_n \end{bmatrix}$ where $y_i = \pi(p_i)$.

- The camera map, $c : \mathrm{SE}(3) \to \mathbb{R}^{2n}$ is the given by $c : H \mapsto Y$.

- Note that, in general $\dim \mathrm{SE}(3) \neq 2n$ except when $n = 3$. In this case c is not into!

to be continued...

Click here to continue: Presentation continued in Power Point



Vision-based Coordinates for Rigid Motion (... continued)

Noah J. Cowan
Department of Mechanical Engineering
Johns Hopkins University
ncowan@jhu.edu

[Cowan et. al., ICRA '99; TRA '02]

Image-based coordinates: 2D world

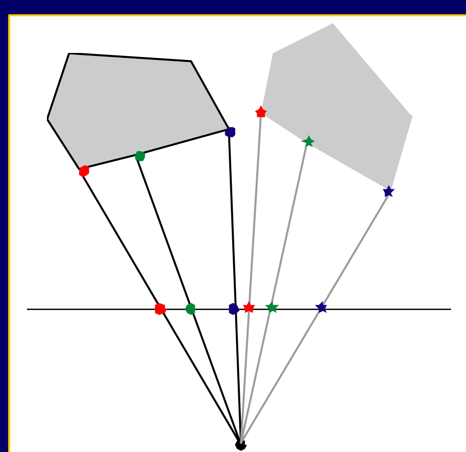
Planar body, $H \in SE(2)$

Collinear features

Pinhole camera (calibrated perspective projection)

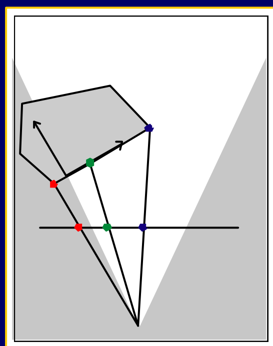
Output $y = c(H) = [y_1 \quad y_2 \quad y_3]^T$

Goal $y^* = c(H^*)$

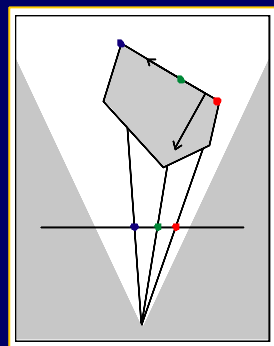


Planar Servoing: visibility

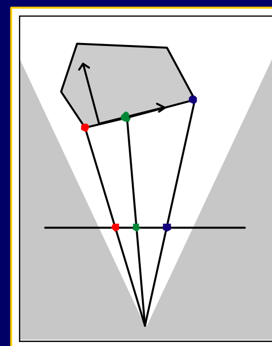
Facing camera, out of FOV



In the workspace, but self-occluded



All features visible, *V* Ì SE*(2)*



Planar Servoing: image coordinates

- The camera map $y=c(H)$ is a *diffeomorphism*

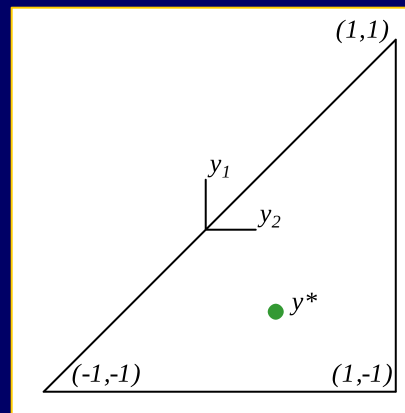
variables to the image plane!

The visibility obstacles appear as the boundary of this simple set!

Planar Servoing: key observations

- The image space is independent of parameters
- We can use image plane coordinates to design a global, dynamical occlusion-free controller

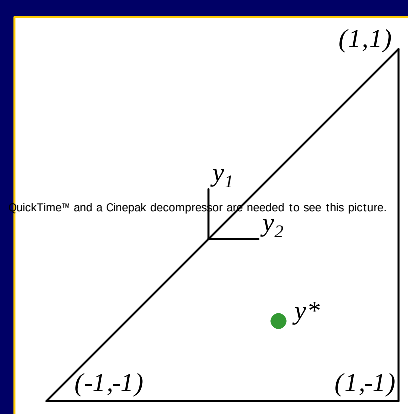
$Z = \{-1 < y_1 < y_2 < y_3 < 1\}$



Planar Servoing: solution

- We would like a controller to keep us within Z
- "Navigation function" for planar visual servoing is simple!

$Z = \{-1 < y_1 < y_2 < y_3 < 1\}$



Planar Servoing: solution

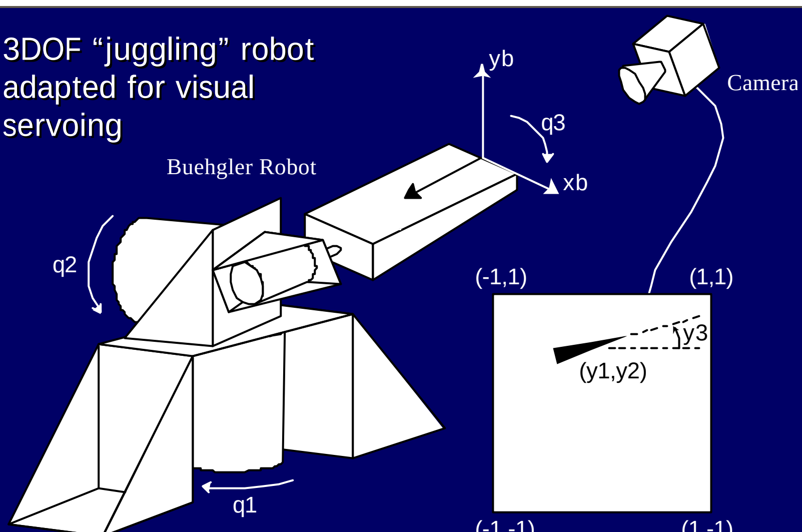
- We would like a controller to keep us within Z
- "Navigation function" for planar visual servoing is simple!
- Resulting controller is global, dynamic and occlusion-free

$$Z = \{-1 < y_1 < y_2 < y_3 < 1\}$$

[Cowan, CCA '01, TRA '02]

Buehgler arm servoing

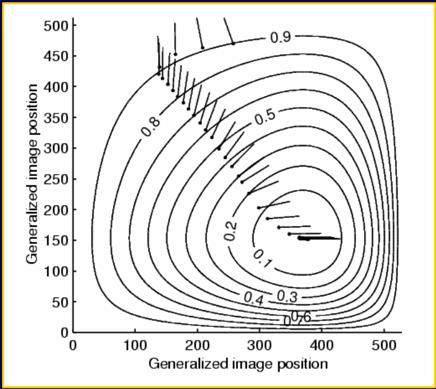
3DOF "juggling" robot adapted for visual servoing

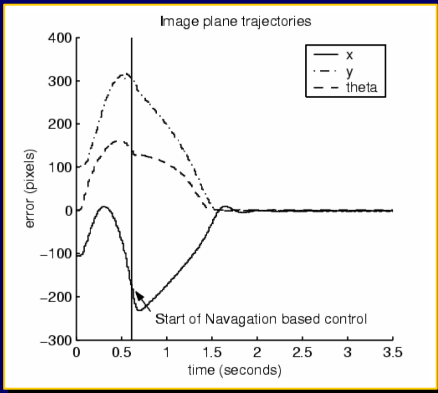


[Rizzi, Whitcomb, Koditschek '92]

Buehgler: typical trials

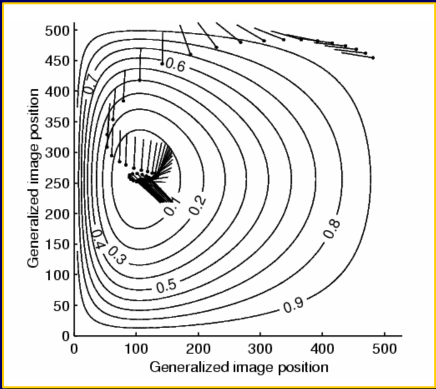
"Critically damped" gains

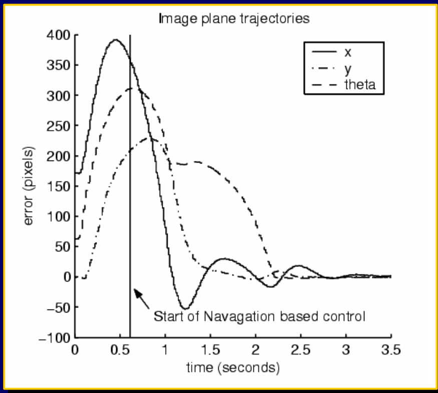




Buehgler: typical trials

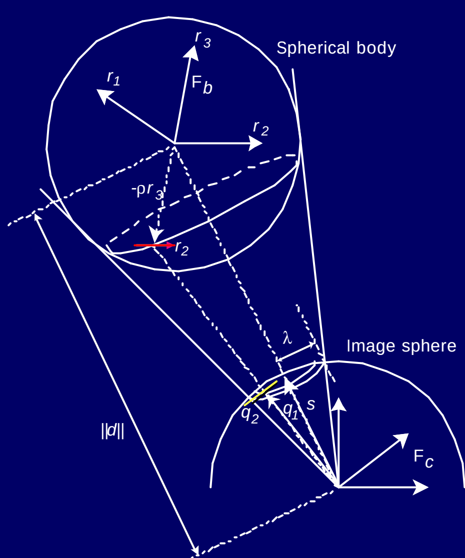
"Under-damped" gains





[Cowan and Chang, CPRA '02]

Generalization for a 6DOF body



$$H = \begin{bmatrix} R & \mathbf{d} \\ 0 & 1 \end{bmatrix} \in V \subset SE(3)$$

$$\boxed{y = c(H)}$$

$$y = (Q = [q_1, q_2, q_3], l, s) \in I$$

$$\subset SO(3) \times R \times S^2$$

Preliminary Simulations

QuickTime™ and a decompressor are needed to see this picture. QuickTime™ and a decompressor are needed to see this picture.

[Cowan et. al, CDC ‘00, TRA ‘02]

6DOF Visual Servoing

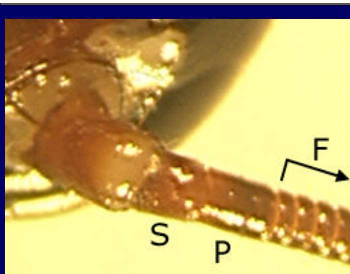
Visual Servoing on RHex

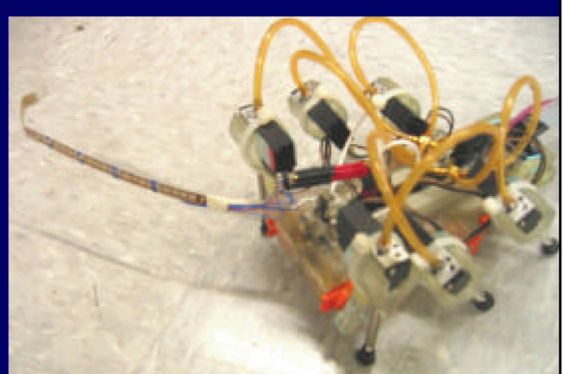
QuickTime™ and a YUV420 codec decompressor are needed to see this picture.

[Lopes, Koditschek et al.]

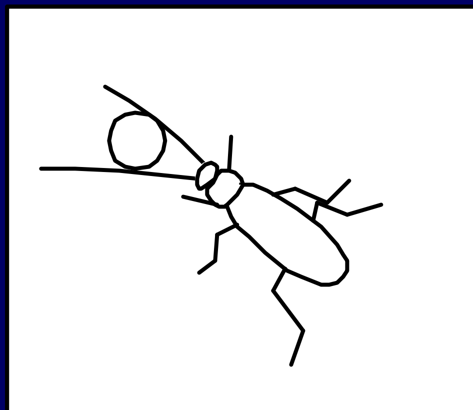
Another sensor-based coordinate system: A Bio-inspired Antenna

[Cowan, et. al., *in review*]





Tactile Orientation



Tactile Orientation: background

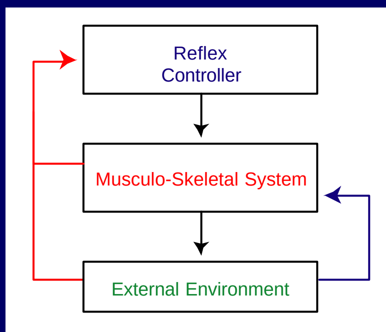
- Cockroach antenna mechano-receptors appear to encode *distance* from and *angle* to an object Okada and Toh [2001]
- A *feedback control hypothesis* has not been formulated for this behavior

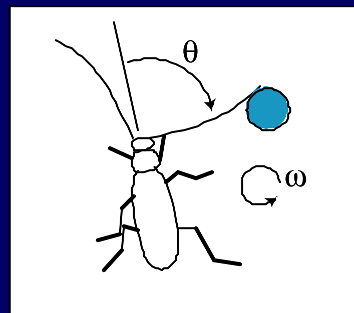
[Okada and Toh 2001]

Tactile Orientation: hypothesis

Hypothesis:

Behavior may be modeled as a 'first-order' reflex

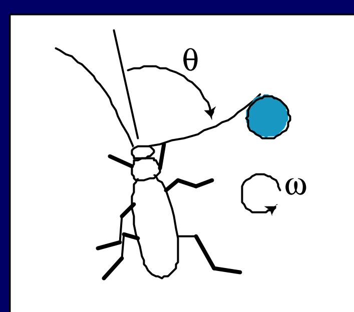




$$\dot{q} = w$$
$$w = -Kq$$

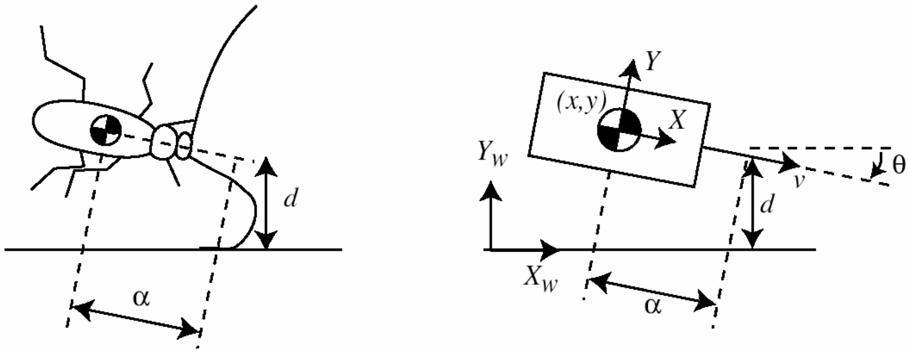
Tactile Orientation: testing

- Data in the literature [Okada and Toh 2001] is consistent with this hypothesis
- I am now designing a set of experiments to attempt to refute this hypothesis
- Anecdotal data seems consistent -- *stay tuned* for more rigorous experiments!



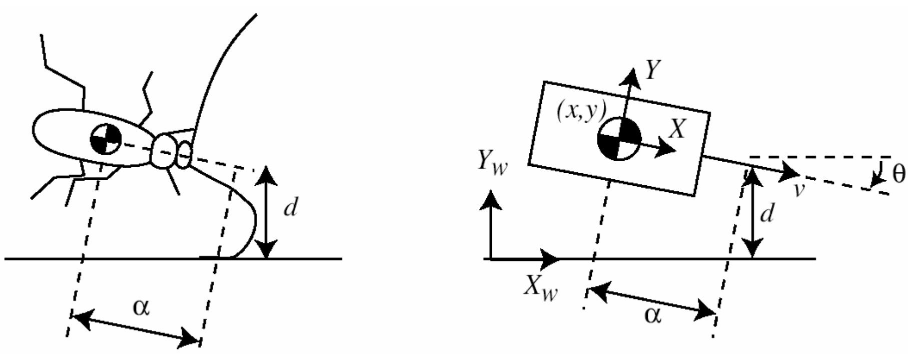
$$\dot{q} = w$$
$$w = -Kq$$

Dynamical wall following



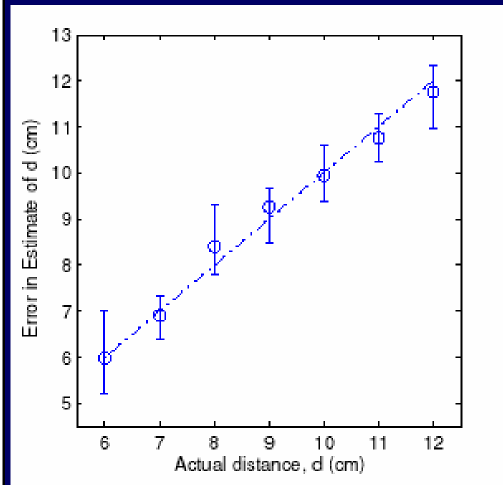
$$G(s) = \frac{D(s)}{U(s)} = \overbrace{\frac{\alpha s + v}{s}}^{\text{sensing}} \cdot \overbrace{\frac{1}{ms^2 + bs}}^{\text{mechanics}}$$

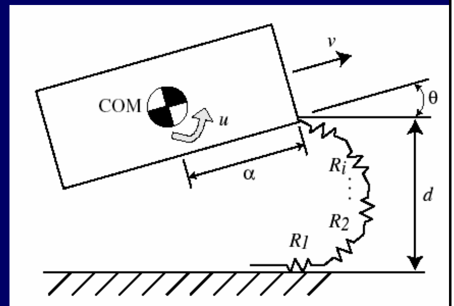
Dimensionless Dynamics

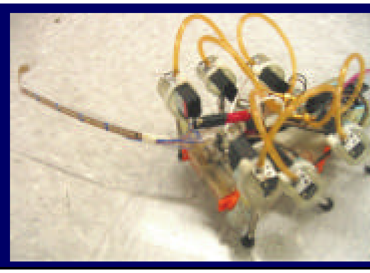


$$\widetilde{G}(w) = \frac{w+1}{w^2(\tau w + 1)} \qquad \tau = \frac{mv}{b\alpha}$$

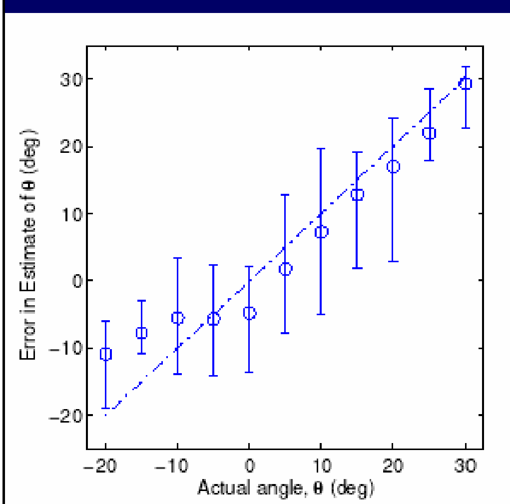
Design:
a novel “distance & angle” sensor

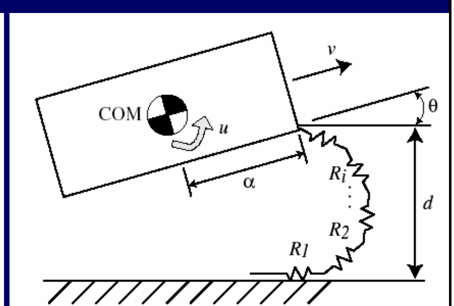


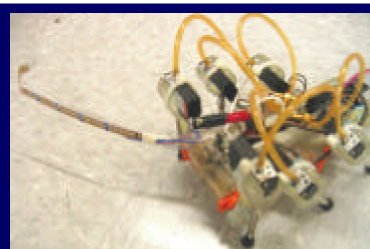




Design:
a novel “distance & angle” sensor







Robotic Integration: wall following

K_P

QuickTime™ and a MPEG-4 Video decompressor are needed to see this picture. QuickTime™ and a MPEG-4 Video decompressor are needed to see this picture.

Result:

Tactile flow

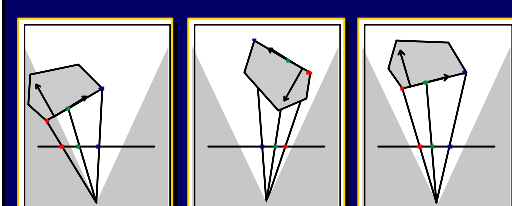
is critical

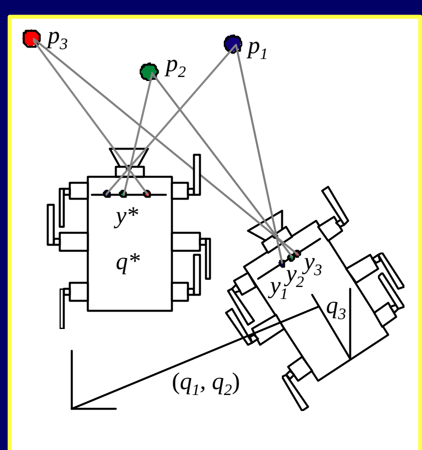
QuickTime™ and a MPEG-4 Video decomp

K_D

Summary

- Vision-based control
  - diffeomorphism from *visible set* to *image space*
  - *visibility* is naturally encoded in image, but not in task space
  - Navigation Functions ensure *global*, *dynamical* convergence while avoiding obstacles





Acknowledgements

- Collaborators
  - Robert Full and the PolyPEDAL Lab
  - Dan Koditschek's Laboratory
  - Omid Shakernia, Rene Vidal and Shankar Sastry
  - Emily Ma, Mark Cutkosky and the Biomimetics Lab
  - Dong Eui Chang
  - Domenico Prattichizzo and Jacopo Piazzi

- Funding Sources
  - ONR/MURI: N00014-98-10669
  - NSF/KDI Grant FD98-73086
  - NSF IRI-9510673
  - Rackham Graduate School (Michigan)