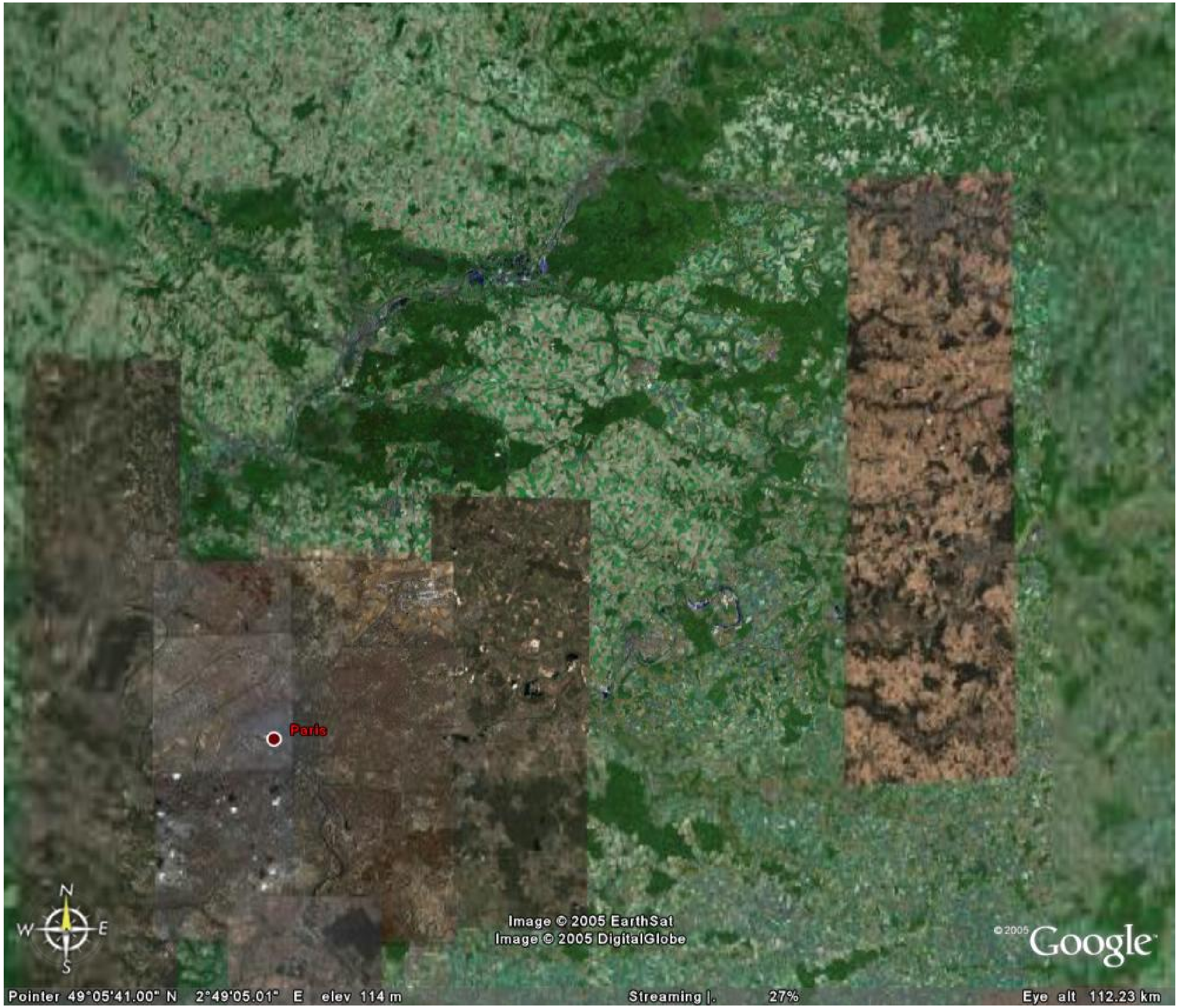
Dynamic Image Based Visual Servo Control:

Applications to Aerial Robotic Vehicles

Robert Mahony
Dep. Engineering,
Australian National University,
Canberra, ACT, 0200.


Paris
N
W
E
S
Image © 2005 EarthSat
Image © 2005 DigitalGlobe
© 2005 Google
Pointer 49°05'41.00" N 2°49'05.01" E elev 114 m
Streaming |. 27%
Eye alt 112.23 km


Compiegne
N
W
E
S
Image © 2005 EarthSat
Image © 2005 DigitalGlobe
© 2005
Google
Pointer 49°25'08.21" N 2°56'32.32" E elev 37 m
Streaming |||||||||| 100%
Eye alt 32.05 km


N
W
E
S
Image © 2005 DigitalGlobe
©2005 Google
Pointer 49°25'47.40" N 2°54'54.35" E elev 41 m
Streaming |||||||||| 100%
Eye alt 5.12 km


11 NOVEMBRE
1914
1918

- Visual servo control: Automatic positioning of a robotic system using data derived from a camera system.

- PBVS: Position based visual servo. The control problem is solved using classical state space representation of the system. Visual data is used as sensor input to a state-estimator that provides state-estimates for the feedback loop.

- IBVS: Image based visual servo. The robot kinematics are mapped into feature kinematics in image space. The control task is posed directly as stabilisation of observed features.

- Dyanmic IBVS. Image based control of a dynamical system

$$\dot{z} = v \qquad \text{Kinematics}$$
$$\dot{v} = g(z, v, u) \qquad \text{Dynamics}$$

Outline of talk

Part I: Some motivation and perspectives

Part II: A short sharp review of classical IBVS

Part III: Dynamic IBVS for a flying brick.

Part IV: Conclusions.

This talk gives only a very narrow path through the field of visual servo control.

There is a large literature in classical visual servo control and IBVS in particular.

There is not much literature on IBVS for dynamical systems.

Part I: A Typical Aerial Robotic System

Sensor systems:

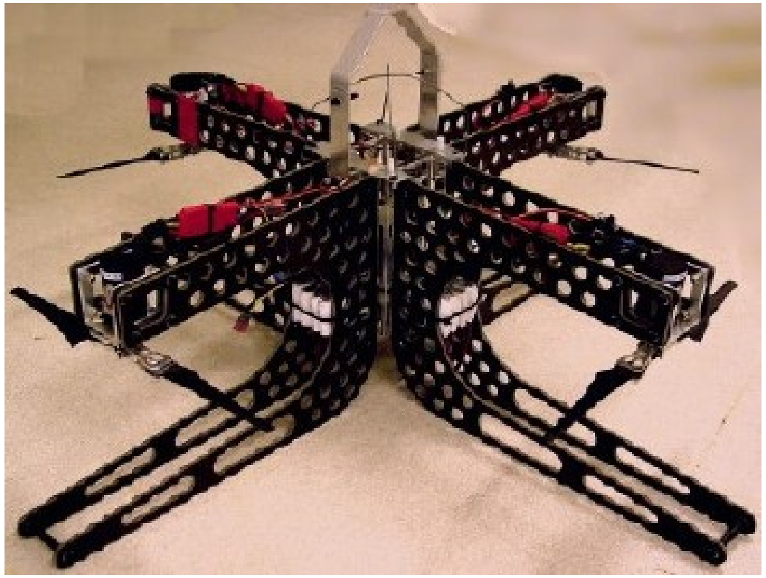
Inertial Measurement Unit (IMU)

- angular rates @ 70Hz
- accelerometers @ 70Hz
- magnetometers @ 70Hz

Fused to provide attitude @ 70Hz

Camera

- image features. @ 10-20Hz
- optic flow. @ 10-20Hz



Motivation against Classical Estimation and Control

The mapping from sensor measurement to state is highly non-linear. The dynamics of the vehicle are non-linear. Sensor noise is often highly correlated due to platform vibration.

&

The closed-loop bandwidth of the vehicle (required for disturbance rejection) demands the full bandwidth of sensor measurements. There is no spare bandwidth for a smoothing filter.

$\Downarrow$

Designing a filter to fuse all sensor readings into a full state-estimate is difficult. Filters designed on extensions of linear methods tend to be highly sensitive and unreliable.

Motivation for sensor based control

- No dynamic lag in measurement process.
- In many applications, the sensitivity of the sensor to error and the required precision of the closed-loop match.
- For task-based missions defined in the sensor domain there is minimal requirement for calibration of sensor systems.

Sensor based control is highly reliable and has extremely good robustness properties

However, representing the system dynamics in the sensor domain can lead to highly non-linear control design problems.

Part II: Classical Visual Servo Control

1. An ensemble of visual features $s \in \mathbb{R}^n$ are extracted from each image.
2. The visual error is

$$\tilde{s} = s - s^*.$$

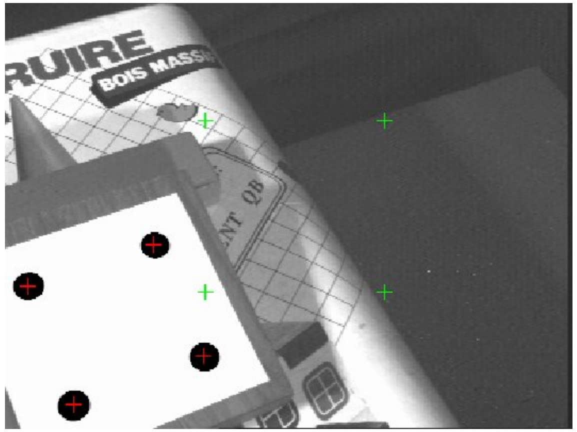
   where s^* denotes the desired image configuration.
3. The visual kinematics are

$$\frac{d}{dt}\tilde{s} = \frac{\partial s}{\partial z}\frac{dz}{dt} = L(s, z)\dot{z}$$

   Where the pose of the camera is $z = (R, \xi)$.

$$L(s, z) = \frac{\partial s}{\partial z}$$

Image Jacobian



▶Observed features $\{s_1, \ldots, s_4\}$ in red
▶Goal features $\{s_1^*, \ldots, s_4^*\}$ in green.

Classical IBVS control design

Linearising control law for a fully actuated kinematic system:

$$v = \dot{z} := -\left(L(s,z)^T L(s,z)\right)^{-1} L(s,z)^T \tilde{s}$$

Leads to the linearised image feature dynamics

$$\frac{d}{dt}\tilde{s} = -L(s,z)\left(L(s,z)^T L(s,z)\right)^{-1} L(s,z)^T \tilde{s}$$

Close to the goal $z = z^* + \xi$, $s = s^* + \eta$ and $\eta \approx L(s^*, z^*)\xi$.

$$L(s^*,z^*)\dot{\xi} \approx -L(s^*,z^*)\left(L(s^*,z^*)^T L(s^*,z^*)\right)^{-1} L(s^*,z^*)^T L(s^*,z^*)\xi$$

$$\Rightarrow \qquad \boxed{\dot{\xi} \approx -\xi}$$

Naive Dynamic IBVS

Consider a simple dynamical system

$$\dot{z} = v \qquad \text{Kinematics}$$
$$\dot{v} = g(z, v, u) \qquad \text{Dynamics}$$

The velocity v is not directly measured.

Try using the visual error flow $\dot{\tilde{s}}$ as a sensor based velocity measurement

$$\dot{\tilde{s}} := L(s,z)v, \qquad v = \left(L(s,z)^T L(s,z)\right)^{-1} L(s,z)^T \tilde{s}$$

Sensor domain system dynamics

$$\frac{d}{dt}\tilde{s} = \dot{\tilde{s}} \qquad \text{Kinematics}$$
$$\frac{d}{dt}\dot{\tilde{s}} = \frac{\partial L(s,z)}{\partial s}\dot{\tilde{s}} + \frac{\partial L(s,z)}{\partial z}\left(L(s,z)^T L(s,z)\right)^{-1} L(s,z)^T \tilde{s}$$
$$+ L(s,z) g\left(z, \left(L(s,z)^T L(s,z)\right)^{-1} L(s,z)^T \tilde{s}, u\right) \qquad \text{Dynamics}$$

Problem with Naive dynamic IBVS

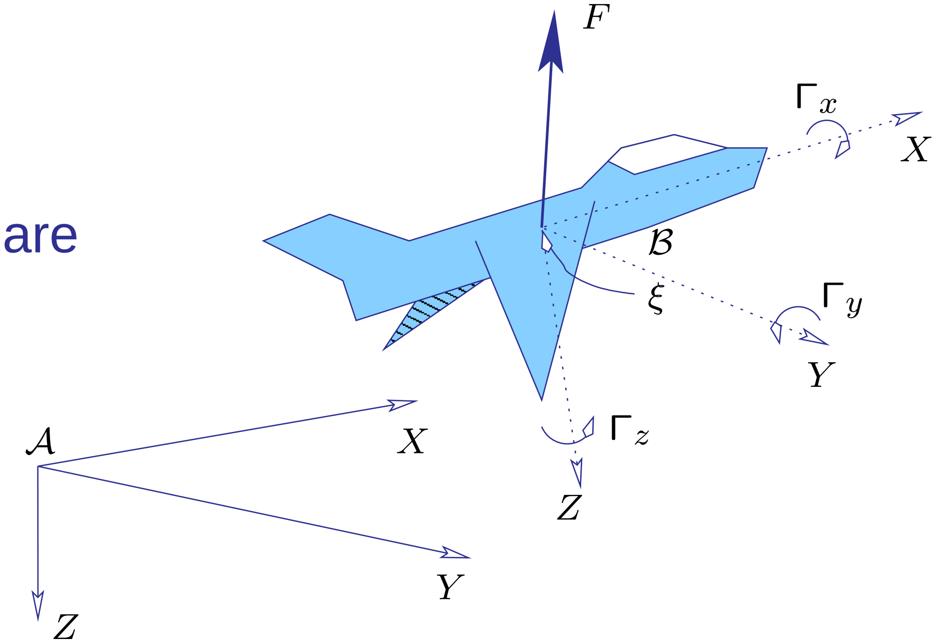
$$\begin{aligned}\frac{d}{dt}\dot{\tilde{s}} =& \frac{\partial L(s,z)}{\partial s}\dot{\tilde{s}} + \frac{\partial L(s,z)}{\partial z}\left(L(s,z)^T L(s,z)\right)^{-1} L(s,z)^T \tilde{s} \\ &+ L(s,z) g(z, \left(L(s,z)^T L(s,z)\right)^{-1} L(s,z)^T \tilde{s}, u)\end{aligned}$$

- Sensor domain dynamics depend on the unmeasured state variable z. In practice, sensor variables depend on unknown depth.
- The system is simply too non-linear for common mortals to understand.
- Overparameterisation of data leads to undesired local equilibrium in closed loop system.

Part III: A specific example in Aerial Robotics: dynamic IBVS for a flying brick

$$\dot{\xi} = RV$$
$$m\dot{V} = -m\Omega_\times V + F$$
$$\dot{R} = R\Omega_\times,$$
$$\mathbf{I}\dot{\Omega} = -\Omega_\times \mathbf{I}\Omega + \Gamma.$$

The exogenous force and torque are denoted F and Γ.

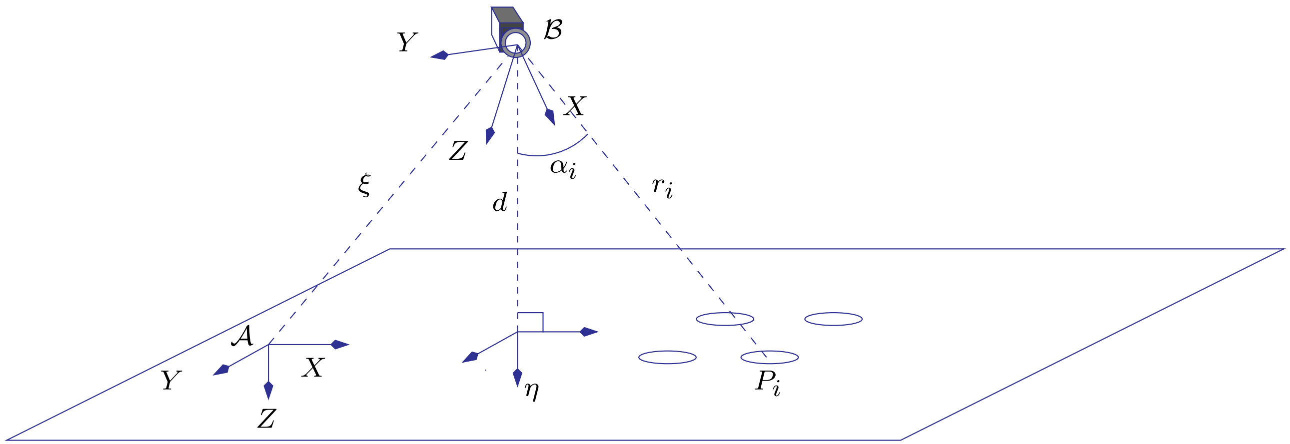


Visual features

▶Target consists of an ensemble of point targets on a flat target plane.

▶Use a spherical projection of visual data

$$p_i = \frac{P_i}{|P_i|} = \frac{1}{r_i} P_i.$$



▶From geometry one has

$$r_i = \frac{d}{\cos(\alpha_i)}$$

d is height of camera from target plane, α_i is angle of observed target from target plane normal.

Feature Kinematics

Image point kinematics

$$\dot{p}_i = -\Omega_\times p_i + \left(I_3 - p_i p_i^T\right)\cos(\alpha_i)\frac{V}{d}.$$

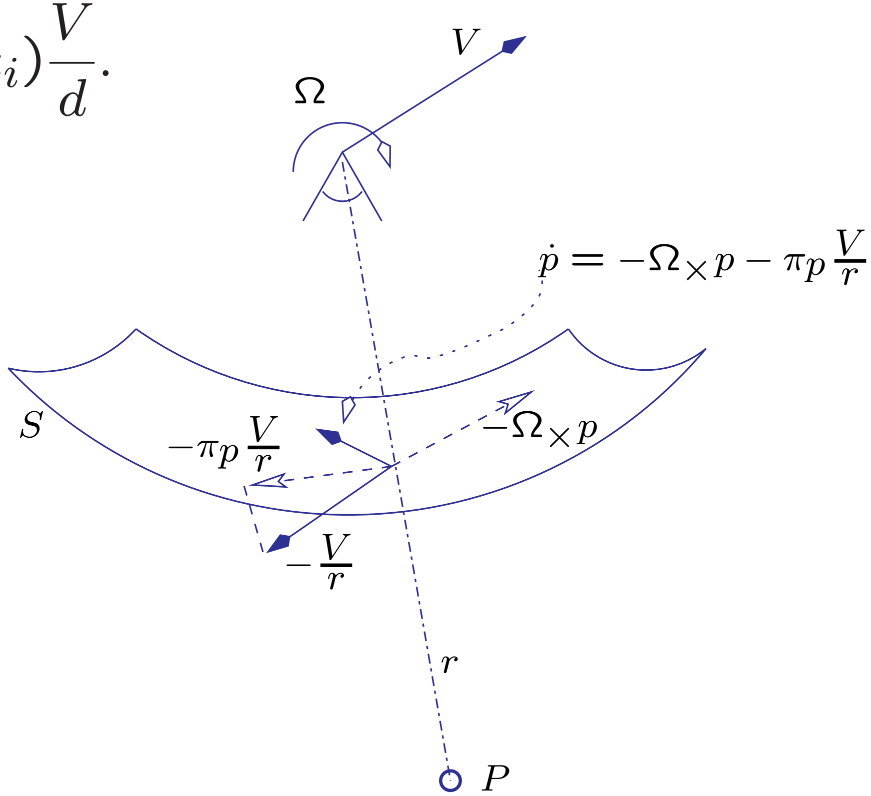


Image feature

$$q := \sum_{i=1}^{n} p_i \in \mathbb{R}^3$$

Feature kinematics

$$\dot{q} = -\Omega_\times q - Q\frac{V}{d(t)},$$

where

$$Q = \sum_{i=1}^{i=n} \cos(\alpha_i)\pi_{p_i}.$$

Note that $Q > 0$ is positive definite and $\Omega_\times$ is skew-symmetric.

Image Error Kinematics

Goal vector q^* (inertial data)

$$\dot{q}^* = -\Omega \times q^*.$$

Image error

$$\delta := q - q^*.$$

The image error kinematics are

$$\dot{\delta} = -\Omega_\times \delta - QW$$

Where W denotes the feature velocity

$$\boxed{W = \frac{V}{d(t)}}$$

feature velocity

Measuring Optic Flow

The visual velocity measure that is used is the *feature velocity*

$$W(t) = \frac{V(t)}{d(t)}$$

The feature velocity can be obtained by differentiating the image feature

$$W = -Q^{-1}(\dot{q} + \Omega \times q) = -Q^{-1}\left(\sum_{i=1}^{n} \dot{p}_i + \Omega \times q\right)$$

If the target plane is highly textured then the integrated optic flow over the whole target plane is

$$W = \kappa \int_{S^2_{\text{lower}}} (\dot{p} + \Omega_\times p)\, dp$$

Average Optic Flow

Feature dynamics

Differentiate W to compute the feature dynamics

$$\begin{aligned}\dot{W} &= \frac{\dot{V}}{d} - \frac{V\dot{d}}{d^2} \\ &= -\Omega \times W + W\langle W, \eta\rangle + \frac{1}{d(t)}\frac{F}{m}\end{aligned}$$

Where

$$\dot{d} = -\langle \eta, V\rangle = -d\langle \eta, W\rangle$$

Image based representation of full dynamics

$$\dot{\delta} = -\Omega_\times \delta - QW \qquad \text{Kinematics}$$

$$\dot{W} = -\Omega \times W + W\langle W, \eta\rangle + \frac{F}{md} \qquad \text{Dynamics}$$

$$\dot{d} = -d\langle \eta, W\rangle \qquad \text{Depth Kinematics}$$

Adaptive control for high frequency gain with integrable dynamics

Let $\widehat{d}$ be an internal controller state with dynamics

$$\dot{\widehat{d}} = -\widehat{d}\langle \eta, W \rangle$$

these dynamics act like *virtual* depth dynamics

Then

$$\frac{d}{dt}\left(\frac{\widehat{d}}{d}\right) = \frac{\dot{\widehat{d}}}{d} - \frac{\widehat{d}\dot{d}}{d^2} = -\frac{\widehat{d}\langle \eta, W \rangle}{d} + \frac{\widehat{d}d\langle \eta, W \rangle}{d^2} = 0$$

Integrating this relationship over time one obtains

$$\frac{\widehat{d}(t)}{d(t)} = a = \frac{\widehat{d}(0)}{d(0)}$$

where a is an unknown constant depending on the error between true depth $d(0)$ and the initial estimate $\widehat{d}(0)$.

Image kinematics for the flying brick

Image based representation of full dynamics

$$\dot{\delta} = -\Omega_{\times}\delta - QW \qquad \text{Kinematics}$$

$$\dot{W} = -\Omega \times W + W\langle W, \eta\rangle + a\frac{F}{m\hat{d}(t)} \qquad \text{Dynamics}$$

$$\dot{\hat{d}} = -\hat{d}\langle \eta, W\rangle \qquad \text{Virtual Depth Kinematics}$$

- The gain a is an unknown high frequency gain.

  This will be dealt with using an adaptive parameter estimation.

- The virtual depth dynamics are an additional dynamic state in the control design. The role is to invert the time-varying response of true depth $d(t)$ so that the adaptation can be undertaken in the space of initial conditions.

Adaptive control design

Let $\rho = 1/a$ and let $\hat{\rho}$ be an estimate for $\rho = 1/a$.

Let $k, c, \lambda > 0$ be positive gains.

Let $\epsilon = W - k\delta$.

The control algorithm is computed according to the following dynamics

$$
\begin{aligned}
\dot{\hat{d}}(t) &= -\hat{d}\langle \eta, W \rangle, & \hat{d}(0) &= \hat{d}_0, \\
\dot{\hat{\rho}} &:= -\lambda \langle U, \epsilon \rangle, & \hat{\rho}(0) &= 1, \\
U &:= Q\delta - W\langle W, \eta \rangle - kQW - c\epsilon, \\
F(t) &:= \hat{\rho} m \hat{d}(t) U,
\end{aligned}
$$

Convergence results

Define the Lyapunov function

$$\mathcal{L} = \frac{1}{2}|\delta|^2 + \frac{1}{2}|\epsilon|^2 + \frac{a}{2\lambda}|\tilde{\rho}|^2$$

Then, the closed-loop trajectory exists for all time and satisfies

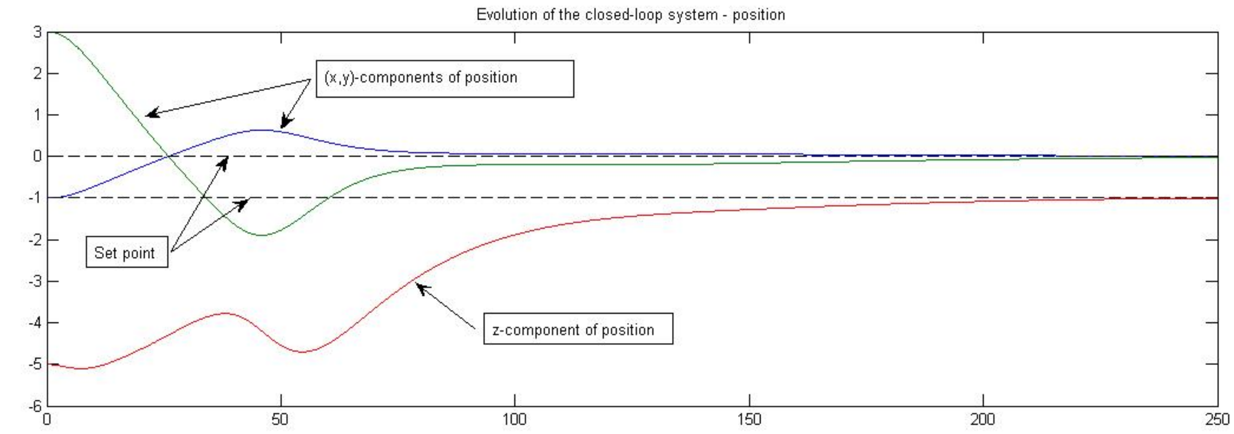
$$d(t) > 0$$

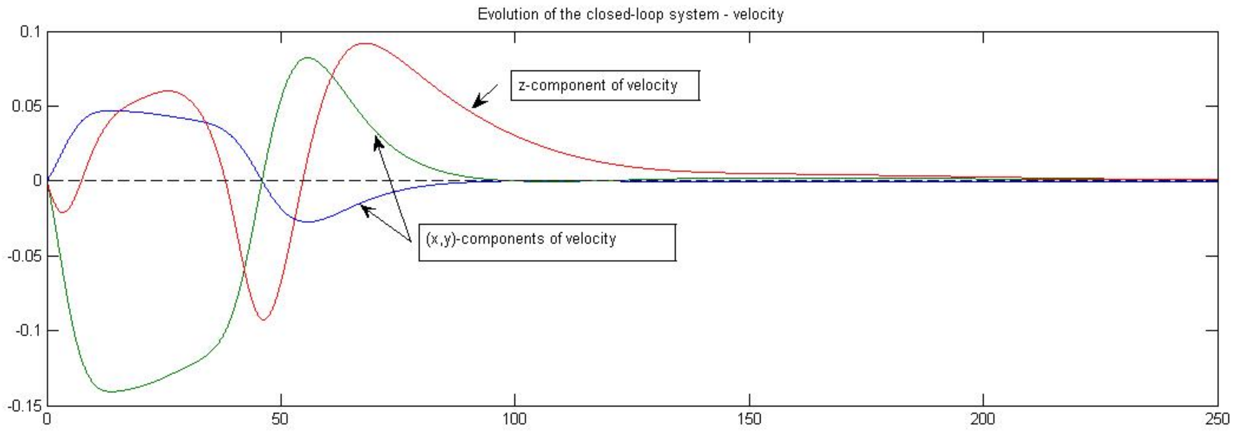
and

$$\frac{d}{dt}\mathcal{L} = -k\delta^T Q\delta - c|\epsilon|^2.$$

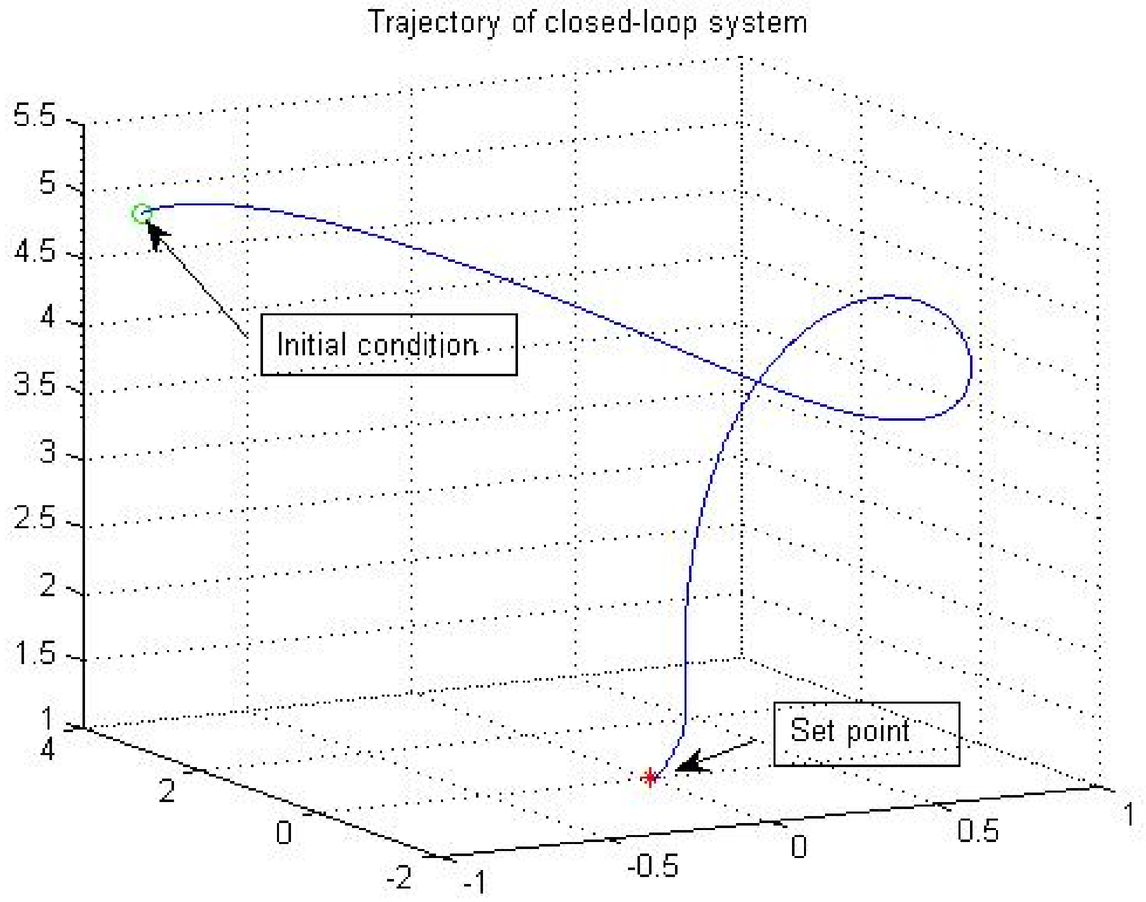
The error terms δ, ϵ are convergent to zero and the system is globally asymptotically stable to ξ_* for initial conditions $d(0) > 0$. The parameter estimate error is convergent $\tilde{\rho} \to$ *const.*.

Simulation - State evolution

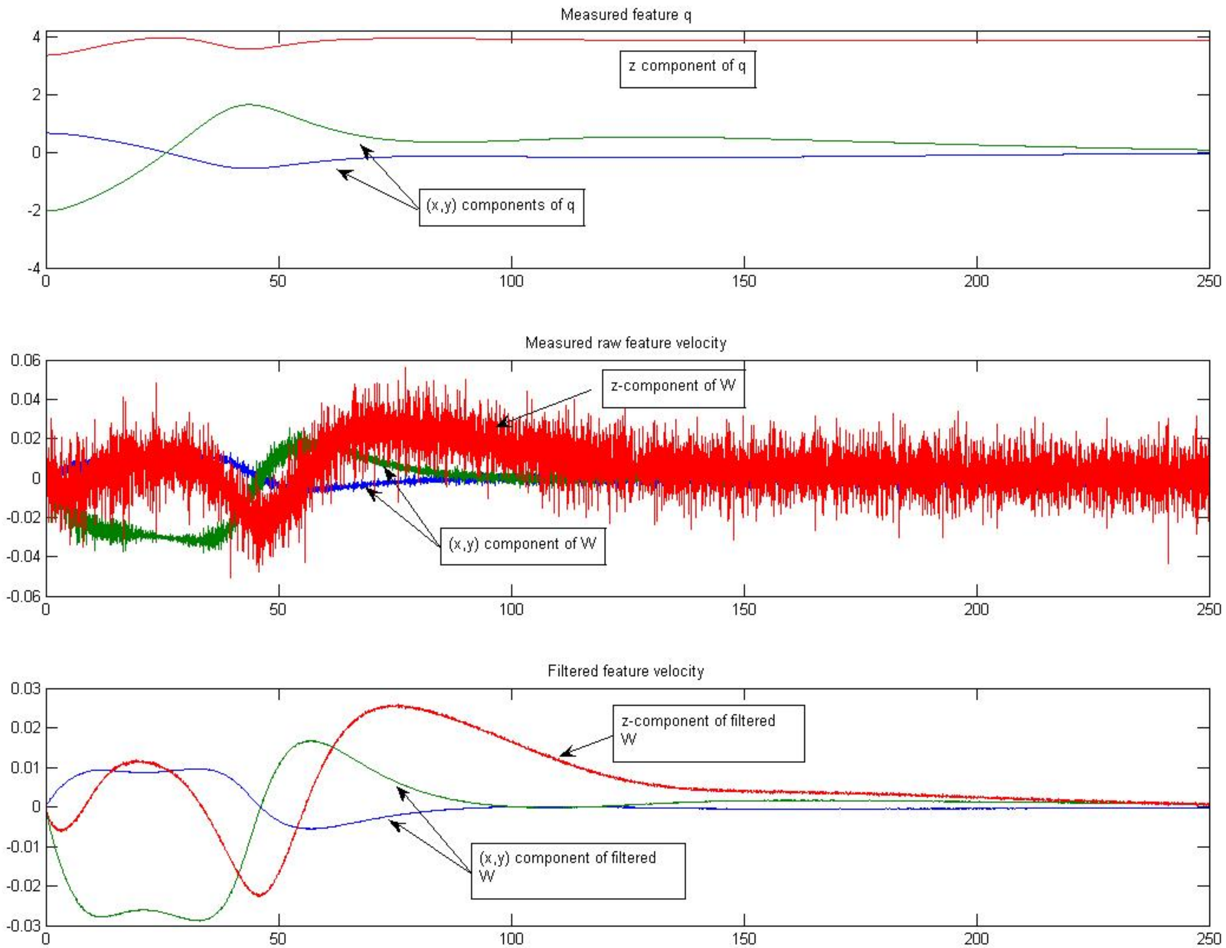




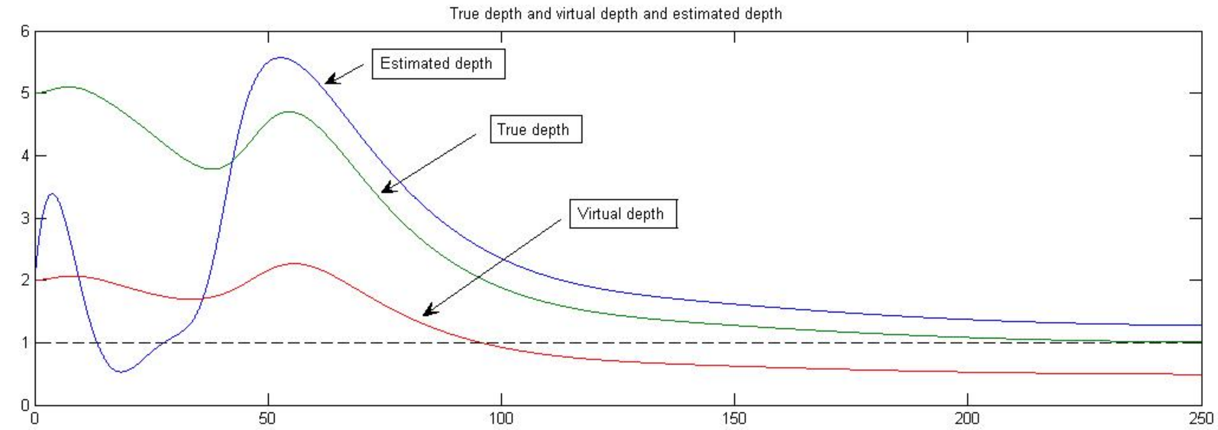
Simulation - State trajectories

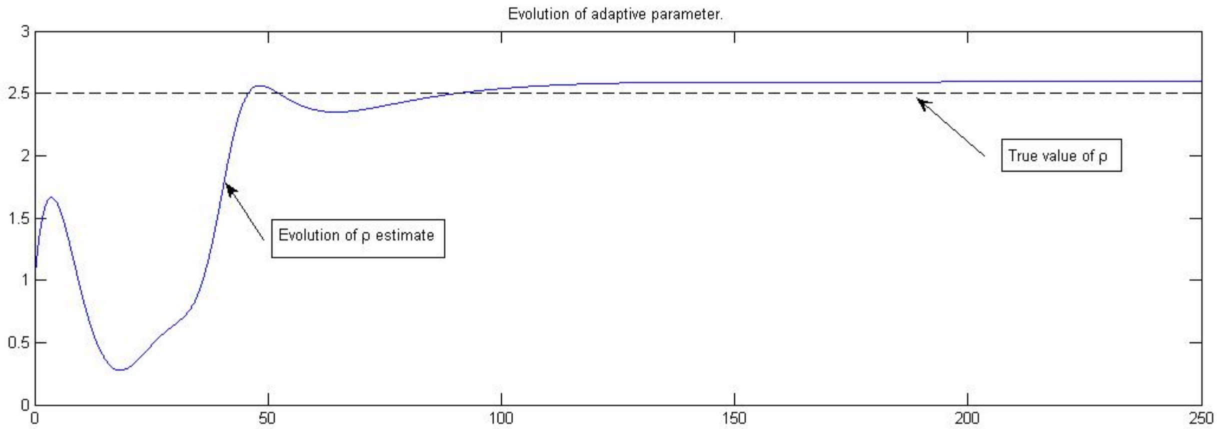


Simulation - Visual data

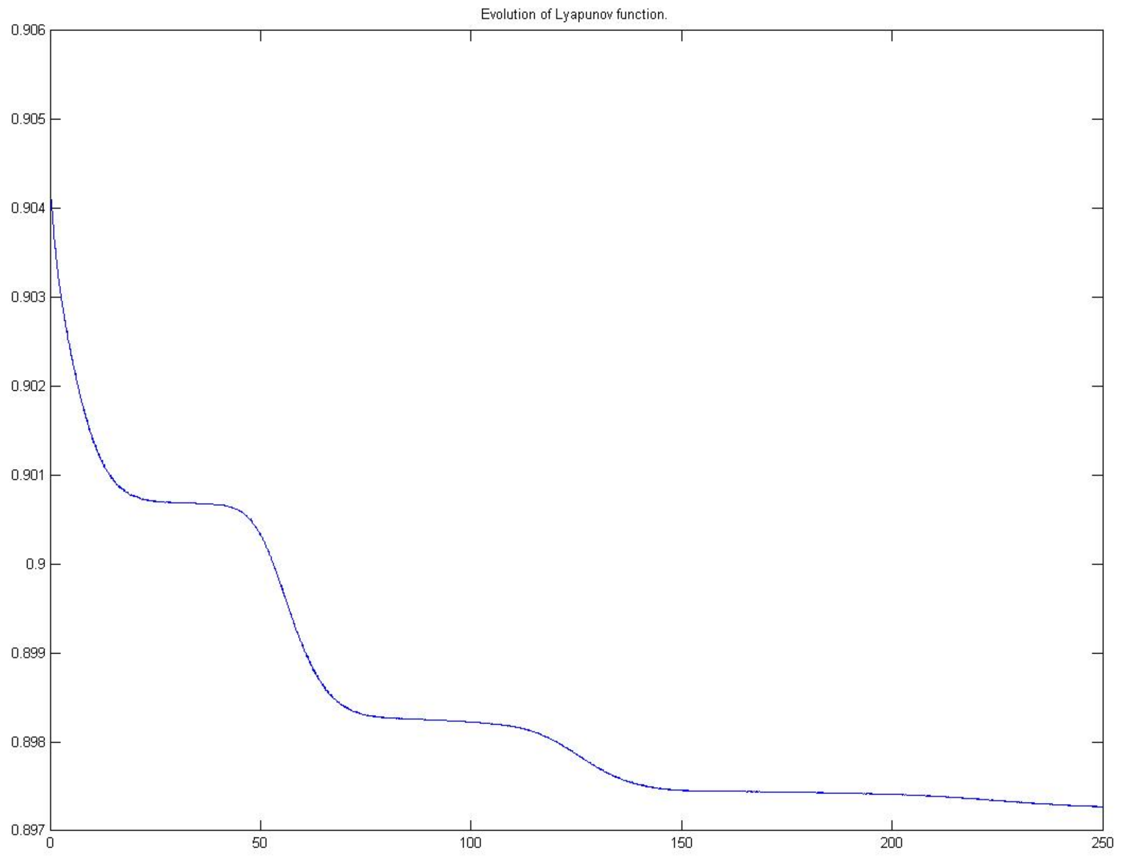


Evolution of internal controller state





Simulations - Evolution of Lyapunov function



Part IV: Conclusions

- Very careful choice of visual features.
  - Spherical projection
  - Averaging of image data into a single image feature such that the image Jacobian is square.
- Use the most robust direct measure of optic flow that can be computed.
- Dynamic inversion of unknown depth dynamics in controller structure to reduce adaptive estimation of unknown depth to an estimation of unknown initial conditions.
- Constructive non-linear control design.

There is still a long way to go in developing practical dynamical IBVS algorithms.

Work presented was undertaken in collaboration with

PETER CORKE

Senior Principal Research Scientist,

Autonomous Systems Laboratory Director,

Robotics.,

CSIRO ICT Centre, Brisbane.

TAREK HAMEL

Professor, I3S-CNRS,

Nice-Sophia Antipolis, France.