17

Homography-Based Control of Nonholonomic Mobile Robots: a Digital Approach

Andrea Usai and Paolo Di Giamberardino
University of Rome "La Sapienza"
Italy

1. Introduction

Why does one need to waste time deriving a discrete-time model from a continuous-time one? The answer is: it depends on how fast the system dynamics is.
The classical approach, especially for nonlinear systems, is to develop a continuous-time control law and then realize the controller by means of a computer implementation.
Such an approach can lead to poor controller performance due to the intrinsic system approximation: the sampling period, at which the controller commands are updated, fixes the sampling period at which the system dynamics is observed.
All the works done in the field of visual servoing, deal with continuous time systems (see, for example, in the case of mobile robots: Chen et al., 2006; Lopez-Nicolas et al., 2006; Mariottini et al., 2006). Image acquisition, elaboration and the nonlinear control law computation are very time consuming tasks. So it is not uncommon that the system is controlled with a sampling rate of 0.5Hz. Is it slow? It obviously depends.
To face the drawbacks of poor system approximation, the trivial solution is slowing down the system dynamics slowing down the controls variation rate. This is not always possible, but when the kinematic model is considered, the system dynamics depends only on the velocity imposed by the controller (there is no drift in the model) and so it is quite easy to limit the effect of a poor system approximation.
In general, taking into account the discrete-time nature of the controlled system can lead to better closed-loop system performances. This means that the control law evaluated directly in the discrete-time domain can better address the discrete-time evolution of the controlled system. This is well understood in the linear case and it is also true for (at least) a particular case of nonlinear systems, the ones that admits a finite or an exact sampled representation (Monaco and Normand-Cyrot, 2001). In fact, the possibility of an exact discretization of the continuous-time model is necessary to lead to the performances improvement previously discussed, since in this case no approximations are performed in the conversion continuous time - discrete time. The commonly used approximations can lead to a discrete-time model with a behavior that diverges from the continuous-time one.
In Section 2, the system model is presented. From this model a dicrete-time derivation is shown in Section 3. In Section 4, the design of a multirate digital control is then described

and discussed, showing that exact solutions are obtained. An improved control (and planning) strategy for the discrete-time system is introduced and applied. Simulation results are reported to put in evidence the effectiveness of the proposed approach.

2. The Camera-Cycle Model

In this section the kinematic model of the system composed of a mobile robot (an unicycle) and a camera, used as a feedback sensor to close the control loop, is presented.
To derive the mobile robot state, the relationship involving the image space projections of points that lie on the floor plane, taken from two different camera poses, are used. Such a relationship is called homography. A complete presentation of such projective relations and their properties is shown in (R. Hartley, 2003).

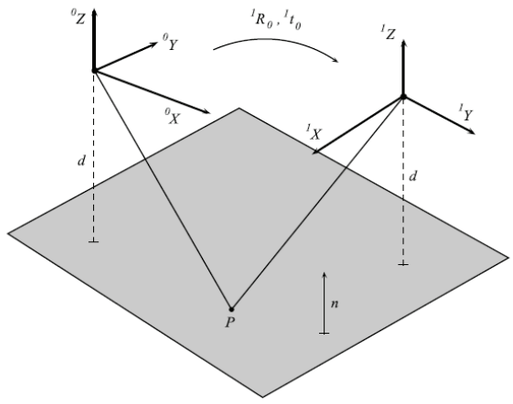

Fig. 1. 2-view geometry induced by the mobile robot.

2.1 The Geometric Model

With reference to Figure 1, the relationship between the coordinates of the point P in the two frames is

$$\begin{bmatrix} {}^1P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^1R_0 & {}^1t_0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} {}^0P \\ 1 \end{bmatrix} \tag{1}$$

It is an affine relation that becomes a linear one in the homogeneous coordinate system. If the point P belongs to a plane in the 3D space with normal versor n and distance from the origin d, it holds that

$$\begin{bmatrix} n^T & d \end{bmatrix} \begin{bmatrix} {}^0P \\ 1 \end{bmatrix} = 0 \quad \Rightarrow \quad -\frac{n^T {}^0P}{d} = 1 \tag{2}$$

note that $d > 0$, since the interest is in the planes observed by a camera and so they don't pass trough the optical center (that is the camera coordinate system origin).

Combining the Equation 1 and the right term of 2, the following relation holds

$$^{1}P = \left(^{1}R_{0} - \frac{1}{d} {}^{1}t_{0} n^{T} \right) {}^{0}P = H \, {}^{0}P \tag{3}$$

The two frame systems in Figure 1 represent the robot frame after a certain movement on a planar floor.
Choosing the unicycle-like model for the mobile robot, the matrix H become

$$H = \begin{bmatrix} \cos\theta & \sin\theta & \frac{1}{d}\left(X\cos\theta + Y\sin\theta \right) \\ -\sin\theta & \cos\theta & \frac{1}{d}\left(-X\sin\theta + Y\cos\theta \right) \\ 0 & 0 & 1 \end{bmatrix} \tag{4}$$

is the homography induced by the floor plane during a robot movement between two allowable poses. Note that $[X,Y,\theta]^T$ is the state vector of the mobile robot with reference to the first coordinate system.

2.2 The Kinematic Model

Taking four entries of matrix H such that

$$\begin{aligned} h_1 &= \cos\theta \\ h_2 &= \sin\theta \\ h_3 &= x\cos\theta + y\sin\theta \\ h_4 &= -x\sin\theta + y\cos\theta \end{aligned} \tag{5}$$

and noting that, for the sake of simplicity, the distance d has been chosen equal to one, since it is just a scale factor that can be taken into account in the sequel, the kinematic unicycle model is

$$\begin{aligned} \dot{X} &= v\cos\theta \\ \dot{Y} &= v\sin\theta \\ \dot{\theta} &= \omega \end{aligned} \tag{6}$$

where v and ω are, respectively, the linear and angular velocity control of the unicycle.
Differentiating the system in Equation 5 with respect to the time and combining it with the system in Equation 6, one obtains

$$\begin{aligned}\dot{h}_1 &= -h_2\omega \\ \dot{h}_2 &= h_1\omega \\ \dot{h}_3 &= h_4\omega + v/d = h_4\omega + {}_d v \\ \dot{h}_4 &= -h_3\omega\end{aligned} \tag{7}$$

that is the kinematic model of the homography induced by the mobile robot movement.

3. From Continuous to Discrete Time Model

In the first part of this section, it will be presented how to derive a the discrete-time system model from the continuous-time one. Afterwards, a control (and planning) strategy for the discrete-time system is introduced and applied. Simulations will be presented to prove the presented strategy effectiveness.

3.1 The General Case

With reference to Figure 1, the relationship between the coordinates of the point P in the two frames is

$$\dot{x} = f(x) + \sum_{i=1}^{m} u_i g_i(x) \tag{8}$$

with f , g_1 , ..., $g_m : M \rightarrow R^n$, analytical vector fields.
To derive a discrete-time system from the previous one, suppose to keep constant the controls u_1 , ..., u_m , by means of a zero order holder, for $t \in [kT, (k + 1)T)$ and $k \in N$. Suppose that the system output is sampled (and acquired) every T seconds, too. The whole system composed by a z.o.h , the system and the sampler is equivalent to a discrete-time system.
Following (Monaco and Normand-Cyrot, 1985; Monaco and Normand-Cyrot, 2001), it is possible to characterize the discrete-time system derived by a continuous-time nonlinear system. Sampling the system in Equation 8 with a sampling time T , the discrete-time dynamics becomes

$$\begin{aligned}x(k+1) = x(k) + T\left(f + \sum_{i=1}^{m} u_i(k) g_i \right)\Big|_{x(k)} + \\ + \frac{T^2}{2}\left(L_{f+\sum_{i=1}^{m} u_i(k) g_i} \right)^2 (Id) + \ldots = \\ = F^T(x(k), u(k))\end{aligned} \tag{9}$$

where $L(.)$ denotes the Lie derivative. It is possible to see that, this series is locally convergent choosing an appropriate T . See (Monaco and Normand-Cyrot, 1985) for details.
The problem here is the analytical expression of $F^T(x(k),u(k))$. In general, it doesn't exist. Otherwise, if from the series of Equation 9 it is possible to derive an analytical expression for

its limit function, the system of Equation 8 is said to be exactly discretizable and its limit function is called an exact sampled representation of 8.
If, better, the series results to be finite, in the sense that all the terms from a certain index on goes to zero, a finite sampled representation is obtained.
Finite sampled representations are transformed, under coordinates changes, into exact sampled ones ((Monaco and Normand-Cyrot, 2001)). As obvious, finite discretizability is not coordinate free, while exact is. Note that the existence of an exact sampled representation corresponds to analytical integrability.
A nonholonomic system as the one of Equation 8, can be transformed into a chained form system by means of a coordinate change.
This leads to a useful property for discretization pointed out in (Monaco and Normand-Cyrot, 1992). In fact it can be seen that a quite large class of nonholonomic systems admit exact sampled models (polynomial state equations). Among them, one finds the chained form systems which can be associated to many mechanical systems by means of state feedbacks and coordinates changes.
So, in general, to be able to get a finite discretization of the nonlinear system of Equation 8, one needs to find a coordinates change (if it exists) that allows to derive an exact sampled model from the nonlinear one.

3.2 The Camera-Cycle Case

Suppose the controlled inputs are piecewise constant, such that

$$\begin{cases} v(t) = v_k \\ \omega(t) = \omega_k \end{cases} \quad t \in \left[kT, (k+1)T \right) \tag{10}$$

where $k = 0, 1, ..$ and T is the sampling period.
Since the controls are constant, it is possible to integrate the system in Equation 7 in a linear fashion. It yields to

$$\begin{aligned} \begin{bmatrix} h_1(k+1) \\ h_2(k+1) \end{bmatrix} &= A(\omega_k) \begin{bmatrix} h_1(k) \\ h_2(k) \end{bmatrix} \\ \begin{bmatrix} h_3(k+1) \\ h_4(k+1) \end{bmatrix} &= A(\omega_k)^T \begin{bmatrix} h_3(k) \\ h_4(k) \end{bmatrix} + B(\omega_k)_d v_k \end{aligned} \tag{11}$$

where

$$\begin{aligned} A(\omega_k) &= \begin{bmatrix} \cos \omega_k T & -\sin \omega_k T \\ \sin \omega_k T & \cos \omega_k T \end{bmatrix} \\ B(\omega_k) &= \begin{bmatrix} \sin\left(\omega_k T\right)/\omega_k \\ \left(\cos\left(\omega_k T\right) - 1\right)/\omega_k \end{bmatrix} \end{aligned} \tag{12}$$

If one considers the angular velocity input as a time-varying parameter, the system in Equation 11 become a linear time-varying system. Such a property allows an easy way to compute the evolution of the system. Precisely, its evolution becomes

$$\begin{bmatrix} h_1(k) \\ h_2(k) \end{bmatrix} = \prod_{i=0}^{k-1} A(\omega_{k-i-1}) \begin{bmatrix} h_1(0) \\ h_2(0) \end{bmatrix}$$
$$\begin{bmatrix} h_3(k) \\ h_4(k) \end{bmatrix} = \prod_{i=0}^{k-1} A(\omega_{k-i-1})^T \begin{bmatrix} h_3(0) \\ h_4(0) \end{bmatrix} + \sum_{j=0}^{k-2} \left(\prod_{i=j+1}^{k-1} A(\omega_{k-i})^T \right) B(\omega_j)\, {}_d v_j \tag{13}$$

where the sequences $\{v_k\}$ and $\{\omega_k\}$ are the control inputs. This structure will be useful in the sequel for the control law computation.

4. Controlling a Discrete-Time Nonholonomic System

Interestingly, difficult continuous control problems may benefit of a preliminary sampling procedure of the dynamics, so approaching the problem in the discrete time domain instead of in the continuous one.
Starting from a discrete-time system representation, it is possible to compute a control strategy that solves steering problems of nonholonomic systems. In (Monaco and Normand-Cyrot, 1992) it has been proposed to use a multirate digital control for solving nonholonomic control problems, and in several works its effectiveness has been shown (for example (Chelouah et al., 1993; Di Giamberardino et al., 1996a; Di Giamberardino et al., 1996b; Di Giamberardino, 2001)).

4.1 Camera-cycle Multirate Control

The system under study is the one in Equation 11 and the form of its state evolution in Equation 13.
The problem to face is to steer the system from the initial state $h_0 = [1, 0, 0, 0]^T$ (obviously corresponds to the origin of the configuration space of the unicycle) to a desired state ${}^d h$, using a multirate controller. If r is the number of sampling periods chosen, setting the angular velocity constant over all the motion, one gets for the state evolution

$$\begin{bmatrix} h_1(r) \\ h_2(r) \end{bmatrix} = A^r({}^c\omega) \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$
$$\begin{bmatrix} h_3(r) \\ h_4(r) \end{bmatrix} = A^{r-1}({}^c\omega) B({}^c\omega)\, {}_d v_0 + A^{r-2}({}^c\omega) B({}^c\omega)\, {}_d v_1 + \ldots + B({}^c\omega)\, {}_d v_{r-1} \tag{14}$$

At this point, given a desired state, one just need to compute the controls. The angular velocity ${}^c\omega$ is firstly calculated such that:

$$\begin{bmatrix} {}^d h_1 \\ {}^d h_2 \end{bmatrix} = A^r({}^c\omega) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{15}$$

Once ${}^c\omega$ is chosen, the linear velocity values ${}_d v_0$, ..., ${}_d v_{r-1}$ can be calculated solving the linear system

$$\begin{bmatrix} {}^d h_3 \\ {}^d h_4 \end{bmatrix} = \begin{bmatrix} A^{r-1}({}^c\omega)B({}^c\omega) & A^{r-2}({}^c\omega)B({}^c\omega) & \dots & B({}^c\omega) \end{bmatrix} \begin{bmatrix} {}_d v_0 \\ {}_d v_1 \\ \dots \\ {}_d v_{r-1} \end{bmatrix} = RV \tag{16}$$

which is easily derived from the second two equations of 14.
Note that, for steering from the initial state to any other state configuration, at least $r = 2$ steps are needed, except for the configuration that present the same orientation of the initial one. More precisely, it can be seen that if this occurs, the angular velocity ${}^c\omega$ is equal to zero or Π/T and the matrix R in Equation 16 become singular.
Exactly that matrix shows the reachability space of the discrete-time system: if ${}^c\omega \neq \{0, \Pi/T\}$ then the vector B is rotated r-times and the whole configuration space is spanned.
Furthermore, if 2 is the minimum multirate order to guarantee the reachability of any point of the configuration, one can choose a multirate order such that $r > 2$ and the further degrees of freedom in the controls can be used to accomplish the task obtaining a smoother trajectory or avoiding some obstacles, for instance. Note that it can be achieved solving a quadratic programming problem as

$$\min_{v \in V} \frac{1}{2} V^T \Sigma V + \Gamma V \tag{17}$$

where Σ and Γ are two weighting matrixes, such that the robot reaches the desired pose, granting some optimal objectives. Other constraints can be easily added to take account of further mobile robot movements requirements.

4.2 Closing the Loop with the Planning Strategy

Let $[{}^d\theta, {}^d h_3, {}^d h_4]^T$ be the desired system state and mark the actual one with the subscript k. Note that, in the control law development, the orientation of the mobile robot is used, instead of the first two components of the system of Equation 11.
Summarize the algorithm steps as

0. Set $r_k = r$.
1. Choose
$$ {}^c\omega = \left({}^d\theta - \theta_k \right) / r_k T$$
2. Compute the control sequence V as
$$\min_V V^T V$$
such that
$$RV = \begin{bmatrix} {}^d h_3 \\ {}^d h_4 \end{bmatrix}$$
with the same notation of Equation 16.

3. If r_k>2 then r_k= r_k+*1* and go to step *1*. Otherwise, the algorithm ends.

The choice of the cost function shown leads to a planned path length minimization. If the orientation error of the point 1 is equal to zero, it needs to be perturbed in order to guarantee some solution admissibility to the programming problem of point 2.
Furthermore, since the kinematic controlled model derives directly from an homography, it is possible use the homographies compositional property to easily update the desired pose, from the actual one, at every control computation step. Exactly, since

$$ {}^{d}H_0 = {}^{d}H_{r-1}\,{}^{r-1}H_{r-2}\ldots{}^{k}H_{k-1}\ldots{}^{1}H_0 \tag{18} $$

it is possible to easily update the desired pose as needed for the close-loop control strategy.
A Simulated path is presented in Figures 3: the ideal simulated steer execution (Figure 2) is perturbed by the presence of some additive noise in the controls. This simulate the effect of some non ideal controller behavior (wheel slipping, actuators dynamics, ...). The constrained quadratic problems involved in the controls computation are solved using an implementation of the algorithm presented in (Coleman and Li, 1996).

4.3 Setting Up the Trajectory Planning

The angular velocity defines the span of the system configuration space by means of the vectors in the matrix *R* of the Equaion 16: varying it during the planning can lead to a better functional minimization. Moreover, it is needed to settle the ω choice in case of no orientation error.
First of all, we need ω is equal to zero at the beginning of the planned path and at the end of it. Consider the function

$$ \omega_0(k) = {}^{\max}\omega_0 \sin^2\left(k\frac{\pi}{r}\right) \quad k \in \{0,1,\ldots,r\} \tag{19} $$

it starts from zero and softly goes to 1 up to return, as softly as before, to 0. For using this function as angular velocity control we have to find its maximum value to take to zero the orientation error. It can be done setting

$$ {}^{d}\theta = {}^{\max}\omega_0 \sum_{k=0}^{r-1} \sin^2\left(k\frac{\pi}{r}\right) \tag{20} $$

In order to face the null orientation error case, intuitively, the mobile robot should first point the desired pose and then compensate for the desired orientation. Take a look at the function in Equation 19 with the doubled frequency

$$ \omega_p(k) = \begin{cases} +{}^{\max}\omega_0 \sin^2\left(k\frac{2\pi}{r}\right) & k \in \{0,1,\ldots,r \setminus 2\} \\ -{}^{\max}\omega_0 \sin^2\left(k\frac{2\pi}{r}\right) & k \in \{r \setminus 2+1,\ldots,r\} \end{cases} \tag{21} $$

where with the symbol \ it is denoted the integer division operation. Note that integral of the function in the previous Equation is zero, in the interval of interest, as the final orientation displacement, due to its contribution. After half interval can be found the maximum velocity needed to cancel the pointing angle displacement as in Equation 20. The angular velocity is chosen as

$$\omega(k) = \omega_o(k) + \omega_p(k) \tag{22}$$

Since we want to get the shortest path, from the planning strategy, we minimize $V^T V$. Furthermore, we would like to minimize the variations between the planned controls ${}_d v_0,\ldots,$ ${}_d v_{r-1}$, in order to have a smoother behavior of the controlled robot movement. It is possible to accomplish to these specifications, solving the quadratic programming problem

$$\min_{v \in V} \left[V^T \Sigma V + \gamma \sum_{k=1}^{r-1} \left({}_d v_k - {}_d v_{k-1} \right)^2 \right] \tag{23}$$

such that

$$\begin{aligned} & RV = b \\ & \left| {}_d v_k \right| < 1 \quad k = 0,\ldots,r-1 \\ & {}_d v_0 = {}_d v_{r-1} = 0 \end{aligned} \tag{24}$$

where the γ is a minimization parameter and the matrix R and the vector b are the same of Equation 16. Note that it is a quadratic programming problem since the second addend of the cost function of Equation 23 can be expressed as

$$V^T \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ldots & \ldots & \ldots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix} V \tag{25}$$

The last two constraints of Equation 24 are added to make explicit that a pose to pose trajectory is planned. Note that the planned trajectory shown in Figure 4 it is shorter than the corresponding one of Figure 2. This happens because the programming problem referred to the trajectory of Figure 4 is more constrained than the other one: varying ω leads to a better suboptimal solution. As expected.

4.4 Trajectory Tracking: a Multirate Digital Approach

Once the trajectory is planned, it is necessary to introduce a tracking technique to execute it. Classical continuous approaches can be obviously applied: for instance, in Usai & Di

Giamberardino (2006), we use a linear controller to track a multirate planned trajectory, again for a visual servoing problem.
In the preceding sections, it has been introduced a close-loop control strategy, iterating the previously discussed planning technique. It will be now discussed an extension of this approach to derive a digital trajectory tracker.
Trivially, if we want to follow a chosen trajectory we can re-apply the multirate planning strategy to take the system from the actual point to a next intermediary one. The incessant desired pose update constraints the system to follow the previously planned path. More precisely, every m steps it is possible to re-solve a quadratic programming problem similar to the one solved for the planning problem but, this time, the desired pose is the one planned τ steps ahead the actual one.
Summarizing the described tracking algorithm, we have

0. Set r and plan the trajectory. Set $K=0$, $m_k=0$.
1. Execute the planned controls at step k and set $k=k+1$ and $m_k= m_k+1$.
2. If $k=r$, the algorithm stops.
3. If $m_k<m$ come back to point *1.*
4. *If* $k\leq(r-\tau)$, update the ω computing

$$\omega(i+1)=\omega(i)+\frac{{}^{d}\theta_{k-1}-\theta_{k}}{mT} \quad i=k,\ldots,k+m$$

And, subsequently, the sequence ${}_{d}v$ minimizing

$$\min_{{}^{m}V}({}^{m}V)^{T}({}^{m}V)+\gamma_1({}^{m}R\,{}^{m}V-{}^{m}b)^{T}({}^{m}R\,{}^{m}V-{}^{m}b)+\gamma_2({}^{d}v_{k+\tau}-{}_{d}v_{k+\tau})^{2}$$

5. Set $m_k=0$ and go to point *1.*

Note that the programming problem has been relaxed in order to have a smoother tracking behaviour. For the same reason it is added a term regarding the desired pose velocity.
Furthermore, the two parameters m and τ influence how the trajectory is tracked. For instance, if $\tau = 2$ the error in the trajectory execution is less than the one obtained with larger values of τ. On the other hand, large values of τ allow many degrees of freedom for the programming problem solution. This determines the deformation of the previously planned trajectory, putting in the programming problem new constraints. For example, these constraints would be useful to let the robot avoid other unexpected obstacles that come out during robot movement. An interesting implementation could be the one in which the parameter τ varies when some event occurs (a new obstacle detected by sensors), allowing the path deformation to a new feasible one. With regard to the parameter m (remark that $m < \tau$), since it influences how many times the control is computed during the tracking, it depends on the computer on which the controller will be implemented.

5. Conclusion

In this chapter, a kinematic model for a system composed by a mobile robot and a camera, has been presented.
Since such a model is exactly discretizable, it has been possible to propose a multirate digital control strategy able to steer the system to a desired pose in an exact way.

As can be seen from the trajectory obtained in the simulations of Figures 2 and 3, there are large differences between the ideal path execution and the perturbed one.
It happens because the iterated planning strategy has is no memory of the previously planned path. The controls are just constrained to take the system to the desired state. There is no control on how get there.
In a real implementation it is advisable to have a certain degree of predictability of the robot behavior during its movement and the respect to some optimal criteria (short paths, smooth movements, obstacles avoidance,...), too. This is why it has been chosen to present a separated planning phase and a subsequent the tracking phase. The effectiveness of the control scheme adopted has been verified by simulations and presented in Figures 4.

6. References

Chelouah, A., Di Giamberardino, P., Monaco, S., and Normand-Cyrot, D. (1993). Digital control of nonholonomic systems two case studies. *In Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on*, pages 2664–2669vol.3.

Chen, J., Dixon, W., Dawson, M., and McIntyre, M. (2006). Homography-based visual servo tracking control of a wheeled mobile robot. *Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on]*, 22(2):406–415.

Coleman, T. and Li, Y. (1996). A reflective newton method for minimizing a quadratic function subject to bounds on some of the variables. *SIAM Journal on Optimization*, 6(4):1040–1058.

Di Giamberardino, P. (2001). Control of nonlinear driftless dynamics: continuous solutions from discrete time design. *In Decision and Control, 2001. Proceedings of the 40th IEEE Conference on*, volume 2, pages 1731–1736vol.2.

Di Giamberardino, P., Grassini, F., Monaco, S., and Normand-Cyrot, D. (1996a). Piecewise continuous control for a car-like robot: implementation and experimental results. *In Decision and Control, 1996., Proceedings of the 35th IEEE*, volume 3, pages 3564-3569vol.3.

Di Giamberardino, P., Monaco, S., and Normand-Cyrot, D. (1996b). Digital control through Finite feedback discretizability. *In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on*, volume 4, pages 3141–3146vol.4.

Lopez-Nicolas, G., Sagues, C., Guerrero, J., Kragic, D., and Jensfelt, P. (2006). Nonholonomic epipolar visual servoing. *In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2378–2384.

Mariottini, G., Prattichizzo, D., and Oriolo, G. (2006). Image-based visual servoing for nonholonomic mobile robots with central catadioptric camera. *In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 538–544.

Monaco, S. and Normand-Cyrot, D. (1985). On the sampling of a linear analytic control system. *In Decision and Control, 1985., Proceedings of the 24th IEEE Conference on*, pages pp. 1457–1461.

Monaco, S. and Normand-Cyrot, D. (1992). An introduction to motion planning under multirate digital control. *In Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, pages 1780–1785vol.2.

Monaco, S. and Normand-Cyrot, D. (2001). Issues on nonlinear digital control. *European Journal of Control*, 7(2-3).

R. Hartley, A. Z. (2003). *Multiple View Geometry in Computer Vision*. Number ISBN: 0-521-54051-8. Cambridge University Press.

Usai , A., Di Giamberardino, P. (2006). A multirate digital controller for non holonomic mobile robot pose regulation via visual feedback. *WSEAS Transactions on Systems*, 5, 1129–1136.

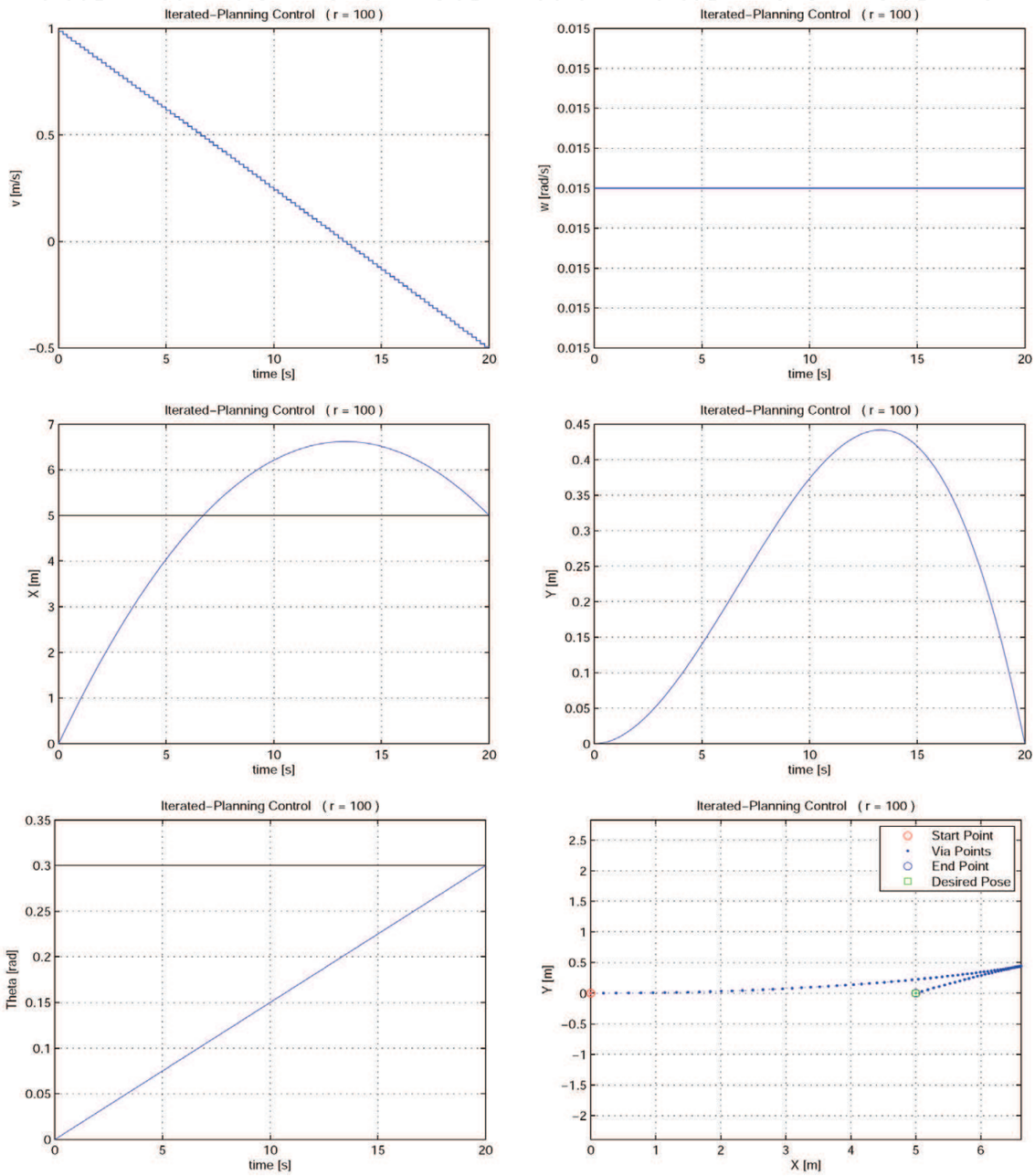

Fig. 2. Multirate control simulation ($d = 1m$). Ideal (no noise) path execution.

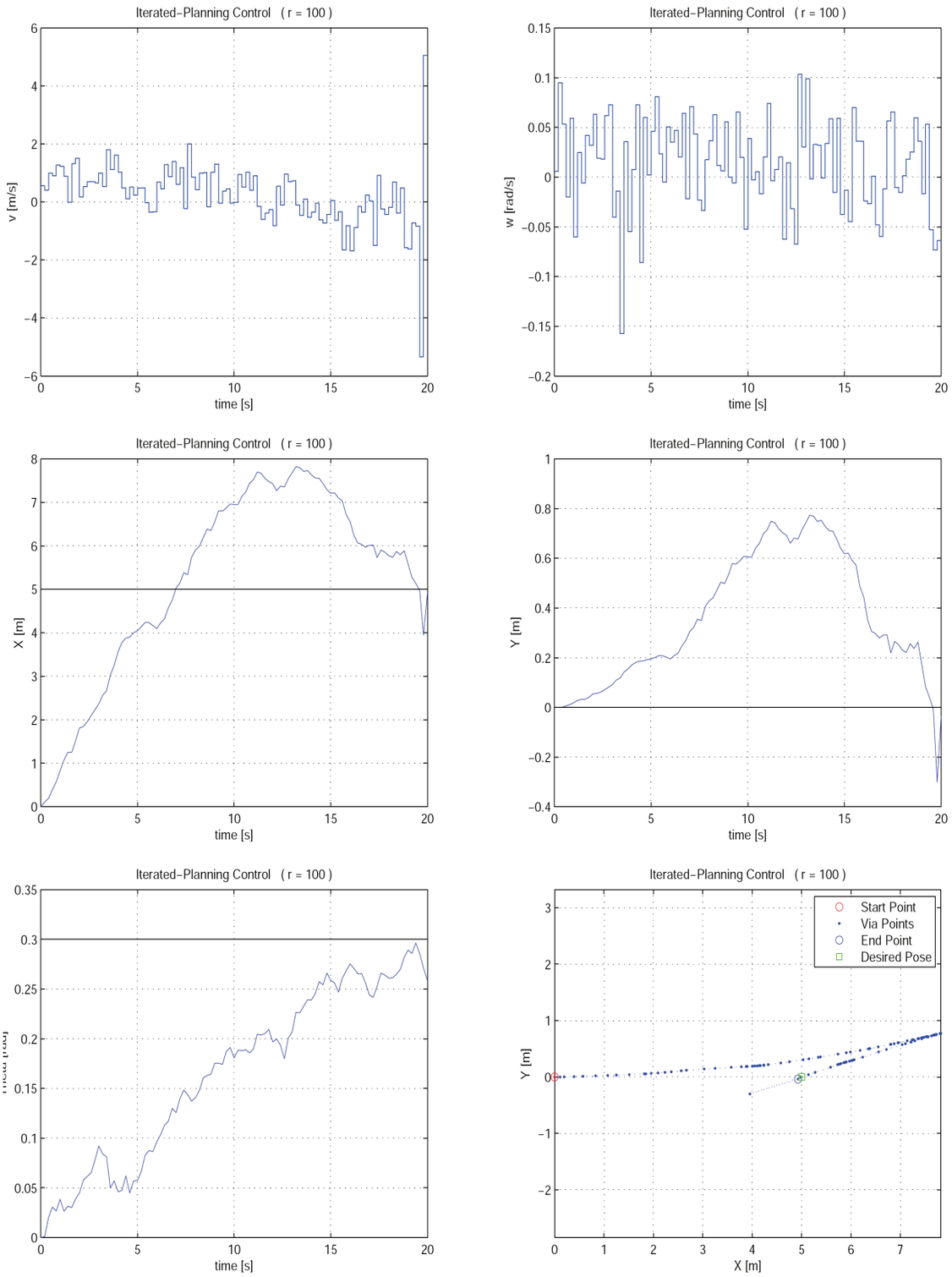

Fig. 3. Multirate control simulation (re-iterated planning, $d = 1m$). Additive random noise on controls (gaussian with std.dev. 0.5 and 0.05, for v and ω respectively).

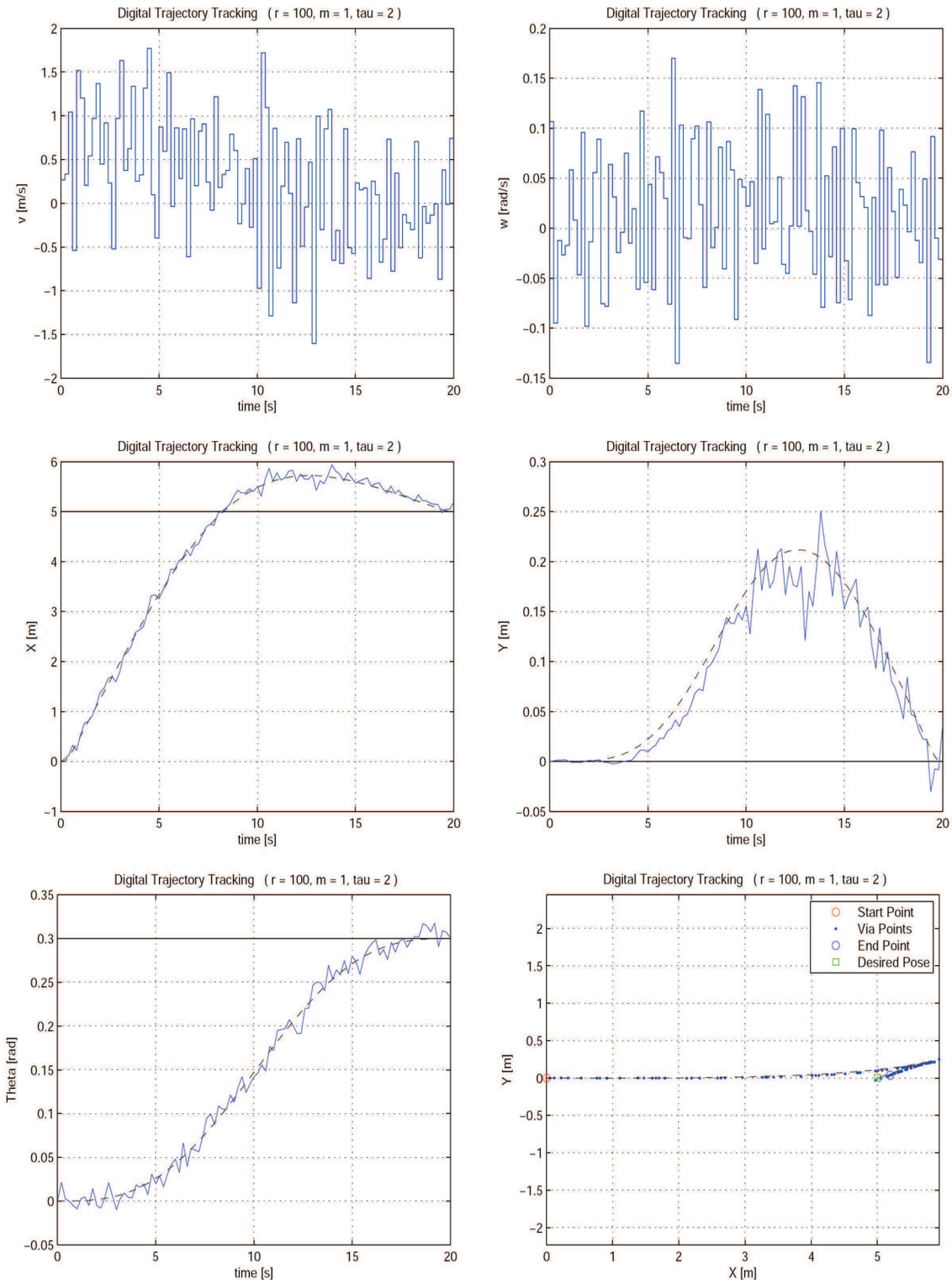

Fig. 4. Multirate control simulation (planning + tracking, $d = 1m$). Additive random noise on controls (gaussian with std.dev. 0.5 and 0.05, for v and ω respectively).

Frontiers in Robotics, Automation and Control

Edited by Alexander Zemliak

ISBN 978-953-7619-17-6
Hard cover, 450 pages
Publisher InTech
Published online 01, October, 2008
Published in print edition October, 2008

This book includes 23 chapters introducing basic research, advanced developments and applications. The book covers topics such us modeling and practical realization of robotic control for different applications, researching of the problems of stability and robustness, automation in algorithm and program developments with application in speech signal processing and linguistic research, system's applied control, computations, and control theory application in mechanics and electronics.

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen