Ultrasonic Localization of Mobile Robot Using Active Beacons and Code Correlation

Marek Peca

Department of Control Engineering, Faculty of Electrical Engineering,
Czech Technical University in Prague, Czech Republic
pecam1@fel.cvut.cz
http://dce.felk.cvut.cz/

Abstract. Ultrasonic localization system for planar mobile robot inside of a restricted field is presented. System is based on stationary active beacons and measurement of distances between beacons and robot, using cross-correlation of pseudorandom binary sequences (PRBSes). Due to high demand of dynamic reserve imposed by range ratio in our specific task, both code- as well as frequency-divided media access has been utilized. For the same reason, 1-bit signal quantization has been abandoned in favor of higher resolution in receiver analog-to-digital conversion. Finally, dynamic estimation of the position is recommended over analytic calculation. The final solution uses the extended Kalman filter (EKF), equipped with erroneous measurement detection, initial state computation, and recovery from being lost. EKF also performs data-fusion with odometry measurement. Unlike the approach in majority of works on mobile robot localization, a model, actuated solely by additive process noise, is presented for the data-fusion. It offers estimation of heading angle, and remains locally observable. Simplistic double integrator model of motion dynamics is described, and the importance of clock dynamics is emphasized.

1 Design Considerations

1.1 Objectives

The system has been designed for a planar wheeled robot navigation inside a restricted, rectangular 3×2.1 m playground according to the EUROBOT competition rules [1]. Around the playground, there may be placed at least three stationary beacons. They may be interconnected together by a wire.

Our robot is a conventional two-wheel vehicle. Its floor projection is a rectangle 258×290 mm. Beacon counterpart is a "sensor area" on top of the robot in rectangle center. Beacon to sensor area distance ranges between $0.19 \ldots 3.6$ m. Rough estimate of maximal robot velocity is $2\,\mathrm{ms}^{-1}$.

1.2 System Fundamentals

The system is based on wave propagation and signal time-of-flight measurement. Ultrasound has been selected for its slow propagation speed, compared to

A. Gottscheber, D. Obdržálek, and C. Schmidt (Eds.): EUROBOT 2009, CCIS 82, pp. 116–130, 2010.

electromagnetic waves. Therefore, cutting-edge high-frequency hardware is not required. Our system is characterized by following fundamentals:

- three static transmitters (beacons), one mobile receiver on the robot
- time displacement evaluation based on pseudorandom binary sequences (PRBSes) correlation
- continuous simultaneous transmission of modulated PRBSes
- both code-divided multiple acces (CDMA) and rudimentary frequency-divided multiple access (FDMA) to provide sufficient dynamic reserve
- Doppler effect ignored

The beacons can be either active (transmitting), passive (receiving), or reflective (returning the signal to the robot). Measurement data need to be collected at the robot, what lead to choice of active beacons.

Distance measurement systems based on wave propagation in general use either short pulses [2], or pseudorandom signals, due to desired correlation properties. Auto-correlation of signals of both types approaches single, narrow peak (Sec. 2.1). Continuous transmission of pseudorandom signals has been favoured over the pulses interleaved by long pauses, due to better utilization of communication channel capacity, and rosistance against short-time disturbances. Only binary pseudorandom signals (PRBSes) have been used, mainly for their easy generation and amplification in simple hardware (Sec. 2.2).

The primary purpose of PRBSes is to provide clear cross-correlation between received and known transmitted signal. Besides this, selection of different PRBSes at each beacon inherently provides CDMA separation. Ie., the PRBSes can be transmitted and mixed together in the same, shared ultrasonic band. Separation improves with increasing sequence length (Eq. 2). On the other hand, longer sequence implies longer measurement period. Therefore, we decided to separate channels also by placing them into different frequency bands (FDMA).

2 Signal Flow

2.1 Modulation and Coding

Both transmitter and receiver in the system use Polaroid 600[1] electrostatic transducers. They offer relatively wide bandwidth ($\sim$ 30 kHz), compared to more common ceramic transducers (typically $\sim$ 2 kHz). The transducers require 150 V DC bias, and a transmitter should be excited nominally by 300 V_{p-p} AC voltage.

Frequency spectrum. Signals are transferred at ultrasonic frequencies 39 . . . 69 kHz, where the transducers offer the strongest response. BPSK (binary phase-shift keying) modulation is used for conversion of base-band PRBS signals into a ultrasonic band. BPSK of a square carrier[2] wave by a binary sequence produces

[1] Acquired and sold now by SensComp, Inc.

[2] Square carrier wave is a good replacement for ideal sinusoid in this case, as higher harmonics of the square wave lies far away from the signal band of interest.

also square wave as a result, simply generable by digital hardware, and suitable to be amplified to high voltage by a simple switching circuitry. Amplitude spectrum of a periodic PRBS of finite length is actually discrete, but in practice, it approaches a spectrum of infinite-length random binary sequence. After modulation by BPSK with carrier frequency f_0, neglecting side lobes mirrored about zero frequency, the resulting amplitude spectrum is $A(f) = A_0 \left|\operatorname{sinc} \frac{f-f_0}{f_b}\right|$, where $\operatorname{sinc} x = \frac{\sin \pi x}{\pi x}$, and f_B is a modulation frequency, ie. bitrate.

Although the main (effective) power content of the signal is transferred within a main lobe $f_0 \pm f_b$, there are still side lobes with amplitude $\leq \frac{f_b}{\pi(f-f_0)}$, interfering with a neighbouring channel. Besides the (slow) side-lobe amplitude decay, there is also an output filter in each transmitter, improving FDMA separation. To keep electronics simple, the output filter is only a 2^{nd} order resonator (Sec. 2.2).

Available transducer bandwidth has been subdivided between signal bands (main BPSK lobes) and "safety gaps" between them. Chosen gap width resulted from trade-off between FDMA separation, and useful bandwidth, proportional to bitrate. The result is displayed at Fig. 1. $f_{1\ldots3}$ are carrier frequencies of the three channels, corresponding to the three beacons. $f_b = 3\,\text{kHz}$, useful bandwidth per channel $\sim 2f_b = 6\,\text{kHz}$, distance between neighbouring carriers $f_2 - f_1 = f_3 - f_2 = 12\,\text{kHz}$.

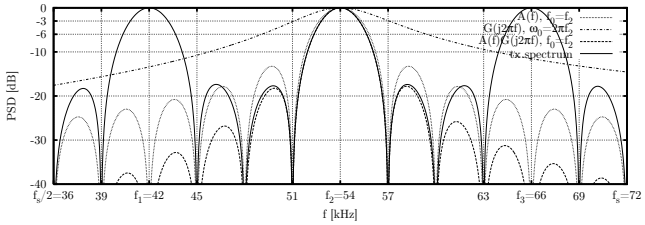

Fig. 1. Spectra of BPSK modulated PRBSes

All the carrier frequencies have been selected to be integral multiples of f_b: $f_1 = 42\,\text{kHz}$, $f_2 = 54\,\text{kHz}$, $f_3 = 66\,\text{kHz}$, and the carriers are coherent with transmitted PRBS. A sampling frequency $f_s = 72\,\text{kHz}$ in the receiver (Sec. 2.3) is also an integral multiple of f_b, although sampling is not coherent with transmission[3]. Also, all the three bands are placed symmetrically with respect to f_s, simplifying digital signal processing in the receiver (Sec. 2.3).

The transfer function of 2^{nd} order filter is $G(s) = \frac{Bs}{s^2+Bs+\omega_0^2}$, where $B\,[\text{rad}\,\text{s}^{-1}]$ is a 3-dB bandwidth. Given $f_0 = f_1$, $B = 2\pi 6\,\text{kHz}$, $\omega = 2\pi(f_0 + 12\,\text{kHz})$, an attenuation provided by the filter is $|G(s \approx j\omega)| \simeq 0.27 \sim 11\,\text{dB}$. Together with side-lobe decay, which is at 12 kHz distance $\approx \frac{1}{4\pi} \sim 22\,\text{dB}$, the FDMA channel separation is 33 dB.

[3] Because the receiver and transmitter clocks are not synchronized.

PRBS Codes. The PRBSes are transmitted continuously as a periodic signal, so the right tool to determine a time difference is a cyclic cross-correlation function

$$R_{xy}[n] = \sum_{m=0}^{N-1} x[m]y[(m+n) \bmod N] \tag{1}$$

of two signals $x[k], y[k]$[4] of period (length) N samples. An estimate of time displacement between x, y is $n_* = \arg\max_n R_{xy}[n]$.

PRBSes used in our system belong to set of Gold codes [3]. Numerical values ± 1 are assigned to PRBS bits. The auto-correlation function of a Gold code is $R_{pp}[0] = N, R_{pp}[n = 1 \ldots N-1] = -1$. Among other codes with the same property, the Gold codes have improved cross-correlation indifference between each other: different codes of the same length have their cross-correlation maxima, ie. false peaks in case of interference (crosstalk), bounded by some small value.

CDMA separation is given by a ratio of this cross-correlation upper bound and auto-correlation maximum. Gold codes are $N = 2^l - 1$ samples[5] long. According to [3], the ratio is

$$r_G = \frac{2^{\lceil \frac{l+1}{2} \rceil}}{2^l - 1}\,. \tag{2}$$

For $N \gg 1$ and l odd $r_G \approx \sqrt{\frac{2}{N}}$, for l even $r_G \approx \sqrt{\frac{4}{N}}$.

The only tunable parameter of Gold codes is the length N. It implies level of CDMA separation, and together with f_b also a measurement period T. Multiplied by a sound velocity c, T gives maximal measurement range D_{max}. We have chosen $N = 2^7 - 1 = 127$, what gives $T = 42.333\,\mathrm{ms}$, $D_{max} = 14.538\,\mathrm{m}$ ($c = 343.4\,\mathrm{ms}^{-1}$). The period is much shorter than mechanical time constants of our robot, and the range also exceeds playground size.

For $N = 127$, the ratio of CDMA is $r_G = 0.13 \sim 18\,\mathrm{dB}$. Together with FDMA reserve, the overall channel separation is approx. 51 dB. Since radiated power is reciprocally proportional to squared distance, the minimal dynamic reserve given by minimal and maximal distances d_{min}, d_{max}, see Sec. 1.1, is $20 \log_{10} \frac{d_{max}}{d_{min}}\,\mathrm{dB} = 26\,\mathrm{dB}$. However, the "extra" dBs gained by FDMA & CDMA are often spent in fight against various imperfect signal propagation conditions, non-ideal processing in a receiver, curved transducer response, and lots of ubiquitous noise.

If a separation of 51 dB should be provided solely by CDMA, the corresponding Gold code length would be in order of 2^{17} or 2^{19}. More realistic choice may be a $N = 2^{11} - 1 = 2047$ code, providing 30 dB. Using such code and BPSK modulation over whole 30 kHz band, $f_b = 15\,\mathrm{kHz}$, would result in $3.22\times$ longer (slower) measurement period. On the other hand, it would offer $5\times$ higher baudrate, thus possibly a better distance measurement resolution.

In our system, we retained the combined FDMA & CDMA scheme, where the separation could be even increased by employment of higher-order transmitter output filters.

[4] If $y \equiv x$, R_{xx} is an auto-correlation function.

[5] Samples or bits, usually called "chips".

2.2 Transmitter

The beacons are interconnected by a wire, providing simple means of their synchronization. Square PRBS & BPSK signals for all three transmitters are generated by one ARM7 microcontroller using one integrated timer-counter.

Microcontroller is set at one beacon, distributing square-wave signals to two other beacons using physical layer of RS485 interface standard. Logic-level square signals are current amplified at 5 V supply by a buffer and transformed to $\sim 200 \ldots 300\,\mathrm{V_{p-p}}$ by a custom pot-core transformer. The transformer coil is a base of the 2^{nd} order LC filter, formed together with a transducer, bandwidth defining resistor, and a tuning capacitor. Another transformer and rectifier provides a bias voltage.

2.3 Receiver

Sampling. Signal flow from the transmitter to the receiver and an estimator is shown at Fig. 2. The three signals travel through the air, and delayed by the path length, they are mixed and received by an ultrasonic transducer at the robot. Received signal is directly sampled by an analog-to-digital converter (ADC) out of the baseband, ie. at sampling frequency f_s lower than double the maximal input frequency. Thanks to the out of the baseband sampling, no front-end downconversion is necessary. On the other hand, a band-pass anti-aliasing filter is necessary instead of common low-pass one. The band-pass filter has been realized as an active 8^{th} order Chebyshev type I, 1 dB ripple filter, built using operational amplifiers.

Input band $39 \ldots 69\,\mathrm{kHz}$ sampled at $f_s = 72\,\mathrm{kHz}$ is aliased (mirrored) as $33 \ldots 3\,\mathrm{kHz}$. High voltage DC bias of the receiving transducer is produced by a transformer, switched at f_s. The switching frequency and its harmonics are aliased at either zero frequency or $\frac{f_s}{2}$, where no useful signal is present.

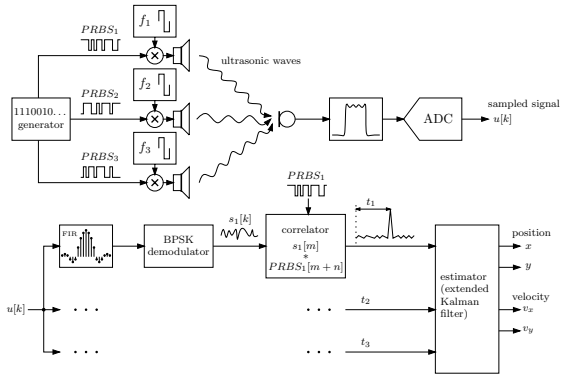

Fig. 2. System diagram and signal flow from beacons to receiver

The sampling and further signal processing at receiver side is by no means synchronized with the transmitter – there are two independent clocks, each driven by its own crystal oscillator.

Filtering. The sampled digital signal is then subdivided into three 6 kHz wide bands, centered around aliased carrier frequencies, $f_1' = 30\,\text{kHz}$, $f_2' = 18\,\text{kHz}$, $f_3' = 6\,\text{kHz}$. A 48^{th} order FIR filters have been designed using least-squares approximation of amplitude frequency response, including transition bands, symmetrically for all the three filters.

Filter coefficients have been quantized into $-127 \ldots 127$ range to allow efficient hardware or software implementation by an $8 \times n$-bit multiplier[6]. Thanks to band symmetries, for each two non-zero filter coefficients at k-th position, equation $b_1[k] = \pm 2^m b_2[k]$ holds for some $m \in \mathbb{Z}$. Moreover, for any linear-phase FIR filter, an even or odd symetry between coefficients holds.

As a result, only 19 multiplications and 73 additions or subtractions, accompanied by bit shifts, suffice to filter one sample by all three FIRs. On ARM7 CPU, the whole filtering period including memory transfers and wait states takes 350 CPU cycles. This gives a 60 MHz microcontroller 42% load at $f_s = 72\,\text{kHz}$.

BPSK Demodulation. Each filtered signal is then BPSK demodulated. Since f_s is integral multiple of f_b, and delays are easy to implement in discrete-time domain, a differential BPSK demodulator has been used. Its (nonlinear) difference equation is $y[k] = x[k]x[k-M]$, where k is a time index, and $M = \frac{f_s}{f_b}$ (number of samples per PRBS bit). Demodulator should be followed by a low-pass filter to suppres upper frequency modulation products. A simple moving-average filter, ie. summing last M samples, has been used.

However, the differential demodulator does not output the same sequence as the modulated one. In effect, each two successive original bits were XORed to produce an output bit: $p[n] = q[n-1] \oplus q[n]$ ($q[n]$ being the original, modulated sequence, $p[n]$ the output, demodulated sequence, and n is the bit number modulo sequence length). Therefore, transmitted sequences have been altered to produce the Gold code PRBSes after differential demodulation.

Demodulated and roughly filtered signal is then decimated, because the effective bandwidth of a binary sequence in baseband is $1.5 \ldots 3\,\text{kHz}$. New sampling rate $f_{s_2} = \frac{f_s}{6}$, ie. 4 samples per bit[7].

Correlation. Demodulated and decimated signals enter the correlation process. After collecting one PRBS period, ie. $N = 4 \times 127$ samples (at this point, bits are oversampled), a peak in correlation between received and known transmitted sequence should reveal the time displacement.

[6] Eg. ARM7 CPU core contains a multiplier, which performs 8×32 bit multiplication in one cycle; the another advantage is, that 8-bit value fits into an immediate machine-code instrucion operand value, thus a memory access is saved.

[7] In theory, 2 samples per bit should be sufficient; faster rate has to compensate poor filtering properties of the moving-average filter.

Before correlation, the whole period of N samples is buffered and normalized to 8-bit range (software "automatic-gain-control"). Then, the correlation $R_{xy}[n]$ should be evaluated for all possible displacements $n = 0 \dots N-1$.

Computation of correlation for all $n = 0 \dots N-1$ following definition (1) involves N^2 multiplications of x and y and $\sim N^2$ additions. Since one of the signals is composed of ± 1, the multiplication reduces to conditional sign change of the second (integer) number. The correlator performs N^2 conditional sign changes and additions. In our application, $N = 4 \times (2^7 - 1) = 508$, so the correlator performs 258064 such operations in one period.

Of course, there exist much more efficient methods to compute the correlation, but they are defined only for sequences with too restrictive properties.

If both $x, y \in \{\pm 1\}$, the multiplication reduces to bitwise boolean XOR. Complexity remains $\mathcal{O}(N^2)$, but implementable in very fast and vectorized fashion on all architectures. However, due to large dynamic range of signals in our system (26 dB), the 1-bit instead of 8-bit quantisation can lead to unusable results.

The correlation can be also very efficiently computed using fast Fourier transform (FFT) algorithm in $\mathcal{O}(N \log N)$. However, the FFT is not applicable for prime-length sequences and almost useless for sequences of length, factorized into few primes. This is exactly the case of all Gold codes. In our case, the signal length is prime-factorized as $508 = 2 \times 2 \times 127$. We could change the sequence length to eg. $4 \times 2^7 = 512$, but the correlation properties of PRBSes would degrade.

Software implementation of the $\mathcal{O}(N^2)$ correlator took ~ 20 ms of CPU time on 400 MHz PowerPC 603e. Given the 42.333 ms PRBS period, the real-time CPU load is 47%. As an alternative, we have developed custom FPGA correlator core as a dedicated computational resource. The correlator has been implemented into Xilinx Virtex-4 XC4VFX12 FPGA. Running at 300 MHz clock, the correlation for $N = 508$ took 0.215 ms. Compared to PowerPC, the processing is done by the correlator FPGA core 93× faster. Moreover, the correlation is performed in parallel, so it does not spend any CPU time.

Interpolation. The correlator output $R_{xy}[n]$ should contain an accentuated peak above a noise floor. A position n_* of the peak is the time displacement, quantized to $\frac{1}{f_{s2}}$. To obtain finer resolution, $R_{xy}[n]$ samples are interpolated.

It follows from the Wiener-Khinchin theorem, that iff one of signals x, y is band-limited, then their correlation R_{xy} is band-limited as well. The received signal is band-limited, although in a non-ideal way. According to the sampling theorem, a continuous correlation function $R_{xy}(t)$ can be fully reconstructed from the sampled one, $R_{xy}[n]$, by ideal low-pass interpolation.

Position of $R_{xy}(t)$ maximum can be an arbitrary real value, t_*. Actually, the precision of t_* is limited by noise in $R_{xy}[n]$. Therefore, the ideal interpolation is fruitless[8], an approximate FIR low-pass interpolation using few samples around n_* is a relevant method.

[8] Although it is possible to find maximum of ideally interpolated $R_{xy}[n]$, eg. using Newton method.

Interpolating FIR filter has been designed using windowing method, spanning 5 neighbouring samples. Interpolating ratio $32:1$ gives resolution $\frac{D_{max}}{32N} = 0.9\,\text{mm}$, far better than precision of mechanical system parts, so there is no need of finer interpolation.

Each of the three interpolators produces the interpolated correlation peak position, t_*, denoted $t_{1\ldots3}$, at a sampling period $T = 42.333\,\text{ms}$. These values enter a position estimation (or calculation) process.

3 Position Estimation

3.1 Calculation vs. Estimation

The final task of the localization system is to convert measured times $t_{1\ldots3}$ to robot coordinates, $x_{1\ldots2}$. The times $t_{1\ldots3}$ are equal to distances[9] between robot and respective beacons, plus a transmitter-receiver clock offset, Δ:

$$t_i = \sqrt{(x_1 - x_{Bi})^2 + (x_2 - y_{Bi})^2} + \Delta\,, \tag{3}$$

where x_{Bi}, y_{Bi} are coordinates of i-th beacon. As mentioned in Sec. 2.3, both clocks are running independently on their own crystals, so they are in each measurement biased by some fraction of period, $\Delta \in [0, D_{max})$. The task can be solved by one of the following ways.

Direct Calculation. The system of three equations (3), $i = 1\ldots3$, is regular, containing three unknown variables, $x_{1\ldots2}$ and Δ. First, equations are subtracted to eliminate Δ – only a time differences, $t_3 - t_1, t_2 - t_1$, remain. The problem is called TDOA (time differences of arrivals) or hyperbolic navigation, since in geometric insight, it relies in intersection of two hyperbolas [4, 5].

The system yields a quadratic equation. One of the two solutions is incorrect. It may be often eliminated after substitution into original equations (3). Otherwise, if both of the solutions are feasible, the choice depends on an external knowledge about the robot.

Under an assumption of absolutely precise measurement of $t_{1\ldots3}$, the result is exact. However, it never holds in practice, since the measurements suffer from uncertainity (noise). This is the reason, why the stochastic methods give much more precise results, than the direct calculation.

In case of noisy measurement, the quadratic equation may have no real solution. It can be overriden by forcing its discriminant to zero.

Although the method does not provide the best possible precision, it can provide an initial guess for the following methods.

Static Stochastic Estimation. Another possibility, how to infer $x_{1\ldots2}$ from $t_{1\ldots3}$, is to minimize (numerically) some norm of the uncertainity in $x_{1\ldots2}$ estimate, eg. $\sigma^2_{x_1} + \sigma^2_{x_2}$, according to the error propagation law, given t_i variance, σ^2_t.

[9] In the following, we assume constant sound velocity c, therefore expressing all time quantities in units of length.

Such an optimization should give slightly better results, than the direct calculation. However, it still does not exploit a substantial property of the system: its dynamics.

Dynamic Stochastic Estimation. The dynamic estimator is supplied with a system model – besides the static information about measurement uncertainity (noise), the model contains information about evolution of variables (state) in time, including uncertainity of the evolution (process noise).

The static stochastic estimation can be regarded as a weighted sum of several contributions with different variances. The dynamic estimation then extends this idea: it weights not only currently measured values, but also past values. Due to the process noise, older values have increased covariance, and therefore lower weights than the recent values.

Roughly speaking, the state variables correspond to a memory (inertia) of the system. In our system, there are two distinct dynamic subsystems: robot motion dynamics, and clock dynamics (Sec. 3.2). The essential improvement in precision over the static estimation consists in exploiting the latter, since the process noise of crystal oscillators, producing Δ, is very low.

Systems that estimate Δ and then determine position by calculation or static estimation are known as TOA (times of arrivals), or spherical navigation systems [4, 5]. Such a case is analogous to oscillator phase-locked loop (PLL), however, the "phase" (Δ) differences are not evaluated explicitly, but they are contained in coumpound measurement, $t_{1\ldots3}$. However, the estimator may profit from motion dynamic model as well.

If a good estimate of Δ is known, the system of three equations (3) is overdetermined in effect. Therefore, the position can be still calculated, if one of the $t_{1\ldots3}$ measurements is missing. This is very important, because one of the beacons can be always occluded by a competing robot. The system should survive "locked" to Δ and its drift for several seconds, while one of the three measurements is missing.

The estimator can evaluate likelihood of individual measurements, thus it may provide simple means of outlier detection, caused eg. by beacon occlusion, or reception of a stronger reflected signal.

The estimator can also process data from other external sources (sensors and actuators), to perform a data-fusion. It can either improve precision, widen bandwidth, or provide additional estimated variables, such as a heading φ by cooperation with odometry.

3.2 System Model

Stochastic System. Design of dynamic estimator consists in choice of system model, and an algorithm. General model of nonlinear stochastic system is

$$\boldsymbol{x}[k+1] = \boldsymbol{f}(\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{v}[k]) \tag{4}$$

$$\boldsymbol{y}[k] = \boldsymbol{g}(\boldsymbol{x}[k], \boldsymbol{u}[k], \boldsymbol{e}[k]) \tag{5}$$

where k is a discrete time, $\boldsymbol{x}$ the system state, $\boldsymbol{u}$ a known input signal (where applicable, eg. motor actuation), $\boldsymbol{v}$ the process noise, $\boldsymbol{e}$ the measurement noise,

and $\boldsymbol{y}$ is a measurement (all values real and possibly vector). Difference equation (4) describes system dynamics, and $\boldsymbol{g}$ is an output function. The noises are specified by their stochastic properties, eg. as uncorrelated Gaussian white noises with zero mean and covariances $\mathcal{E}\{\boldsymbol{v}\boldsymbol{v}^T\} = \boldsymbol{Q}, \mathcal{E}\{\boldsymbol{e}\boldsymbol{e}^T\} = \boldsymbol{R}$.

In our design, we managed with a much less general and simpler model

$$\boldsymbol{x}[k+1] = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{v}[k] \tag{6}$$

$$\boldsymbol{y}[k] = \boldsymbol{c}(\boldsymbol{x}[k]) + \boldsymbol{e}[k] \tag{7}$$

– the only nonlinearity is $\boldsymbol{c}(\boldsymbol{x}[k])$, based on (3). The dynamics is linear, as well as contribution of both noises. No known system inputs are considered, all information is propagated through the measurement.

Integrator Model. An integrator is a simplistic model of 1st order system with infinite time constant. It expects, that $x[k]$ will be near previous $x[k-1]$, with uncertainity given by the process noise. In terms of (6), $\boldsymbol{x} = (x), \boldsymbol{A} = (1)$. If x is expected to move with almost constant velocity (or drift) v, it can be modelled by double integrator: $x[k] = x[k-1] + v[k-1], v[k] = v[k-1]$, or $\boldsymbol{x} = (x, v)^T, \boldsymbol{A} = \left(\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right)$.

The single or double integrator submodels have been used in our system for both clock and motion dynamics.

Clock Dynamics. Clock offset Δ modelled by single integrator expects, that clock drift $\Delta' = \frac{\partial \Delta}{\partial t}$ is small. Better is to use the double integrator, estimating Δ' along with Δ. Drift is then considered to be constant, varied by an unmeasured process noise (mostly temperature changes in reality).

Motion Dynamics. Precise motion dynamic model contains mechanical time constants given by mass, inertia, and frictio. Moreover, the motion dynamics often contains substantial kinematic nonlinearities (anisotropic friction of wheels, dependent on heading φ). The motor actuation signal may, or may not be included in the model. Obtaining model parameters may be difficult (system identification task).

On the other hand, the integrator dynamics offer simple, although suboptimal approach. It can not gain as good precision as a more realistic model. However, it describes well the situations, where acceleration is occasional and robot remains in unchanged motion most of the time. The double integrator motion models are popular, where a very little is known about object dynamics (computer vision, tracking, [6]).

Measurement Model. Ultrasonic measurement is modelled by the function $\boldsymbol{c}(\boldsymbol{x})$ with an additive noise $\boldsymbol{e}$. The noise represents uncertainity of correlator and interpolator output. Its variance is fixed.

Odometry Fusion. Cartesian coordinates $x_{1\ldots2}$ may be estimated from ultrasonic readout $t_{1\ldots3}$. To obtain an estimate of heading φ, we have added odometry, ie.

data from wheel incremental sensors, to the system. Such a data fusion of relative positioning (odometry) with an absolute one (ultrasonic system) is a common procedure in mobile robotics [7–9].

Essentially, the increments over fixed period of time represent wheel velocities, v_L, v_R. Obviously, a trajectory reconstructed from these velocities suffer from cumulative error, and it is up to the estimator to correct it using absolute measurements.

Very often the odometry is treated as a known actuation signal $\boldsymbol{u}$ to the system, and the absolute positioning information is treated as a measurement $\boldsymbol{y}$, [7–9]. However, the local linearization (Sec. 3.3) of such a model with additive process noise is not observable – φ remains obscured by cumulative noise of odometry. Neither the transition matrix $\boldsymbol{A}$, nor an output $\boldsymbol{c}(\boldsymbol{x})$ provide coupling between φ and a rest of the state vector, therefore it can not be corrected by ultrasonic measurements.

To overcome this problem, we treat the odometry as a second measurement $\boldsymbol{y}_2 = \left(v_L, v_R\right)^T$, beside an ultrasonic measurement $\boldsymbol{y}_1 = \left(t_1, t_2, t_3\right)^T$, $\boldsymbol{y} = \left(\boldsymbol{y}_1^T \boldsymbol{y}_2^T\right)^T$. The system is considered to be actuated only by the process noise. This approach seems to be more realistic, as the odometry is treated as a velocity measurement with additive noise. Using this model, all the system states are observable, including φ and its derivative, angular velocity ω.

The final 8^{th} order model, based on $4\times$ double integrator dynamics, see Fig. 3, is:

$$\begin{aligned}
\boldsymbol{x} &= \left(x_1, v_1, x_2, v_2, \varphi, \omega, \Delta, \Delta'\right)^T \\
\boldsymbol{Q} &= \operatorname{diag}\left(0, \sigma_x^2, 0, \sigma_x^2, 0, \sigma_\omega^2, 0, \sigma_\Delta^2\right) \\
\boldsymbol{y} &= \left(t_1, t_2, t_3, v_L, v_R\right)^T \\
\boldsymbol{R} &= \operatorname{diag}\left(\sigma_t^2, \sigma_t^2, \sigma_t^2, \sigma_{LR}^2, \sigma_{LR}^2\right)
\end{aligned} \qquad \boldsymbol{A} = \begin{pmatrix}
1&1&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0\\
0&0&1&1&0&0&0&0\\
0&0&0&1&0&0&0&0\\
0&0&0&0&1&1&0&0\\
0&0&0&0&0&1&0&0\\
0&0&0&0&0&0&1&1\\
0&0&0&0&0&0&0&1
\end{pmatrix} \tag{8}$$

$$\boldsymbol{c}(x_1, x_2, \Delta, v_1, v_2, \varphi, \omega) = \left(c_{1\,1}, c_{1\,2}, c_{1\,3}, c_{2L}, c_{2\,R}\right)^T \tag{9}$$

$$c_{1i} = \sqrt{(x_1 - x_{Bi})^2 + (x_2 - y_{Bi})^2} + \Delta \tag{10}$$

$$c_{2L,R} = \frac{1}{2}\left(v_1 \cos\varphi + v_2 \sin\varphi \pm k_\omega \omega\right) \tag{11}$$

3.3 Estimation Algorithm

Nonlinear Estimation Methods. Estimation of a state of a linear system, ie. a system defined by (6, 7) with linear output $\boldsymbol{c}(\boldsymbol{x}) = \boldsymbol{C}\boldsymbol{x}$, is a straightforward task, solved by linear filtering. The linear estimator, performing filtering of data $\boldsymbol{u}[k]$, $\boldsymbol{y}[k]$ and producing $\boldsymbol{x}[k]$, is called observer in general, and the Kalman filter, if optimized with respect to $\boldsymbol{Q}, \boldsymbol{R}$ covariances. In case of nonlinear system, there are several approximate estimation methods, from which we have tried two: the particle filter, and the extended Kalman filter (EKF).

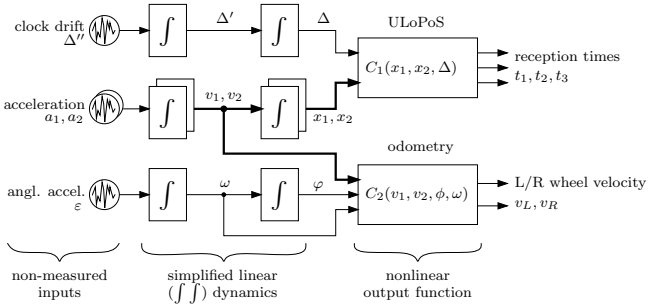

Fig. 3. Block diagram of 4× double integrator system model with both ultrasonic and odometric measurement

In both of the estimators, state is represented by a probability density function (PDF) $P(\boldsymbol{x})$. Measurement is represented by a likelihood function $L(\boldsymbol{x}|\boldsymbol{y}) = P(\boldsymbol{y}|\boldsymbol{x})$ (function of $\boldsymbol{x}$ for known $\boldsymbol{y}$). Within each sampling period, $P(\boldsymbol{x})$ evolves in two steps. In prediction step, $P(\boldsymbol{x})$ is transformed and convolved with $\boldsymbol{v}$ according to (4). After new measurement, corrective step arises. $P(\boldsymbol{x})$ is multiplied with $P(\boldsymbol{y}|\boldsymbol{x})$ and normalized, where $P(\boldsymbol{y}|\boldsymbol{x})$ is given by (5).

Particle Filter. Also known as Monte Carlo estimator [10], approximates $P(\boldsymbol{x})$ by a large set of randomized points (particles) in state space. In the prediction step, each particle is transformed by (4) using randomly generated $\boldsymbol{v}$. In the correction step, particles are weighted by $P(\boldsymbol{y}|\boldsymbol{x})$, then replicated and eliminated to form next $P(\boldsymbol{x})$.

Particle filter provides global PDF approximation. Therefore, it does not need intial guess in situations with large prior uncertainity (startup, error states). Also, it allows to preserve any non-Gaussian, even multimodal PDFs, what may slightly help during erroneous measurements.

The major disadvantage of the particle filter is its computational exigency. While processing real data by 4^{th} order estimator, the number of particles required for smooth filtering was $\sim 10^4$. It has been impossible to reach real-time estimation using C language implementation on embedded computers available in our project (400 MHz PowerPC 603e). Although the estimation ran only offline, it has been useful during system design. The particle filter has been implemented according to [10] (Sec. 4.1, Algorithm 1), applied to models with 1^{st} order motion dynamics together with 1^{st} or 2^{nd} order clock dynamics.

Extended Kalman Filter. EKF [11], is based on local linearization. $P(\boldsymbol{x})$ is considered to be Gaussian, desribed by mean $\boldsymbol{x}$ and covariance matrix $\boldsymbol{P}$. Equations (4, 5) are linearized in point $\boldsymbol{x} = \boldsymbol{x}[k]$, $\boldsymbol{u} = \boldsymbol{u}[k]$, $\boldsymbol{v} = \boldsymbol{e} = 0$. The resulting linear system is used to estimate next $\boldsymbol{x}[k+1]$, $\boldsymbol{P}[k+1]$, like in an ordinary linear Kalman filter. Gaussian variances are assumed to be small enough with respect to local linearization.

Since the approximation is only local, it requires a feasible initial state estimate to work properly. Otherwise, convergence is not guaranteed.

EKF can not cope with arbitrary PDFs. This does not really matter in our system, as during proper operation, all measurement PDFs intersect together in one narrow peak. An ambiguous multimodal PDF can occur, when some measurements are missing (because considered faulty), or when a fusion with ambiguous sensory data (eg. local playground color) would be requested.

To eliminate both of the drawbacks mentioned, the estimator is equipped with error recovery procedure (see below).

Algorithm of EKF used in frame of our system follows. The local linearization of the model (6, 7) consists in linearization of output function (9):

$$\boldsymbol{y}[k] \approx \boldsymbol{c}(\hat{\boldsymbol{x}}) + \boldsymbol{C}(\hat{\boldsymbol{x}})(\boldsymbol{x} - \hat{\boldsymbol{x}}) + \boldsymbol{e}\,, \qquad \boldsymbol{C}(\hat{\boldsymbol{x}}) = \left.\frac{\partial \boldsymbol{c}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}} \tag{12}$$

where $\hat{\boldsymbol{x}}$ is a current state estimate (linearization point), and $\boldsymbol{C}(\hat{\boldsymbol{x}})$ is linearized output matrix. The prediction step follows linear dynamics:

$$\boldsymbol{x}'[k] = \boldsymbol{A}\boldsymbol{x}[k-1]\,, \qquad \boldsymbol{P}'[k] = \boldsymbol{A}\boldsymbol{P}[k-1]\boldsymbol{A}^T + \boldsymbol{Q} \tag{13}$$

Then, the correction step using $\boldsymbol{C}(\boldsymbol{x}'[k])$ is performed:

$$\boldsymbol{C}[k] = \boldsymbol{C}(\boldsymbol{x}'[k])\,, \qquad \boldsymbol{M}[k] = \boldsymbol{C}[k]\boldsymbol{P}'[k]\boldsymbol{C}^T[k] + \boldsymbol{R} \tag{14}$$

$$\boldsymbol{L}[k] = \boldsymbol{P}'[k]\boldsymbol{C}^T[k]\boldsymbol{M}^{-1}[k] \tag{15}$$

$$\boldsymbol{x}[k] = \boldsymbol{x}'[k] + \boldsymbol{L}[k](\boldsymbol{y}[k] - \boldsymbol{c}(\boldsymbol{x}'[k]))\,, \qquad \boldsymbol{P}[k] = \boldsymbol{P}'[k] - \boldsymbol{L}[k]\boldsymbol{M}[k]\boldsymbol{L}^T[k] \tag{16}$$

First, $P(\boldsymbol{x})$ mean $\boldsymbol{x}$ is shifted by the transition matrix, then, covariance $\boldsymbol{P}$ is transformed according to the error propagation law, and widened by the process noise in (13). The optimal Kalman gain matrix $\boldsymbol{L}$ is calculated in (15). A difference between measured output $\boldsymbol{y}$ and modelled output $\boldsymbol{c}(\boldsymbol{x}')$ is multiplied by the Kalman gain, and added as a correction to $\boldsymbol{x}$ in (16), and $\boldsymbol{P}$ is narrowed by information contribution of the measurement.

EKF has been implemented in C language, floating-point math. Inversion involved in (15) is calculated using Cholesky factorization, since $\boldsymbol{M} = \boldsymbol{M}^T, \boldsymbol{M} > 0$. There are two operations, vulnerable to irregular input data: square-root, and division. They have been protected against negative values, and near-zero divisor, respectively.

Application Specific Details. EKF must be provided with feasible initial state. Therefore, the initial state is set to $\boldsymbol{x}[0] = (x_1, 0, x_2, 0, 0, 0, \Delta, 0)^T$, where x_1, x_2, Δ are the closed-form solution of TDOA (Sec. 3.1).

The ultrasonic measurement subsystem does not provide self-reliant detection of erroneous measurements. Actually, it is almost impossible to distinguish between correct and false measurement, eg. in case of direct signal path occlusion and reception of a reflected beam. Therefore, the error detection relies on the estimator. Obviously, it is not able to reveal all possible errors, because some

of the erroneous values may lie close enough to correct ones, and confuse the detector.

Likelihood of individual i-th measurement $P(y_i|\boldsymbol{x})$ can be evaluated before the correction step. A similar, yet more simply computable measure, excentricity ε, has been used to judge faulty measurements. In the correction step (16), contribution of i-th measurement y_i to the state is $\Delta\boldsymbol{x} = \Delta y_i \boldsymbol{l}_i$, where Δy_i is difference between measured and predicted y_i, and $\boldsymbol{l}_i$ is i-th column vector of $\boldsymbol{L}[k]$. ε is a ratio of $|\Delta\boldsymbol{x}|$ to standard deviation of $\boldsymbol{x}$ in direction of $\Delta\boldsymbol{x}$:

$$\varepsilon^2 = \Delta y_i^2 \boldsymbol{l}_i^T \boldsymbol{P}^{-1}[k] \boldsymbol{l}_i \tag{17}$$

If ε is greater than specified threshold ($\varepsilon^2 > \varepsilon_{thr}^2$, for eg. $\varepsilon_{thr} = 30$), the measurement is considered faulty (too unlikely). Corresponding column $\boldsymbol{l}_i$ is reset to zero, to protect state vector against damage.

While only one of the three ultrasonic measurements is considered faulty, the system should continue operation. Position estimation should manage with estimated clock offset and drift. It may be advisable to set temporarily the clock process noise σ_Δ^2 to zero.

If two or more measurements are evaluated as faulty, the system may get lost (diverge) due to accumulated errors. When the system detects more than N_{lost} successive iterations with more than one faulty measurements, it declares being lost, and tries to reinitialize next step with a new initial state.

4 Results

System Performance. Errors occured, when ultrasonic signal reflections were stronger than signal received by direct path. If the reflections were weaker, then they made no problem, since corresponding correlation peaks are lower. We have not been able to measure and adjust directional characteristics of the transducers to eliminate the problem entirely. In our experimental room, there sometimes occured stronger reflections from plastic files on a shelf, too near to playground boundary. A woolen sweater cast onto the files solved the problem.

In some areas, occasional noisy measurements occured, causing temporary deviations of measured position of $\sim 1 \ldots 10$ cm. It may have been incurred by slightly broken beacon mechanics. But most of the time, the measured position or trajectory were apparently clean.

Precision during robot movement has not been evaluated yet, due to lack of a suitable reference positioning system.

Position measured at stationary point shown standard deviations $\hat{\sigma}_{x_1} = 1.8$ mm, $\hat{\sigma}_{x_2} = 4.4$ mm. Compared to manually measured position, maximal error was 17.6 mm and mean bias 4.5 mm during 84 samples long measurement. It should be pointed out, that mounting precision of mechanical parts (beacon holders) has been $\sim \pm 10 \ldots 25$ mm. Real conditions at a competition may be similar.

Conclusion. In stationary measurements, the precision is in order of mounting precision of playground parts. The immediate following step is to close a positional feedback to robot motion control along a specified trajectory.

Next, the system precision should be evaluated using camera system, and maybe also a pencil-paper trajectory recording.

In the future, we may try to search another type of transducers or reflectors, to provide better and easier beam alignment.

Work supported by the Ministry of Education of the Czech Republic, project 1M0567.

References

1. Eurobot: EUROBOT 2009: Temples of Atlantis (2009), http://www.eurobot.org/
2. Chudoba, J., Přeučil, L.: Localization using ultrasonic beacons. In: 3rd International Congress on Mechatronics 2004, Praha, ČVUT v Praze, FEL (2004)
3. Linnartz, J.P.M.G.: Wireless Communication. Kluwer Academic Publishers, Dordrecht, http://www.wireless.per.nl/reference/chaptr05/cdma/codes/gold.htm
4. Linde, H.: On Aspects of Indoor Localization. PhD thesis, Fakultät für Elektro- und Informationstechnik, Universität Dortmund (August 2006)
5. Jiménez, A.R., Seco, F., Ceres, R., Calderón, L.: Absolute localization using active beacons: A survey and IAI-CSIC contributions. White paper, Instituto de Automática Industrial – CSIC, Madrid, Spain (2004)
6. Davison, A., Reid, I., Molton, N., Stasse, O.: MonoSLAM: Real-Time Single Camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(6), 1052–1067 (2007)
7. Fox, D., Burgard, W., Dellaert, F., Thrun, S.: Monte Carlo localization: Efficient position estimation for mobile robots. In: Proc. of the National Conference on Artificial Intelligence (1999)
8. Winkler, Z.: Barbora: Creating mobile robotic platform. In: MIS 2003. Matfyzpress (2003)
9. Köse, H., Akın, H.L.: The reverse Monte Carlo localization algorithm. Robotics and Autonomous Systems 55(6), 480–489 (2007)
10. Carpenter, J., Clifford, P., Fearnhead, P.: Improved particle filter for nonlinear problems. IEE Proceedings Radar, Sonar and Navigation 146(1), 2–7 (1999)
11. Ljung, L.: Asymptotic behavior of the extended Kalman filter as a parameter estimator for linear systems. IEEE Transactions on Automatic Control 24(1), 36–50 (1979)