IT++

C++ software library in NEWCOM

Zbigniew Długaszewski
Poznań University of Technology
IT++

• originates from the former department of Information Theory at Chalmers University of Technology, Sweden
• developed by researchers\(^1\) and used by researchers both in the communications industry and at Universities

\(^1\) project administrator is Tony Ottosson, Chalmers University
What is IT++

• a library of routines and modules for simulation of communication systems and signal processing
• uses existing open source mathematical libraries for increased functionality, speed and accuracy
• released under the GNU GPL license
IT++ platforms

- works on GNU/Linux, Sun Solaris, Cygwin,
- Windows 2000/NT/XP (with Microsoft Visual C++ .Net), Mac OS X
IT++ properties

• fast
• modular
• fairly well documented (using Doxygen)
• highly influenced by Matlab (naming and features: vectors, matrices, functions)
Welcome to IT++!

release-3-8-0

Introduction

IT++ is a C++ library of mathematical, signal processing, speech processing, and communications classes and functions. It has been developed by researchers in these areas and is widely used by researchers both in the communications industry and at Universities. Since 2004 IT++ is also being developed as a part of the European network of excellence NEWCOM.

The kernel of IT++ are templated vector and matrix classes and lots of functions for vectors and matrices. As such the kernel is similar to the Matlab functions. IT++ makes extensive use of existing open source libraries for increased functionality, speed and accuracy. In particular the libraries ATLAS (or CBLAS), LAPACK, and FFTW are used. For more details about these packages and how to install them see the installation section.

The IT++ library originates from the former department of Information Theory at Chalmers University of Technology, Gothenburg, Sweden. Since the library is coded in C++ the name of the library seemed like a good idea at the time. While departments come and go, IT++ have developed a life of its own and is now released under the GNU GPL license for you to enjoy.

IT++ works on GNU/Linux, Sun Solaris, Cygwin, Windows 2000/NT/XP (with Microsoft Visual C++ .Net (more testers needed)), and Mac OS X (more testers needed).
IT++ kernel

- templated vector and matrix classes
- functions for vectors and matrices
- open source mathematical functions
Predefined data types

- standard C++ types
- complex <double>
- bin: binary (0,1) data
- vector types
- matrix types
- Array class
Vector types

• a vector can in principle be of arbitrary type (that support addition, subtraction, multiplication and division), since the general vector class `Vec<TYPE>` is templated

• the predefined vector types are:
  – vec: basic vector type containing double
  – cvec, ivec, bvec, svec
Matrix types

• similarly to vector classes, the general matrix template class is \texttt{Mat<TYPE>}

• the predefined matrix types are:
  – mat: basic matrix type containing double
  – cmat, imat, bmat, smat
External libraries

• BLAS - Basic Linear Algebra Subprograms
• ATLAS - Automatically Tuned Linear Algebra Software
• LAPACK - Linear Algebra PACKage
• FFTW
# CVS repository

**cvs: itpp/it++/src**


Files shown: 1

<table>
<thead>
<tr>
<th>File</th>
<th>Rev.</th>
<th>Age</th>
<th>Author</th>
<th>Last log entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>base/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>comm/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fixedpoint/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>graphics/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>matlab/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>protocol/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>srccode/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Makefile</td>
<td>1.57</td>
<td>8 days</td>
<td>tonyottosson</td>
<td>Small fixes</td>
</tr>
</tbody>
</table>

Show files using tag: Select Branch: [ ]

Back to SourceForge.net  |  ViewCVS and CVS Help
Powered by ViewCVS 1.0-dev
IT++ classes

• sources, random generators,
• channels,
• modulators,
• encoders and decoders,
• filters and many more ...
# IT++ Class Index

### Alphabetical List

| A       | B       | C       | D       | E       | F       | G       | H       | I       | J       | K       | L       | M       | N       | O       | P       | Q       | R       | S       | T       | U       | V       | W       |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| AR1_Normal_RNG (itpp) | AR_Filter (itpp) | Array (itpp) | Audio_File (itpp) | AWGN_Channel (itpp) | B       | BCH (itpp) | BERC (itpp) | Bernoulli_RNG (itpp) | bfstream (itpp) | bfstream_base (itpp) | bfstream (itpp) | bin (itpp) | BLERC (itpp) | Block_Interleaver (itpp) | bofstream (itpp) | BPSK (itpp) | BSC (itpp) | C       | CFix (itpp) | CFixed (itpp) | Extended_Golay (itpp) | F       | Factory (itpp) | Fading_Generator (itpp) | FastICA (itpp) | Filter (itpp) | FIR_Fading_Generator (itpp) | Fix (itpp) | Fix_Base (itpp) | Fix_Factory (itpp) | Fixed (itpp) | G       | GF (itpp) | GFX (itpp) | GMM (itpp) | Gold (itpp) | Hamming_Code (itpp) | I       | I_Uniform_RNG (itpp) | IFFT_Fading_Generator (itpp) | Impulse_Source (itpp) | M       | MA_Filter (itpp) | Mat (itpp) | Modulator (itpp) | Modulator_1d (itpp) | Modulator_2d (itpp) | Multicode_Spread_1d (itpp) | Multicode_Spread_2d (itpp) | N       | Name (itpp) | Normal_RNG (itpp) | OPDM (itpp) | P       | PAM (itpp) | Parser (itpp) | Pattern_Source (itpp) | PlotPDF (itpp) | PSK (itpp) | Pulse_Shape (itpp) | Punctured_Convolutional_Code (itpp) | Q       | Rice_Fading_Generator (itpp) | Rice_RNG (itpp) | Root_Raised_Cosine (itpp) | S       | Sawtooth_Source (itpp) | Sequence_Interleaver (itpp) | Sine_Source (itpp) | SND_Format (itpp) | SND_In_File (itpp) | SND_IO_File (itpp) | SND_Out_File (itpp) | Sparse_Mat (itpp) | Sparse_Vec (itpp) | Spread_1d (itpp) | Spread_2d (itpp) | Square_Source (itpp) | Stack (itpp) | Stat (itpp) | T       | TDL_Channel (itpp) | Timer (itpp) | Triangle_Source (itpp) | }
Doxygen documentation

• Doxygen is a documentation system for C++, C, Java, ...

• can generate an on-line documentation browser (in HTML) and/or an off-line reference manual (in LaTeX) from a set of documented source files
HTML documentation example

itpp::BSC Class Reference

[Communication Channel Models]

A Binary Symmetric Channel with crossover probability p. More...

#include <channel.h>

List of all members.

Public Member Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSC (double in_g=0.0)</td>
<td>Class constructor. Sets the error probability to p</td>
</tr>
<tr>
<td>void set_prob (double in_p)</td>
<td>Set crossover (internal) probability.</td>
</tr>
<tr>
<td>double get_prob () const</td>
<td>Get crossover (bit error) probability.</td>
</tr>
<tr>
<td>bvec operator() (const bvec &amp;input)</td>
<td>Feed input through the BSC channel.</td>
</tr>
</tbody>
</table>
Detailed Description

A Binary Symmetric Channel with crossover probability \( p \).

Input and output are of type bvec with 0 and 1. Example:

```c
#include "itcomm.h"

int main() {
    //Initiate the BSC with cross-over probability 0.1
    BSC bsc(0.1);

    bvec transmitted_bits = randb(100);
    bvec received_bits = bsc(transmitted_bits);
}
```

Definition at line 743 of file channel.h.

The documentation for this class was generated from the following files:

- channel.h
- channel.cpp
Future goals

• IT++ as the main tool for simulation of communication systems
• physical, link, protocol layers, cellular environment
• new way of making abstractions (for a few users across layers)
Roadmap from version 3.8.0

- **3.8** will include minor changes and bug-fixes
- **3.9** (September 2005) support for protocol simulations (especially TCP/IP)
- **4.0** (June 2006) support for cellular environment simulations
Call for participation

• USE IT++!
• test it, find bugs
• report bugs or fix them
• improve and expand documentation
• help to port IT++ into new platforms
Support developers

• help to improve existing algorithms (e.g. faster implementation)
• propose new functionalities for existing classes
• expand existing code
• share your own code