Second Generation Dutch Pulsar Machine — PuMa-II

Ramesh Karuppusamy and Ben Stappers
Netherlands Foundation for Research in Astronomy (ASTRON),
P.O. Box 1, 7990 AA Dwingeloo, The Netherlands

Phone: +31 (0) 521 595 255 Fax: +31 (0) 521 595 408
E-mail: ramesh@astron.nl

Kees Slump
Signals and Systems Group, Electrical Engineering, University of Twente
P.O.Box 217, 7500 AE Enschede, Netherlands

Michiel van der Klis
Astronomical Institute "Anton Pannekoek" and Center for High Energy Astrophysics, University of Amsterdam, Kruislaan 403, NL-1098 SJ Amsterdam, Netherlands

Abstract—**The Second Generation Pulsar Machine (PuMa-II) is under development for the Westerbork Synthesis Radio Telescope. This is a summary of the system design and architecture. We show that state of the art pulsar research is possible with commercially available hardware components. This approach obviates the need to develop specialized custom hardware, decreasing project cost and time. PuMa-II will increase the sensitivity by a factor of $\sqrt{2}$ to 4, depending on the mode of operation. The project is an interesting combination of technology, computational astrophysics and astronomy. It is expected that the prototype to be tested by end 2004 will deliver excellent results. Once completed PuMa-II offers exciting possibilities in pulsar studies using the Westerbork Synthesis Radio Telescope.**

Keywords— **pulsar research; radio astronomy; signal processing**

I. INTRODUCTION

The ***W**esterbork* ***S**ynthesis* ***R**adio* ***T**elescope* (WSRT) will be equipped with the next generation ***Pul**sar* ***Ma**chine* (PuMa-II) capable of exploiting the full bandwidth offered by the WSRT frontends. The WSRT is a synthesis aperture instrument, capable of imaging the sky at various continuum and spectral line frequencies. The WSRT can be operated in a special mode to observe Pulsars using dedicated backends. PuMa-II is a pulsar backend receiver, and will be the successor of PuMa [1]. It is designed around low cost PCs in order to reduce the overall system cost. The system will handle a peak throughput of 640 Mbytes/second up to 12 hours, requiring a total of 27.6 Terabytes of storage space. In addition to storage of data at a sustained rate of 640 Mbytes/sec, we intend to develop an efficient coherent dedispersion algorithm that can be used on the pulsar data acquired by PuMa-II. A modular approach is adopted for the design and the final instrument will be made up of eight identical modules, each recording at 80 Mbytes/sec. This results from the 8-bit samples of two polarizations each of 20 MHz bandwidth from the added the WSRT signals.

Pulsar is an acronym for Pulsating Radio source. The pulsed radiation is emitted by a rapidly rotating neutron star and appears like an amplitude and phase modulated noise signal. The pulse rate can be anywhere from a few milliseconds to a few seconds, depending on the type of pulsar. Pulsar signals are smeared as the signal propagates through the interstellar medium, due to the dispersion effect of the tenuous plasma and interstellar charged particles. In radio astronomical observations, the minimum detectable signal is proportional to the bandwidth and integration time of the measurement. This is expressed as:

$$\Delta T_{\min} = \frac{K_s T_{sys}}{\sqrt{t\Delta n}} \qquad (1)$$

Where $\Delta T_{\min}$, is the minimum detectable signal, T_{sys} is the system noise temperature, K_s is a dimensionless proportionality constant, t is the integration time in seconds, and Δn is the observation bandwidth.

Pulsars signals are very weak radio sources and wide instantaneous bandwidths become necessary to improve the signal to noise ratio. Wide bandwidths are used for measuring all radio astronomical signals, as seen in expression (1). The effect of bandwidth and interstellar medium smearing the pulsar signal and can be described by the following equation:

$$\Delta t = \frac{8.3\times 10^{3} \times DM \times \Delta n}{n_{mid}^{3}} \qquad (2)$$

Where t, measured in seconds, can be many times the pulse period, DM is the integral of the number of electrons along the line of sight and is proportional to the distance of the pulsar from the observer; Δn is the observing bandwidth in MHz and n_{mid} is the center frequency in MHz of the band of observation.

Traditionally, splitting the observation band into smaller pieces using analogue filters reduced the smearing problem. However, most analogue filterbank systems consist of, at most, 256 filters. Moreover the filters are hard to make, expensive, not easily configurable, and still suffer from dispersion within the finest channel. It is also not possible to achieve a time resolution better than the inverse of the channel bandwidth.

Smaller, configurable filter bandwidths can be obtained if data is digitally stored, and processed. This was the approach used by the previous generation pulsar machine, PuMa [1] in the filter bank mode. However, the smearing dispersion in the finest channel still remains. Applying an inverse chirped filter to the digitized signal stored in computer as complex voltages can further reduce the effect of smearing. This process is called coherent dedispersion, and PuMa-II will use this as a standard mode of operation.

The present pulsar machine in the WSRT, PuMa, can handle a maximum of 80 MHz analog bandwidth in filterbank mode, and 10 MHz in baseband recording mode for offline coherent dedispersion. PuMa-II can handle 160 MHz of analog bandwidth giving a minimum sensitivity increase of $\sqrt{2}$ in filter bank mode. In the baseband recording mode, PuMa-II handles 16 times more bandwidth resulting sensitivity increase of four for coherent dedispersion. PuMa-II records data as 8-bit samples, unlike PuMa, which uses 2-bit samples. This increased resolution also shows up as improved sensitivity, and can used to reduce the effect of radio frequency interference (RFI) when RFI suppression algorithms are available. PuMa-II will be a fully WSRT-compatible backend, integrating seamlessly with the existing backend interfaces. An additional hardware component is being developed for this purpose is the PuMa-II interface Card (PiC).

A pulsar signal simulation is being prepared in the MATLAB environment. Various properties of the signal, such as frequency of observation, dispersion measure, and bandwidth can be controlled in this simulation. The artificial pulsar signal will be used to test various coherent dedispersion algorithms.

II. ARCHITECTURE OF PULSAR MACHINE

A. The WSRT Setup

The WSRT is an array consisting of 14 25-metre parabolic dishes in a 2.3-km East-West line. The facility can be used in “phased-array” mode for synthesis imaging of the sky, and a “Tied-Array” mode to simulate a single 94-m dish, by adding the signals from various telescopes in phase. In synthesis mode, a hardware Correlator is the workhorse of the observatory. For pulsar research, the WSRT is used in “Tied-Array” mode, with a pulsar backend receiver. The resulting 94-m dish signal equivalent improves the sensitivity by a factor of ~16, when compared to a single 25-m dish.

A simplified block diagram of the relevant sections of the WSRT is shown in figure 1. The signals from the telescope frontends are 160 MHz each centered at the frequency of observation. This band is converted to eight video bands (option from 156 KHz to 20 MHz in steps of 2^n) by the IVC (Intermediate to Video Frequency

Converter[1]) for each telescope giving a total of 2 (polarizations) X 16 (telescopes) X 8 (subbands) = 256 outputs. This signal is then sent to the Analogue-to-Digital Converter (ADC) subsystem, where each IVC output is digitized at 2-bit resolution and sampled at 40 MHz. Outputs of the digitizer can be directly fed to the Data Distributor Unit (DDU), and/or can be sent to the Tied Array Adder Module (TAAM).

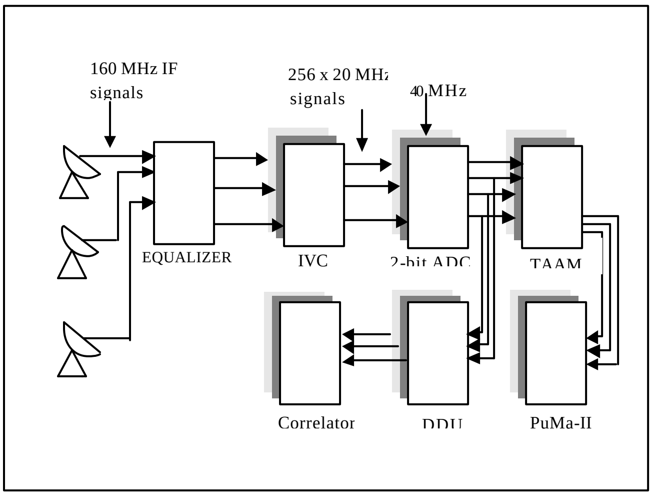

Figure 1 Block diagram of the WSRT

The PuMa-II accepts inputs directly from TAAM, where 2-bit digitized signals are added and scaled telescope-wise resulting in 8-bit values. The DDU is a switching matrix that supplies data to the correlator units for synthesis imaging measurements.

The digital data are transported via optical links among various subsystems (ADC, DDU and Correlator). The effective data rate is 960 Mbits/sec. The TAAM has eight serial outputs, where each output is the two polarizations of 20 MHz added signals.

A one-pulse-per-10-second (1 pp10s) synchronization signal (not shown in figure 1) is distributed to all subsystems, and it is used to timestamp the data samples accurately. This signal, called SYSTICK, is derived from the local Hydrogen Maser standard, and all clocks in various subsystems are referenced to this standard. A finer grained variety of SYSTICK is called the BOCF and that is used inside the hardware subsystems. This serves as a local data marker. All synchronization is based on SYSTICK and BOCF.

B. *PuMa-II System Design*

Figure 2 shows the basic building blocks of PuMa-II. Each module handles the two polarizations of 20 MHz analog bands and to record the serial data stream of 640 Mbits/sec, a minimum continuous write speed of 80 Mbytes/sec is needed. If this data rate is to be sustained up to 12-hours of observation, we need 3.4 Terabytes of storage space. With these requirements, each PuMa-II module is chosen to consist of a powerful desktop PC equipped with high performance PCI buses, a high-speed DMA (***D**irect **M**emory **A**ccess*) card, a PuMa-II Interface card (PiC), and two fast RAID (***R**edundant **A**rray of **I**ndependent **D**isks*) controllers connected to a 4 Terabyte-RAID disk pack.

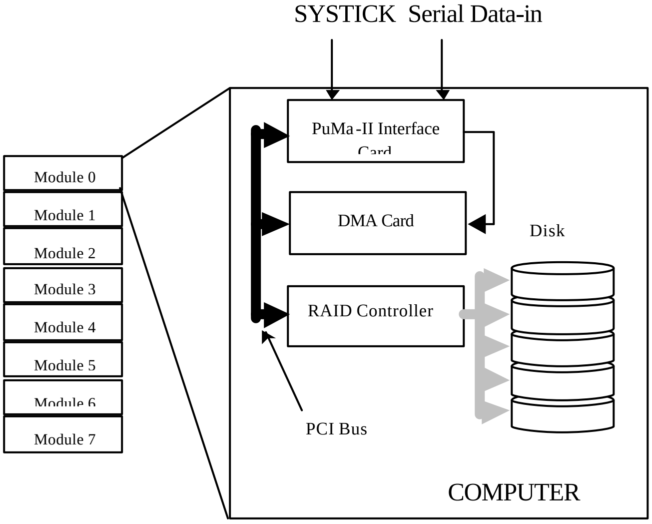

Figure 2 PuMa-II Block Diagram

The PiC is built to de-multiplex the serial data, and to synchronize the output to SYSTICK. Synchronized data is then sent as LVDS (***L**ow **V**oltage **D**ifferential **S**ignaling*) signals to the DMA card, along with the clock signal. LVDS is the industrial standard for high-speed digital design, and it is chosen here to transport signals across cards. The DMA card is from EDT Inc., USA. This is a high performance PCI add-on board, capable of 210 Mbytes throughput across the PCI Bus. The card comes with a Linux device driver [2].

The acquired data can be pumped via a gigabit network to a cluster of computers for further processing.

---

[1] In radio astronomical jargon, video frequency refers to the baseband signal down-converted from the intermediate frequency.

An additional flexibility of PuMa-II is that, a digitizer daughter board on the PiC, along with a synchronizing signal, enables PuMa-II to be placed in any observatory capable of providing analog outputs.

The software per module consists of a shared memory segment, and three concurrent threads. A fourth thread can be scheduled, if some processing is needed. The three core threads are tied to data acquisition, data writing and monitoring tasks. All codes are developed using the C programming language, while data processing is handled by C++ codes. A standard SuSE Linux distribution will be used as the operating system. The PiC has a device driver, and some rudimentary data monitoring is done via the driver.

C. *PuMa-II Interface Card (PiC)*

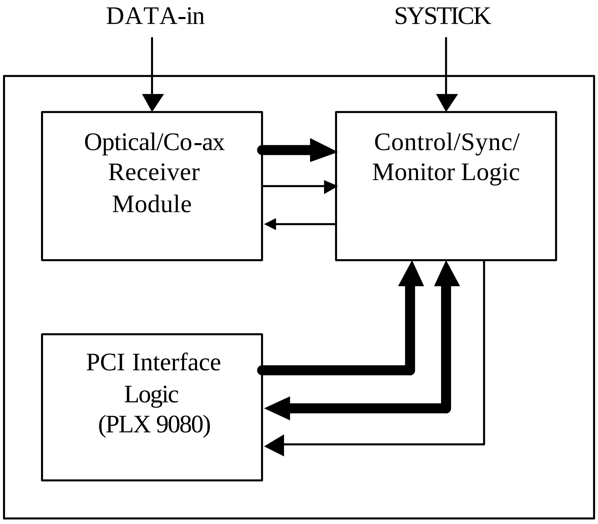

Figure 3 PuMa-II interface card details.

The PiC will be the hardware component that connects PuMa-II to the rest of the WSRT. It is built around the PCI 9080 interface chip from PLX Technologies, and Altera ACEX FPGA. The card has a daughter board to de-multiplex serial data from the TAAM (Figure 1). The output of serial receiver module is a 16-bit de-multiplexed data, a clock and the local marker signal. The marker signal (BOCF) and SYSTICK are used to synchronize the data. Board control, error detection, synchronization and status monitoring are basic tasks done in the FPGA. The PCI interface is used to communicate with the upper layer application software. The PiC is realized in an 8-layered printed circuit board.

In Figure 3, the PiC is shown in a block diagram. The optical receiver units are identical to the receiver units found on other subsystems like the ADC/DDU and Correlators. The control signals to this module are generated by the FPGA. The error and status of the serial receiver module is also handled here. The FPGA is connected to the PLX 9080 interface chip by a local de-multiplexed 16-bit Address/Data bus. A Few more control signals from the PLX 9080 are used to generate interrupts to the PCI bus, and to send the status/monitor information to the software layer above.

D. *Software*

The application software relies on the driver of the DMA card for data acquisition, and uses the PiC driver to start and stop a measurement. A manual start command, or a remote start command is issued to the application. Once a start is issued, the application instructs the PiC to synchronize and raise an interrupt at the next SYSTICK. When the interrupt arrives, the application reads the UTC (Universal Coordinated Time) from one of the network timeservers, and time read is rounded off to the nearest 10-second. This value will be used to timestamp the data acquired. When starting the PiC, the DMA card is also armed to acquire at the arrival of a dedicated signal, IDV (generated by the PiC, after synchronization). The DMA card driver acquires data in a ring buffer maintained in the Linux kernel memory space. The application copies data from kernel buffers to a larger ring buffer in the computer's main memory. This data is then written to the hard disk by a second thread. The files are written in the order it was acquired. Each file is 800 Mbytes in size corresponding to 10 seconds of data. The acquire-write process continues till a stop is issued, or till the specified observation duration is elapsed. An open-source variant of Silicon Graphics' XFS filesystem [6] with realtime subvolume option is used to achieve fast write-performance to the physical surface of the hard disk. The filesystem has journaling capability and is now is a part of the standard Linux Kernel distribution. We have noticed raw write speeds of 110 Mbytes/sec to the disk.

III. SIGNAL PROCESSING REQUIREMENTS IN PUMA-II

All radio signals propagating through the interstellar medium are affected by dispersion and scattering. Dispersion can only be measured with time varying signals, like those from pulsars and is similar to the effect of light rays passing through a prism – the lower frequencies travel slower than the higher frequencies. Dispersion distorts pulsar signals and effectively limits time resolution. Figure 4 show a typical frequency-time plot of the Vela Pulsar (PSR B0835-45). In this plot the x-axis range is the pulsar rotation period, represented as pulse phase.

We see a band across the plot, and this is the result of pulses arriving earlier at higher frequencies when compared to pulses at lower frequencies. This is the typical signature of a pulsar that differentiates it from any terrestrial impulse like signals. One can clearly see that summing up the all signals in this bandwidth in order to improve signal to noise ratio, it is necessary to correct for the delays.

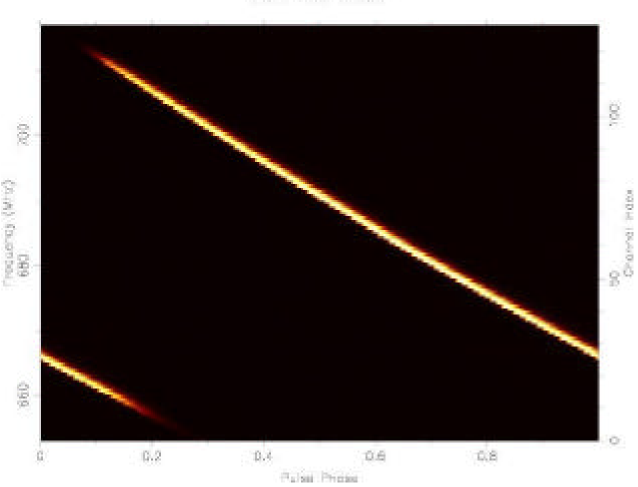

Figure 4 Frequency-Time plot of the Vela Pulsar

Dispersion removal can be done before or after detection of pulsar signals (known as incoherent or coherent dedispersion, respectively). To incoherently dedisperse pulsar data, the receiver band is divided into smaller channels (e.g. using a filterbank), and each channel is delayed appropriately before summing up the channels. The disadvantage in this method is the finite width of the channel, as this always limits the time resolution achievable.

In the process of coherent dedispersion [3], the received signal is passed through a digital filter having an inverse transfer function of the interstellar medium. In this case the time resolution attained is limited only by the inverse of the sampled bandwidth, and it is 25ns in our case. The signal is sampled and stored as complex voltages before applying the filter. In figure 5, Mode 0 is the data that is coherently dedispersed. In figure 5 dedispersed profiles of the fastest known pulsar, B1937+21 are compared. In addition to improved signal to noise ratio, the "sharpness" of profiles is significantly better.

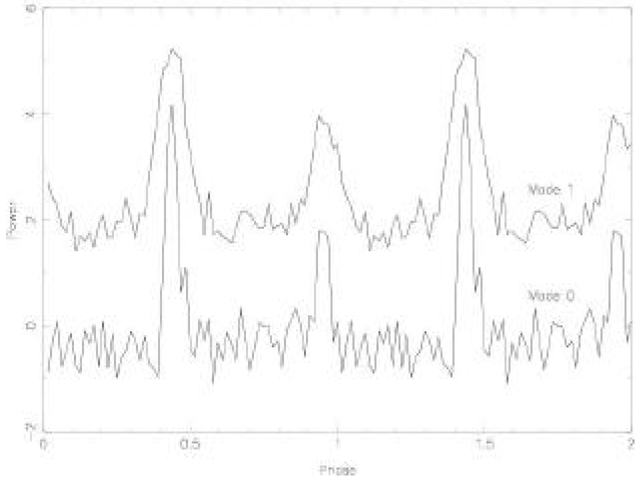

Figure 5 Comparison of Coherent (Mode 0) and incoherent (Mode 1) dedispersion for PSR B1937+21 at 800 MHz, 10 MHz observation bandwidth.

The FFT algorithm is employed to digitally filter pulsar data. If X_k and H_k represent the Fourier transforms of a sequence of data stream and the impulse response of the dedispersion filter, then the product $Y_k = X_k H_k$ is a convolution of the data with the filter response function. The inverse Fourier transform of Y_k will the dedispersed data. The length of the FFT size (chosen to be a power of 2) is calculated using the dispersion measure of a pulsar. Table 1 shows this for a few pulsars at 400 MHz, and a 20 MHz of observation bandwidth. Note the usable points column. Due to the cyclic nature of convolution through FFT, both edges of the convolved data are polluted by wrap-around effect [5]. The usable points are the number of points that are not polluted by this effect. This number can be small depending on the pulsar and its dispersion measure. Hence a proper FFT length should be chosen to improve computational efficiency.

TABLE 1 FFT SIZES FOR A FEW PULSARS

Pulsar	Dispersion Measure ($cm^{-3}pc$)	Pulse Period (s)	FFT size	Usable points
PSR0531+21	56.7	0.0331	8388608	2499506
PSR1937+21	71.0	0.0015	8388608	1014247
PSR0329+54	26.8	0.7145	4194304	1410742
PSR1154-62	324.2	0.4005	67108864	33436081

Pulsars are like giant flywheels, with very precise period of rotation. Timing study of pulsars is an important area of pulsar research. The changes in pulse period can used to calculate the age of a pulsar. If timing data is available for at least a year, then the position and proper motion of a pulsar can be computed. In the case of pulses from binary systems, timing study provides rich information on the gravitational field in the binary system. The precise timing information from these signals highlights the need to timestamp the samples accurately. And a good signal to noise ratio is a pre requisite for such studies. Coherent dedisperion is the technique used to improve the SNR of pulsar signals.

IV. SIMULATION

Pulsar signals can be seen as amplitude and phase modulated noise signals. In a plot of time-frequency diagram, the pulse appears to marching off across the frequency channels. This is the signature of pulsar, and the effect is called dispersion. A sample pulsar signal is simulated in MATLAB. A sequence of random numbers is generated, whose length is based on the center frequency, bandwidth and dispersion measure. A square pulse or a pulse profile modulates the generated sequence. This will be the amplitude modulation of the signal. Convolving with the dispersion function disperses the signal, and this is phase modulation. Both these modulations model the effect of the interstellar medium, whose transfer function can be expressed as,

$$H(n+n_0)=\exp\left[-\frac{i2pDn^2}{n_0^2(n_0+n)}\right] \qquad (3)$$

Where, n_0 is the center frequency, D is dispersion, and is related to Dispersion Measure by the relation, $DM = 2.41\times 10^{-16}\times D$.

The dispersed signal will be generated in the method similar to the coherent dedispersion technique outlined in the previous section using the FFT algorithm. This signal will be representative of the signal received by a telescope on earth. The generated data will be used to test the dedispersion algorithms.

V. CONCLUSION

We sum up the results of current research as follows:

i) It is possible to build sophisticated state-of-art pulsar machine with off shelf hardware components, minimal custom hardware and at a comparatively lower cost.

ii) When PuMa-II is operational, it will be one of the pulsar machine capable of coherent dedispersion with largest bandwidth available in the world.

The PuMa-II project and one of the authors' [RK] PhD program are funded by Nederlandse Onderzoekschool voor Astronomie (NOVA) [7].

REFERENCES

[1] J. L. L. Voute *et al.*, "*PuMa, a Digital Pulsar Machine*", Astronomy&Astrophysics, 2002, pp.385-733

[2] PC-CDa and PC-CD (High Speed DMA Interface) User's Guide

[3] T.H. Hankins & B.J. Rickett, "*Pulsar Signal Processing*", Methods in Computational Physics, pp. 64 -81

[4] J.D. Kraus, "*Radio Astronomy*", 2nd Edition, Cygnus-Quasar Books.

[5] W. H. Press *et al.*, "*Convolution and Deconvolution using FFT*", Numerical Recipes in C: The Art of Scientific Computing, pp.538-545

[6] Silicon Graphics open source home page: http://oss.sgi.com/projects/xfs/index.html

[7] Information on NOVA: http://www.astronomy.nl