Approaches to Improving Acoustic Communications on Autonomous Mobile Marine Platforms

Toby Schneider and Henrik Schmidt
Center for Ocean Engineering
Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA 02139
tes@mit.edu and henrik@mit.edu

***Abstract*—Autonomous underwater vehicles (AUVs) are quickly advancing in adaptive sensing and feature detection capabilities. These missions increasingly require timely transmission of data over the only feasible undersea link: acoustics. Thus far, the underwater communications and autonomous vehicles research communities have pursued their problems more or less independently. This paper reviews and presents a number of techniques for integrating the two disciplines by applying the artificial intelligence available on the AUV to the problem of improving communications over the high-latency, low-throughput acoustic link.**

These techniques are split into two broad categories: 1) Disruptive: maneuvers and autonomous decisions that require some disruption to the broader vehicle mission, and 2) Non-disruptive: methods that do not affect the motion of the vehicle. In the non-disruptive category, an application of arithmetic encoding to substantially reduce the telegram size of a vehicle's position report is presented.

***Index Terms*—autonomous underwater vehicles, acoustic communications, robotic networks**

I. Motivation

Users of mobile marine platforms such as autonomous underwater vehicles (AUVs) and unmanned surface vehicles (USVs) are one of the major beneficiaries of improved acoustic communication capabilities, since the need to move often precludes the use of fiber optic communication tethers. These vehicles are also becoming increasingly "intelligent"; they are outfitted with substantial computational ability and are capable of fulfilling complex mission components or entire missions autonomously; for examples, see [1]–[3]. Meanwhile, underwater acousticians and signal processing researchers have characterized many of the detrimental effects of the ocean acoustic environment on successful transmission of datagrams; the effects are summarized in [4], [5]. However, there has been little crossover between underwater autonomy and acoustic communications, with the former community generally treating the physical link as a "black box" that sends bytes from one point to another.

This work was funded by the Office of Naval Research (ONR) under projects N00014-08-1-0011 and N00014-11-1-0097.

This paper aims to show that much can be gained from improved awareness of a physical acoustic link and its related hardware by the AUV's decision making software. Several techniques are reviewed or suggested for improving communications by making use of the vehicle's intelligence; they are split into disruptive (requires movement of the vehicle that may be orthogonal to the overall mission objectives) and non-disruptive (no negative effect on the mission objectives). Here, "improving communications" is defined as increasing the unit information throughput per unit power ratio. A summary of the methods discussed here is given in Table I.

II. Disruptive Techniques

Since, by definition, AUVs are mobile, the possibility exists for motion (or lack thereof) to effect a change in the physical communications situation. In general, the movement required would be partially orthogonal to the movement goals of the overarching mission (e.g. environmental sampling, hazard detection).

One technique, detailed in [6], is having the AUV move to improve signal-to-noise ratio over a modeled path to a receiver whose position is known or modeled. It is possible to improve the quality of the received signal by modeling the transmission loss (TL) from the vehicle to a known receiver (along with the noise (NL) at the receiver) and then moving the vehicle to a local minimum in the resulting predicted signal-to-noise ratio (SNR), where

$$SNR = SL - TL - NL \tag{1}$$

TABLE I: Techniques to improve acoustic communications available to an artificially intelligent AUV

Description	Disruptive	Improvement	Requires	Advantages	Disadvantages
Move to expected beneficial position	yes	higher SNR or reduced multipath	receiver position, propagation model	potentially large gains (e.g. SOFAR channel)	very disruptive
Stop to transmit or receive	yes	less Doppler, lower noise	source transmit time, self-noise model	minimally disruptive	synchronization with transmitter
Frequency selection	no	higher SNR	receiver position, propagation model, broadband or multi-channel hardware		extra hardware
Transmit power selection	no	lower power	receiver position, propagation model	easy to implement	accurate modeling difficult
Entropy-based source encoding	no	increased information per bit	data model	complementary to other techniques	data modeling time consuming or difficult

with source level (SL), TL, NL, and SNR in decibels [7].

This maneuver has the potential to increase the data throughput of the acoustic link since the signal strength directly relates to the (theoretical) channel capacity C via the Shannon-Hartley theorem

$$C = B \log_2(1 + 10^{SNR/10}) \tag{2}$$

where C is in bits per second and bandwidth B is in Hertz, assuming white noise [8]. Depending on the modulation scheme employed, progress towards this unattainable theoretical limit may not limited by the signal-to-noise ratio, but rather by another environmental factor such as multipath echoes (which cause inter-symbol interference) or Doppler shifts. One drawback to this technique is that it requires having sufficient environmental data or statistics to adequately model the propagation path between source and receiver. Such information is sometimes not available to the vehicle.

Another technique involves slowing or stopping the vehicle to mitigate self-noise and/or Doppler effects. Certain modulation schemes, such as orthogonal frequency division multiplexing (OFDM), are highly sensitive to Doppler shifts. Since vehicle speeds (order of 10^0 m/s) are not negligible compared to the speed of sound in sea water (order of 10^3 m/s), normal vehicle motion can be disruptive to successful communications. Whether this technique is useful or not depends on the autonomy system understanding the requirements of the acoustic modem; it makes no sense to stop the vehicle if the modem's modulation is immune to the relevant frequency shifts. A second reason to arrest the motion of the vehicle is just prior to receipt of a telegram to reduce the vehicle's self noise. Zimmerman, et al [9] found that the self-noise of a typical mid-size AUV (the Bluefin Odyssey IIb) was 20 to 50 dB higher than the ocean background noise levels in the 10Hz to 10kHz range; the upper end of this band is commonly used for AUV communication. Most of this noise is motion related (propeller and turbulence), suggesting that much of this noise could be removed by stopping the vehicle temporarily to receive communications. However, such a scheme requires knowledge of incoming transmissions, which can be pre-arranged (e.g. fixed time division multiple access (TDMA) medium access control (MAC)) or negotiated (e.g. request-to-send/clear-to-send style MAC schemes).

Both of these techniques have the obvious disadvantage of taking time and power away from the core mission for the purpose of communications. However, many missions (especially detection of mines or submarines) are inherently useless without timely reports of collected data. Thus, when using these disruptive techniques some form of multi-objective optimization needs to be used. Otherwise, there is a risk of the mission collapsing to the degenerate case where all of the mission time is spent communicating useless data.

III. Non-disruptive Techniques

Some methods that do not require potentially unacceptable changes to the mission include selecting modem parameters based on propagation models and use of entropy source coding, such arithmetic coding.

A. *Transmitter parameter selection*

Rather than moving the vehicle as suggested in the previous section, the vehicle can select an optimal bitrate, transmit power or frequency band. Given that the acoustic absorption per unit distance increases with frequency, whereas the ambient noise level generally decreases with frequency, a maximum in the expected signal-to-noise ratio exists varies as a function of carrier frequency as shown in [10]. Thus, a vehicle aware of the range to its communicating partner can select the available carrier frequency that is expected to be closest to this maximum SNR. Similarly, the power can be adjusted to reach an expected acceptable threshold for a given modulation scheme and bit rate. Due to the realities of designing broadband tranducers, very wide band modems required to make this technique feasible are not presently available. However, a vehicle equipped with two tranducers with different bands could select between the two based on the range to the receiver.

B. Arithmetic coding

Unlike the other approaches given here, this technique is agnostic of the particulars of the physical link, assuming error-free channel coding is provided, as is typically the case. Thus, it is complementary with the other approaches discussed here. Here, a overall model of the source data is provided, and shared knowledge of the vehicle's mission is used to determine a reduced model for use by an entropy source coder, such as an arithmetic coder. This coder encodes the source data into a near minimal lossless representation. The drawback to this approach is that it requires knowledge (i.e. a model) of the physical origin of the data collected, which can be time consuming or difficult to obtain. Fortunately, in many cases the model can be generated (and adaptively updated) by prior statistics.

Arithmetic coding was chosen over various alternatives because

- assuming an accurate model, it produces a nearly optimal encoded bitset.
- the modeling process is separate from the coder design. This allows a single implementation of an arithmetic coder to function on many distinct sources of data.

The main drawback is that arithmetic coding has a reasonably high computational cost. This is generally not a concern for the underwater vehicle domain since available computing resources typically far outpace the throughput of the acoustic channel.

1) Choosing and representing source data: Here we will examine an implementation and its performance transmitting hypothetical messages pulled from a experimental dataset (shallow water GLINT10 experiment in the Tyrrhenian Sea) containing in excess of sixty hours of cumulative dive time. Specific quantities from this experiment or chosen here for this example are given in Table II. The desired transmission in this example is a Cartesian representation of the vehicle's position $(x[n], y[n], z[n])$ with each discrete step n separated by a predetermined time τ, where

TABLE II: GLINT10 Parameters

Parameter	Value
Time between messages (τ)	10 s
Number of full transmissions (N_f)	199
Mean size of full transmission	60 bits
Number of delta transmissions (N_d)	24420
Size of delta transmission	See Fig. 6
Vehicle speed (s)	1.5 m/s
Water depth (D)	110 m
Experiment datum (lat_d, lon_d)	42.45667°N, 10.875°E
Transmitted x, y, z precision	1 m
Delta model bounds: $[dx_{min}, dx_{max})$	$\pm 5s\tau$ = [-50, 51) m
Delta model bounds: $[dy_{min}, dy_{max})$	$\pm 5s\tau$ = [-50, 51) m
Delta model bounds: $[dz_{min}, dz_{max})$	$\pm D/10$ = [-11, 12) m

$$(x[n], y[n]) = UTM_{WGS84}(lon[n], lat[n]) - UTM_{WGS84}(lon_d, lat_d) \quad (3)$$

and $z[n]$ is the negative of the pressure-derived vehicle depth. UTM_{WGS84} is the Universal Transverse Mercator transformation using the WGS'84 ellipsoid [11], $lon[n], lat[n]$ are the vehicle's longitude and latitude, and lon_d, lat_d are the latitude and longitude of the experiment datum, a reference used for convenience (unrelated to the UTM zone datum).

The position of the vehicle is a commonly desired quantity during operations, as it lends assurance to the operators of the vehicle's correct functioning, as well as being necessary to geolocate any instrument data transmitted concurrently. This technique can be extended for transmitting many types of scalar data whose source can be modeled, but for brevity the focus will be on transmitting the vehicle's position alone.

The data used are given in Fig. 1 and Fig. 2, and represent one AUV performing a variety of data collection and adaptive autonomy missions. The details of the missions are not of interest here, as the goal is to develop a technique for communicating position data regardless of the vehicle's mission.

Position measurements were transmitted as one of two types of messages:

- Full transmissions: $(t_f, x(t_f), y(t_f), z(t_f))$ including the time and full position of the vehicle relative to the experiment datum.
- Delta transmissions: $(dx[n], dy[n], dz[n])$, where $n = t_f - [0, 1, 2, 3...]\tau$. Sent continously following a full transmission or prior delta transmissions until

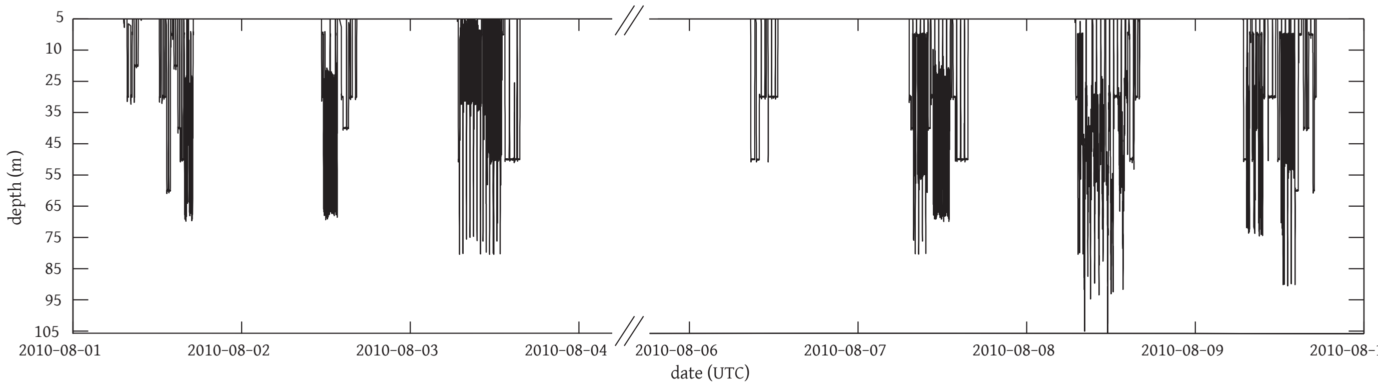


Fig. 1: z position (negative depth) of the "Unicorn" AUV for the x and y positions given in Fig. 2.

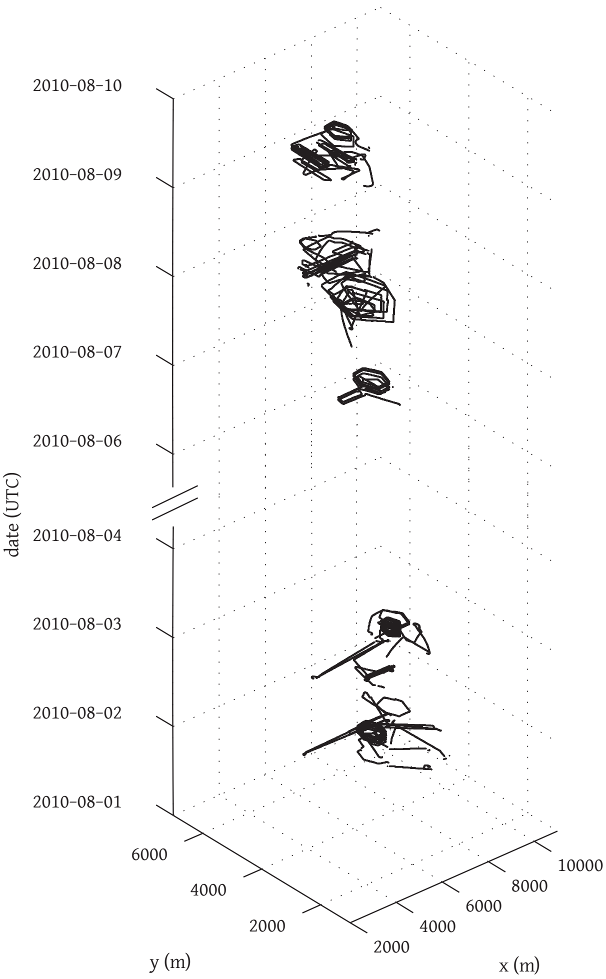
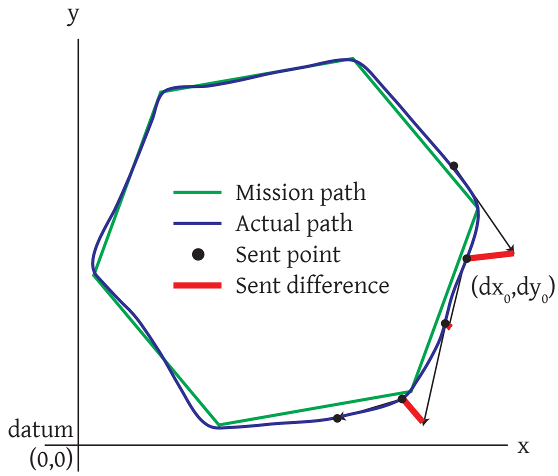
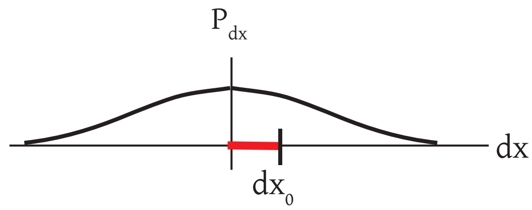


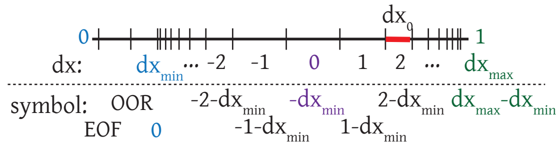
Fig. 2: Cartesian x and y positions from the "Unicorn" AUV during the entire GLINT10 trial. Many different mission types were run, providing a rich dataset of actual AUV maneuvers. Only positions where the vehicle was at depth ($z > -2$) are included, as the vehicle has access to much higher quality links (e.g. IEEE 802.11 wireless Ethernet) than acoustic on the surface.



(a) Generation of differences (dx and dy) from the vehicle's actual position from the extrapolated position (generated on both sender and receiver using tracked positions previously transmitted).



(b) Example of the probability model used, represented the error between the actual and extrapolated positions.



(c) Arithmetic coding symbol intervals (each dx is mapped to a symbol with 1-meter precision).

Fig. 3: Example illustration mapping vehicle position (a) to a given probabilistic model (b) used to generate the symbol intervals required for arithmetic coding (c).

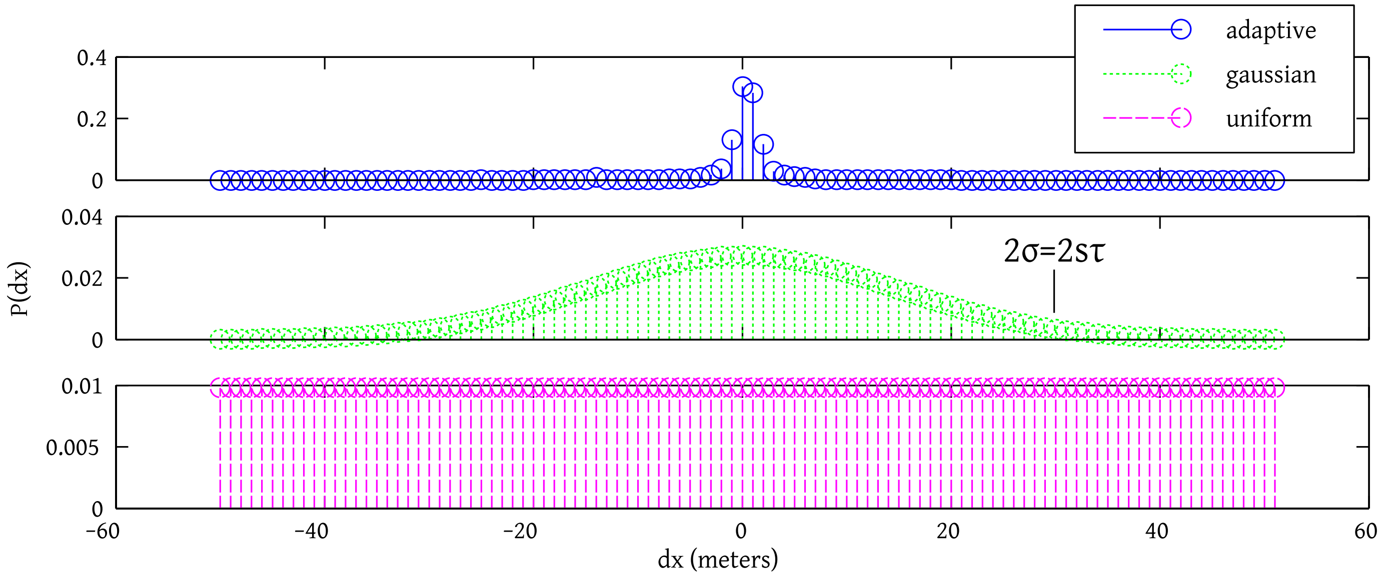


Fig. 4: The three models used for the results given in Fig. 6.

the vehicle was removed from operation for greater than τ seconds, after which a full transmission is sent to reinitialize the receiver's state. The differences are computed symmetrically on the vehicle and receiver using a simple tracker: the prior two transmitted positions were used to determine yaw Θ where

$$\Theta = \tan^{-1}\frac{y[n-1]-y[n-2]}{x[n-1]-x[n-2]} \tag{4}$$

The vehicle's last position is extrapolated using this heading at the fixed speed s, and this is used as a reference for the difference (or error) to the actual vehicle position to be transmitted, such that

$$(dx[n], dy[n]) = (x[n] - (x[n-1] + \tau|\vec{v}|\cos\Theta), \\ y[n] - (y[n-1] + \tau|\vec{v}|\sin\Theta)) \tag{5}$$

This operation is visualized in Fig. 3a. For depth, the last difference is used:

$$dz[n] = z[n] - (z[n-1] - z[n-2]) \tag{6}$$

n does not need to be transmitted, assuming the lower layers of the network stack can provide in-order receipt of messages without duplicate packets. In this case, the decoding simply increments n on each message received. As this can be accomplished with automatic repeat request (ARQ) with a single alternating bit to discard duplicates, this is a reasonable assumption.

2) Generating a source model: The next step in this process is identifying a suitable model. The full transmissions were encoded using a uniform probability distribution, since the vehicle could be reasonably be redeployed anywhere in the operation region. Since the delta transmissions make up the vast majority of transmissions from this dataset ($\frac{N_d}{N_d+N_f} = 99.2\%$), we will focus on these messages. The process of mapping the source delta data from the previous section is sketched in Fig. 3b.

A priori, it seems logical that the probability distribution governing the delta values (dx, dy, dz) would be zero mean, since any pattern the vehicle makes will have an equal number of negative and positive position differences (for example, see the hexagon in Fig. 3a. The negative dx on the east side will be offset by the positive dx on the west side.). The shape of the distribution is unclear, however, and depends substantially on the manuevering choices the vehicle makes (tight circles would lead to high error using the dynamic model given in Equation 5, straight lines would be low error). Thus, three distributions were compared (all shown in Fig. 4):

- Uniform:

$$P[dx] = \begin{cases} \frac{1}{dx_{max}-dx_{min}-1} & dx \in [dx_{min}, dx_{max}) \\ 0 & dx \notin [dx_{min}, dx_{max}) \end{cases} \tag{7}$$

- Normal,

$$P[dx] = \begin{cases} \mathcal{N}(0, (s\tau)^2) & dx \in [dx_{min}, dx_{max}) \\ 0 & dx \notin [dx_{min}, dx_{max}) \end{cases} \tag{8}$$

The standard deviation was chosen so that the "worst case" scenario (vehicle makes a 180° turn immediately after the preceeding transmission) has about 95% of the probability mass, that is

$$1.96\sigma \approx 2s\tau \tag{9}$$

- Adaptive: This model starts processing the dataset with the uniform distribution given above, and then equally incorporates the statistics of all previously transmitted symbols. Thus, an accurate model of the vehicle's prior positions is built up to encode future positions. At any sample m_0, the model is

$$P[dx] = \begin{cases} \frac{N_{dx}+1}{m_0+dx_{max}-dx_{min}-1} & dx \in [dx_{min}, dx_{max}) \\ 0 & dx \notin [dx_{min}, dx_{max}) \end{cases} \tag{10}$$

where N_{dx} is number of prior transmissions over $[dx[0], dx[m_0 - 1]]$ that had the value dx, and $m = 0$ is the start of the experiment (so that $[dx[0], dx[m_0 - 1]]$ contains all decoded transmissions, both full and delta). The model is updated after encoding and after decoding so that the sending AUV and the receiver can share the same state.

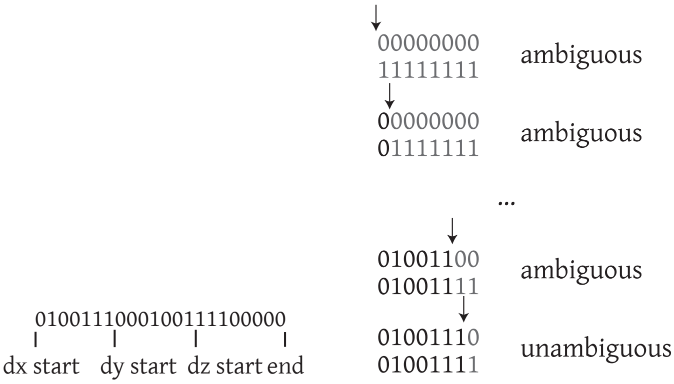
3) Implementing and using the arithmetic coder: To ensure this work can be easily fielded on AUVs in the near future, the arithmetic coder was implemented in the Dynamic Compact Control Language (DCCL), part of the Goby2 project [12]. The details of the arithmetic coder was based on the widely used integer implementation by Witten, Neal, and Clearly [13]. While the integer implementation is used in the code, this paper uses the floating point notation that involves encoding a range from [0, 1) using normalized probability models.

The mapping from delta values to symbols S (shown in Fig. 3c) is given by

$$S[dx] = \begin{cases} dx - dx_{min} & dx \in [dx_{min}, dx_{max}) \\ \text{out-of-range} & dx \notin [dx_{min}, dx_{max}) \\ \text{end-of-file} & dx \in \emptyset \end{cases} \tag{11}$$

with two special symbols: end-of-file (EOF) used to indicate the end of encoding, and out-of-range (OOR) used to indicate any value outside $[dx_{min}, dx_{max})$. An EOF symbol is not required if the number of messages encoded per packet is arranged between sender and receiver ahead of time.

However, one innovation was required to conform to the DCCL requirement that decoders consume exactly the same number of bits as the encoder produces. The implementation of an arithmetic coder given in [13] and elsewhere assumes that the decoder can safely read nonsense bits past the end of the file, until the actual end-of-file symbol is decoded. This will not work with DCCL since extra bits used in decoding end up being taken from those required for the next field. Thus, in



(a) Example DCCL bitstream for a single encoded delta message. The "dy start" and "dz start" markers are given for illustration only; the way DCCL distinguishes the start of one field is where the last field's decoder left off.

(b) Example decoding dx from this example bitstream. Both the upper and lower bounds are tracked, consuming a single bit at a time until the range unambiguously identify a symbol.

Fig. 5: Example of the arithmetic decoder for DCCL, showing tracking of decoded ranges to ensure the number of bits consumed by the encoder and the decoder are identical.

the DCCL implementation used here, the decoder tracks both the upper (current bitset followed by all ones) and lower (current bitset followed by all zeros) bounds of the current symbol, adding bits one at a time until the symbol is unambigously decoded. An example of this process is given in Fig. 5. Relatedly, the end of the bitstream must be encoded exactly so that the decoder does not leave extra bits in the stream that would corrupt the next field of the message. The authors of [13] always use two bits to indicate which middle quarter (either [0.25, 0.5) or [0.5, 0.75)) is wholly contained by the final encoder range. However, when at least one ends of the encoder range is at one of the bounds (low = 0 and/or high = 1), fewer bits may be required. The exact number of end bits (e) is given by

$$e = \begin{cases} \emptyset & \text{high } = 1, \text{low } = 0, \text{ no follow bits} \\ 0 \text{ or } 1 & \text{high } = 1, \text{low } = 0, \text{ follow bits} \\ 1 & \text{high } = 1, 0 < \text{low } < 0.5 \\ 0 & 0.5 < \text{high } < 1, \text{low } = 0 \\ 01 & \text{low } < 0.25, \text{high } \geq 0.5 \\ 10 & \text{low } < 0.5, \text{high } \geq 0.75 \end{cases} \tag{12}$$

plus any follow bits accrued from prior center expansions around [0.25, 0.75). This is consistent with the (rounded-up) information entropy $H = -log_2(P)$ for those cases which is given by

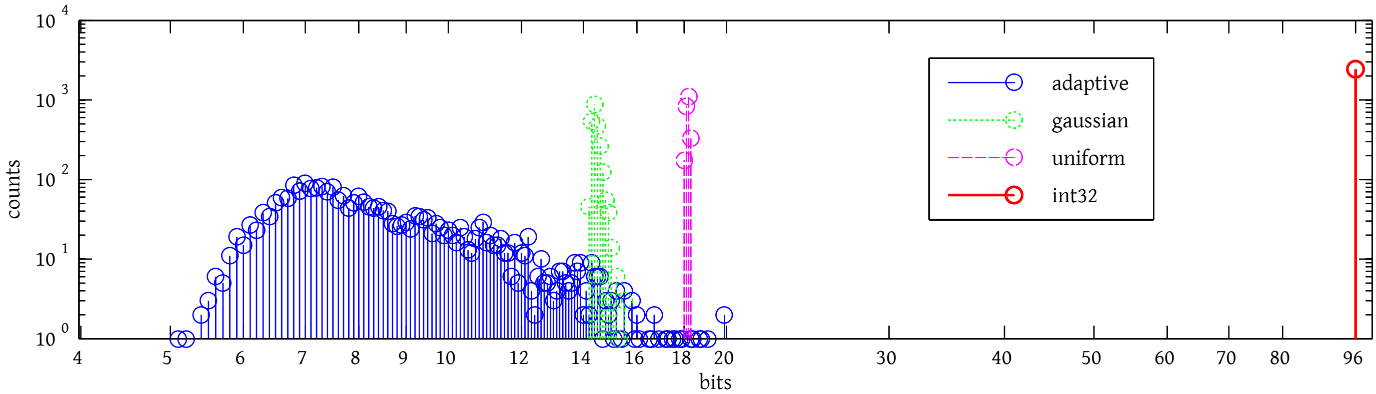


Fig. 6: Log-log plot of the number of messages generated with a given size for the three models given in Fig. 6 each operating on the dataset given in Figs. 1 and 2 along with a 32-bit integer control for comparison.

$$\lceil H_{bits} \rceil = \begin{cases} 0 & \text{high} \ = 1, \text{low} \ = 0 \\ 1 & \text{high} \ = 1, 0 < \text{low} \ < 0.5 \\ 1 & 0.5 < \text{high} \ < 1, \text{low} \ = 0 \\ 2 & \text{low} \ < 0.25, \text{high} \ \geq 0.5 \\ 2 & \text{low} \ < 0.5, \text{high} \ \geq 0.75 \end{cases} \tag{13}$$

4) Results: Each of the three models given in Fig. 4 was used with the arithmetic coder discussed in the prior section to encode the dataset given in Figs. 1 and 2. The resulting size of each message was recorded and the statistics plotted in Fig. 6, along with the uncompressed 32-bit integer as a reference point. As expected, the Gaussian model performed better than the uniform distribution since it makes use of the dynamic model from Eq. 5 where low errors are more probable than high errors (the vehicle in general continues on a similar path of motion). However, this model overstates the error significantly from the adaptive model, as seen by the difference in standard deviation between the two models in Fig. 4. Once the adaptive model was initialized with sufficient data, it easily outperforms the other two models, with a mean of 8.66 bits per message, an improvement of 91%, 52%, and 40% from the uncompressed (int32), uniform, and normal models, respectively. Furthermore, this technique using the adaptive model is an improvement of 86% over the widely used Compact Control Language [14], which uses 61 bits to encode a vehicle position in the "MDAT_STATE" message.

IV. Conclusion

Several techniques were reviewed or presented for using AUVs' positional knowledge and/or mobility to increase the useful *bits per Joule* sent from the AUVs and successfully received. These techniques fall broadly into two categories: disruptive and non-disruptive, which do and do not affect the motion of the vehicle respectively.

Since the acoustic channel presents many challenges, it is essential that the underwater robotics community collaborate fully with the underwater communications community, especially as the latter begins to standardize. The sensitivities of a specific acoustic modem system to various physical channel quantities (e.g. signal-to-noise ratio, multipath, Doppler) should be quantitatively available to the underwater roboticist, who can use this knowledge to make well-informed decisions on the tradeoffs between various disruptive behaviors. For example, it makes no sense to waste power transiting to improve SNR if the available bit-rate is instead limited by multipath.

References

[1] Y. Zhang, M. Godin, J. Bellingham, and J. Ryan, "Using an autonomous underwater vehicle to track a coastal upwelling front," *IEEE Journal of Oceanic Engineering*, vol. 37, no. 3, pp. 338 –347, Jul. 2012.

[2] S. Petillo, A. Balasuriya, and H. Schmidt, "Autonomous adaptive environmental assessment and feature tracking via autonomous underwater vehicles," in *OCEANS 2010 IEEE - Sydney*, May 2010, pp. 1 –9.

[3] M. Purcell, D. Gallo, G. Packard, M. Dennett, M. Rothenbeck, A. Sherrell, and S. Pascaud, "Use of REMUS 6000 AUVs in the search for the air france flight 447," in *OCEANS 2011*, Sep. 2011, pp. 1 –7.

[4] M. Chitre, S. Shahabudeen, and M. Stojanovic, "Underwater acoustic communications and networking: Recent advances and future challenges," *The State of Technology in 2008*, vol. 42, no. 1, pp. 103–114, 2008.

[5] J. Preisig, "Acoustic propagation considerations for underwater acoustic communications network development," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, pp. 2–10, 2007.

[6] T. Schneider and H. Schmidt, "Model-based adaptive behavior framework for optimal acoustic communication and sensing by marine robots," 2011. [Online]. Available: http://gobysoft.com/dl/gram2011.pdf

[7] R. J. Urick, *Principles of underwater sound*. McGraw-Hill, Apr. 1983.

[8] C. E. Shannon, “A mathematical theory of communication,” *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 5, no. 1, p. 355, Jan. 2001. [Online]. Available: http://doi.acm.org/10.1145/584091.584093

[9] R. Zimmerman, G. D’Spain, and C. Chadwell, “Decreasing the radiated acoustic and vibration noise of a mid-size AUV,” *IEEE Journal of Oceanic Engineering*, vol. 30, no. 1, pp. 179 – 187, Jan. 2005.

[10] M. Stojanovic, “On the relationship between capacity and distance in an underwater acoustic communication channel,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 11, no. 4, p. 3443, 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1347373

[11] NIMA, “Department of defense world geodetic system 1984: Its definition and relationships with local geodetic systems. second edition, amendment 1,” NIMA, Tech. Rep. TR8350.2, 2000, available at http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf, accessed January 3, 2010.

[12] T. Schneider and H. Schmidt, “Goby-acomms version 2: extensible marshalling, queuing, and link layer interfacing for acoustic telemetry,” in *9th IFAC Conference on Manoeuvring and Control of Marine Craft, Arenzano, Italy*, 2012.

[13] I. H. Witten, R. M. Neal, and J. G. Cleary, “Arithmetic coding for data compression,” *Communications of the ACM*, vol. 30, no. 6, p. 520540, 1987. [Online]. Available: http://dl.acm.org/citation.cfm?id=214771

[14] R. P. Stokey, L. E. Freitag, and M. D. Grund, “A compact control language for AUV acoustic communication,” *Oceans 2005-Europe*, vol. 2, p. 11331137, 2005.